UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE OPTICA Y OPTOMETRÍA

TESIS DOCTORAL

Desarrollo de instrumentos y tecnologías de e-Salud para el análisis de Big Data en lentes de contacto para presbicia

Development of e-Health instruments and technologies for Big Data analysis in contact lens for presbyopia

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Youssef Marrakchi Chikri

DIRIGIDA POR

José Manuel López Alonso

Madrid

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE ÓPTICA Y OPTOMETRÍA

TESIS DOCTORAL

Desarrollo de instrumentos y tecnologías de e-Salud para el análisis de Big Data en lentes de contacto para presbicia

Development of e-Health instruments and technologies for Big Data analysis in contact lens for presbyopia

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Youssef Marrakchi Chikri

DIRECTOR

José Manuel López Alonso

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE ÓPTICA Y OPTOMETRÍA

TESIS DOCTORAL

Desarrollo de instrumentos y tecnologías de e-Salud para el análisis de Big Data en lentes de contacto para presbicia

Development of e-Health instruments and technologies for Big Data analysis in contact lens for presbyopia

PROGRAMA DE DOCTORADO EN ÓPTICA OPTOMETRÍA Y VISIÓN

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Youssef Marrakchi Chikri

DIRECTOR

José Manuel López Alonso

DEDICATION AND ACKNOWLEDGEMENTS

DEDICATION AND ACKNOWLEDGEMENTS

To my family, whose unwavering support and encouragement have been my greatest source of strength throughout this journey.

To my friends, who have stood by me in both the challenges and triumphs.

And to all those who believe in the power of knowledge and perseverance.

First and foremost, I would like to express my deepest gratitude to my tutor, **Dr. David Madrid-Costa**, for his invaluable guidance, support, and encouragement throughout this research. His insights and expertise have been instrumental in shaping this work.

I am also profoundly grateful to my director, **Dr. José Manuel López Alonso**, for his continuous support, thoughtful advice, and dedication. His constructive feedback and mentorship have greatly enriched this thesis.

I would also like to thank **Nuria Garzón Jimenez** and **Mariano González Pérez** for their collaboration, support, and dedication during my predoctoral stage through discussions, documentation for the ethics committee, and other scientific documents.

I extend my appreciation to my colleagues and peers, whose discussions and collaboration have contributed significantly to my academic growth.

Finally, I would like to thank my family and friends for their endless patience, understanding, and motivation. Their belief in me has been my driving force, and this achievement would not have been possible without them.

RESUMEN

RESUMEN

Objetivo:

El objetivo de esta investigación es desarrollar un sistema de monitorización y análisis continuo de los factores que influyen en la comodidad y el ajuste de lentes de contacto, a través del desarrollo de una aplicación para móviles, tablets y ordeandores. El sistema pretende proporcionar datos en tiempo real para ayudar a los médicos a adaptar las prescripciones de lentillas en función de diversos parámetros, como el comportamiento de parpadeo y las condiciones ambientales. Mediante el uso de tecnologías accesibles, como vídeos autograbados y datos de estaciones meteorológicas, la investigación busca ofrecer un enfoque práctico y escalable para la monitorización continua de los usuarios de lentillas.

Materiales y métodos:

Se ha desarrollado una aplicación móvil híbrida utilizando bibliotecas HTML5, CSS3, JavaScript, Angular e Ionic para garantizar la compatibilidad entre plataformas Web, iOS y Android. Esta aplicación se conecta a estaciones meteorológicas cercanas de libre acceso para recopilar datos ambientales, como los niveles de humedad y contaminación, sin registrar ni mostrar la ubicación del paciente, lo que garantiza la privacidad. La aplicación también permite a los pacientes grabar videos de sus ojos, que luego se analizan en el entorno MATLAB para evaluar la dinámica de su secuencia de parpadeo y verificar el enrojecimiento ocular.

Las características del parpadeo se extraen utilizando un algoritmo diseñado para replicar los resultados que se obtienen típicamente con cámaras de alta velocidad de cuadros, lámparas de hendidura y dispositivos de seguimiento ocular, que se utilizan comúnmente en entornos clínicos. Este proceso permite la recopilación de valiosos datos de parpadeo fuera del entorno clínico, lo que proporciona más flexibilidad y acceso tanto para los pacientes como para los médicos. Las características del color de la esclerótica para la detección del enrojecimiento ocular se obtienen utilizando un algoritmo propio que detecta la esclerótica a partir de los marcos de los ojos.

Con los datos obtenidos del algoritmo de parpadeo y la aplicación desarrollada, se crea una base de datos de 208 secuencias de vídeo y se analizan mediante técnicas de big data.

Resultados y discusión:

Los resultados obtenidos a partir de la validación preliminar de la aplicación indican que los factores ambientales, en particular la humedad, desempeñan un papel importante a la hora de influir en la dinámica del parpadeo. El análisis de los vídeos recopilados de los participantes ha demostrado diferencias notables en los patrones de parpadeo cuando se llevan lentes de contacto en comparación con cuando no se llevan. En concreto, los sujetos que llevaban lentes de contacto mostraron un movimiento de párpado reducido, velocidades de parpadeo más lentas, un mayor número de parpadeos incompletos y patrones de intervalo de parpadeo alterados. Estos cambios sugieren una relación directa entre la incomodidad de las lentes de contacto y las modificaciones en el comportamiento natural del parpadeo. Además, el aumento del enrojecimiento ocular, en particular en la región de la esclerótica temporal, destaca el impacto fisiológico del uso de lentes, posiblemente debido a la irritación, la sequedad o la fricción mecánica.

La capacidad de monitorear y clasificar secuencias de parpadeo mediante videos grabados directamente por los pacientes tiene el potencial de proporcionar información valiosa a los médicos sobre la adaptación de lentes de contacto, particularmente entre pacientes con presbicia. Los resultados indican que los sujetos sin lentes de contacto muestran un patrón de parpadeo más uniforme, formando un grupo distinto con el 85% de los datos registrados. En contraste, las personas que usaban lentes de contacto demostraron una mayor variabilidad, particularmente los pacientes con presbicia. Esta heterogeneidad está influenciada en gran medida por las diferencias de color de esclera en las regiones nasal y temporal, así como por variaciones en la dinámica del parpadeo, lo que sugiere un proceso de adaptación más complejo entre este grupo demográfico. Estos hallazgos refuerzan la necesidad de estrategias personalizadas de adaptación de lentes de contacto para mitigar la incomodidad y el posible daño a la superficie ocular.

Conclusiones:

Los hallazgos del estudio destacaron la importancia de desarrollar herramientas digitales capaces de monitorear de manera continua la dinámica del parpadeo y los cambios en la superficie ocular en condiciones del mundo real. La implementación de una aplicación móvil híbrida que utiliza los marcos Angular e Ionic permite una recolección de datos eficiente, integrando variables ambientales sin comprometer la privacidad del paciente. Los resultados confirmaron que la humedad es un factor dominante que influye en el comportamiento del

parpadeo, lo que debe tenerse en cuenta en el desarrollo de materiales para lentes de contacto y estrategias de adaptación.

Además, las diferencias observadas en las características del parpadeo entre usuarios de lentes de contacto y no usuarios indican que la incomodidad inducida por las lentes se puede evaluar de manera efectiva a través del enrojecimiento de la esclera y las desviaciones de la frecuencia del parpadeo. La capacidad de clasificar a los sujetos en función de la uniformidad del parpadeo y las características del color de la esclera proporciona una nueva vía para evaluar la adaptación de las lentes de contacto, particularmente en pacientes con presbicia. El estudio también demuestra la viabilidad de usar cámaras de teléfonos móviles para el análisis del parpadeo, lo que abre posibilidades para la recolección de datos a gran escala y aplicaciones de big data en optometría.

En conclusión, los hallazgos resaltan la necesidad de enfoques personalizados en la adaptación de lentes de contacto, especialmente para personas con presbicia que presentan una mayor variabilidad en la dinámica del parpadeo y la coloración de la esclera. Se necesitan más estudios que incorporen tamaños de muestra más grandes y diversas condiciones ambientales para refinar estas metodologías y mejorar la toma de decisiones clínicas con respecto a la comodidad y el ajuste de las lentes de contacto. La integración del análisis del parpadeo con los síntomas oculares informados por los propios pacientes puede proporcionar una evaluación integral, mejorando en última instancia los resultados del paciente y la experiencia del usuario de lentes de contacto.

ABSTRACT

ABSTRACT

Objective:

The objective of this research is to develop a system for continuous monitoring and analysis of the factors that influence contact lens comfort and fit, through the development of an application for mobile phones, tablets, and computers. The system aims to provide real-time data to help physicians adapt contact lens prescriptions based on various parameters, such as blinking behavior and environmental conditions. By using accessible technologies, such as self-recorded videos and weather station data, the research seeks to offer a practical and scalable approach for continuous monitoring of contact lens wearers.

Material and methods:

A hybrid mobile application was developed using HTML5, CSS3, JavaScript, Angular, and Ionic libraries to ensure compatibility across Web, iOS, and Android platforms. This application connects to nearby free-access weather stations to gather environmental data, such as humidity and pollution levels, without recording or displaying the patient's location, ensuring privacy. The application also allows patients to record videos of their eyes, which are then analyzed in the MATLAB environment to assess the dynamics of their blinking sequence and check the eye redness.

The blinking characteristics are extracted using an algorithm designed to replicate the results typically obtained with high-frame-rate cameras, slit lamps, and eye-tracking devices, which are commonly used in clinical settings. This process allows the collection of valuable blinking data outside of the clinical environment, providing more flexibility and access for both patients and clinicians. The sclera color features for eye redness detection are obtained using a self made algorithm that detects the sclera from eye frames.

Jointly with the data obtained from the blinking algorithm and the application developed, a database is created from 208 video sequences and analyzed using big data techniques.

Results and discussion:

The results obtained from the preliminary validation of the application indicate that environmental factors, particularly humidity, play a significant role in influencing blinking dynamics. The analysis of videos collected from participants has demonstrated notable differences in blink patterns when wearing contact lenses compared to when they are not.

Specifically, subjects wearing contact lenses exhibited reduced eyelid movement, slower blink speeds, an increased number of incomplete blinks, and altered blink interval patterns. These changes suggest a direct relationship between contact lens discomfort and modifications in natural blinking behavior. Furthermore, the increase in eye redness, particularly in the temporal sclera region, highlights the physiological impact of lens wear, possibly due to irritation, dryness, or mechanical friction.

A critical aspect of the study was the validation of the blink characterization algorithm against high-frame-rate cameras traditionally used in clinical environments. The findings support the feasibility of using ubiquitous smartphone cameras for blink analysis, thereby facilitating data collection outside the clinical setting. The ability to monitor and classify blinking sequences using self-recorded videos has the potential to provide valuable information to clinicians regarding contact lens adaptation, particularly among presbyopic patients. The results indicate that subjects without contact lenses exhibited a more uniform blink pattern, forming a distinct cluster with 85% of the recorded data. In contrast, individuals wearing contact lenses demonstrated higher variability, particularly presbyopic subjects. This heterogeneity is largely influenced by scleral color differences in the nasal and temporal regions, as well as by variations in blink dynamics, suggesting a more complex adaptation process among this demographic. These findings reinforce the necessity of personalized contact lens adaptation strategies to mitigate discomfort and potential ocular surface damage.

Conclusions:

The study's findings underscore the importance of developing digital tools capable of continuously monitoring blinking dynamics and ocular surface changes in real-world conditions. The implementation of a hybrid mobile application using Angular and Ionic frameworks allows for efficient data collection, integrating environmental variables without compromising patient privacy. The results confirm that humidity is a dominant factor influencing blinking behavior, which should be considered in the development of contact lens materials and adaptation strategies.

Moreover, the observed differences in blink characteristics between contact lens wearers and non-wearers indicate that lens-induced discomfort can be effectively assessed through scleral redness and blink frequency deviations. The ability to classify subjects based on blink uniformity and scleral color features provides a new avenue for evaluating contact lens fit, particularly in presbyopic patients. The study also demonstrates the feasibility of using

smartphone cameras for blink analysis, opening possibilities for large-scale data collection and big data applications in optometry.

In conclusion, the findings highlight the need for personalized approaches in contact lens adaptation, especially for presbyopic individuals who exhibit greater variability in blink dynamics and scleral coloration. Further studies incorporating larger sample sizes and diverse environmental conditions are necessary to refine these methodologies and enhance clinical decision-making regarding contact lens comfort and fit. The integration of blink analysis with self-reported ocular symptoms can provide a comprehensive assessment, ultimately improving patient outcomes and contact lens user experience.

INDEX

INDEX

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AAC. Advanced Audio Coding
ADC. Analog-to-digital converter
API. Application Programming Interface
BB. Bounding Box
blob. Binary Large Object
CAT. Computer Adaptative System
CBT. Cognitive Cognitive Behavior Therapy
CDF. Cumulative Distribution Function
CFL. Precision Light Source
CIE. Internationa Commission on Illumination
CL. Contact lens
CLI. • Command-line interface
CNN. Convolutional Neural Network
CRS. Corneal Refractive Surgery
CSS. Cascading Style Sheets
DDD. Drowsy Driver Dataset
EAR. Eye Aspect Ratio
EBR. Eye Blink Ratio
ECG. Electrocardiogram
EEG. Electroencephalogram
EFV. eye feature vector
EMG. exponentially modified Gaussian
EOG. Electrooculogram
FDA. Food and Drug Administration
FFMPEG. Fast Forward Moving Picture Experts Group
FN. False Negative
FP. False Positive
FTP. File Transfer Protocol
HOG. histogram of oriented gradients
HSV. Hue, Saturation and Value
HTML. Hypertext Mark-up Language
HTTP. Hypertext Transfer Protocol
IBI. Inter-Blink interval, Inter-Blink Interval
iOS. iPhone Operating System
IP. Image Processing
IT. Information Technology
JSON. JavaScript Object Notation
KLT. Kanade Lucas-Tomasi
MCD. Maximum Closure Duration
MDR. Medical Device Regulation
MFV. mouth feature vector
MMORPG. Massively Multiplayer Online Role-Playing Game
NCA. Neighbor Component Analysis
OCT. Optical Coherence Tomography
OCTaVIA. Optical Coherence Tomography Visual Atlas

OO. orbicularis oculi
OOP. Object-Oriented Programming
PCA. Principal Component Analysis
PDF. probability density function
PERCLOS. The percentage of time that the eyes are more than 80% closed
PHP. Hypertext Preprocessor
PNG. Portable Network Graphics
PRO. Patient Reported Outcome
PSD. Power Spectrum Density
RAM. Random Access Memory
RG. red-green
ROI. Region of Interest
SARS. Severe Acute Respiratory Syndrome
SNR. Signal to noise ratio
SQL. Structured query language
SVD. Singular Value Decomposition
SVM. Support Vector Machine
TN. True Negative
TP. True Positive
WEBM. Web Media File
WHO. World Health Organization
XML. Extensible Markup Language

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

HYPOTHESES AND OBJECTIVES

1. HYPOTHESES, OBJECTIVES AND STRUCTURE

1.1 Work Hypothesis

With the use of smartphones, tablets, computers and environmental sensors through internet connectivity, it is possible to create a system to collect and monitor clinical data, patient-reported outcomes, and environmental data to characterize the performance of contact lenses.

1.2 Principal Objective

To develop an application for mobile phones, tablets and computers that is capable of collecting large-scale data on ocular behavior with contact lenses, including objective factors through blinking and visual characteristics of the eye and subjective data on patient comfort.

1.3 Secondary Objectives

1. Develop a web application to collect objective and subjective data from the patient; videos of the eye and users' information, and its environment; temperature, and humidity.
2. Connect to weather and air contamination stations using geo-localization to access additional environmental information.
3. Develop algorithms using image processing and artificial intelligence to extract eye physics parameters from eye videos.
4. Generate an extensive database by automatically connecting the web application and the workstation used to process the eye videos.
5. Produce preliminary results on a small data set to study the feasibility of the project. Use big data techniques to statistically analyze the small pilot database generated for classification, diagnosis, and prediction purposes related to the adaptation of different types of contact lenses.

After an introductory chapter with a bibliographic review on the subject, the methodology developed in the doctoral thesis to meet the previous objectives is introduced, consisting of: a description of the computer development of the application (app) developed, imaging algorithms for blink characterization with and without contact lenses, color-based imaging

algorithms for monitoring the impact of contact lens use on the sclera and conjunctiva, and a method for generating and classifying data for the database to be generated. Finally, the results chapter details the initial analyses on the generated database, as well as several big-data algorithms that could be used. The thesis is completed with conclusions, bibliography, and appendices.

2 INTRODUCTION

2. INTRODUCTION

2.1 Main Topic

The framework in which this thesis is presented is given by the participation of 3 institutions, Alain Afflelou Portugal, Mark'ennovy England, and the Complutense University of Madrid, in a European Project that aims to solve the problem of presbyopia in a way personalized.

This project is based on the premise that presbyopia is an ailment that affects the entire population from an advanced age that we can place at 50 years. Because the European population in general is one of the oldest and has a very high life expectancy (1), it can be stated that this population lives and will live a large part of its life with presbyopia (2).

However, although presbyopia has been the effect of several studies and there are also numerous techniques that provide a partial solution to this defect, currently there is no method that can provide an ideal solution to this problem that involves the restoration of correct vision at any distance (3).

The objective of this European project can finally be summarized as offering a therapeutic solution for presbyopia based on the particularities of each patient. The proposed solution is the use of new contact lenses, which means taking into account all personal characteristics and particularities, including the environment in which each person develops and the activities they practice daily. The proposed solution includes two different objectives. On the one hand, the development of new contact lenses that have the characteristics more personalized to the presbyopia population, which can be split into two different objectives. The first is the development of new optical designs with superior quality and the second objective is the development of new materials to limit the discontinuation of the tear film layer and reduce its early evaporation (4). On the other hand, the second main objective is the development of new technological tools to predict the best contact lens for each patient and monitor the fitting process beyond the optometrist’s consultation room, which is divided into three different tools. The first tool consists of a Patient Reported Outcome (PRO) instrument, a computer adaptive system (CAT) with better precision and sensitivity to obtain subjective information from the patient for further clinical diagnosis. The second tool consists of a computer model to simulate the interaction of the contact lens with the eye, predicting the response of the eye and the adaptation of the new developed contact lenses. The third objective consists of the developed and proposed thesis and aims to develop tools to collect data.

Each one of the objectives stated above will be achieved in two steps. The first one consists of the development of new technology, and the second one is a validation in a clinical trial of the method or the obtention of new information that will measure the improvement of the therapeutic solutions proposed.

Finally, the work developed in this thesis consists of collecting all the possible information that allows us to identify and monitor the meteorological and air pollution conditions in the environment of the presbyopic patient, as well as the clinical parameters and subjective results provided by the patient about the use and comfort of contact lenses.

To carry out this work, first we investigate with an explanation the phenomenon of presbyopia to subsequently identify all objective parameters related to the environment and air pollution, as well as signs of discomfort or relevant clinical parameters. Patient adaptation to new contact lenses.

2.1.1. What is Presbyopia?

The normal behavior of the accommodation of the eye occurs when the ciliary muscle contracts to make the lens thicker, decreasing the lens focal length and increasing the lens power, to see clearly near objects. When the ciliary muscle is relaxed, the lens becomes thinner, increasing the lens focal length and decreasing the lens power; so far, objects are clearly visible. Eye focusing is the ability to see clearly an image at a certain distance because of the correct recollection of its light rays and is intrinsically related to the lens focal length.

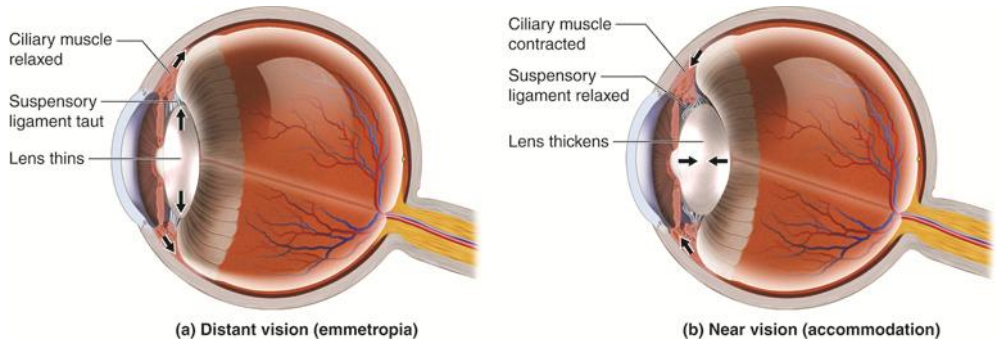

Figure 2-1

Figure 2-2: Accommodation of the lens with distant and near vision. This work by Cenveo is licensed under a Creative Commons Attribution 3.0 United States (http://creativecommons.org/licenses/by/3.0/us/).

Presbyopia is the physiological inability of the ciliary muscle and lens system to focus on a target near the target (5). Inefficacity in accommodation is associated with aging of the eye that results in progressively worsening ability to focus clearly on close objects. With age, the ability

to restore focus from near to far is also affected. Presbyopia is due to lens atrophy and its inability to change shape and change lens focal length through mechanically applied changes in zonular tension (tension due to ciliary muscle contraction) (3,6).

To restore the accommodation, different methods are available very different from each other that where tested and have their own advantages and disadvantages (3,4). Their focus is related to dry eye symptoms and symptoms due to aging and predisposition to suffer from this disease (7). From the methods stated, special attention was paid to the cost-effective techniques stated in the literature, which are spectacles lenses and contact lenses. Furthermore, both techniques have not suffered sufficient changes during the last decades, and innovation on quality of vision has barely improved, and no additional patents or methods have been implemented to reach continuous improvement by adapting the solution to each unique human eye condition (8).

In terms of quality of life improvements and optical limitations due to the relative position of the lens provided by spectacles (3), and in the same line as concluded in the study (9) which aims to determine the health-related quality of life associated with presbyopia, the justification for the study of the contact lens is made. Furthermore, from a study performed to compare the preferred vision correction, it was concluded in both cases that patients with presbyopia will prefer the contact lens solution if comfort and visual quality are reached (10).

2.1.2. Contact lens fitting. Main issues and variables of influence retrieved from literature.

In addition to improving quality of life and, specifically, better appearance and fewer restrictions on activity, the adjustment and fitting of contact lenses for a particular use for each specific patient becomes tricky. Contact lens use, especially for presbyopic patients, can be associated with many complications (11), ranging from induced dry eye and discomfort (12), to mucine ball, corneal neovascularization, keratitis, among others that cause discontinuation of use. Several studies have investigated the use of contact lenses by patients to study factors that affect discontinuation of use. The main factors of abandonment are the appearance of ocular symptoms. The main ones as cited above, are dry eye and discomfort (11–14).

As the principal objective of the contact lens in presbyopic patients is to correct the vision, and it is the first requirement imposed by the patient, the optical design is an important aspect to consider. A recent study involving presbyopic patients has been conducted using different optical designs, going from spherical designs to multifocal designs (15). It shows that the

multifocal designs are penetrating the market for this age group and the main daily problems encountered can be easily solved by additional counselling and instruction. This is in accordance with the following review (16) that enlighten us with the principal visual problems found while fitting multifocal lenses allowing the patient to have the best visual acuity in the different vision ranges (near, intermediate and far).

Another key factor to the fitting of contact lenses related to comfort is the material type and the frequency of replacement. We can find different contact lens types going from soft to rigid ones. In particular, the soft contact lenses group is composed of silicone contact lenses or non-silicone contact lenses (17). Additionally, different disposal liquids for contact lens care are available that are intrinsically related to the type of material and the frequency of renewal. All the different aspects related to material and the care products of contact lenses are there to provide the best experience for the final user and adapt to each different patient's eye.

Considering the results of the small review literature related to contact lens fitting, we approach the next chapter of this thesis considering a further and deeper study on the main variables to obtain from the eye patient and environmental conditions and the illnesses related to discomfort and contact lens discontinuation for the purpose stated on the objective of this thesis and this introduction chapter.

2.2 Telemedicine and optometry

The use of informatic technology in the field of medicine plays a crucial role in improving the healthcare system. Well-structured information about patients and easy access to these data provided by the use of Information Technology (IT) plays a crucial role in the efficiency of patient attention and the efficacy of the proposed treatment that results in improved health and satisfaction of patients(18). In the healthcare field, mobile apps are used to prevent, diagnose, and treat patients. The use of mobile applications by patients in a medical environment aims to expand patients' knowledge of his health and to provide different tools to improve his quality of life, to obtain medical resources and assistance in real time.

The World Health Organization (WHO) defines eHealth as the cost-effective and secure use of information and communications technologies in support of health and health-related fields, including healthcare services, health surveillance, health literature, and health education, knowledge, and research (19). The mHealth concept defined also by the WHO and is related to the use of mobile phones in healthcare will be the topic of this brief introduction.

Mobile apps are being used in many fields of medicine nowadays in all 3 differentiated stages that a patient goes through while facing a disease. Examples of different mobile phone apps for healthcare will be shown related to the main medical fields that involve the diseases related to the leading risk of global death (20).

Related to the chronic diseases of aged patients we can find eCAALYX ® as an example of an app. It is a remote monitoring system for the elderly who have various chronic diseases. This application (monitoring system) acts as an informed intermediary between the sensor worn by the patient and the web accessible by clinicians. This system, on the one hand, alerts the clinician once the measurement and location of the sensor (geographical position of the subject) has been collected, also notifying the patient. This system also performs a small analysis of the patient's health status through the data collected, detecting anomalies in the functioning of the cardiovascular system (tachycardia) or respiratory deficiencies related to respiratory infections. In addition, there is a graphical interface for the patient/user that allows viewing medical reports, collected data, and contacting health professionals (21).

One of the leading risks of the chronic disease and an important factor in global death is obesity (22). An example of a study implying mobile phones is ‘My Meal Mate’ (23). The design of an app to help patients lose weight that self-monitors physical activity, diet, and weight and gives feedback via text messages. It has a large database of UK-branded food products. The use of a smartphone implies that the patient is more likely to continue with the good habits of health and have access to this information easily and trustfully as the videos of good habits related to weight loss were validated (24).

Related to the disease of young patients, the resources and information of the patient are crucial. An effective management of chronic disease allows the patient's health to increase, and it avoids hospitalization. The integration of the smartphone into our daily life made it possible to assess patients about their disease allowing a better understanding of the action that needs to be taken to increase the health status of patients. This pilot study intends to design and test the implementation of a Telemanagement diabetes system involving healthcare professionals, patients, and patient's family members as the diabetes type 1 illness affects children. This study implemented a system to make the patients change is behavior by acting on their treatment protocol and lifestyle to improve their health based on the data collected on blood glucose levels (25).

Related to mental health diseases, smartphone devices offer diverse opportunities for enhancing mental health care, such as collecting dense and varied data for diagnosis and monitoring, utilizing machine learning for personalized insights, and connecting users to clinical care, peer support, emergency services, and innovative therapies beyond the app itself. In this field, there are two types of data, active and passive. Active data typically involves smartphone-based surveys like symptom monitoring or ecological momentary assessment, offering insights into patients' clinical course. Passive data, automatically collected through sensors, offer insights with reduced patient burden, enabling understanding of mental health experiences in context. An example of this type of data is the location of the patient to understand the life journey (26). The following article (27) shows all the results of the current studies where smartphone-supported psychological interventions reduce anxiety. Another example of use related to mental health is the Woebot (28): The study demonstrates that a text-based conversational agent designed to mirror the therapeutic process has the potential to offer an alternative and engaging method of delivering CBT (Cognitive Behavior Therapy).

In addition to mobile apps, wearable devices use 9DoF sensors (accelerometer, magnetometer, gyroscope) for sports fitness and health monitoring. The following review (29), covers motion tracking devices (e.g., MotionNode Bus, Opal) and wrist-worn commercial devices (e.g., Fitbit Flex, Withings Pulse) for fitness tracking accuracy and ease of use, addressing challenges and solutions in comprehensive health monitoring.

Following the thesis purpose, many apps have entered the marketplace and are accessible to the general public on ophthalmology. The development of tele optometry holds innovative promise for the future; Its purpose is to prevent, diagnose, and treat in time, to provide timely attention to any kind of visual disability (30). In the review cited, the following applications have been studied by the three differentiated purposes.

Regarding the prevention of diseases, numerous applications have been found in the market mainly related to facilitating the interaction of patients with eye problems and visual disabilities with different electronic devices such as computers, mobile phones, and tablets to facilitate the performance of tasks such as reading messages, learning processes, or simply making calls. Furthermore, in these same applications and others, services have been found, such as primary care, information on ocular pathologies related to systematic diseases such as diabetes, hypertension, or thyroid. Numerous apps have also been found related to educating patients

about preventive eye care and correction systems, such as contact lenses, to prevent or help monitor and evaluate ocular pathologies.

Related to diagnosis, the apps that can be found on the marketplaces and or take part in clinical studies, focus on acuity, perception, and the visual field. These applications seek to clearly and precisely analyze and identify the ocular problem or pathology that must be treated. Examples of applications are stated in the next paragraphs that were reviewed on the literature.

SmartOptometry and Peak Acuity are two apps to measure visual acuity without the need for a clinician or a consultancy room or neither clinical knowledge. Both apps includes several tests to be done by the patient in an easy way that offers acceptable correct visual acuity estimation results. The methodological study to furthermore validate these 2 applications concluded that the use of these applications were feasible and accurate to estimate near and distance visual acuity although more extended clinical studies involving more patients will be needed to validate the results (31).

VisionApp is an app for telemedicine that provides their users with booking an ophthalmic teleconsultation but also provides through diverse tests the best corrected visual acuity. This application self-testing has been studied for validation as it proposes an innovative way to test the best corrected visual acuity through smartphone and it is intended for a pediatric use. For the results of 30 participants the study concluded that the results obtained with the proposed method agrees with the classic clinical measurements but further validation studies were needed to propose this innovative telemedicine solution to the children. (32)

Applications in the field of ophthalmology to treat patients, in the literature, 3 types have been found: firstly, a digital tool designed to support users of contact lenses in adapting, controlling, and monitoring; secondly, apps aimed at enhancing patients' comprehension of cataract surgery procedures and improving their satisfaction; and thirdly, apps intended to oversee the treatment progress of individuals with glaucoma.

The success of applications in the ophthalmic field, focusing on the patient, is intrinsically related to a good eye care routine and can be translated to the mobile app domain to generate a promotional approach in self-care activities based on visual health, with visual screening, detection of visual alterations and ocular pathologies. The final purpose of mobile phone technology in medicine for the clinician is to be able to assist in real time, personalizing medical care options and monitoring the progress of the disease, to expand patient knowledge and support physician-patient communication, which contributes to improving treatment (30).

2.2.1. State of the art

In the optometry field, more related to the final aim of the work of the developed thesis, 12 apps will be summarized to make an idea of the current state of the art. These apps are the most used in the optometry and ophthalmology fields.

The Eye Handbook app offers different features for ophthalmologists and eye care workers. The features proposed go from educational videos for patients, evaluative tools to make eye tests, reference, and educational tools for physicians and optometrists' calculators, and other other-office-based tools. The features proposed are immense from near vision cards and color vision plates, passing by a huge dataset including eye diagrams and eye movies among high-resolution images to illustrate anatomy and all different pathologies to patients, translation of commonly used ophthalmic terminology, a list of diagnoses as a color-coded diagram of retinal drawings, questionnaires for commonly encountered ophthalmic diseases, and a summary of benchmark studies in ophthalmology, dry eye calculators, ocular trauma score calculator, age to bifocal add among others. A forum is also available where all the users (physicians, optometrists, and ophthalmologists) can interact with patients' cases and media features are present to learn about the latest techniques used in the field. At the moment the application has been downloaded more than 4.18 million times and already accounts for more than 60.18 thousand registered users with more than 22 thousand average active users per month, and 1 million total unique user devices. The app entered the marketplace in 2009 and was developed by Cloud Nine Development (33–35).

Eyedock(36) is a user-friendly design app designed by optometrists for optometrists. It's a resource with all the information related to eye care. It includes data related to treatments for ocular disease, contact lens brands or drugs, and tools to calculate CL (Contact lens) power. it is regularly updated to help the optometrist find information. To unlock all the features proposed, a monthly membership payment is required. The additional features are advanced contact lens parameter search, rigid gas permeable and specialty contact lens database with lens detail and fitting guides available, additional ocular conditions database where are available 200 ocular conditions with symptoms, signs, differentials, workups, treatments among other information, ICD-10 codes database explained including diabetic retinopathy coding widget and finally advanced tools and calculators as oblique cross cylinders, Parks 3-step utility, Plaquenil Risk, and Meridian power calculators. The app was released on 11 February 2010 and was developed by Todd M. Zarwell OD, FAAO.

By the same developer as Eyedock, the Eyescholar(37,38) web app a tutorial for Optometrists doctors reached the internet on May 29, 2020. With a yearly subscription fee, you can access a tool that helps optometrist practitioners enhance their skills in refractive error measurement with refracting instruments and retinoscopy with trial lenses without the need for the instrument or the patients.

In addition to the Eyescholar and the Eyedock Apps Todd M. Zarwell OD, FAAO also developed a very used application called the Parks Three-Step Test(39) (also known as the Parks-Bielschowsky Three Step Test as well as the Parks-Helveston 3 Step test). It is used in the ophthalmic field to isolate a paretic extraocular muscle in acquired vertical diplopia or double vision. It is not effective for use in assessing horizontal diplopia and it is not reliable in patients with prior strabismus surgery. The test is more reliable with recent onset diplopia. Because of the spread of concomitance, eventually, the muscle actions that the test relies on become more difficult to isolate. By calculating the vertical deviation in the primary gaze, left and right gaze, and right and left head tilt, Park's Three-Step Test can eliminate each muscle's primary field of action. Eventually, the affected muscle will be the only one remaining. It was released on April 29, 2014.

OpTranslate is a mobile phone app that helps optometrists conduct eye exams on non-English speaking patients. Developed by ocular media, it has translation in 12 languages including Hindi, Mandarin, and Farsi among others to help train eye care professionals into competent multi-lingual practitioners, expanding their range of practice and enhancing the doctor-patient relationship through improved communication. Its final purpose is to train practitioners to extend their knowledge without the limitation barriers of languages. The app includes a full comprehensive eye exam, contact lens exams, binocular vision testing, and pre-written diagnoses. It has also an educating the patient module that includes helpful written explanations and diagrams to enhance the patient's knowledge of the problem suffering the patient to get a better result on the communication behavior between eye care professionals and patients. Furthermore, an Educating the Doctor module to enhance the language of the clinician on foreign language terms with quizzes and tests to enhance the doctor's capabilities and communication with patients who only speak a foreign language. Finally, a podcast in French and Spanish to accompany the written lessons in this translation guide is also included. At this moment the application is free for usage and doesn’t need any payment to unlock the full features provided. It links to additional tools as Google Translate to help patients and optometrists. IT was released on July 25 2018 and it was developed by Dmitry Richter (40).

EPOCRATES is a web service and mobile application used by healthcare professionals. First, started as a pharmacology reference but furthermore, expanded to include a variety of medical reference materials and tools such as calculators from reference tables. It is also used as a clinical decision-support system because it has features such as history and exam findings, diagnostic confirmation and treatments, and many references and calculators. The final purpose differs if used as a free or paid tool. The free access version provides reliable, comprehensive, and easily navigable service for looking up information about medications uses, dosing contradictions, reactions, interactions, pregnancy, pharmacology, and more. For the professional version, decision support tool, including disease information, support for alternative medicines, infectious disease treatment and laboratory panels, and normal values. The information is accessible through the search taskbar and appears as bullet points or other shorthand. The information is displayed on three or four pages depending on if it is used as a free or professional app. The app was released on January 14 2009 and it was developed by Epocrates (41).

GoodRx is an App for healthcare professionals that promotes awareness, access, and adherence. It helps the clinician to compare savings across pharmacies, classes, and manufacturer copay cards, confirm brand drug is covered by patient insurance, and help the patient save without being part of the brand advertisement. It proposes patients discover affordable alternatives by easily comparing brand and generic options, along with other drugs in the same class. Its purpose is to help the patients with the financial aspect, proposing to the clinician a tool to check the insurance coverage of the patient and to share coupons with discounts on the drugs needed, and easily compare brands and generic medication among pharmacies in an area through zip code. The final goal is a way for consumers — insured or not — to sort through our confusing, frustrating, and expensive healthcare system. And find the treatment they need at a price they can afford. To access the app a paid membership is required. The app was released on August 10, 2016, and was developed by GoodRx (42,43).

OCTaVIA(44,45) (Optical Coherence Tomography Visual Atlas) is a collection of OCT (Optical Coherence Tomography) images and reference information for multiple ocular diseases and retinal conditions. Its purpose is to help the clinician to diagnose 100 different ocular diseases from the use of OCT images. It includes a dataset of OCT reference images to aid in patient diagnosis and care. The images can often reveal retinal disease conditions before the patients become symptomatic. Additionally, information related to disease descriptors of OCT images is available and links to other additional information are available. The app

focuses on retinal diseases. The app targets optometry and ophthalmology students and practitioners to easily get OCT images with specific diseases to train and enhance the diagnosis stage. The date of release was October 31, 2018, and it was developed by Elena Biffi, OD, MSc, FAAO.

ICD10 Consult is an app that proposes an easy way to understand electronic medical records without the need for extensive research. Includes more than 10,000 acronyms and synonyms. The features available are:

- Deep code analysis allows easy handling of laterality, encounter, and staging codes, letting the app do the work of finding the perfect code
- Add your custom notes to any code which will appear seamlessly within search results
- Tens of thousands of medical acronyms and synonyms are transparently cross-referenced to let you search using the terms you are used to; e.g. "MI" to "Myocardial Infarction"
- Search term flexibility understands that, for example, 'vertebrae' matches 'vertebra' and 'ovarian' matches 'ovary'
- ICD-10 tabular and alphabetical indexes integrated and simultaneously accessed to maximize results
- Instant linkage within code descriptions
- Coding guidelines as well as section and chapter instructions tightly integrated
- Know a glance if a code is valid for reimbursement or if more specificity is needed
- Browse diagnoses by chapter, section, and subsection; search within specific hierarchies
- Favorites list allows quick access to your most common codes, with groups to organize them
- Convert ICD9 codes to ICD10 by name or by code number

The app was released on March 4, 2010, and was developed by Evan Schoenberg.

9 Gaze (46) is an easy, quick, and efficient way to document eye motility and strabismus in the 9 cardinal positions of gaze. The app provides an on-screen guide for eye positioning to assist with image consistency through all 9 photographs. 9Gaze app automatically creates a composite image of the 9 photographs for easy exporting. Its purpose is to assist the clinician

in the diagnosis step by providing a tool to document eye motility and strabismus in the 9 cardinal positions of gaze. Related features are:

- Ability to use the selfie cam (iOS (iPhone Operating System) only)
- Formatted for "notched" iPhones (iOS only)
- On-screen guide for placement of eyes
- Automatic creation of the composite image of all 9 positions of gaze
- Ability to skip gaze directions
- Re-take any picture
- Turn the flash on or off
- Landscape and portrait modes
- Ability to document name, medical record, date of birth, and date of composite
- iPad and Android compatible

It was released on March 20, 2017, and was developed by See Vision, LLC.

Eyetube (47) is a video-sharing ophthalmology Platform where you can find high-quality ophthalmic surgical videos, educational series, spotlights, etc. Eyetube's mission is to foster an engaging physician community across all of the eye care, powered by user-generated content. Its feature is an organized video-sharing platform where it is possible to connect with a global community to advance patient care and promote innovation. More than 135,000 members around the world watch 2,500 videos on Eyetube every day. It was released in March 2008 and was developed by Fyra Digital.

Feyenally (48) is a camera-based eye test for remote eye examination app. MDR (Medical Device Regulation) and FDA (Food and Drug Administration) regulations certification has been submitted for further acceptance. Its purpose is to obtain the eye refractive error of patients remotely and automatically through a mobile phone app to make this task less resource-demanding to reach the population that can't afford professional eye care services. Features and solutions included in the app are eye refractive error verification with a mobile device. It is a unique combination of objective methods used to define the refractive error of the eyes. The test takes just a few minutes and requires nothing more than your mobile device so, there is no need to visit the specialists in public places while taking the test. It was released on January 4, 2024, and developed by Gepetto Sp. z o.o.

Not far from describing the most important apps in the market for optometry, an important study of visual acuity apps found in the ios market in 2021 concluded that neither of the current

applications were able to be used in a telemedicine purpose even if some of the applications were accurate and reliable to many different mobile versions(49). The fact is that the technology needed cannot be still reached as the amount of pixels displayed on the screen is not enough to develop a test able to produce a low average mean error at small distances (35-50 cm). The only application that was able to produce an accurate result needed a distance of 4 meters between the mobile phone and the user. Additionally, most of the applications had randomized letters as targets for their tests and this will not make possible to corroborate the results by the clinician in a teleconsultation.

In summary, despite the quick rise of eye care apps, there's a noticeable lack of professional engagement in their development and a deficiency of peer review post-publishing. This underscores the necessity for adopting evidence-based guidelines and standards in app creation within this evolving field of modern eye care, aiming to enhance the quality of future mobile health applications (50).

2.3 Digital Image processing

In 1920 the first image was sent through a transatlantic wire using telegraph codes. In the first stage the image only had 5 levels of grey that 1929 they were able to improve to 15 levels. The use of the transatlantic wire reduced the time to of travel of an image between the two continents basically, to be able to publish news with pictures in a faster way.

The first digital image is released in 1957, is a picture of a baby with 176 pixels in one side. The first manipulation of an image is related to 1964 when the reception of images from the Ranger 7 spacecraft from the Ranger Program that successfully transmitted to earth the first pictures of the moon. Due to the system used for image transmission, 3 different important image correction were performed with the use of computers. The first one was to solve the distortion caused by the differences of image scanning of the vidicon in the spacecraft and the cathode ray-tube used to reconstruct the image. The second one was due to the irregular response of the cathode ray-tube and the third one due to interferences of the TV signal caused by the spacecraft electronic systems.

Figure 2-3: The first digital image © Walden Kirsch, CC BY-NC-SA 2.0

Three different type of image processing exists and solved different types of problems during the late XX century. In 1990, Hubble telescope was launched to stratosphere to perform different missions that were related to expand the knowledge about space bodies such as galaxies and pluton planet. As first, it was found that the lens suffered from spherical aberrations but indeed the quality of the image obtained was slightly better than the best planetary telescope. The images tried to be enhanced with computers (digital processing) but in the 1993, optical processing was performed reaching the telescope and correcting the aberrations of the lenses. The last processing technique is the analogic, and this one is better understood as the different options for enhancing the quality of the images of TV changing the contrast or the luminosity intensity from the TV menu and performed with electronic devices.

In fact, a digital image is the one consisting of a two- or three-dimensional matrix array that its values are discrete. A sample is an image consisting of a matrix array with continuous values that correspond to the integral of the radiation during the exposure time of the electromagnetic sensor for example, an image from a television. A very good example of an optic image correction is the use of current lenses or contact lenses to enhance the visual quality.

Digital image processing considering the definition stated on the paragraph below, the analysis or processing of an image is the numerical representation, actions and operations with the numerical information to obtain a transformed image. The elements of the numerical distribution of a digital image are called pixels and the values usually are related to the intensity of gray level.

Digital image processing is a cross discipline in nature. Knowledge needed to process images are vision science, computational science and numerical computation among others and a minimum information related to the domain is going to be applied. Different fields are used to

produce valuable information from digital image analysis that goes from medicine and its different branches through automative and robotics, military defense and any other field were images are involved and valuable information could be extracted. In addition, digital image storage in computers are done in a certain way to enhance the efficiency conserving the amount of information the discrete image is proposing. To understand better the image and video formats intrinsically connected to their properties, the storage system needs to be explained.

As we already know, the image display can be analogic or digital depending if the brightness value obtained from the sensor has been discretized or displayed based on a 3 2-dimensional function continuous to continuous. When the values are discretized for computational analysis, the discretization in grey level (most common) is done according to 256 values that is the number of levels that can be labeled with 8 bits (or 1 byte) that goes from black 0000 0000 to white 1111 1111. It is known that gray level images are 8 bits deep. With this explanation we can relate the computer science to the image processing as it is very known nowadays that capacity and storage of different devices are measured in megabytes, gigabytes and terabytes. An example of storage devices nowadays most used are the computer hard disks that typically have hundreds of gigabytes.

The image processing follows the next flowchart where the steps are clearly differentiated in the next subchapters. We have to consider that this process includes a collection of hardware (equipment) and software (computer programs).

2.3.1. Image acquisition system

It can be different types of systems depending on the sensors used and type of rays captured. Most common image acquisition systems are related to electromagnetic spectrum and captures visible lights as photography. Radiofrequency, microwaves, infrared wavelengths, X-rays and gamma rays produce respectively magnetic resonance imaging, radar imaging, thermography, medical or astronomical or industrial imaging and nuclear medical or astronomical observations. Additionally acoustic waves can be captured for geological exploration or echography depending on if it is high or low frequency sounds. Electrons can also be captured for microscopy and synthetic images nowadays can be created using Generative Adversarial Networks a machine learning framework.

2.3.2. Image storage

This is possible after converting the continuous image using an analog-to-digital converter (ADC) to store the digitalized image. The image can be stored temporarily on the RAM (Random Access Memory) of the computer or permanently on the different optic storage devices. Different software available in the market can organize the image set in an efficient way in mobile and desktop devices.

2.3.3. Image analysis

In this step, once the image is obtained and stored, processing software is needed to manipulate the image. Examples of image processing software are OpenCV, Lightroom, Matlab, ImageJ, Photoshop, GPUImage, Pytorch, Keras among others. The software selection is considerably related to the hardware characteristics available for this purpose. The use of mobile phone for image processing is different from the use of a Personal Computer with a dedicated graphic card, as the computational resources available are higher in that case.

The optimum choice hardware - software is intrinsically correlated with the final analysis purposes, as the different techniques for image transformation require different equipment and software. To understand better the needs, it is very important to know the objectives and the capabilities of the resources available. There are five different classes of image processing techniques that are image enhancement, image restoration, image analysis, image compression and image synthesis. Each one of these classes will be developed in the next paragraphs with examples of different techniques.

2.3.3.1. Image enhancement

Image enhancement in other words is to make the image look better for the observer or to be in better shape for further image processing techniques as for example image feature extraction. The types of image enhancements mostly used are contrast adjustments, histogram equalization, denoising, image filtering and decorrelation (to enhance color differences).

2.3.3.2. Image restoration

In comparison to imagen enhancement, image restoration uses probably the same techniques but for an objective purpose, to eliminate and correct all the different degradation suffered through the digitalization process. It includes very important phenomenon called noise. The noise in an image is a variation of brightness or color information. The identification of the noise type and the cause of the noise can be case of study. In general, the most common

differentiation of noise studied are the spatial domain noise model and the frequency domain noise model with the exception of the sinusoidal noise functions.

The spatial noise are random numbers characterized in general by a probability density functions or similar and by the cumulative distribution functions. Specific functions in different image processing software are used to simulate noise but in general, with the application of random functions to generate random matrices to assign values for some pixels of the entire or part of the digital image to corrupt it are the most common technique. For example, assigning a value of 0 (dark pixel) to some random pixel of the image is called pepper noise and if we assign the value 1, it is known as salt noise. An image containing white pixels in dark areas and black pixels in bright areas is called salt and pepper noise and it is due to analog-to digital converters error.

Gaussian noise is another spatial domain noise that arises during the acquisition of the digital image. Here the noise follows a gaussian distribution and it is due to the sensor used for the acquisition. The principal causes are the interferences produced by electronic circuits and the temperature. Shot noise is due to photons reception variation due to sensor characteristics that varies at a level of light intensity. Compared to the gaussian noise shot noise follows a Poisson distribution. Lastly the mapping to convert a sample into a digital image is called quantization and produces uniform noise.

In the frequency domain and spatially dependent noise, a typical example is the periodic noise. This is found when the image is captured with electrical interferences. It is spatially dependent because we can observe the results of the noise applied to the entire image following a spatial pattern. As this noise repeats in the same way and equally spaced pixels, in the frequency domain after the application of the Fourier transforms to histogram of intensity levels, the visualization of the noise can be characterized as a spike. The periodic noise is usually visible in the image and can be corrected if it is characterized (peaks detection) in the frequency domain function or if there is enough knowledge about the frequency of the interference.

To restore the image and reduce the noise effects, for the spatial domain noises, there are different filters to apply and can be classified as linear, non-linear and order-statistic filters. Average, median, maximum and minimum, erosion and dilatation filters are an example of spatial denoising. In addition, adaptative spatial denoising, a more complex filtering resulting in a better outcome, can reduce noise effectively and the output of the filter changes not only due to the transfer function (average, median, maximum value) but also depending on the

parameters set as a threshold (the variance of noise distribution) or a change in the window size depending on the value of the reference pixel in comparison of the median, minimum and maximum value.

For the images affected by periodic noise, three types of filters using frequency domain are used (band reject, band pass and notch). As the noise can be characterized as a peak in the frequency domain using for example, the Fourier Transform function, the objective in this case is to isolate periodic interference. The most used is the notch filter and it consists of modifying the Fourier transform of an image and then computing the inverse transform to obtain the processed result in the spatial domain. To understand the effect of the notch filter first it is important to know that low frequencies in the transform function are related to slow varying intensity components in an image while high frequencies are caused by sharp transitions in intensity, such as edges and noise. In general, a low-pass filter will attenuate high frequencies while passing low frequencies resulting in an image blur. In contrary a high-pass filter will enhance sharp details but cause a reduction in the contrast. In this way notch filters transfer function is similar to high-pass filter transfer function whose centers have been translated to the centers of the notches, meaning that a rejection of the periodic noise frequency (51).

2.3.3.3. Image analysis

The principal purpose of image analysis is to measure objects inside an image. To perform the process, generally, the first step consists of segmenting the image to isolate to object to measure furthermore. The principal techniques for segmenting and feature extraction are edge detection, thresholding and color image processing.

Edge detections techniques (52,53) are based on the pixel intensities discontinuation detection using first and second order derivatives to separate objects on an image that meet this requirement. For the first derivatives a threshold is needed to locate all the important gradients of intensities that are more significant than the threshold given. For the second derivative, the edges are detected when the value of the second derivative has a zero crossing. From the edges detected, geometrical forms can be located using the Hough transform function. The purpose of its use is to link edges that can be discontinued by the noise available in the image and to detect geometrical features like circles and rectangles to extract features from images. An example of Hough transform use, is to extract the iris from an eye image using ellipse or circular Hough transform to find the best candidate to segment the image (54).

Thresholding (55) is a technique used to select intensity values from the histogram of an image (pixel intensity versus the number of pixels available in the image). The identification of separated valleys in a threshold signifies that there are 2 regions with different intensity values that will probably segment the image into two different objects (generally separating the object of study for its background). When the histogram doesn't have a bimodal with a wide and deep valley, optimized thresholding can be used to segment the image taking the histogram as an input. The Otsu method (56), optimize the threshold considering the value that maximizes the between-class variance. This value is obtained using the 0 and first cumulative function.

Adaptative thresholding (57) is a method that consists of the selection of a dynamic threshold that it value will be different depending on the neighbor pixels. The adaptative thresholding is used considering that a small region is less likely to suffer from non-uniformity of light intensities or shadowing that will alter the results of the final thresholding misclassifying the object region from the background.

The use of the color space is also possible for image segmentation to separate the different features from an image that their main characteristic differentiation can be from the color channel (58).

Additionally, to region segmentation can also be done using K-Means clustering, a machine learning technique. This technique also can be used for color image processing segmentation. In the following machine learning chapter, a more extensive explanation of the algorithm is developed.

Gabor filter (59) is used for image segmentation. In the image restoration subchapter, we studied spatial and frequency domain denoising filters. For the segmentation purpose, we highlight the gaussian kernel and the sinusoidal kernel that are part of the output Gabor filter function.

2.3.3.4. Image compression

In this step, the purpose is to optimize the memory required for image storage. The images tend to have redundant information and are more likely to be sent electronically from a device to another one. To have an idea of the memory needed a 512x512 pixels image is the same size as a 40 pages text document.

To optimize the amount of data required to represent an image, it is important to understand that the most used technique to represent digital image by given a value of intensity of each

pixel of the image is not optimal for storage. The redundancy of information found in this system has the following types: coding, spatial and temporal, and irrelevant information. To store using computational language, the 8-bit codes have more bites than the needed to represent the different intensities. The correlation between the different pixels intensities spatially located and between previous and posterior frames (temporal domain), can be used to simplify the image storage and eliminate redundant pixel information. The use of information of images that are not detected by the human eye visual system can be eliminated.

Considering the different types of redundant information, different coding are available to reduce the image storage needs by eliminating information or optimizing the storage by simplifying or reducing the data. Two different compression types are loss less and lossy predictive coding.

The Huffman coding are the most used techniques nowadays by software developers because it is optimized considering the probability of use of certain symbol (intensity value or intensity mapping) and the number of bits needed to code. This is done by creating a map of the probabilities of use of each for example, pixel intensity value, and the code length of each symbol that is translated furthermore to 0-255 intensity value will have a number of bits inverse proportional to its appearance probability. This results generally in a final compression rate above one without loss of information, which leads to obtaining the same image quality.

2.3.3.5. Image synthesis

The image synthesis is the creation of new images from the information available of analyzed image or the data obtained from the analysis. Image synthesis is a more complex concepts and it involves some or all the different steps seen before it is basically the obtention of knowledge from the extracted features from one or different images, the measurement of an object from the image obtained or the classification of all the images available in a set further feature recognition. One of the most important image synthesis models used nowadays is the obtention of information for a machine learning model to predict and classify the objects available in the images. In the next chapter, the different machine learning models will be explored and the relation between image synthesis and the purposes of machine learning will be highlighted.

2.3.4. Human eye visual system and artificial vision

To understand the image processing, the human eye vision system should be explained. The final purpose is to achieve automatically, using computers, an analysis of images to obtain the same knowledge and information as humans do with the observation of the world.

The human eye vision system depends on the eyes in the same way as the brain. The eyes receive light rays and basically transform patterns into images. The light rays bounce on an object that reflects the light on the eyes. The light ray enters through the eye by the cornea and goes through the pupil that its main function is to let the correct amount of light intensity. Behind the pupil is located the lens of the eye to focus the light on the eye that helps to see the object as a clear sharp image. On the back of the eye, in the retina, there are tons of tiny cells around (130 000 000) that are sensitive to light that turns the light received into electrical signals. The optic nerve function is to transport the electrical signal to the brain. In the retina, a rough image is formed upside down of each eye, but the brain turns both slightly different images to the correct position and combines them into a unique image. The brain additionally adds a lot of details to observe complex shapes, movements, depth and many colors. In addition, the recognition of the object is performed in the brain.

Having considered how the human visual eye system works, the image analysis can be divided into 3 types of processes. In this sense, the division in levels of artificial vision can be also a clear understanding of the deepness of the process. In the first low level vision class, it includes all the transformations related to the acquisition software and hardware used to transform the object into a digital image and prepare it (pre-processing) for further transformations. The second level, corresponding to the intermediate vision class, includes all the functions and transformations for the segmentation purposes. In this step we look forward to separating the different textures, color or forms. It is usually the second step processing for furthermore interpretation or automatic detection of specific objects corresponding to the high-level vision. The last class is the nearest one to human vision, as the algorithms can interpret the object requested.

2.3.5. Display the image

Finally, the transformed image needs to be displayed generally on a computer or television. In this step the image is again transformed to continuous analog signal intensity, and it is possible due to the use of the ADC for video display.

2.4 Machine Learning

Machine learning is commonly used nowadays for the purpose of predicting and clustering data to obtain additional information to make decisions. It is used in the medical field to diagnose stock trading to predict the market fluctuation, in the power plants to regulate the

energy load based on the electricity consumption and for weather forecasts among others. The main objective of machine learning is to learn from experience to recreate a model to predict the future in an accurate and precise way. To obtain a model with the characteristics stated, it is very important to have a huge and varied dataset. The model will improve as the information given is more important and increases regularly to predict the future tendencies.

The use of machine learning with the raise of big data, has been developed in almost every field from computer finance to image processing passing by manufacturing, transportation, agriculture and natural language processing among others. This is because there is no need of a function or formula to explain the phenomenon to be studied at it is based on the experience, in other words, in the availability of data. The information available for the study of phenomena usually is divided onto 2 groups or types, the prediction variables and the response data or in other words, the information that can be measured or determined of a phenomenon and the resulting or class expecting that will result from the object of study. If only information related to the prediction variable and the class or group is not known, the model is classified as non-supervised machine learning. If the variables from the experience can be classified or continuous variable response can be obtained, for example, the prediction of the age of a patient from the analyzed face feature (previous data of face features “predictive variables” and age of the patient “response variable” are already known), the model is classified as supervised learning. The supervised learning models depend on the characteristics of the response variables. If the response expected by the model is a class, in other words, the variable is discrete, for example, instead of predicting the age of the user we predict the generation where he belongs, it is known as classification model. If the exact age of the user is giving in number of years, it is known as a regression model.

The choice of the use of machine learning algorithms is intrinsically related to the amount of information available for the study but also if the variation on the response doesn’t follow any function or scientifical model, or if all the variables are not available and the error rate can be neglected and won’t affect the system. Another important case to consider machine learning is when the phenomenon is constantly changing and there is a need to consider a variability if the response while using more recent data and in general the significant variable of interest is not already recognized.

The first case, and an important accomplishment related to the understanding of epidemiology through the information obtained from data was accomplished in 1854 by a prestigious doctor

from London that helped to determine the cause of cholera. The use of maps to account the deaths caused as a consequence of cholera in the south of London was used principally to find the deaths cause (60). It was furthermore found that in the North of London there were no deaths because the water was not contaminated and the popular belief that the transmission was airborne was found erroneous. Nowadays numerous authors study the data using artificial intelligence to create models

From Taiwan covid-19 consequences were less dramatic for their citizens among there are very near to the city of Wuhan were the epidemic started. With their past experience of SARS (Severe Acute Respiratory Syndrome) virus happened on the 2003 in China, they were prepared to give a quick and accurate response to restrict the contamination. We must highlight that population of Taiwan is somehow 27 million and almost 5% lives and works in China. Additionally, the pandemic started near a holiday period (new lunar year on February 5 2020) were it was expected that most people travels from and towards Taiwan for a short period of time. The creation of a new institution related to public health on 2004 after the pandemic of SARS, allowed a quick reaction that was translated into a rapid sharing of information between the immigration custom and the public health insurance to create big data analytics to identify possible cases, instore a plan for quarantine to reduce and prevent the spreading. The set up of emergency protocols and specially for data information sharing is crucial to allow that the concerned teams for prevention and diagnosis of patient will be able to proceed with the last updated knowledge (obtained from current time data analysis using big data techniques and machine learning models) to perform the best measures. This was the case for Taiwan where it was expected that this island will be one of the highly affected regions due to it closeness to the main source of covid-19 and being one of the nearest and preferred vacation region by Chinese on the holidays, in fact didn't suffer the dramatic consequences as it was expected. The fast response, before most countries on the world considering that the World Health Organization was informed of the first cases of "pneumonia of unknown cause" on 31 December 2019, in Taiwan they already started to monitor the patient coming from Wuhan. On 5 January, travelers having visited Wuhan in the past 14 days and had symptoms were quarantined and tested for 26 different viruses, including the Middle East Respiratory Syndrome and SARS, on 10 January, it was found that the virus was related to these respiratory syndromes. In the same case, in another study completely different but related to the data and public health is the importance of rapid share of new genome discovers to enhance the possibilities of stopping the spread and to allow patients to recover from ebola (61).

2.4.1. From raw data to model training

The main challenges found in big data analysis are the amount of data available, the different variations of types of data, and the speed of the data transmission are the main concern.

Handling big volumes of data can be challenging, as nowadays, a computer has a limited storage source. The data is usually stored in the cloud, which is split into parts and stored in different platforms so that servers can be located in various countries. Nowadays, there are small regions dedicated only to the storage of big data, where they have buildings of servers and also the primary system to perform all the conditioning to make the servers work adequately.

The different variations of data available in the market can cause the system to work partially if it can analyze the videos, the signal, and the dictionaries in the same way or to extract the information needed as data from the signals and the videos. As we saw in the previous chapter, the improved knowledge of digital image processing will enable us to analyze the videos and synthesize the information needed. This step of adapting all the information to the model for analysis is known as preprocessing, and it can be from the image but also related to the signals it was explained in the previous chapters. The obtention of the localization of signal peaks that can have immense importance can be stored as an array, and the signal synthesis would have been performed. The knowledge obtained can be stored in a structured database.

Speed has been enhanced with new technologies related to 5g and optical fiber for internet connectivity. When a large amount of data is stored on different servers, speed is important when the model needs actual real-time data to improve the prediction and obtain an accurate result.

In summary, the complexity related to the three different concepts or challenges found in machine learning, the selection of the model to consider the different types of data, the amount of data to be analyzed, and if it will be automatically retrained to consider the last data available can be the main challenge for machine learning analysis.

2.4.2. Types of machine learning models

Different types already exist depending on the data available and the preprocessing techniques used to adapt the information and knowledge stored in the database and on the purposes that we intend to reach with the implementation of the model. They are famous and present in the literature. Three types of machine learning models can be differentiated considering the

purposes and output: Classification, regression, and clustering. We must assume that the first corresponds to supervised machine learning models that need to be trained to obtain the desired results on the same structured database but with new data or, as we will state, the data test. The clustering models correspond to non-supervised machine learning, and this is because there is no need for a response variable. After all, the final objective is to find tendencies or clusters in the data, meaning that we can classify them into different groups depending only on the input variables.

2.4.3. Unsupervised Learning Techniques

Two different types of models are available: hard and soft clustering. The hard clustering will separate all the entries (each entry is composed of a set of different predictive variables) into different groups depending on the value of their predictive variables. In the case of soft clustering, instead of separating the entries, a value will be given to each one of the entries of the probability of being part of one or another group, making the separation less clear and having the possibility of input to be part of 2 different classification groups.

2.4.3.1. Hard clustering models

In hard clustering, we can differentiate four types of typical models: k-means, K-Medoids, Hierarchical Clustering, and Self Organizing Map.

2.4.3.1.1. K-Means clustering

K-means clustering is done in the basis of distance in between samples. This means that if we have an available set with different samples, clustering will be done to calculate the distances. Each sample will be assigned to the category in which the mean value is closest. Meaning that n-samples belonging to the same cluster have a mean value that will be the minimum total distance to all the samples.

This model is used when we have available a large dataset and the number of clusters we want to group is already known.

2.4.3.1.2. K-medoid clustering

This method is very similar to the k-means clustering but instead the mean value that will be the center of the cluster it is always a sample (usually the closest sample to the center of the cluster calculated with the k-means). The main difference between the k-medoid and k-mean is the center of each cluster will always be a sample of the data in the k-medoid, but in the k-means technique, the center of a cluster will be the exact center of the cluster (mean value).

It is mostly used when the number of clusters is known, and a large computational resource is not needed as the model is fast enough to scale large data sets. It is also recommended for categorical data, as the center of a cluster will be a category.

2.4.3.1.3. Hierarchical clustering

This method consists of creating a dendrogram and making binary decisions each time the data can be separated into two groups. The final decision to make different clusters will depend on the user and will separate the dataset in the number of clusters considered optimal. This technique is very useful when there is no information about the number of clusters before the analysis, and in the same way, it proposes a visualization guide to determine the number of clusters. Generally, considering a small number of clusters will result in a more general model, while considering a higher number of clusters makes the model more specific.

2.4.3.1.4. Self-organizing map

With this model, it is possible to visualize high-dimensional data (more than three variables) in 2D or 3D. These models allow the topology to be preserved and the data's dimensionality to be deduced. The output is similar to a honeycomb representation in 2D, showing the shape of the dataset.

2.4.3.2. Soft clustering models

An example of soft clustering is generally used to analyze gene expression data from microarrays to understand the genes better. It is commonly used to reveal structures hidden in large gene expression data sets, as one gene can be involved in several biological processes. An example is using soft clustering to study the genes involved in diabetes 2. The model clustered variant-trait associations from publicly available genome-wide association studies for 94 known T2D variants and 47 diabetes-related traits and 5 clusters of T2D loci differentially enriched for relevant tissue-specific enhancers and promoters (62).

2.4.3.2.1. Fuzzy c-means

Partition-based clustering is used when data points tend to belong to more than one cluster. This is especially useful when clusters overlap. It's actually used when the number of clusters is already known for pattern recognition.

2.4.3.2.2. Gaussian Mixture Model

The Gaussian mixture model helps predict the clusters softly, giving each sample a probability to belong to each cluster. In this model, each cluster is treated as a Gaussian distribution, being

the samples in the center have a great probability of being part of the cluster, and the further the sample is to the center, the more likely it is to belong to more than one cluster. This model is used when clusters have different sizes and correlation structures.

2.4.4. Improving Models with dimensionality reduction

Dimensionality reduction consists of reducing the number of predictor variables using two techniques: variable reduction or variable transformation. When the datasets consist of a large set of variables, most of the characteristics will be correlated or intrinsically dependent. A study of variable significance is important to consider in order to reduce the computational resources needed to train and test the classification or regression model. It is considered that the techniques I will discuss are classified as preprocessing, meaning that it provides processing of the raw initial data to obtain an optimal result without additional time-consuming. The three most well-known dimensionality reductions are Principal Component Analysis (PCA), Factor Analysis, and Non Negative Matrix Factorization.

PCA will transform the variables given in a dataset so that the first new principal variable will be the most significant of the data set and will be the principal cause of model variability. The second variable will be the second most important, capturing less variability than the first but more than the third component.

Factor Analysis's objective is to find the correlation between the different predictive variables. Factors will give an idea of the linear relationship between the variables observed being those linear functions of the factors. The variability reduction will be done by representing the factors more representative of each variable, usually much less than the observed variables, and representing almost all the information available in the data set. The Factor Analysis it’s the inversion of the PCA, considering that PCA new variables are created in function of the observed variables. Physically the explanation of factor analysis will have more significance as most of the case the factor will represent a new phenomenon that is affecting the observed variables and can be the reason why some variables are related. PCA in the other hand will give information of which variables are more significant to the variability of the dataset but each observed variable will affect more than one principal component.

Nonnegative Matrix Factorization is usually used when the factors needed must be positive, cause there is a probability that those factors will model physical phenomena or count words in a linguistic analysis. The purpose of the technique is the same as for Factor Analysis but with result of positive factors for interpretation of the analysis done.

The reduction of the data dimensionality and the clustering are most of the time part of the prediction of the response variable, being this one continuous or discrete (numerical or categorical).

2.4.5. Supervised Learning Techniques

The principal purpose of supervised learning is to create a model based on the data available, not only of the observed variables but also of the response obtained in the experience while considering the value of each of the predictive variables.

As stated before, two types of supervised Learning techniques depend on the response variable's source. If the variable is discrete, usually the model classifies the different samples, constituted of the observed values in different categories. Those are known as classification models. In the other case, if the response variable is continuous, the models are known as regression models. An example of a continuous response is the prediction of the power electricity demand at a current time, knowing that it will be different depending on the use at each moment.

The choice of a supervised learning model depends on the type of response variable and the time and resources needed to train the model. This often translates into the computational power required to obtain a high-performing model. The memory needed is also closely related to the available resources, including data and computational capabilities.

Moreover, the model's accuracy when predicting new data sets is crucial; a robust model will yield results similar to those seen during the training phase. Typically, using a larger data set can improve the accuracy of new predictions.

Lastly, the transparency or interpretability of the model is essential and is inherently linked to the type of model used and the knowledge of the database. If the model is based on a recognizable function related to a specific phenomenon, the interpretation of the results is generally clearer.

We can classify the supervised learning model depending on the number of categories. It is important to understand the response expected and to know if it will be a binary classification (63) problem or multiclass. Some of the models in this subchapter will work only for a binary classification problem, but others are more complex and demand more computational resources but give greater results for multivariant categories.

2.4.5.1. Common Classification Algorithms

2.4.5.1.1. Logistic Regression

It is used to predict a binary response, and the data will correspond to a class or, if not, to the other one. It is based on a linear equation when data can be separated using a line. It is mainly used at the beginning of the analysis because of its simplicity and lowest time consumption giving a preliminary classification and an idea of the complexity of the dataset (a baseline method) (63).

2.4.5.1.2. K-Nearest Neighborhood

K-Nearest Neighborhood is done referring to the closest neighbor. This means that if we have an available set with different samples that already we have clustered in various groups or clusters and we add a new sample, considering the mapping of two variables “x” and “y” and “k” as a categorical variable that will account as the group of each one of the variables. The new sample will have the same category, “k,” as the nearest sample in the dataset. The distance between the samples can be calculated using different methods; we highlight Euclidean, cosine and Chebyshev (64).

The explanation given before is straightforward. The model gets more complex when, instead of providing the new sample the same category as the nearest sample, we use more than one neighbor to assign a category to it.

It is used when a simple algorithm is needed to establish benchmark learning rules. It is computationally resource-demanding; it needs a lot of memory, and the processing time is more than that of other models for data training.

2.4.5.1.3. Support Vector Machine

Support Vector Machine (SVM) classifies data by finding (hyperplane) that separates all data points of one class from others. To penalize points on the wrong side of the hyperplane a loss function is used. Kernel transformation on the SVM model is used to transform nonlinearly separable data into higher dimensions where a linear decision boundary can be found (65).

SVM is used for multiclass classification in addition to binary classification, using error-correcting output codes. In addition, when there is no clear linear separation of high-dimensional data, the use of SVM can give an accurate result. The classifier used is simple and easy to interpret.

2.4.5.1.4. Neural Network

The Neural Network model is inspired by the human brain for classification purposes. It has highly connected layers formed by different neurons that connect in various stages from the input to the output. In each neuron of each layer or step, a transformation of the input data is performed. The number of neurons in each layer is related to the predictive variables, and the number of layers gives an idea of the complexity of the model. In general, the neural network is very complex; the model cannot be interpreted and requires higher computational resources. Among its inconveniences, if the data can be constantly added, the model will adapt to the results of the new input, being this model very flexible to changes in the input and predicted outputs. It is very used with great results for modeling high nonlinear systems (66).

2.4.5.1.5. Naïve Bayes

In the Naïve Bayes model, the classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other features. It classifies a new dataset depending on the probability of belonging to a particular class. It is used for small datasets with many features. Due to its probabilistic nature, the classification is easy to interpret, but also, while facing new data that involves scenarios not encountered in the training data, the model prediction won't suffer from more inaccuracies (67).

2.4.5.1.6. Discriminant Analysis.

Discriminant analysis classifies data by considering the combination of linear features and the relationship between classes. Every class is regarded as a Gaussian distribution; in this case, the samples on the border of each class will give information related to the linear combination of features that will be the boundaries between the different classes. The distribution parameters can be linear combinations or quadratic functions used to calculate the boundaries. The model calculates the Gaussian distribution of each class (68,69).

The discriminant analysis is a simple model and easy to interpret. It requires low memory, and it is used to make fast predictions.

2.4.5.1.7. Decision Tree

The model predicts data following a decision tree. Branching the tree means giving a condition where the value of a predictor is compared to a trained weight. In other words, if the predicted value is above a threshold given, the sample is classified in a category, and if the value is less, the sample has another category. The total number of branches gives an idea of all the categories that can have a data set (70).

The decision tree model is easy to interpret and fast to fit. It requires low memory and not high predictive accuracy.

2.4.5.1.8. Bagged and boosted decision trees

This model uses several decision trees weighted to work as a combined decision tree ensemble. Each tree is trained separately with bootstrapped data from the training set to furthermore combine the results by boosting the ensemble, giving different weights to each decision tree to adjust and eliminate misclassified samples (71,72).

This model is used when there are plenty of computational resources and the time consumption for training the model is less important. It is used when the input variables are discrete or behave nonlinearly.

2.4.5.2. Common Regression Algorithms

2.4.5.2.1. Linear Regression

The linear regression model will usually be the first to fit because of its interpretability and low memory consumption in the training stage. It requires minimum time to fit, will give information related to the tendencies of the database, and will be the baseline for more complex regression models (73,74).

The main characteristic of the linear regression model is to describe the linear behavior of the continuous response variable.

2.4.5.2.2. Nonlinear regression model

The nonlinear regression model fits the continuous response variable to a nonlinear function. The nonlinear model is known as parametric because the function supplied needs parameters that the model will calculate (75).

It is used when data has strong nonlinear trends and cannot be fitted into a linear space, or when complex models need to be fitted to the data.

2.4.5.2.3. Gaussian Process Regression Model

It is used to predict the value of a continuous response variable without parameters. It is a non-parametric model used in the field of spatial analysis for example in geology for level boundaries or hydrogeological for the prediction of water paths. It is also used in automation to optimize engine design. It is used when the response variable cannot correctly be fitted to an equation (76).

2.4.5.2.4. SVM regression

This model is very similar to the Support Vector Machine classification model. Instead of finding a hyperplane to classify the data, it finds a model that deviates from the measured data by a value non greater than a small amount, with parameter values that are as small as possible to minimize sensitivity to error (77,78).

It is used for high-dimensional data, which gives many predictor variables.

2.4.5.2.5. Generalized linear model

It is a special case where linear methods fit nonlinear models. It involves fitting a linear combination of the inputs to a nonlinear function. The model gives the linking function of the outputs (79).

It is used when the response variables have nonnormal distributions.

2.4.5.2.6. Regression Tree

This model is very similar to the Decision Tree but adapted for predicting continuous response variables. It is used for nonlinear prediction response (70).

2.4.6. Feature selection and Hyperparameter tuning

One of the techniques already explained to improve the model is dimensionality reduction. As stated in the previous subchapter, this dimensionality reduction can be performed by performing a transformation on the prediction variables or by selecting the most influential ones to reduce the computational resources and complexity of the model and get to the model training step faster.

In addition to the techniques stated before, feature selection can be a step for preprocessing the data to reduce the computational resources needed to make the model work efficiently. The methods used for feature selection are stepwise regression, sequential feature selection, regularization, and Neighbor Component Analysis (NCA).

The stepwise regression consists of adding and removing features in the training model step to obtain a better predictive model accuracy. The model is trained every time with a different number of predictive variables, and the model with optimal predictive accuracy is chosen to test the results.

Sequential feature selection consists of iteratively adding and removing predictor variables and evaluating the effects of each change on the performance model. It differs from stepwise regression because the model is tested with different predictive variables.

Regularization consists of using shrinkage to reduce the weight of the less important variables or the ones that are highly correlated to others to zero.

NCA is a method that provides the weights of each predicted variable for obtaining the predicted response. It discards the less significant variables to obtain a more efficient model that is slightly less accurate but much faster.

The feature selection process is usually done before the variable transformation, but it is seen in this introduction after the supervised model explanation because most of the techniques used require that the model be selected and already trained in a previous step to learn the weight of each variable for predicting the response.

Once the model is selected, the features are selected and transformed, and the enhancement of the model is done by tuning the model's hyperparameters. It is an iterative process that usually needs to train the model several times to understand and find the best optimal hyperparameter values. The three methods used to optimize the tunning hyperparameters are Bayesian optimization (80,81), grid search, and gradient-based optimization.

2.5 Blink detection algorithms, state of the art

Once the cited concepts are studied in the introduction, chapters 3 and 4, an extensive study of algorithms that use image processing techniques and machine learning models has been done, covering all the studies published in indexed journal databases over the past ten years.

Several applications of the blinking algorithms have been found that are different from the purpose of the thesis. Even though the principal features to extract from images and videos don’t differ much, all these techniques will be useful for developing the blink detection algorithm stated in the methods chapter. The applications found in the studies covered over the past ten years can be observed in the Figure 2-4.

Considering that different sciences study the blinking and taking into account all the studies reviewed, a summary of the principal characteristics and type of blinks that can be useful to know and also to characterize the blinks are going to be developed as an introduction to the different types of blinking detection methods.

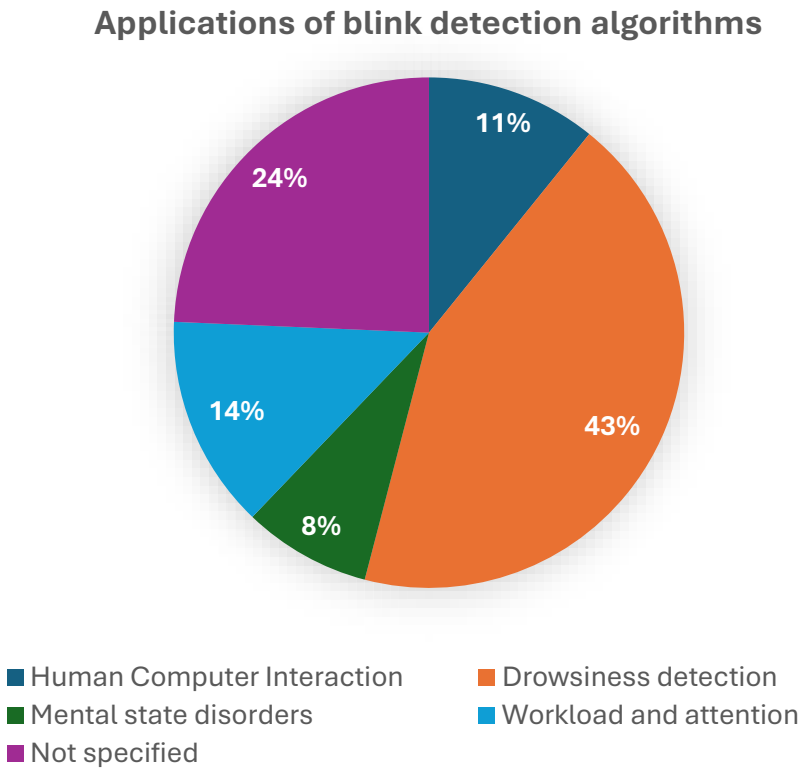

Figure 2-4: Applications for blinking detection algorithms.

2.5.1. What is a blink? Types of blinks and characteristics.

It is essential to understand that the principal functions of blinking are to moisten the surface of the eyeball and remove dirt, allow alteration to take place in the tension of the ocular muscles to eliminate early fatigue, protect the eye from injury, and to protect the eyes from continuous light exposure (82).

Different blinks can be found once the functions of blinks are recognized, depending on the function performed or the occurrence of the blink cause. First, the blinks can be classified as voluntary if the user is in a cognitive state, closes his eyes, and opens them just after, and as involuntary blinks. The involuntary blinks, considering the eyelid functions, can be reflexive if the blinks occur due to a response to protect from an injury or from continuous light exposure or spontaneous blinks where the reason can differ. Still, it is more of a transient movement or unconscious. Additionally, considering the amplitude of the movement, the blinks can be classified as twitch, incomplete, and complete blink. Twitch blink consists of an eyelid closeness that covers less than 30% of the eye cornea. Incomplete blinks consist of the same eyelid movement with cornea covering from 30 to 75%. Complete blinks are considered when the cornea is covered by the eyelids over 75 %. Considering that incomplete blinks usually play

a minor role, consider the primary purposes and functions that the blinking performs and the complications related to dry eye and contact lens discontinuation.

Additionally, if the movement is done only by one eye, it is called a wink. Still, it is not contemplated further in our cases because it is more of a voluntary movement not generally related to contact lens accommodation.

The characteristics extracted from the blinks when the blinking is under study are, independently of the nature or application of the study, the following:

- Blink rate and blink interval.
    - Inter blink interval distribution.

Different authors have contributed to measuring this healthy pattern in various studies. It was found that the value for adults was around 20 blinks per minute (83). This measurement for healthy participants tends to change during life, going from 6 to 8 blinks per minute to stabilize during adulthood (84). In addition, considering all the studies analyzed in the following subchapter to find the best techniques to extract the blink dynamics using event-based video cameras, it was found that some tendencies affect the blinking interval as the drowsiness state, that reduces the number of blinks per minute, the mental state disorders that can affect the occurrence of the blinking or the correlation in between the level of difficulty in a task performed and the variation of the blinking rate, see subchapter 2.5.2.4. Indeed, different Factors such as air quality, defensive responses, cognitive processes, situational demands, and individual differences influence the blink rate (85).

When it comes to studying the blinking rate phenomenon more deeply, it was found that the inter-blink interval distribution is skewed to the left, with most of the blinks occurring between 0.5 and 2 seconds and a minority occurring above 5 seconds.

- Blink duration.
    - Reopening time and closure duration.
    - Percentage of eye closure.

Regarding the blink duration, spontaneous blinks are recorded considering different patients to move from 150 to 400 milliseconds, and the wholly closed eye duration that most of the studies calculate as a percentage of eye closure during a certain amount of time is considered to last for 50 milliseconds (85). Studying the duration more in depth, it was found that the reopening time duration is double the closing time (86). It has to be clear that voluntary blinks are usually

longer in duration, and reflexive blinks are a bit longer than spontaneous blinks (85) but the alert state due to visual task demands of the user will decrease the duration value of the blink (87).

- Blink amplitude.

Concerning the blink amplitude, upward gaze direction has longer amplitudes since the upper eyelid is pushed upwards before the blink (85). The obtention of the amplitude of the blink from a signal measure (as occurs when analyzing the frames of an eye video from event-based cameras), the best approach will be to find the gradient (start of the downward phase) and the highest point on the curve. From this point, we can obtain the amplitude. (88) uses this for expressing blink amplitude. Other studies involved in the blinking detection with medical laboratory equipment,

The measurement of each one of the following characteristics is usually done in milliseconds for the blink duration and interval, while the blink amplitude, depending on the technique used for the phenomenon study, will be in millimeters or degrees, representing the last one, the angular displacement of the eyelid considering the diameter of the sclera.

While the principal characteristics have been stated, extracting the dynamics and cinematics values from the blink is one of the primary purposes of the thesis. The study of the relative position of the upper lid is very important to check the expected behavior of the physiological eye objectively, specifically the interaction between the eyelids and the surface of the eye. The addition of an external foreign thin layer in between the surface of the eye and the internal layer of the eyelid can change the dynamic and cinematic of the eyelid movement during the blink, resulting in a poor accommodation of the eye to the contact lens.

To characterize the blinking phenomenon both dynamic and cinematically, using a signal that is intrinsically related to the position of the upper eyelid is a good choice for a noninvasive method. With the obtention of this relative position in a blinking sequence, we can obtain furthermore, if the framerate of samples is known, essential features as the speed of the downstroke and upstroke, the maximum and minimum acceleration during the opening and closing movement, the time at blink completion occurrence (fully closed eye), time at maximum power developed by the orbicularis oculi (OO) involved in the downstroke movement, and the levator palpebrae (LP) muscle involved on the opening of the eye. An example of the obtention of a particular significant timeline point in the blinking sequence is

done in the study (89) using a high-speed camera and light refraction measurement to obtain the relative position of the upper eyelid for 20 seconds for 26 subjects.

Once the most used characteristics are known, the different techniques used for each purpose will be stated in the following subchapter. The principal differences will be highlighted to find the tendencies, gaps where improvement can be made, and the most used algorithms and techniques.

2.5.1.1. Blink study justification

To better understand the eye blinking phenomenon, particularly spontaneous blinking, this thesis aims to study the influence of the external environment and the activities performed during the blinking process (89–91). One key objective is to obtain objective information related to patient adaptation to different types of CL and to investigate the influence of CL use, as seen in studies by (92) and (93). The first study highlights the importance of evaluating physiological changes and subjective comfort in patients after long-term wear of various contact lenses. The second compares the behavior of the blink-in contact lens wearers under two controlled environmental conditions simulating the office building conditions (gentle conditions) and aircraft cabins during flights (extreme conditions) by checking the blink rate and the ocular surface conditions. The subjects were wearing two different contact lenses for comparison purposes, even less the blister properties of the contact lenses were similar. It was concluded that the tear film osmolarity decreased due to adverse conditions, considering or not the previous ocular surface state of the patient. This decrease in the osmolarity also was related to a higher blink rate during the desiccating environment. A decrease in comfort by the patient resulted from the increase in blink rate and decrease of tear film osmolarity due to the adverse conditions that resulted in a significant increase in corneal staining and limbal conjunctival.

Given the primary focus on tracking spontaneous blinks, it has been found that there is a significant relationship between the frequency of lacrimal layer fracture and spontaneous blinking. Various factors affecting blinking, related to cognitive state, have been studied. A decrease in corneal sensitivity is recognized as a major factor impacting tear film stability post-corneal refractive surgery (CRS). Blink rates decrease with reduced corneal sensitivity after CRS, affecting tear film stability since blinking maintains ocular surface moisture and contributes to the lipid layer's secretion and distribution (94). Incomplete blinks can lead to inadequate lipid distribution and increased evaporation, emphasizing the need for further study on blink patterns and tear film stability after CRS.

Patient comfort in contact lens wear and continuation of lens use are often hindered by symptoms of dry eye (12,95). Dry eye symptoms impact contact lens comfort, which has led to applications in ophthalmology focusing on symptoms and diagnosis of dry eye (30,96).

Special attention to blinking characteristics is crucial for objectively comparing eye videos of the same patient with and without contact lenses. (92) explored the short-term impact of contact lens use on spontaneous blink patterns, noting effects on interblink interval, speed, duration, and palpebral aperture height. The completeness of blinks remained unaffected, indicating that analyzing spontaneous blink characteristics can provide a sensitive and non-invasive assessment of contact lens effects on the ocular surface. Various lens types, including spherical soft, toric soft, and rigid corneal lenses, were studied, showing diverse effects on blink metrics. Notably, rigid corneal lenses significantly narrowed the palpebral aperture height.

Blinking characteristics are influenced by mental state, attention, activity, ocular surface exposure, and environmental conditions (94).

A review of techniques for the automatic detection of blinks and the study of blink dynamics is justified, as patient comfort with contact lenses is closely linked to lens movement on the eye surface. The alteration of blinking is a primary consequence of using contact lenses (92). Furthermore, dry eye, caused by insufficient lacrimal thickness, is a primary reason for discontinuing contact lens use (14). Dry eye results from reduced lubrication, which can be due to aging (7) or medications affecting the lacrimal layer composition, manifesting as altered blink dynamics.

2.5.2. Blink detection sequence

The most used steps for the successful detection of an eye blink from an event camera consist of first, detecting the eye from the face image. This is usually done by a face detection algorithm with a feature selection to locate the eye. In the next step, when the eye region is located and isolated, the detection of blink will usually consist of giving a category to each frame, considering the eye state is open or closed. As we can see, this is the most used technique, but not the most useful. Once the eye state detection is performed, the results are shown considering if an image is successfully categorized and if all video blinks have been detected.

Considering the blink detection algorithm flowchart, six different characteristics of each blink study will be analyzed for comparison. One of the characteristics has already been stated at the

beginning of the chapter 2.5 and it is the app location or the purpose of the study, which is related to the field where the blink algorithm is used.

2.5.2.1. Face and eye detection techniques.

For this step, we can classify the algorithms used considering if they are using machine learning (see chapter 2.4) or not. If this is the case, as stated in the corresponding chapter, the algorithm must be trained from an available database. In this case, it will be a set of images where already it is annotated where the face and or the eye is localized, and the algorithm will predict the corresponding location by getting the information of the experience (trained dataset) to help to identify the region of interest (face or eyes). The most used algorithm for this purpose is the Viola-Jones Algorithm (97,98) for face and eye detection (99)Another algorithm very used for face and eye detection is based on machine learning models, specifically convolutional Neural Networks. It is called the Dlib library. It is a pre-trained model using Deep learning that is faster than the Viola-Jones Algorithm.

To enhance the knowledge of both techniques, the differentiation between each one considering the results is that the first one gives a rectangular region of each one of the features of the face, and the second one proposes 68 face landmarks that characterize all the face features. Considering the methodology, the Viola-Jones algorithm uses Histogram Oriented Gradient. This technique consists of calculating the magnitude and direction of the gradient of the pixels of an image to obtain furthermore a histogram with the information calculated. Furthermore, a feature extraction from the histogram is done, and it is compared to one of the models of the object to see if the gradient features are similar. An SVM model is usually used to corroborate that the image analyzed is the same object as the one wanted to recognize. Dlib library uses a convolutional neural network trained directly by the images of faces. As seen in the machine learning chapter, neural networks can't be interpreted, and it is very accurate if the dataset is very large, and this is the case with the Dlib library. Finally, the Dlib library has the possibility to be used with histogram of oriented gradients (HOG) and SVM, making this algorithm very robust for this task.

Other studies have used algorithms that don't involve techniques from machine learning but only image processing. A color threshold was used to classify the skin pixels from non-skin regions and consider face proportions and eye localization using morphological operation in a threshold binarized image (100,101). Morphological operations in grayscale images were used

in conjunction with edge detection and projection of gray vertical intensity to locate the pair of eyes in a recorded face video (102).

For tracking purposes, once the face is located, several algorithms use techniques to estimate optical flow and track the pixels' location of successive frames from a recorded or streamed video to reduce computational resources. With the Lucas Kanade Method, the user's eyes and faces can be tracked, and the region of interest is not lost in between frames.

2.5.2.2. Blink detection method

In this step, we will discuss the continuation to the eye detection region to get the final information about the eye state classification or the quality of the blinks available in a dataset or recorded video.

The easier-to-understand corresponds to the first block as the algorithm that used the Dlib software to get the 68 face landmarks to characterize the eye's opening in each frame. In this case, because 6 of the 68 landmarks characterize one of the eyes, it is easy to measure the distance between the two eyelids. All the algorithms using this technique will use the Euclidean distance to obtain a measure of the eye aspect ratio or leave the distance between the eyelid as a measure of the coverage of the cornea to determine the quality of the blink (complete or incomplete).

Different techniques are used in each of these studies to make it possible the classification of the frame eye state possible and have the best accuracy. The image processing techniques, signal processing, or supervised learning techniques will be stated in the following list:

- Smoothing curve using a moving average filter, furthermore obtaining a two standard deviation of the signal in a window of 5s as a threshold for eye state classification (Biondi, Saberi et al., 2023).
- Low-pass filter with a cutoff frequency heuristically set to 3hz and an adaptative blink threshold chosen as a function of the maximum eye aspect ratio of the filtered signal (104).
- Selection of the Eye Aspect Ratio (EAR) threshold (105–108).
- Baseline correction of the EAR signal using the asymmetric least squares smoothing approach (109).
- A support vector machine is applied to the resulting EAR value obtained for eye state classification(110).

- After obtaining the Euclidean eyelid distance, signal processing, smoothing, and noise avoidance using, respectively, the Savitzky Golay filter, median filter, and local peaks detection and further SVM for eye state classification (111).
- The discrete signal of EAR is filtered between 1 and 3 Hz for noise removal, and a threshold is calculated as the quadratic mean of the signal among each experimental condition (112).

Considering the different techniques that didn't give a characteristic of the eyelid position or the eye corner, several methods were used with very good results to find the eye state in each frame.

A technique used by most studies is the Viola-Jones algorithm to find the location of the eye region using only image processing techniques by determining the skin areas used binarization. This general technique gives a threshold to the gray-level image to separate the darker areas from the brighter ones. Even though this technique is very susceptible to the light available while recording, several studies used it. The technique is general, but the results were different depending on the author, the morphological operations performed afterward, or the classification method used for eye state determination. The different techniques used were pixel counting comparison between the number of significant pixels in the upper and low half of the eye image (113,114), number of significant pixels plus a threshold for eye state recognition (100,115–117), determination of eyelid distance using the binarization (101) and also vertical projection derivative function afterward (101), determination of pupil position using edge detection (118,119) or determining the iris position with the help of the Hough transform circle or ellipse (120,121). For the iris or pupil detection, generally, if it is not located, the eye is considered closed.

The literature also contained original, more elaborate techniques in addition to the image processing and classification techniques.

For example, (122) describes a pixel motion analysis obtained from comparing each of the frames of the video eye with a reference frame with an open eye. The optical flow technique, used to find the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene, is used to detect the motion of the pixels. The amount and direction of significant motion vectors and the duration of the blink are three conditions studied to detect voluntary or involuntary blinks.

Another example is OKOA Vision, a commercial technique used to detect eye movement and blinks, which produces a waveform motion detected by measuring the opening and closing distance of the eyelids. Signal processing techniques are performed for further blink detection are noise filtering and peak detection functions (123).

The interlace technique is also found in the literature that consists of separating an image into two frames (124,125) of the eye image for blinking detection purposes. An eye blink pattern is obtained progressively by calculating the opening eye area using color information embedded in the eye image.

Another specific technique used is to keep the ratio of the palpebral fissure area to the temporal median iris diameter and propose two thresholds for the classification of complete and incomplete blinks. The deep learning encoder decoder algorithm is used to obtain the palpebral fissure area and the diameter of the iris during the eye detection process (126).

In this study, an efficient technique is developed (127). Considering the eye image and the HSV (Hue, Saturation and Value) color system, considering only the mean value of channel V, a signal is obtained. Furthermore, with a threshold to the gradient of the signal, blink detection is performed.

Convolutional Neural Network (CNN)-based methods that automatically obtain the eye state at the same time the eye region is obtained are also used for blink detection (128,129).

2.5.2.3. Databases are used to train and test the different algorithms

To perform the blinking detection algorithms, a set of images or a video collection needs to be tested, and if required, the algorithm needs to be trained. Blinking detection from event-based cameras must be tested in videos or image sets to corroborate the method's efficacy. For comparison purposes between the different algorithms, a set of videos or images is usually published for various other authors to enhance the accuracy obtained with the previous methods. In this subchapter, the different database characteristics found in the literature are going to be developed.

- The ZJU data set (130) includes 2424 subjects, among which 1192 subjects with eyes closed and 1232 subjects with eyes open.
- The UBFC-RPPG database created using a custom C++ application for video acquisition with a simple low-cost webcam (Logitech C920 HD Pro) at 30fps with a resolution of 640x480 is also used (131).

- Not that popular, a database of videos of 145 recruited incarcerated males was used in (132).
- The Eyeblink8 dataset can be found in (111,133) 408 eyeblinks are made on 70992 annotated frames with a 640 x 480 pixels resolution.
- The data set Talking Face consists of images of one subject sitting and talking in front of the camera. There are 5000 frames captured with 25 fps with a resolution of 720×576, in a total of 200 seconds of the video stream. This data set is used by (Al-gawwam & Benaissa, 2018; Bergasa et al., 2004; Dewi, Chen, Jiang et al., 2022; Fogelton & Benesova, 2016; Matjaz Divjak and Horst Bischof, 2009).
- A covered face-eye database is also used (136).
- Closed eye in the wild (137) with a number of 1194 (closed eyes), 1239 (open eyes) of 100x100 pixels resolutions.

In the databases listed before, a criterion was used to eliminate all the studies that used medical devices or high-speed cameras, or infrared cameras as the study's purpose is to find techniques that will work with ubiquitous cameras. When the quality and frame rate is enhanced, the techniques found to extract blinking and eye features were less robust and more susceptible to environmental light intensity when used in videos from ubiquitous event cameras instead of the infrared camera used to create the database.

2.5.2.4. Results and conclusions

From the review of the main techniques used to extract the blinks from a set of images or video recorded or inline streams, the results proposed in the different studies depends on the application field and purposes of the study. It must be highlighted that the algorithms that have not stated a field of application study have shown results more explicitly to corroborate the accuracy and efficiency obtained. The most used metrics in this type of study were to classify the state of the eye in each of the frames of the video. With the classification done, a misclassification rate can be calculated with the accuracy, precision, and F1-score given in most of the studies. This is performed with the help of the annotations in the different databases and knowing if each frame is classified correctly for a blink or not having the information at the end of each image's True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

Compared to all the other studies that specified a field of application, it was interesting to see how each one of the studies calculated, for example, the percentage of closure to study the

drowsy state or the duration of the blink in relation to the amplitude to classify the blinks as voluntary and involuntary for human-computer interaction.

The misclassification rates in the different studies were related to various environmental and patient eye characteristics.

Comparing the studies based on event cameras and the ones using electrooculograms, for example, the characteristics extracted were much less for the first type of study. This is done due to the poor quality of data and the framerate of the samples of each video analyzed. Considering this tendency, one of the principal ideas we wanted to test concerning an enhanced blink detection algorithm is the extraction of characteristics such as maximum blink speed to obtain a better characterization of the eye blink, considering that nowadays even the most ubiquitous cameras can achieve a 30 Hz frame rate.

From the studies analyzed covering the past ten years on articles from journals in indexed databases, the principal causes of misclassification and the difficulties found by the authors are:

- Noise available in the signal obtained (depending on the method, can be the density of pixels, the distance between the eyelids, among others) due to the differences in the patterns found between voluntary and spontaneous blinking.
- Subject looking down. An eye movement is usually translated into a movement of the eyelid if it is in a downward direction. Due to the technique used (camera recording), the eye state will be classified as closed when it can be open due to the position of the camera towards the user.
- Excessive head movement increases the misclassification rate due to the blur produced when the frame rate is low but also to a poor eye detection algorithm.
- The use of glasses will confuse the eye detection algorithm due to the reflection produced by the glasses. It can make eye detection impossible.
- A larger distance from the eye to the camera is translated into fewer pixels of the eye region, which will make it more difficult to detect the eye, and furthermore, misclassification of the eye state is more likely to happen.
- A lower image quality is translated into worse eye detection and worse blink detection.
- Poor environmental light conditions can cause the Viola-Jones algorithm to detect the eyebrows instead of the eyes.

The different studies have discussed proposed solutions to the problems stated before to make more robust algorithms or enhance the quality of the input images and videos to fulfill the requirements of the study. Studies (117,129) proposed to test the algorithm in the worst conditions, and others stated that their methods were robust to environmental light conditions (101,108,109). Additionally, (110) collected more data on the eye open state in the same conditions for comparison purposes to make his method more robust as it has more information about the eye open state of the same patient. To consider the issue related to the ambient light environment and its results in blink misclassification (120) preferred to introduce a threshold factor specific to each individual and its environment. The factor will not change if the illumination condition is similar.

The state-of-the-art algorithm-based event cameras generally have small discrepancies with the gold standard method, the eye tracking system. Using the Dlib library brought a solid eye detection method that many authors have used in their studies independently of the application, as it is also possible to extract mouth and head position and rotation. The threshold for determining the state of the eye in the different studies was similar, resulting in a good accuracy.

Finally, comparing the different studies is barely possible as the applications are different, and the extracted features are not always the same. Additionally, only a few articles have used the same database, but even the same database has different annotations differing on some frames due to each author's interpretation. Some authors proposed one or two algorithms with different methodologies and have recreated other algorithms from various authors and proposed a comparison to see which technique performed better (100,101,110,116,122,126).

2.5.3. Summary of the articles involving algorithms for blink detection from event-based cameras.

Table 2-1: Summary of articles related to studies of algorithms of blinking analysis from ubiquitous video cameras.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Human-Computer Interaction	Face detection Haar cascade classifier algorithm with eye detection Adaboost classifier	From the eye image, a binarization with 70 intensity pixels is considered as a threshold, and the number of black pixels determines the state of the eye (open or closed).	Face video recorded through an app.	1 subject.	(99)	2 Dataset containing eye images and without the object for training purposes of the eye detection algorithm.	Eye tracking using center pupil and corneal reflection	Overall accuracy, detection accuracy, TP, TN, FP, FN.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
Intelligent mouse	The frames are turned into red green (RG) model and after a dimensionality reduction to 50 bins, a threshold is applied to detect frames considered as skin. Afterwards, the eye is detected considering all the nonskin region sizes that correspond to (mouth, nose, eyebrows, and eyes) Finally, a tracking system is used after the frame is turned to a grayscale. An integration of the eye variance feature and the Gaussian Mixture Model classifier is performed.	The binarized eye image is calculated from the grayscale image after morphological operations to eliminate noise. The vertical eyelid distance is obtained using the variance projection derivative function. The vertical projection derivative function is used to select the true values of the eyelid eliminating the non-valid ones. Exterior vertical projection derivative value. An SVM is used to classify eye states.	ZJU data set and 360 images of the eye and non-eye data set of each 30 subjects. Videos collected from lab settings and videos obtained from archives of different news reports.	30 different subjects	(101)	ZJU data set (This data set contains 2,424 subjects, of which 1,192 subjects with both eyes closed and 1,232 subjects with eyes open.) and 360 images of eye and non-eye data sets of each 30 participants. Blink detection comparison with other techniques that use different blink detection methods: simple gray level, Gabor wavelet, local ternary patterns (LTP), histogram of Oriented Gradients (HOG), and multiscale histograms of principal oriented gradients (MHPOG) along with SVM classifier.	None.	FP, TP, FN, sensitivity, predictive value, eye detection accuracy.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
Mouse operations	The Viola-Jones algorithm is used to detect the eyes. Every single eye is cropped equally to size 102x54	Pixel motion analysis is applied to compare each analyzed frame with the reference frame (1st frame). The optical flow technique is used to detect the motion of the pixels. The number of significant motion vectors (speed), the direction of the significant motion vectors, and the duration of the blink are three conditions studied for each frame to detect voluntary or involuntary blinks.	Video oculography. Uses a webcam and a light source to enhance precision (CFL).	10 healthy users.	(122)	The ZJU database is used to corroborate the result of the algorithm. The results of the blink detection techniques on ZJU databases were compared (135,138–140)	No additional features were extracted.	Overall accuracy, detection accuracy, false alarm rate, overall success rate.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Fatigue Detection	Video decomposition, grayscale processing, image smoothing, and edge detection. Face detection is obtained using the Matlab PCA package. Eyes are detected by the boundary gradation threshold from the gray vertical integral projection feature.	Pixel height of the palpebral fissure. The iris and eyelids are obtained from image binarization, and the height of the pixels is calculated on the eye axis. The percentage of time that the eyes are more than 80% closed (PERCLOS) value is obtained.	Logitech c310 5MP webcam, 1280x720 pixels resolution, 30 Hz.	14 subjects were male and female from 23 to 27 years of age.	(102)	Comparison of blink rate between rested (conscious state) and fatigue.	None.	Face detection rate, eye detection rate.
Assess the cognitive load of the user during a computer task	HOG for face detection. D-lib facial landmark prediction algorithm for face shape key point detection for eye detection	EAR is obtained from 6 eye landmarks. Smoothing of curves with moving average filter. A 5s window is used to obtain the 2 std value for the blink classification threshold.	Ubiquitous cameras	25 subjects, 18 men, and 7 women. Mean age 23 and std 5.3	(141)	Comparison with the eye tracker using Bayesian analysis.	Cognitive load assessment using blinking frequency and eye tracker system with data collection frequency 60 Hz.	Discrepancies between blink frequency between eye tracker and event-based camera blinking algorithm proposed method.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Classify the severity of depressive disorders	Cascade Haar Classification using Dlib library, to extract 68 facial landmarks.	EAR is obtained using the six eye landmarks. A low-pass filter is applied to the EAR signal with a cutoff frequency heuristically set to 3hz. An adaptative blink threshold is chosen as a function of the maximum eye aspect ratio in the filtered signal.	1 dataset A1 and divided into 2 subsets of the same dataset (39 in total, 29 (A2) and 25 (A3) in the 2 subsets. The first subset excludes patients with glasses, and the second subset excludes glasses, hair occluding the eyes, and half close the eyes during a long period.	39	(104)	The UBFC-RPPG dataset is used. No other blink detection techniques were compared as this is the only technique tested in the current database.	Facial blood volume is obtained from the cheeks, nose and forehead without eyes, and ROI of the eyebrows ROI.	Precision, recall, and F1-score are obtained from Datasets A1, A2 and A3. From the UBFC-RPPG dataset
Assess psychopath traits in prisoners	Facial landmark detectors based on regression trees (Kazemi & Sullivan, 2014) included in the Dlib (King, 2009) C ++ library.	EAR is obtained from the 6 eye landmarks.	A dataset of 125 videotapes of prisoners' faces with good illumination is used.	Not specified.	(132)	None.	Additional study of the correlation between blink pattern and deceptive personality traits	Correlation to blink patterns from deceptive personality traits.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
Predict the engagement status of students	Haar feature selection, creation of integral images, AdaBoost training, and cascade classifier. An open CV library is used over the converted grayscale frame. Facial landmarks are obtained jointly with the Region of Interest (ROI) of the eye. The Dlib library was used to train the algorithm.	EAR is used to check the eye state. The EAR threshold was selected as 0.25. The blinking detection was only a part of the work done as the main purpose is to detect emotions.	The camera records students while assisting in class for a one-hour session and is processed at 20 frames per second.	10	(107)	None.	Observation of the gaze from the nose, chin, pupil, mouth, and upper head position.	Accuracy, misclassification rate, precision, recall, specificity, and F1-score of facial emotion performance not for blink detection. Confusion matrix.
Detect fatigue in workers	The Dlib toolkit is used to detect 68 face landmarks to obtain the position of the eye and mouth for further analysis and considers the bottom and the upper corner of the face for the detection of the head angle.	EAR is used to detect the change in the state of the eye from open to closed and to evaluate the fatigue status considering the percentage of time that the eyes are more than 80% closed (PERCLOS)	Camera-based, 640x480 pixels are the resolution of the videos.	9 subjects	(108)	None.	The mouth opening ratio is calculated using the mouth landmarks, and it is possible to detect the yawning frequency. HEA (head Euler angle is also calculated) to obtain the nodding frequency	Blink Frequency (BF), PERCLOS, yawning and nodding frequency. Blink detection accuracy, yawning and nodding.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
monitoring the neurophysiological parameters of potential patients and assessing their mental state.	The Dlib library with Adaboost classifier (based on the YCbCr color model) is used to collect 68 face landmarks.	The eyelid distance is obtained from the six landmarks of each eye by calculating the EAR.	common webcam.	15 students (8 males and 7 females) of 30.6 +- 3.7 years old.	(112)	A comparison is made for the video-based techniques and the Electrocardiogram (ECG) and Electroencephalogram (EEG) techniques used to accurately measure in the laboratory EBR and Heart Rate	In addition, heart rate was measured by extracting the red component from the image detected by the face.	For the eye blink ratio, a difference of 4.5% and 4.8% during different tasks.
Quantify visual attention	The Dlib library with Adaboost was used to extract 68 face landmarks to retrieve the position of the eye and face from each frame of the video.	EAR was calculated from six landmarks from each eye. A signal is obtained, and blinks are detected. A threshold of each subject is calculated to classify all the frames as open or closed state. The raw EAR signal from the analyzed video is modified by performing a baseline correction. An asymmetric least-squares smoothing approach is used.	RGB camera	37 subjects	(109)	None.	Fixation of gaze from yaw and angle of pitch from head position.	Attention tasks related to blink rate variability are proposed, and data related to blink detection performance are obtained.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
No application was stated.	Eye detection is performed with the Viola-Jones algorithm. The left eye is detected, and the eye tracking algorithm becomes active until the eye disappears, Kanade Lucas-Tomasi (KLT) feature algorithm tracking.	Iris detection is considered for blink detection. First, luminance normalization using a 31x31 pixel median filter of the eye ROI tracked. The Otsu optimal threshold binarization is applied. The eyebrow is erased with a mask. Dark points after performing morphological operations are considered to have a circular shape for detecting the iris	17 1-minute videos. The illumination range was from 320 to 600 lux. At least the 720p resolution.	17	(120)	None.	None.	True and false positive rates are calculated.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
Comparison with electro-oculogram, and visual counting.	OKAO vision software is software that enables facial recognition and detection of various parts of the human face. Facial landmarks such as eyebrows, pupils, nostrils, and mouth angles are identified by detecting brightness around these landmarks using 3D model fitting and statistical identification methods based on a large training data set.	aVTRa is based on the image processing technology used in OKAO Vision. used OKAO Vision to produce a waveform of the blinking motion detected by measuring the opening and closing distance of the subject's eyelids. Individual blinks were then detected from the blinking waveform through noise filtering and peak detection algorithms.	WEB camera, 30 fps still images	(123)	(123)	Electrooculogram (EOG) detection and visual count for comparison purposes	none.	Measured blinking in both static conditions, where the subject was sitting still with his head fixed on the table, and dynamic conditions, where the subject’s face was not fixed, and natural communication was held between the subject and the interviewer. Percentage of concordance between proposed technique vs. visual count, proposed technique vs. EOG, EOG vs. visual count

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
No application was stated.	Zface is a robust validated technique. With Zface, 6 eye landmarks are obtained for each eye characterization.	The vertical distance between the eyelids is obtained from the 6 eye landmarks. Signal processing techniques, smoothing, and noise avoidance. -Savitzky Golay filter. -Median filter. -Local peaks. SVM for eye state classification	Several validated datasets from the literature	None. Images and videos from well-known databases are being used to train and test the algorithm.	(111)	ZJU dataset, Eyeblink8, Talking face	None.	Precision, recall, average blink duration, ground-truth blinks, detected blinks, ground-truth blink rate
No application was stated.	The Viola Jones algorithm is used to detect the ROI of the face for the input image. Head yaw and roll angles are obtained using geometrical image processing techniques obtaining the position of important face landmarks as head corner, nose corner, and eye corners using Harris feature points.	After Viola Jones' pair of eyes or single eye is detected, image binarization is done. Gradient application is performed, and a comparison between upper and lower black pixels number and a classification of eye state is done.	Video or still images of events cameras. Validation: 15 video sequences holding about 1125 frames in various lighting conditions and different subjects. 30 fps (55-60 cm distant).	15 different topics.	(113)	Customized database, UPNA, and ORL databases.	Head angles, yaw, and roll angles.	Recall, precision, TP, FP, FN.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
No application was stated.	Automatic eye ROI Detection algorithm consisting of turning to the YCbCr color space where a map is applied to differentiate skin pixels from nonskin. An Otsu binary mask is applied, and afterward clear border function and disk-based morphological erosion and dilatation. Blobs related to nonskin regions are considered.	The iris binary area obtention and the size of the signal are proof of blinking.	Ophthalmic forehead chin rest, iPhone attached to Cartesian machinery to allow relative position changes between the user and the camera. Video is acquired at a low frame rate(60Hz)	4 subjects aged between 20 and 28 years, 2 women and 2 men.	(100)	Optical flow estimation is used between 2 consecutive video frames. Gunnar-Farnebäck is applied to detect moving objects. When a blink occurs, the estimated vertical component vectors of the velocity are higher. Head movements are considered when all motion vectors vary similarly in direction and magnitude. 3 different other techniques are also developed and compared.	The correlation between frames and local binary patterns, additionally with optical flow and the proposed blink technique method described are compared using the same video collection.	Precision, F1-score, sensitivity.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
No application was stated.	Two options were coded: The corners of the eye are detected manually from dataset annotations. Using CLandmarks and VJ algorithms, the corner of the eye are also detected automatically.	Using Gunnar-Farnebäck, motion vector of each pixel in the eye region are estimated. Vertical components are analyzed to differentiate eye blinks from head movement. Normalization of motion vectors using intraocular distance is performed and mean vertical component and its standard deviation are the 2-signal interest for blink classification using support vector machine	ZJU dataset, Eyeblink8, Talking Face.	The ones present in the different datasets	(133)	ZJU dataset, Eyeblink8, Talking Face and Researcher's night datasets. Ground truth of blink is available in each video.	None.	Precision, Recall, Count of detected blink, False negative, False positive, True positive.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
Blinking classification (voluntary and involuntary)	No techniques are used for eye and face detection. Direct eye frames are used.	The interlace image technique is used to separate an image into 2 frames. An eye-blinking wave pattern is obtained progressively by calculating the open-eye area using color information embedded in the eye image.	Event camera-based eye video.	10 subjects.	(125)	None.	The blink classification depends on the duration of the blink.	A classification rate of 4 different voluntary blink types.
Blink classification (complete and incomplete)	The segmented image is done in each frame with the application of a deep learning encoder decoder algorithm. Iris diameter, sclera area and palpebral fissure height are obtained.	2 thresholds were applied for classification purposes of incomplete and complete blinks, to the ratio of the current palpebral fissure over the temporal median value of the iris diameter of the corresponding eye.	536 images training set (481 images of the iris dataset and 55 additional images of partially or fully closed eyes) Test video with 105 complete blinks and 54 incomplete ones with a duration of 4 min.	1 video of 4 minutes	(126)	Two different other algorithms with the same dataset and video collection were tested for comparison purposes (133,142).	Blink classification (complete and incomplete)	F1-score, sensitivity, specificity, precision, negative predictive value, false positive rate, false discovery rate, false negative rate.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
Voluntary and involuntary automatic detection blinks	Not specified, the eye image is probably obtained using the Viola-Jones algorithm.	Calculate the visible eyeball area under typical room lighting. The blink detection system consists of a novel measurement called the frame splitting method, where a measurement is performed with two field images obtained by dividing one interlaced image into two images. The state of the eye is calculated using thresholds dependent on the mean difference value of size (measurement values) of the open-eye area at eye opening and standard deviation (Template match).	Commercially sold high-definition video cameras and PC	15 subjects.	(124)	None.	None.	Classification rate, classification error. Duration, maximum amplitude, and integrated value of the amplitude.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Drowsy Driver Detection	CNN to distinguish the human eye by returning the bounding box. Dlib library to obtain 68 landmarks from face ROI.	EAR is calculated from 6 landmarks of the ROI of the eye. The EAR threshold was empirically obtained for each data set for eye state classification. Best threshold (0.18)	Several validated datasets from the literature	None. Images and videos from well-known databases are used to train and test the algorithm.	(143)	3 videos from the Eyeblink dataset, TalkingFace, and Eyeblink8 datasets. Comparison is made with referenced blink detection methods.	None.	Precision, recall f1-score, confusion matrix.
Help Saudi Arabian women improve their driving skills and prevent accidents.	The CNN model is used for face and eye segmentation and blink detection.	CNN.	Event-based camera monitoring drivers in a continuous manner.	Data set used for pair of eyes images from videos.	(144)	Comparison with reference techniques. Database of covered faces (136)	No additional features were extracted.	Event detection, detection precision sensitivity, variation error, detection time. Accuracy.
Preventing the drowsy driver	Haar cascade classifier for ROI face detection. Eye tracking method for successive frames using the OpenCV library mean-shift method.	Pupil detection using OpenCV Hugh Circle Transform. The pupil feature image is successively binarized and filtered. Edge detection is applied using the Canny method, and the classification state of the eye is decided using	An infrared camera that can operate at night. It used Python and C languages for its development environment, and it used a Raspberry Pi 3, an infrared camera, speaker, microphone, carbon dioxide sensor, Galaxy S4, and the automobile.	No information was available.	(119)	None.	Carbon dioxide detection. Speech-to-text recognition technology developed.	No results related to the blink detection algorithm.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Drowsy Driver Detection	Intel's Haar cascades classifier.	Pupil detection is done using morphological operations (erosion, dilatation, and closing/opening operations) binarization, and finding circle contours. PERCLOS, BF, and MCD are extracted from the localization of the size of the eye pupil in each frame.	Resolution 640x480, 15/30 fps, 192412 frames divided into 4 segment collections.	Not specified, use a 192412-video frame dataset.	(118)	Six different related techniques with the same drowsy detection purpose were compared to two different methods proposed by the authors.	Yawning and mouth opening, facial features to enhance drowsiness detection	The proposed results are global and related to the accuracy of detecting drowsiness. The article did not explicitly show values related to the precision, sensitivity, or f1 of PERCLOS, MCD (Maximum Closure Duration), and BF precision, sensitivity, or f1-score.
Drowsy Driver Detection	The Viola Jones algorithm is used for face detection. Eye detection is performed using skin color detection with the Viola-Jones algorithm.	The explanation of the blink method is not well described. A threshold value is calculated for each frame dynamically and the threshold is compared with the drowsy threshold.	The VGA camera is used for frame-video collection. Matlab software is used for the simulation of the detection method.	Not specified.	(145)	Viola Jones to extract ROI from the mouth.	Lip detection for a percentage of open mouth.	Percentage error of images for eye detection and closed eye detection. Average accuracy of the blink detection system.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Drowsy Driver Detection	The Open-Source Computer Vision Library (Open CV) is used to implement the Haar cascade classifier.	The eye aspect ratio formula is used to calculate the state of the open eye. Six points characterize the eye feature, and the vertical distance between the upper and lower eyelids related to the longitudinal palpebral fissure is obtained.	The Raspberry Pi camera sensor and Raspberry Pi 3 Model B Algorithms for eye blinking detection are developed with OpenCV. 1920x1080 pixels resolution videos.	6 subjects. 3 males and 3 females.	(106)	None.	No additional features were extracted from the videos.	Mean EAR with open eyes and mean EAR with closed eyes for each participant.
Drowsy Driver Detection	The OpenCV framework is used to analyze the captured video. First, the input video frame is turned into a grayscale. Face detection was implemented using the Haar classifier to obtain a rectangular region over the eye. The rectangular region can be manually adjusted to obtain.	The rectangular eye region is masked further to eliminate the noise arising from the eyelashes and eyebrows. A smaller rectangular mask without the corners (like the palpebral fissure geometric form) is used for this purpose. The image result is binarized using the OpenCV *threshold* function. White pixels are counted using countNonZero().	Use of the Samsung LG-K330 mobile phone camera. The videos recorded have a frame rate of 30 fps.	One male (24 years old) and one female (23 years old).	(117)	None.	None.	No explicit data related to blink detection.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Drowsy Driver Detection	A Dlib facial detector is used and extracts 68 features from the face of the individual. The face is cropped from the input image. The face image is resized to a 244x244 pixels frame.	CNN. The deep learning model takes the face images as input and results in an output specifying whether the eyes in the image are open or closed.	The open data set published by 'Xiaoyang Tan' from Nanjing University of Aeronautics and Astronautics is used to train CNN. The webcam-based prototype was developed to test the model trained using Dlib facial landmarks.	Not specified.	(129)	Data set used: closed eyes in the wild (CEW). - No of images: 1194 (closed eyes), 1239 (open eyes) - Total: 2433images (100X100) JPG - Total number of epochs: 5 - Training images: 2181 - Testing images: 252	11 other drowsy detection methods were compared with the same data set.	Accuracy.
Drowsy Driver Detection	Haar cascade classifier for face and eye detection. ROI of 1 eye is the final output.	From the HSV color model, the mean value of the V channel of the ROI is obtained and monitored in the eye ROI is obtained. A threshold is used to obtain the state of the eye and applied to the gradient of the V-channel time function.	60hz video frame rate with a duration of 60s	Ten subjects were involved in the study and a 60 second video was recorded for each patient.	(127)	The visual blink rate of the observer is obtained for comparison purposes.	Additional monitored systems with different sensors are used to detect drowsy drivers and are not related to blinking.	An accuracy of 89% is obtained while comparing the blink rate from the algorithm and the observed from a visual count.

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
Detect drowsy driver	A pair of eyes detected with the Viola-Jones algorithm	The Hough transform is used to detect the position of the iris. When the iris is not detected, the eye state is classified as closed, and when the duration of a normal blink is overpassed, the system alerts the driver of a drowsy state.	Web camera connected to Matlab software. Used during day and night shifts.	Not specified, it seems to be a pilot study with one or two subjects.	(121)	None.	None.	No results related to the accuracy of the system.
Detect drowsy driver from blinking and yawn detection.	The Haar feature method is used for face detection and, more specifically, the Viola-Jones algorithm. The Kalman filter is applied for face tracking reducing computational cost by estimating the position of a moving object based on its historical values in the next frame.	Eye detection is first performed by dividing the face image region by 2 and taking the upper half. The image is turned to grayscale, and further, it is binarized using the adaptive threshold technique. The iris is successfully tracked using a Kalman filter. The eye state is obtained considering the number of black pixels in the eye region.	A web camera is connected to Matlab software to prepare a snapshot function to acquire images from streaming video. 1280 × 720 resolution at 30 fps.	7 (3 from online videos and 4 from the MRL image dataset MRL).	(115)	YAWD video data set online and MEL eye image data set analysis.	Drowsy system: it uses the data obtained from the eye and mouth states. It is a fuzzy model that depends on the classification of the first ratio of eye closure and the last ratio of mouth open.	Eye detection accuracy, eye status detection accuracy.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Detect drowsiness	Improved YOLOv3-tiny network for face detection. The Dlib toolkit is combined to extract facial feature parameters (68 face key points), by which they calculate the 128-dimensional feature vector, eye feature vector (EFV) and mouth feature vector (MFV) of the driver's face in the image.	EAR is obtained to detect blinks. Two phases are done; the first biometric and classification input is done by the driver to enhance computational time and resources for online blink recognition. Blink detection is performed online with a trained classifier (eye open threshold) and biometric data for driver recognition.	Web cameras collect biometric images, eye classification images, and mouth classification images.	10 samples. DSD data set.	(110)	(http://wider-challenge.org/2019.html) The WIDER FACE database is used to train YOLOv3-tiny to locate drivers' faces. 2 other blink detection techniques were compared (146,147).	Mouth Aspect Ratio	PERCLOSE, BF

Purpose	face/eye detection	Blink detection method	Video Source	Subjects	Ref.	Databases or comparison	Additional Features Extracted	Results
Detect drowsy driver	The big pair eye function of the viola jones algorithm is performed to locate the boundary box of the pair of eyes from the driver participating in the study.	Afterward, the image of the eyes is converted into binary by determining the threshold using the OTSU method. The image is divided into upper and lower parts. When the eyes are open, due to the color of the pupils and eyelashes, the ratio of dark pixels in the upper part of the eye is greater than the state of closed eyes (hidden pupil). The resulting signal is multiplied by 3. A threshold is used for state classification.	Commun web camera	Not specified.	(114)	None.	Drowsy level detection using neural network	Explicit results only for the accuracy of neural networks for drowsy detection. Value of 93%
Help the disabled population. Human-computer interaction.	Viola Jones algorithm to detect the eyes	CNN is used to detect the state of the eye.	Images from different datasets.	7 subjects from a hand-made dataset. Uses popular datasets.	(128)	The TensorFlow framework is used for the head mouse control system. Handcrafted data set of 853 images of different head poses. Closed eyes in the wild data set for eye state classification.	Use of head direction to move the mouse accordingly with head direction	Root Mean Square Error (RMSE)=0.0575 Accuracy for eye state classification= 97.42%

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Detect drowsy driver	Face and eye detection with the Viola Jones algorithm. The equalization of the grayscale and adaptative histogram of the image was previously performed.	Pupil and eye detection. This is performed using morphological operations on the image after binarization of the eye image.	640x480 video dataset IR videos 15/30 fps.	36 subjects. Drowsy Driver Dataset (DDD)provided by NTHU. 17 individuals tested from the data set	(116)	Comparison with other 13 techniques that are using the neural network to extract the drowsy state.	Mouth features. Position and orientation of the head.	Explicit results of the drowsiness classification. Algorithms give implicit results of PERCLOS, BF, EAR, AND MCD.
Comparison to normal blink rate. Detection of abnormal eye defects early stage.	The Kalman filter is used first to denoise the image. YOLOv5 to crop eye image from face video UNET.	Y-UNET feature extraction particularly focus eyelid extraction. The Euclidean distance method is used to extract the vertical distance between eye lids obtained from eye landmarks.	30 fps 320x320 rescaled resolution	Not specified.	(148)	Comparison with R-CNN	None.	Mean average precision value for the algorithm.

Purpose	**face/eye detection**	**Blink detection method**	**Video Source**	**Subjects**	**Ref.**	**Databases or comparison**	**Additional Features Extracted**	**Results**
Drowsiness detection.	The Viola Jones algorithm is used to detect the face. Proportional eye crops are used to detect the eye.	The eye image is turned to grayscale and then normalized in mean and variance (0=mean value, 1= variance). The eye image is finally binarized using a specific threshold. 2methods: • The first method uses landmarks in the binarized eye. • The second method uses an MLP neural network.	low-cost reverse parking infrared camera. 320x240 pixels.	8 different people.	(149)	Comparison with different techniques and obtention of accuracy from image set from 8 different people	None.	Accuracy
Human Drowsiness Detection System	Dlib library to obtain 68 face landmarks. A pi camera is used with the template matching system for intrusion purposes.	Euclidean distance to calculate vertical diameter between eyelids. Raspberry pi + OpenCV lib	30 fps VGA webcam	Not specified.	(150)	None.	Sensor to detect drowsy drivers. Special camera with template matching for additional intrusion detection	Percentages of hits detect the number of predicted blinks as a function of the number of blinks in the video.

3 METHODS

3. METHODS

3.1 Project design

This chapter explains all the technologies developed to reach the main objectives stated in the first chapter, HYPOTHESES, OBJECTIVES.

The technologies developed can be divided into three groups, following the functions that will be developed. First, the recollection application is used to obtain all the necessary information related to the personal eye characteristics, all the variables used to characterize the environment where the subject is evolving, and the subjective information given by the patient through patient-reported outcomes and questionnaires. The second part involves all the transformations used to extract the different features from the data obtained with the application. It consists of several algorithms developed and coded in the Matlab entourage to extract the feature from the eye videos to characterize the blinking, diagnose the bulbar eye redness, and compare those results while using or not using contact lenses. The third part is all the machine learning models trained and tested to find tendencies in the data obtained to predict the accommodation of the contact lens without needing the patient to reach the optics or the clinic where the tests are used in clinical environments.

As it is a big data project, the flow chart that this study will follow has its basis on a big data project. In a big data project, 4 intrinsically related steps can be found. First, the definition of the project must be stated considering the objective to be achieved. In this thesis, the goals and definition of the project were stated on the HYPOTHESES, OBJECTIVES chapter. It is possible to come back to the definition of objectives as new trends were obtained in the data after performing the analysis, so it can achieve or find out results that were not considered before. The following step consists of selecting features to analyze and find the whereabouts of the storage of all the data needed. In this step, it is essential to determine what information will be needed to achieve the defined objective. It is also important to know the different types of data available and how to store them, as it is well known that video images and information data stored as dictionaries will need a special database known as nonstructured to handle large amounts of data that has not the general structure of a table. In this step, the nature of the data and the characteristics extracted from the patient using the recollection tool need to be adapted for the next step. This step is called preprocessing, and it has many levels. Adapting the information to the data used in the analysis model step requires special knowledge of the

recollection tool and the model analyzed. The next step consists of analyzing all the data obtained using the recollection tool from the patients, which is preprocessed to be suitable for the analysis. It should be highlighted that since deep knowledge of both the model and the recollection tool needs to be reached to obtain a robust system with great results and reach the objective stated in the project definition step, all the different stages are connected and it is not necessary to complete and have a clear defined project to start working on the model and the recollection tool. The final step consists of the improvement reached or the value gained while completing the objective defined on the project or a step to enhance further the model analyzed, adapt the recollection tool to extract additional characteristics, to transform the information collected from the patient into more valuable data for the analysis model, or to obtain an additional goal to complete the project definition.

3.1.1. Data recollection tool: APP

The recollection tool consists of an application that works on all platforms (Computers, mobile phones, Android and iOS and tablets). It is coded and developed using a hybrid platform, ionic platform, and the typescript language Angular. It uses the PHP language to run the back end of the application and save all the data obtained from the questionnaire and different information introduced by the patient and the data collected from the environment. All the features available in the application will be stated in detail furthermore.

3.1.1.1. Code Development

The principal component of the application is the video collection source, a video interface to collect the objective information from the patient's eye. With the collection of a video of the eye, different parameters and characteristics could be extracted, analyzed, and compared in different weather conditions while using or not contact lenses. The interface, we will call front-end furthermore, and it is related to all the different features and characteristics of the application that the user can see and interact with, will be programmed using HTML5 (Hypertext Mark-up Language) (151), for the appearance of the features and the easy accessibility for the user CSS3 (Cascading Style Sheets) (152) is used for the development.

To have a clear understanding of how the application is coded, it is best to know how to perform or work with Object Oriented Programming (OOP). An object represents element properties, which are variables, and methods, which are functions. For further understanding, if you used to play MMORPG (Massively Multiplayer Online Role-Playing Game) games, the first thing you will be asked before starting the adventure is to create your principal character. This

character usually has different classes that can be chosen, meaning it can be a magician, warrior, templar, or thief. Another characteristic can be the number of live points, the health of our character, and a pseudonym. When an object is created, different properties, such as the characteristics stated for the personage created, are given. Additionally, our fictional character in the game can perform different actions, for example, attack another character, walk, run, buy and sell goods, or just make some dance movements. Those actions in the programming object are stated as methods. In general, when programmers talk about implementing methods, they are programming new functions for an object. In JavaScript and TypeScript code, objects are identified by using brackets for the definition {}.

The OOP is an abstract concept, meaning that all the elements available in a developed application or website are objects. An essential characteristic of the OOP is that objects can be literal objects, constructors, prototypes, or class objects. The literal object is used to transfer information. Suppose a property is set on an object and the data is needed on a different instance. In that case, the literal object is created to store the information (property of an object), and to transfer this information, for example, to the user (through the console), we will use a literal object. Constructors are used to easily create the maximum number of objects with the same structure. The constructor has the same code writing as the functions, and the properties we want to assign to the new object from a constructor are introduced as inputs inside the parentheses (). It makes it easier to create new objects from objects with the same structure and to insert objects as properties inside the parentheses of the constructor to assign an object as a property of a more significant object.

In JavaScript, the OOP uses prototypes instead of classes. A class will have inside, as explained for the object, methods, and properties, but also a constructor. To understand the prototypes better, we can try coding a simple object, using or not a constructor; we will observe that an additional property is created automatically. This property is, in fact, an object and has many properties inside. The most crucial property inside the prototype is the constructor. In this property, inside the prototype property, we can find the constructor used to create the object studied. If the object created was not done using a constructor, a parent constructor is found and responsible for creating all the objects in JavaScript (it is known as Object). In the same case, if we used a constructor inside the constructor property of the previous prototype, we would find a prototype with a constructor property named Object, the constructor parent. Object use is possible, enabling the creation of an empty constructor and assigning new properties and methods to use on the constructor. In the prototype property constructor, we will

find the methods created for the constructor, and they can be used even if the principal object is created but the method is not defined.

The most used function of the object prototype is the use, in a constructor, of a function (method) defined in another constructor. Even if the prototype is determined automatically, the object of the prototype can include methods as functions created in other constructors, allowing the objects created from a constructor to use a method available in another constructor by calling the foreign constructor; this is known as inheritance.

The class objects are introduced in TypeScript, a superset of JavaScript, meaning that all JavaScript programs are valid. Still, the syntax can slightly differ for safety reasons. The angular framework used in the development of the application uses TypeScript language. The class objects are objects with a defined structure that forms, together with other classes, what we know as the interface. In the class object, a constructor is defined to include other classes to allow inheritance, meaning that methods from the classes called in the constructor can be used in the current class. An example of a class is a page in the app development. Inside this class, a constructor will be defined with different classes going from ones enabling routing (page changing), authentication service (to detect if the authentication was made correctly), data service (to access and use data from a remote database), translation (select the correct JSON (JavaScript Object Notation) file to show the page in the language chosen) among many other classes that we will see in the application different coded pages.

Considering the OOP concept and inheritance between classes, we can know how the application's different pages, buttons, connections, and interfaces will be connected and work in a system called an App.

The HTML code is usually used in other technologies such as JavaScript, TypeScript, and CSS. It is the most basic component of the web, and it is used to display all features visible to the user. The elements that can be displayed are buttons, texts, and input boxes. The different components are developed using container elements marked with the quotation marks left (<) and right (>). The most common container elements are <head>, <title>, , <img>, <video>, <textarea>, <form>, <button> among many others. Each of the containers has different properties that can be given to the element container as the access of a specific JavaScript class or a method to perform at the click of a button. Furthermore, the different properties will be explained while they are being used. Inside the container, text is usually written to display, for example, the name of the button or to show the title in a <title> container

element. For the pages designed with the ionic platform, new containers are available to enhance the view for all the different platforms and work better when the application is finally built to work on Android, iOS, or as a web application. <ion-grid>, <ion-header>, <ion-row> are examples of attribute elements available in the HTML files for each page as a default option that will help to display web pages on a smartphone correctly.

To access the final application, please go to the following link: https://app.eyeh2020.com/

3.1.1.2. Video Interface

When researching the existing applications with a video interface, different plugins were found using the ionic framework and the angular or react platforms. The native ionic media capture plugin has also been tested, but it is currently only available for Android and iOS but not for the web. We must clarify that the plugins are libraries with different classes where methods and properties can be accessed to perform and reach the intended goal. In this process step, we must consider whether the library is compatible with the version of the angular and ionic frameworks being used.

Using an angular core library where the VideoElementRef property is available inside the ElementRef class has been the best choice for the project's needs and is easy to use. With the *ElementRef* from Angular, the recording is done via streaming. The size of the stream can be changed, and the stream's position can be easily programmed. This allows it to show the stream and the activity monitored while recording on the screen simultaneously. A text to read while the patient is being recorded (90) or an objective to stare at.

The video page inside the application is programmed using Angular with the Ionic Platform. The Angular framework is specific for creating a front-end app with a single page, while HTML and TypeScript are used for coding.

3.1.1.2.1. Libraries used in the video page

Firstly, the first libraries imported were Angular Core with *Component* (an interface; a set of classes), *ViewChild,* and *ElementRef*, which are classes being used on the page that is needed for the coding of *Components* and *Childs-specific* options for angular to allow the connection between the TypeScript logic coded and the visualization by the user of the different interfaces developed. The component has a selector, a set of other files that includes the HTML and the CSS codes (for visualization by the user of the various instances), and all the different TypeScript pages where the logic is coded. Two TypeScript files (*.ts*) are necessary to

highlight: the module, which is essential for organizing and managing applications by grouping related components, services, and other artifacts, and the page where all logic is coded and developed, where the methods and properties are used from each one of the imported classes. In the module TypeScript file all the libraries seen in the TypeScript file are also imported into the module file to make possible access to the classes, properties, and interfaces, among others.

In the page TypeScript file, we will talk furthermore only about the codes developed available on this file; the ViewChild decorator calls the video element class and the record video element class to perform the visualization on the HTML file of the stream captured using the front camera and the recorded video. The properties inside the video page TypeScript file *videoElement* and *recordVideoElement* will store the *HTMLVideoElement* interface that stores furthermore allowing the visualization in HTML of the stream and the recorded video when the recording is triggered and finished. Additionally, in the video element interface far behind the eyes of the user, a connection to the foreign air contamination and weather data is made possible on this page where the patient eyes are being recorded. The *UserService* (service class that differs from a page because the HTML code is not required) is being injected into this page and called in the constructor of the video interface page to access the different properties and methods inside the user service. A better explanation of this service is provided in the sub-chapter 3.1.1.4 related to the backend, where the connections are made.

The ionic/angular library is imported, and the *Platform* class is used in the constructor. This class is very important for recording because, at the time, it is easy to distinguish between videos recorded using the web from a computer or an Android device or using Safari. The final purpose of the *Platform* class is for the app to distinguish between Android, Web Explorer, and iOS. The difference in the platform used is found in the codecs (encoder-decoder of the digital signal) that the application records in the video files. In all common web browsers, the codecs available in the recorded video are WEBM (Web Media File). This video file property WEBM is not supported by the Matlab Environment (the vital fact that we will see further for the analysis of the video collected), so the analysis of these videos is not possible in the Matlab Entourage. The Safari browser records the videos using mp3 and h265 codecs compatible with the Matlab Platform, so the codecs to add to the video were not similar when the web application was accessed using an iOS device instead of an Android. To solve the problem, we used FFMPEG (Fast Forward Moving Picture Experts Group), which converts videos from any extension or codecs to mp3+h265lice (mp4). The FFMPEG was first tested in Windows and

Mac but was recently implemented using a Matlab FFMPEG toolbox by Takeshi Ikuma. The BSD (Berkeley Source Distribution) license is provided jointly with the programmed code.

The Angular/fire/auth library was imported to use the class Auth to ensure the user is identified as using the app. Extended information about the library and its use is available in the Authentication page subchapter 3.1.1.3.3.

The DataService page is being injected to send all the information to the MySQL and FTP (File Transfer Protocol) servers. Like the Auth class, the data service is explained deeply in the backend subchapter, where all the connections are made to the external weather and air contamination databases and the storage system developed.

The Lottie-web library is imported to use *Lottie* in the HTML file. Lottie is an animated picture that takes less memory space than the well-known gifs.

The Translate service detects the user's chosen language. Depending on that, the monitored activity and buttons are written in one language.

The ionic storage library is used jointly with the translation service. The language selection is stored using this library to make the variable available on all the needed pages. The Storage class allows different classes to access variables and properties easily from the classes available in the App entourage.

3.1.1.2.2. VideoPage class

Second, the properties of the *VideoPage* class are declared.

- The video element and the recorded video are declared using the decorator property *Viewchild* to allow access to the view properties in the HTML page. As seen in the subchapter before, this will enable us to visualize the user's stream and the recorded video.
- The properties for device location (longitude, latitude), air contamination data (airdata), and *videoStorage* are declared public.
- All the properties needed for the recording and streaming of the HTML video element are initialized. *videoElement* and *recordVideoElement* give the HTML video properties. The *mediarecorder* class will store the video. *recordedBlobs* property inside the *videoPage* class will store the blob info from the video. *isRecording* is false and will be a statement property to know if the recording has started. *downloadUrl* is used to store and create basically the video once the recording is finished. Stream is used to save the

html video element and is shown in the screen while the page is loaded indistinctly if the recording is on or off. The *Buffer* property will store the blob (Binary Large Object) where binary data containing the information related to the recorded video are stored, once the recording is over.

- Animation options are coded following the following instructions available in the Lottie library “https://www.npmjs.com/package/ngx-lottie”.

The properties of the imported classes are declared inside the constructor. The constructor will include all the library classes imported, except for the ones used in the decorator ViewChild because they are part of the angular core library. The properties of the constructor are the following classes: Router, UserService, Platform, Auth, TranslateService, Storage and DataService, and *ProcessXmlService.*

Inside the method *ionViewWillEnter* used to set up the page, three different things will be done before any other method or function once the video page is loaded. This is the purpose of the use of the method *ionViewWillEnter.* The first processes that are inside are getting the information stored in the language variable and a code to connect furthermore the data obtained from the video of the video to the PRO Instrument to correlate the subjective data received from the Patient Reported Outcome (for further information, see backend subchapter related to the databases connection 3.1.1.4). Once this step is performed, the user will now be able to see the stream. Then, the *isRecording* property is set as false, and it starts to get the data from the weather station using the user service. The *getConstraints* method will declare the ideal values for the video recorded. This is done to reduce the time to wait until the video is sent and ensure that the quality is sufficient to get the data from the patient's eye. Once all that is done, the recording starts with a timeout of 1 second.

The methods coded in addition to *ionViewWillEnter* are:

- The method *get_data_api* uses the injectable service from the connection to the weather station and the device location.
- The *animationCreated* method to use the Lottie animation of the camera.
- The *startRecording* method, where the recording is done from the stream using the *MediaRecorder* variable and setting up the codecs. Using the statements *try* and *catch*, the recording will consider the native platforms and the hybrid one, giving the mime type WEBM to the web browser and MP4 to the native platform. The recording

countdown is also programmed, giving 60000 msec to the recording and 1000 ms to set up the page before the recording.

- The *stopRecording* method has the instruction to stop the record. It is called once the countdown of 60000ms is finished.
- The *playRecording* method is called to play the video recorded, it is called once the record is over.
- The method *OnDataAvailableEvent* ensures that the stream data recorded is not lost during the record.
- The method *OnstopRecordingEvent* creates the video from the data generated during the recording. The video is stored as a blob. Depending on the platform, the codecs will be MP4 or WEBM. The video URL is also saved to be replayed before sending confirmation in the variable *recordVideoElement.src*.
- The conversionBlobToBase64 method converts the blob to base64 code and sends the data to the FTP *se*rver.
- The method validate will consider the user and create a unique id for the current data collection. It will send the video using the *save_video.php* to decode the base64 video into a video file. It will also send the data patient to the MySQL server using the *add_data_patient.php*.

3.1.1.3. App features

The different functions performed with the app that are not related to the video interface are as many as the different data needed and obtained for the project, which were included in a successive process to obtain the best quality from the patients' data. The framework used to develop the different functions and features was chosen depending on the version of the frameworks and languages used to develop the video interface.

The different functionalities developed will be stated in the next sub-chapters. We have to take special consideration of how the Ionic platform is able to develop an app with different pages, where the Angular framework is used to develop each one of the pages, as the purpose of the creation of the Angular framework is to be able to develop and code a single page.

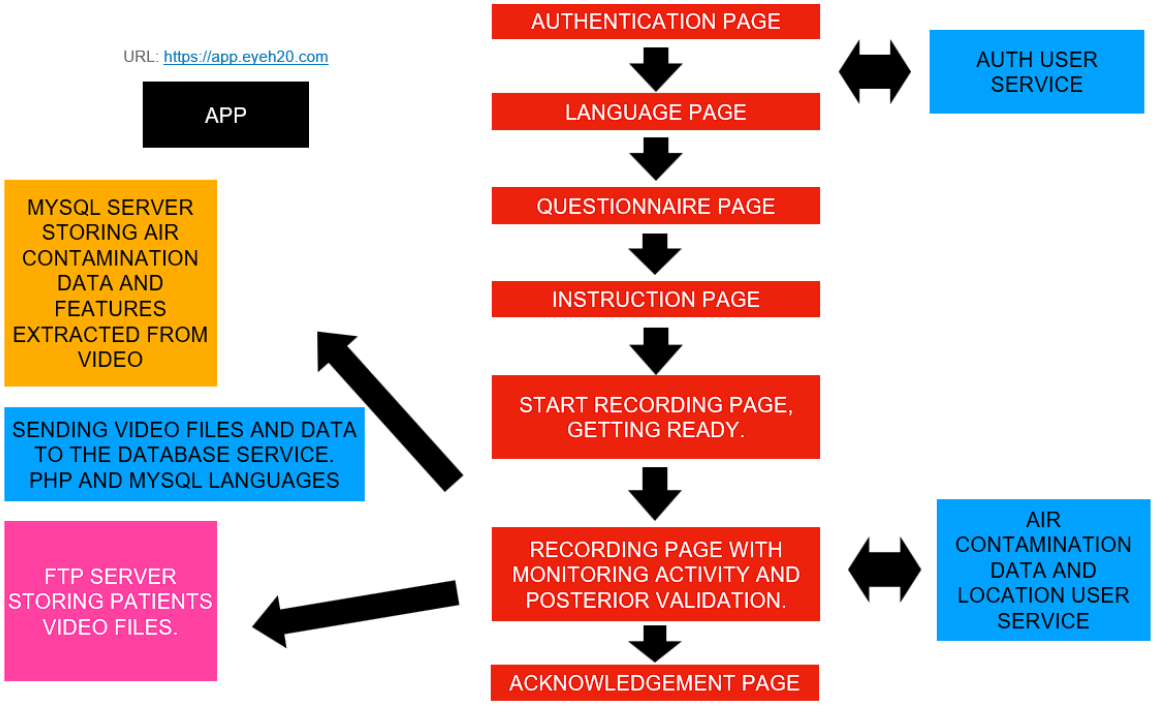

Figure 3-1:App structure

3.1.1.3.1. Installed programs for coding and development

- Node.js is used to interact with the ionic ecosystem.
- Visual Studio Code as a code editor.
- Command-line interface/terminal (CLI):
    - Windows users: command line (cmd) or Powershell CLI, running in Administrator mode.
    - Mac/Linux users: Terminal.

Using the terminal, for each library to work, they must be installed with the *npm* command. In the *package*.json file are each of the necessary modules. With the command $*npm install package.json*, all libraries will be installed. To see all the libraries installed to make the app work on all 3 platforms, as it is possible to build the application as an iOS App, as an Android App, and as a Web application. For instance, the web application was finally decided as it is better to perform this task with the protection of the patient privacy to keep track of the authentication on the application and ensure with a human verification step that all the information related to the privacy and purposes of the application is read and understood. There is a way to contact the principal researchers who are developing the study by the patients. Among these, building an application and reaching the Apple and Android marketplaces is possible.

3.1.1.3.2. Main characteristics, home page.

In addition to the package.json file, other important files that require attention and are related to the app's main functioning are the app.module.ts, app-routing.module.ts, and *app.component.ts*.

The *app.module.ts* includes all the libraries used on the home page, which, in our case, corresponds to the ribbon at the top of the app that appears on all the pages.

In the *app-routing.module.ts,* all the different pages accessible to the app have their routes defined using a constant, which is an object that includes constant methods and properties that represent a route configuration for the router service. The library of authentication guard of Firebase, a service from Google used to manage the authentication service, that will be developed deeply on the authentication page subchapter, has implemented the methods *redirectUnauthorizedToLogin* and *redirectLoggedInToVideo* that are used inside the routes constant object to check if the patient is logged correctly into the app and restrict the access to the pages where the correct authentication is a must.

The *app.component.ts* file includes all the logic of the ribbon and in the same way for the video interface explained on the previous subchapter(include reference), the translate service, the router and authentication services are called in the constructor and in the *ionViewWillEnter* function the storage is accessed to see the language selected by the user to show the different options available; in this case go to the language page selection and logging out. This option is available as the *router* class is defined in the constructor and the *logout* function uses the method *logout* from the *authService* class, and the *navigateByUrl* method is used from the *router* class to redirect to the log-in page.

3.1.1.3.3. Authentication service and log-in page

Once the video interface how to code knowledge is acquired in the preliminary step (considering that the explanation in the subchapter 3.1.1.2 is already complete) and there is knowledge of how the routing of the app has been made a subchapter (3.1.1.3.1 and 3.1.1.3.2). The authentication service further allows access only to informed patients of interest; sign-up and login options are necessary. This is also done to let only authorized users store data on the project server. The authentication service and the login page are explained in the following paragraphs.

The language used is the same as for the video interface (Ionic platform and Angular framework) but also uses the PHP (Hypertext Preprocessor) and MySQL languages.

To create the service, the Auth interface and the promises *signInWithEmailAndPassword*, *createUserWithEmailAndPassword*, *signOut*, and sendPasswordResetEmail are imported from the Firebase library. Promise's purpose is to handle asynchronous operations, providing better control over the code flow.

The log-in function is created to allow the registered user to be verified. The try and catch method is used to catch angular errors. What is inside the try will be compiled, and if it is correct, the user will be returned. If the user variable is assigned an incorrect or null value (code error), null will be returned.

User check-in Firebase is done using the promise *signInWithEmailAndPassword* imported from Firebase, and it requires 3 inputs: the authentication class, the email input of the user, and the password input. This promise has 2 methods or functions that will accept or reject the logging. The access will be done if the email and the password inputs are found in the firebase authentication database. The function created in the authentication service to use this promise from @angular/fire/auth library is called *log-in.* In the same way, the registration process is handled by Firebase if the input of email and password follows the rules of creation template of user and password, and the email input is not found on the database. In the same way, *signInWithEmailAndPassword* promise needs the same 3 inputs and the output of the promise is two (registration or not of the new user). The function created in the authentication service to use the promise *signInWithEmailAndPassword* from @angular/fire/auth library is called *register*. *sendPasswordResetEmail* is the third promise used and only needs the *auth* class initialized on the constructor of the *AuthService* class and the email input by the user. If the typo constraints are fulfilled and found in the Firebase authentication database, the Firebase authentication service email is sent automatically using a template to the user email. The method created in the *authenticationService* class to use the promise *sendPasswordResetEmail* from @angular/fire/auth library is *resetPassword*. The last promise is *signOut, which* requires only the *auth* class as input. The method created in the *AuthService* class to use the promise *signOut* is *logout.*

The login page is a form to fill out correctly. Its structure is very simple. If the filled-in fields are correct, the page redirects you to the instructions, and if they are wrong, it asks you to try again.

- First, the angular/forms libraries are imported, the router is able to redirect, *AlertController* and *LoadingController* are alerts for the user that the page is being loaded, and finally, the authentication service is created previously. The *get* function is used to read the fields entered by the user.
- Subsequently, the form is created in the *ngOninit* function, which is characterized by the fact that it is executed first when the app page is compiled.
- The asynchronous *register* function allows the app page to register the user. This button inside the login page will redirect to a page where the consent is available before calling the *AuthService* class and checking the boxes of the permission of acceptance of the approved protocol (Subchapter 8.1).
- In the same way, the *resetPassword* asynchronous function will send the email to the registered user for recovery password. It is implemented in an HTML file under the button 'forgot password.'
- The *login* function allows checking the validity of the data entered in the authentication form. The login function, in turn, uses the login function programmed in the authentication service to check the veracity of the data. If the data are correct, the router property and the *navigateByUrl* method redirect to the instructions page. If the data are unavailable in the FireBase database, an alert message with the following text: "Login failed, please try again" appears on the screen.

The HTML programming of the log-in page allows the following display:

- First, the *logo_user* image is inserted using the *img* class. Then, the login page's typescript-programmed form (.ts) is inserted. Two <*div*> containers to highlight are the *email* class and the *password* class <*div*>, where the input area of the form is programmed in HTML.
- The <*div*> *sec-2* class container includes the email input zone. In it appears an icon, a text zone with an example « username@gmail.com,» and an « email » control so that only texts in the form of email addresses are valid. Red notes also appear if the prior validity check is not met.
- In the <*div*> *sec-2* class container, within the *password* class, an icon is programmed, a text zone with an example "……..." and a password type control. The password must be at least 6 characters as it appears in the programmed note below the text area if the check is not met.

- The control programmed in the text zone of the email and the password come from the *validators* class of the *angular/form* library.

All the information stated in the html file and the functions in the TypeScript file, results on the login page displayed in Figure 3-2.

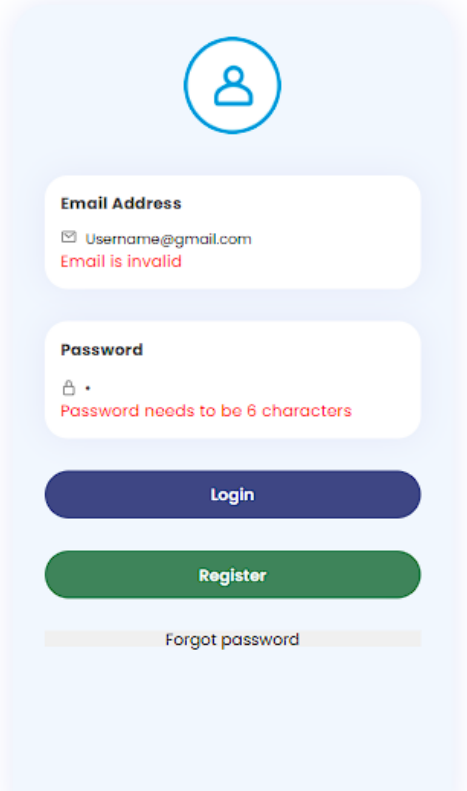

Figure 3-2: Login page

3.1.1.3.4. Language page selection

Once the patient introduces the credential correctly, the language selection page is displayed in Figure 3-4. The language page selection has four flags and four options: Spanish, English, French, and Portuguese. The most critical import for this page is the ionic storage. The storage allows us to store variables that can be used on all the app pages. For language visualization, we use the property *langue* (language in French) to store the string corresponding to each language. The language is saved as a variable in the common app for all your pages. This variable selects the text to be displayed from a file for each language. Each language has a document with the translated (where all the texts being printed in the app, besides the questions, are translated in each language). We select 'uk' for English (default), 'es' for Spanish, 'port' for Portuguese, and 'fr' for French. The method *langue is an asynchronous function in the class LanguePage.* In this method, the storage is created for the variable "langue" using the *storage*

class from the library Storage to store information accessible from any other class or component inside the application, such as the information about the language selection. We also have to note that in this step, we create the *storage* for the *timedata* constant that with the information of the current time when the user selects the language, we can furthermore locate all the information given by the patient going from the questionnaire we will show in the following subchapter (see questionnaire page), the video recorded (see video interface) and finally the weather and air contamination data obtained from the different external databases using the location of the nearest station. The *timedata* constant code will be different each time the user enters the application to fulfill all the steps (questionnaire, video, and if it is the case, the external Patient Reported Outcome).

In the HTML file 'langue.page.html', where the user interface is coded, every country flag is a button. In our case, Spain, the United Kingdom, Portugal, and France. Those buttons are programmed so that if you click on an image flag, the property *langue* will get its value to select the corresponding idiom file. The language page is performed to remove all barriers so that patients can use the applications in an international language environment like a big country capital (in our case, Madrid). The container used to display the flag button is <*img*>, and the file location provided for each one of the Portable Network Graphics (PNG) files is done using the attribute <*src*>. The attribute <*style*> is used to set the display characteristics, for example, the margin length in pixels and the width in pixels, among others.

The language selection page is shown in Figure 3-4. Both visualizations are proposed, and web and mobile phones are indistinctly Android or iOS.

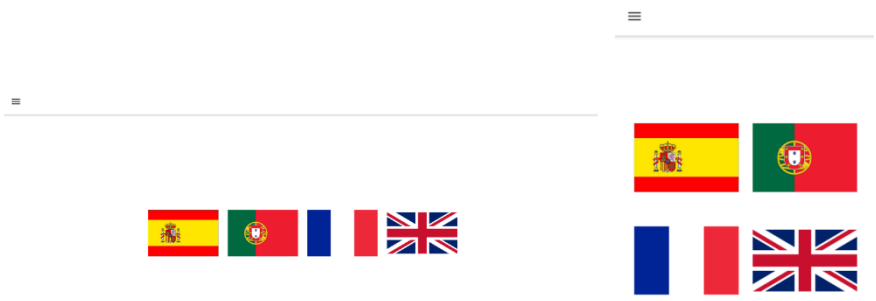

Figure 3-3: Language selection page left web view, right mobile view.

3.1.1.3.5. Questionnaire page

After the language selection, the app may include a questionnaire about the subjective quality of vision, comfort, and objective variables about the patients (age, ethnic group, etc.). One of the future uses of the app will be to assess patients' subjective comfort when wearing contact lenses, so some questions have been included in this field to obtain the subjective behavior of the contact lens that interacts with the physiology of the patient's eye.

The questionnaire gets different types of information about the user and saves it to the MySQL database. For example, personal data, demographic data, or quality of life the patient has while using contact lenses. The technique used to code the questionnaire can be described as follows. First, a table with all the questions and the corresponding answers in different languages is uploaded to the database. The questions have the same structure as PRO-Instruments developed to test the impact of dry eye or contact lenses on quality of life (153), (154), (155); every question has five possible responses, but the patient can choose only one option. In addition to the options, the question language is also identified in the database. A specific column is used to store the language info. This ensures that different patients can complete the questionnaire worldwide and that the language barrier is removed. Currently, 21 questions are available for patients in four different languages: English, Spanish, French, and Portuguese. The questions are divided into three groups. The first group is answered once when accessing the app for the first time (longitudinal size of the eye, contact lens brand, ethnicity, age, biological sex, if they are a contact lens user and since when, and periodicity of use). The second group is related to visual comfort. This set of questions appears only if the patient has used a contact lens during the day. The third group corresponds to the one that appears every time the user is accessing the app. The questions on this group correspond to temperature and humidity if the user is performing the data recollection inside the conditions are known, and the current state of use of the contact lens during the video or during the day.

The development of the questions of the three groups presented have basically the same structure for the html and typescript files. The difference on the front end will not depend on the appearance of the different questions (demographic question and current state of contact lens use) but if an image is displayed for the understanding of the question and the answer given by the user requires a text or a selection of possible already present answers but not if those questions are only presented to the user on the first time or every time the user accesses the application for the recording of a new video.

3.1.1.3.6. Eyesize page

The questions with pictures and one possible input answer will appear first and be located on the *eyesize* page. The methods inside the *eyesizePage* class are *ionViewWillEnter*, *ngOnInit loadQuestions, nextQuestion, save_question,* and *nextQuestionnaire.*

In the *ionViewWillEnter* method, as on almost all the pages that display texts, the storage class is being accessed to use the methods get, set, and remove to obtain the language stored in the language selection page and to access the *timedata* code set on the language page selection to keep track of the answers given by the user. Once the data of the *language* and the *timedata* code are available, the *loadQuestions* method is called with both the *timedata* and language input variables.

The ngOninit method is used to create access to the *formBuilder* class and use the method *group* to create a *FormGroup* instance. It is a formulary that enables the correct storage of the input made by the user. The set of validators will allow only information to be stored and sent to the database if it is within the constraints specified by the developer. For example, in the eyesize *FormGroup* class, only one property is set called lonpalp, and the use of symbols and letters is not allowed. As this form will be used to send information related to the longitudinal palpebral dimension of the right eye measure of the eye patient in millimeters, only numbers from 0 to 100 will be allowed. In the other *formGroup brand* class, as it is used to store and send the brand's name, special characters, letters, and numbers are allowed.

Inside the *loadQuestions* method of the *eyesizePage* class, using let prefix *parametre* variable is called with three different properties, language, user credential, and *timedata* code. Those parameters (parametre in French) will be the input for the dataService method *get_oneanswer_questions,* which functions to send a request using a php file language to access the MySQL database information stored related to all the questions in the database with the structure of one input text or number answer. Looking inside the php request and considering the access of the MySQL database, different tables are available. The *q_eye_size* will include the questions that only appear on the first time the users access the database. The *q_temp_hum* includes all the questions with one possible text or number input that will appear every time the user access to record a new video of the eye. The loading of the first table of questions *q_eye_size* will be done if no answers related to these questions is found on the *answers_eye* table where all the answers from the patients to the questions with input answers are stored. The properties *all_questions, cont_questions* and *currentQuestion* stores all the data available

in the respective question table from the database, access the number of questions available and the current question, respectively. Those properties are accessed from the html file to correctly display each of the questions and their index.

The *saveQuestion* method inside the *eyesizePage* class, its principal goal is to save the input answer of the patient into the MySQL server. Property *id_question* stores the property *id* from *currentQuestion* class, and the property *selectedValue* stores the answer input using the get method from *eyesize* or *brand formGroup* instance declared on *ngOnInit*. In the same way as for the question loading, the body class is used to read and send properties through a php post to send the information to the database. The answers will be stored in the table *answers_eye.* The information stored in the table corresponds to the user credential code found in the currentUser property from the *auth* class injected into the *eyesizePage* class. The id of the question, the current question, the response stored on the *selectedValue,* and *timedata* code obtained on the *ionViewWillEnter* method from the *storage* class. The POST request is sent through the method *save_eyesize* from the *dataService* class injected into the *eyesizePage* class.

The *nextQuestion* method from *eyesizePage* class will be triggered by the user when the previous question answer option is chosen and saved. Reset the property *response* from the class *eyesizePage* giving the null value for safety purposes, so the input is empty in the next question. *CurrentQuestionIndex* is changed to the next unity to display the next question. An if/else statement is coded to check if the index of the next question is still inside the boundaries considering all the questions read from the database. if not, the *currentQuestion* property is set to false. This property is read in the html file, when it is not a number quiz completed text input is shown, and the button that triggers the *nextQuestionnaire* method is available. When the *currentQuestionIndex* is a number inside the constraints set by the length of the questions extracted, html file will load the question considering the image to display using the **ngIf* statement to select the correct image and the correct form group.

3.1.1.3.7. Question page

The same structure is used for the question page; in this case, only the differences between them will be explained. We must consider that in those questions, no image has to be displayed. We must also take special consideration of the number of possible response choices in each question loaded from MySQL, knowing that the maximum number of different responses is six.

This time, we will not find the *ngOnInit* method in the typescript file as there is no need for a *formGroup* instance to store user input. The possible answers will be listed, and the *selectValue* property of *questionPage* will get the text of the answer chosen. The answer chosen will be stored on this property and sent through PHP to the MySQL server.

The different possible answers are displayed in the html file using the *ngIf statement to see if the response column in the questions table on the MySQL server loaded using the *loadQuestion* method is empty or has an array of characters. The option will not be displayed to the user if it is empty. Suppose response 1, response 2, and response 3 contain an array of characters, but columns 4,5, and 6 are empty. In that case, the question displayed will have three possible answers, and the user can only choose between those 3. In the other case, six options will be displayed if all six columns have text.

The questionnaire pages are shown in Figure 3-4, Figure and Figure 3-5. Both visualizations are proposed, and web and mobile phones are indistinctly Android or iOS. For further information, go to the annex subchapter of the question page to check the developed code.

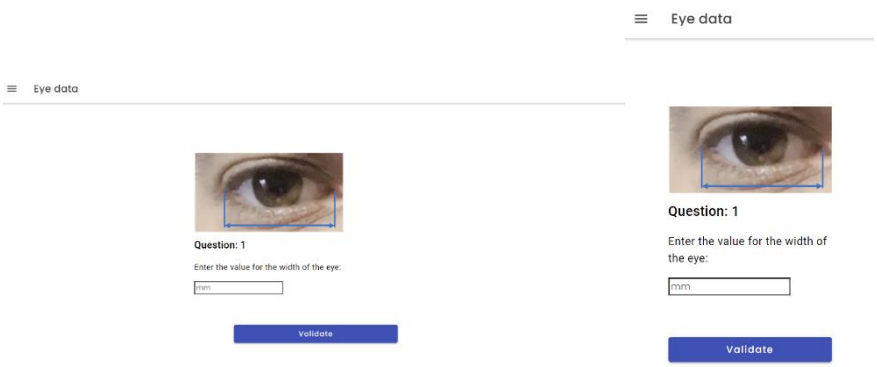

Figure 3-4: Questionnaire page numerical input left web view, right mobile view.

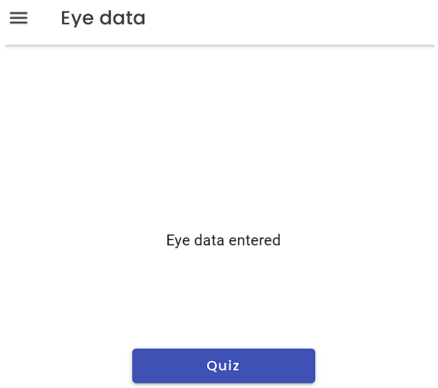

Figure 5: Display of the size page when all the questions of one possible answer are already answered.

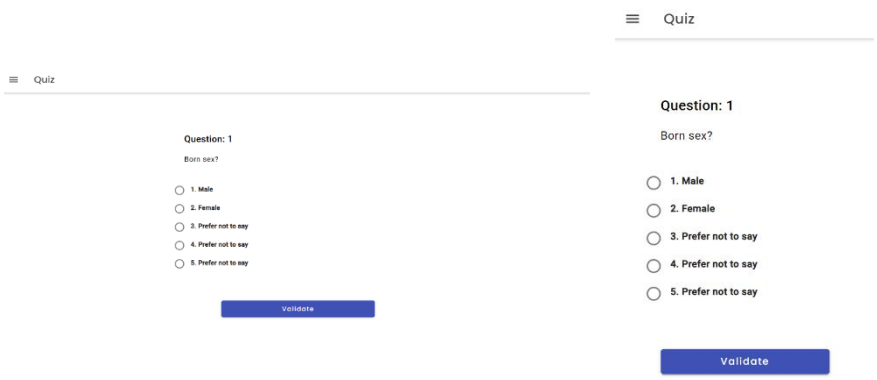

Figure 3-5: The questionnaire page options input left web view, right mobile view.

3.1.1.3.8. The instructions page

The instruction page is the first page that appears after correctly entering the user data. The only special thing about this page is the use of the "stepper," which the user uses to correctly set each instruction. On this same page, using the stepper, it is possible to divide it into four sections: before recording, camera positioning, during recording, and finally, after recording.

This instruction page is made as follows:

- First, the angular Component, *OnInit* interfaces, and the *NgZone* and router classes are imported.
- The first Component interface is essential for the creation of each of the pages. The *OnInit* interface is used to first execute all classes and instances within this interface.

- The *router* class is used to redirect each of the pages and connect them. It is essential on each of the pages that we want the user to visit. A page without the import of the *router* class cannot be accessed by a button. It is possible only if the URL of the address is known.
- The stepper imports from the angular library in the *cdk* folder. In the component, it is declared as a provider.

The most important thing on the instructions page is the ability to select the language seen on the previous pages, as the procedure is to first read the stored value of the language property (the same for every page that has texts that we want to display in different languages), and then simply display the instructions with the images as seen in Figure 3-6. Both visualizations are proposed, web and mobile phone indistinctly Android or iOS.

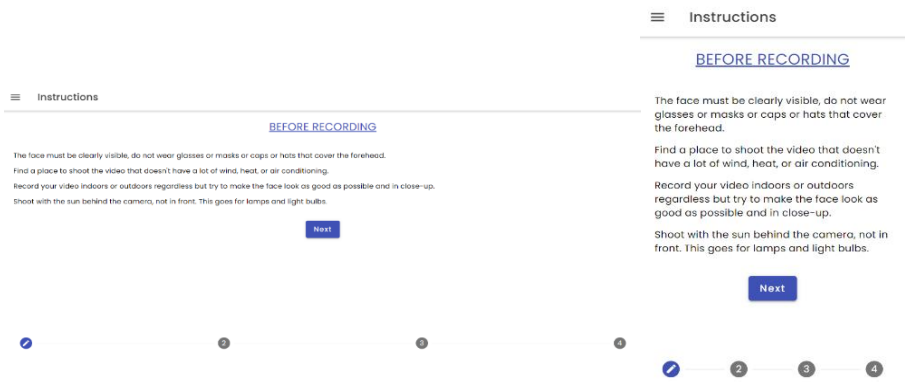

Figure 3-6: Instructions page left web view, right mobile view.

3.1.1.4. Back end. MySQL database and ftp server.

A MySQL database and an automatic connection have been created to store all the data extracted from each patient. An ID is given to each video the patient takes using the app, previously known as the *timedata* property stored using the *Storage* class. The same ID is used for the data collected from the current time and position during the video recording. To every air contamination data id, a video file with the patient's eye physics is available. The video file is stored on the ftp server. The data service job is to send the videos to the server, and the air contamination data collected and stored as the property *airdata* inside the *VideoPage* class from the atmospheric station to the MySQL database.

- The HTTP (Hypertext Transfer Protocol) client, as is the location and contact service with the atmospheric database, is imported first.
- Then, the rxjs map class is imported. Map “Applies a given project function to each value emitted by the source Observable and emits the resulting values as an Observable”. In other words, an input value is given with the help of mapping an output value dependent on this input. This output value is returned as an observable. Finally, we import the injectable angular, since it is a service injected into any app page or component. In our case, we will see that this is injectable on the video recording page.
- In the add patient function, we can see how the map function adds the values collected from the atmospheric station to the database.

The PHP files found in the php folder are as follows:

- The data service assigns the data variable to the data. In this PHP file, the content of the variable is detected. For example, if there is no e-mail in position 0 of the data, the data collected is incorrect, so the data variable takes a null value. Subsequently, the user's ID, email, air pollution data, and the time in which the measurements are made are collected.
- Next, to correctly enter the data into the MySQL database, it is verified that each of the requested fields of the meteorological data is present at the time the station collects data. If this is not the case, as is normally the case, the variable is assigned a null value and entered as null in the MySQL database.

	id	1	id_iaqi	dew	h	no2	o3	p	pm10	pm25	so2	t	w	wg	b
Editar Copiar Borrar		109	1693086243083			9	87								
Editar Copiar Borrar		108	1693083867919			4	86								
Editar Copiar Borrar		107	1693082988254			15	91								
Editar Copiar Borrar		106	1693082607194			15	91								
Editar Copiar Borrar		105	1693081865057			8			17	3					
Editar Copiar Borrar		104	1693069355736			4			63	5					
Editar Copiar Borrar		103	1693066667529			9	80								
Editar Copiar Borrar		102	1693044966810			13	67								
Editar Copiar Borrar		101	1693057820507			13	67								
Editar Copiar Borrar		100	1692972341496			12	88								
Editar Copiar Borrar		99	1693001073396			7			64	26					
Editar Copiar Borrar		98	1692994189387			20			49	11					
Editar Copiar Borrar		97	1692992343615			20	64								
Editar Copiar Borrar		96	1692988702580			16	72								
Editar Copiar Borrar		95	1692987829666			17			72	21					
Editar Copiar Borrar		94	1692974665319			9			47	14					

Figure 8: Air contamination data table in the MySQL database, where id_iaqi is the time code of each data record. No2: nitrogen dioxide, o3: ozone, pm10 and pm25 particles of 10 and 2.5 microns.

	id	id_iaqi	email	ce	ta	vmax	vv	dv	prec	pres	hr
Editar Copiar Borrar	39	1689353433347	tgb6zdgddm99vaqk6mcs@eye.com		27	7.6	3.2	190		857.2	36
Editar Copiar Borrar	38	1689352788580	tgb6zdgddm99vaqk6mcs@eye.com		27	7.6	3.2	190		857.2	36
Editar Copiar Borrar	37	1689352613183	gaaj5gxvor3tprqhmk8s@eye.com		34.9	10.7	4.4	263		937.3	16
Editar Copiar Borrar	36										
Editar Copiar Borrar	35	1689324616866	gaaj5gxvor3tprqhmk8s@eye.com			3.3	0.6	150			
Editar Copiar Borrar	34	1689290790152	gaaj5gxvor3tprqhmk8s@eye.com		26.8	2.8	1.4	4		940.8	39
Editar Copiar Borrar	33	1689290374882	gaaj5gxvor3tprqhmk8s@eye.com		29	4.8	1.2	305		940.7	34
Editar Copiar Borrar	32	1689290374882	gaaj5gxvor3tprqhmk8s@eye.com		29	4.8	1.2	305		940.7	34
Editar Copiar Borrar	31										
Editar Copiar Borrar	30	1689286026478	gaaj5gxvor3tprqhmk8s@eye.com		29	4.8	1.2	305		940.7	34
Editar Copiar Borrar	29	1689242741086	gaaj5gxvor3tprqhmk8s@eye.com			4	1.1	16			
Editar Copiar Borrar	28	1689237468255	gaaj5gxvor3tprqhmk8s@eye.com			3.1	1.4	55			
Editar Copiar Borrar	27	1689176837614	gaaj5gxvor3tprqhmk8s@eye.com		32.3	9.9	3.1	274		940.4	17

Figure 3-7: Weather data table in the MySQL database. The email used is anonymized using 20 alphanumeric usic codes. ta: temperature, vmax: maximum wind speed, vv: mean wind speed, dv: wind direction, pre: precipitation precipitation, pres: pressure, and hr: relative humidity.

- Subsequently, using the file 'config.php', where all data from the server and the MySQL database are stored, each of the selected meteorological data is inserted. In this case, the id values of the user (different code given to each of the videos regardless of the user) is the same code for video and collected data), temperature, maximum wind speed, mean wind speed, wind direction, rain precipitation, pressure, and relative humidity.

src > assets > json > config.php

```php
<?php
    header("Access-Control-Allow-Origin: *");
    header("Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept");
    ini_set('display_errors','on');

    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "youssef_app";

    $conn = new mysqli($servername, $username, $password, $dbname);
    $conn->set_charset("utf8");
?>
```

Figure 3-8: Configuration.php file to connect the application to the MySQL database. In the picture, the database stated is on a local machine.

3.1.1.4.1. Access to the weather and air quality database.

Once the interface video is set on and the main purpose achieved and explained in this paper, the next objective was the identification and connection of accessible environmental databases (temperature, humidity, pollution, etc.). To complete this task, we started looking at existing databases that calculate environmental conditions such as temperature, humidity, and, specifically, pollution. Numerous databases, such as Apple and Microsoft Weather, were consulted, and EuropeAirApp was found in the iOS store. As the need was to obtain the conditions precisely but also collect the highest amount of data quickly and to be comfortable for the patient, using APIs (Application Programming Interface) that connect to global databases of air contamination and weather data was the solution (156). As the study will take place in Madrid, the best way to avoid using an API that will require monetary resources is to find the data in the Spanish open data set finally.

As the targeted patients of this first study live in Madrid, and due to possible commercial agreement in terms of license, we used the following API (157) that connects with the air quality database of the city hall of Madrid, where the data are free for use (158). To extract the atmospheric parameters, the AEMET(*Agencia Estatal de Meteorología*) OpenData API is used (159). AEMET is the Spanish National Weather Agency.

To connect the web application with video interface to the Madrid Air Quality Database (158), the following API available on the following website API is used (157). The setup is available on the website; no token is needed to connect to the Madrid air quality database. The documents are available at the current time (González, 2019).

Following the documentation, we could connect to the database, creating a service to get the device's current location and date to contact the HTTP client with the station and date to get

the information from the nearest location center. All station's locations were uploaded to the APP to find the nearest station with the corresponding number. The distance is calculated considering the position of the device of the user to the 24 stations located in Madrid uploaded with the corresponding value of longitude and latitude. To find the device location, the geolocalization method from the JavaScript web API (160) is used. Once called, it stores the position of coordinates longitude and latitude in different variables after printing the values on the console. Then, with location position known, the *getData* method gets the information t from the nearest statiohrough php request (the php file location is given inside the *dataService* class) using the class *HttpClient* class from the angular common library *http* to send the php request. Using the get method inside the *HttpClient* class and the php file location to access furthermore the air contamination database through the link of the Madrid air quality data API (157) to perform a successful connection. We extract all the information from the station using the php file. The data obtained were the concentrations of nitrogen dioxide, ozone, 10-micron particles, 2.5-micron particles, benzene, carbon monoxide, and sulfur dioxide concentrations. After some months of working, this method was finally shut down due to an update of the main web page where the data were collected. The new method, which involves only changing the php file, consists of reading an XML (Extensible Markup Language) file provided at the following link: https://www.mambiente.madrid.es/opendata/horario.xml. We must consider that this method no longer needs the use of an external API and a service is developed in the app to access the data in XML format. To keep the structure of the video interface page somehow intact, we used the same method, named *getData*, to access the specific php file and read the XML file to get the actual air contamination data. This information is updated each hour, and the previous lecture can also be accessed. The last verified input of data is recognized with the analysis of the XML file, and the explanation provided by the open data service is used. For the obtention of the nearest station, the same technique is used as the database accessed is the same one used with the previous database, so the station codes are the same. The main change is to read the information using JSON from an XML interpreter. In the PHP file, the URL is loaded using the XML interpreter and converted to JSON code with the function *json_encode* to decode and read it as an array. The structure of the resulting array is known, and the data of the last our are accessed for each component of the nearest station. This time, the XML file accessed has the information on all the stations, and all the measured components of each station are there. For each statement, we need to extract all the different measurements of the nearest station.

To extract the data available from the AEMET open data set, we used the PHP language to connect to the webpage using the *cURL* library (libcurl) to transfer data using various network protocols. As for the air quality data provided by the Municipality of Madrid, the location is used to find the nearest weather station in Madrid that is owed by the Spanish national weather agency. To extract the data, the AEMET open data is provided with the station number in addition to the API token provided following the official access of the Spanish Weather AEMET agency. Data provided by each AEMET station in Madrid include temperature, highest wind speed, mean wind speed, wind direction, humidity, pressure, and rain precipitation.

After the data are recorded, the patient's location is erased so that it cannot be accessible or recorded.

The backend consists, in addition to the tables shown on the MySQL database, where all the data related to the weather and air contamination are available, the tables with the questions, answers, patient connections registration, and consent acceptance conforming to the storage system where the information is retrieved by the app for questions display and stored, from the app front end (air contamination, weather, answers to questionnaire and patient video metadata).

3.1.2. First Algorithms developed for the analysis of video files for eye feature extraction

3.1.2.1. Short summary of the thesis section corresponding to the video analysis collection.

This document explains each of the steps that the app-recorded video files follow as soon as they enter the eye project FTP server. They are downloaded to the main researcher's local personal computer for subsequent analysis until objective results are obtained from the patient's eye. It should be considered that the video record that is finally reflected on the server is the video of the image of the eye, not that of the face.

3.1.2.2. Explanation of the cropping system used for privacy safeguard.

In the first place once the video of the face from the patient is sent to the server, as programmed and coded in the app recollection tool, the first step done in the local analysis machine accessible only by the principal investigator, and almost constantly connected to the remote app server property of the eye project and subsequently to the Complutense University of

Madrid, is to program a scheduled automatic execution of the following task. Using the Task Scheduler software of the windows 11 of the local machine, we program the execution of a python script that has a clear purpose of first, download each 30 min all the videos loaded in the remote ftp server.

The programming of the task is done using the interface of the Task Scheduler software of windows with the create task option.

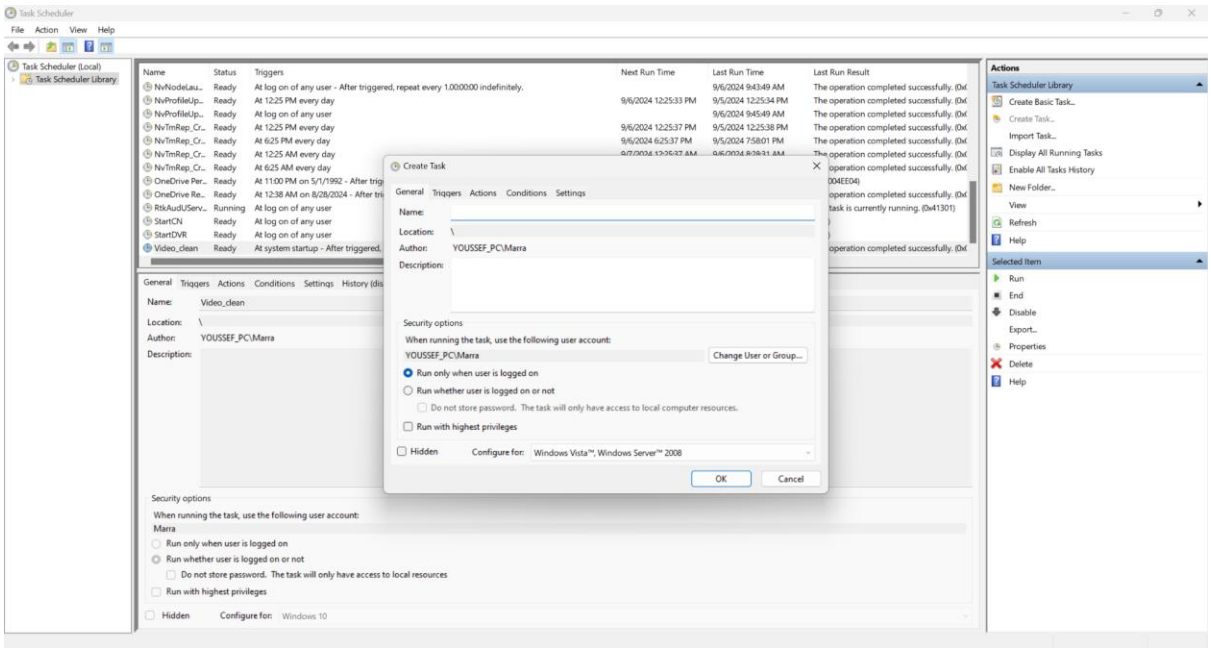

Figure 3-9: Interface of the Task Scheduler software

The file address with the python script is included in the actions tab, and on the triggers tab it is coded when the task is going to be performed. In the first place the script is executed at startup of the computer, then after triggered, the action is going to be repeated every 30 minutes indefinitely.

The script explanation is as follows. First, the FTP library is imported from the FTP class, which specifies the client protocol and makes it possible to connect to the FTP remote server. To make it work, first, a variable with the FTP server address as a string is defined, the username to be used to access the server, the specified password needed, the local directory where all the files will be stored, and finally, the remote directory of the FTP server where the available face videos are stored.

After defining the variables, the function whose purpose is to download and delete face video files is defined and coded.

The first step of the download and delete process is the connection to the FTP server. This is done with the *ftp* class method using the address of the FTP server as input property, and the output class is now *ftp*. The login method of the ftp class is used to introduce the properties user and password to complete access to the remote server. with the *cwd* method of the *ftp* class, we access the app's folder where the videos are stored. To retrieve the information of all the videos available in the directory folder, we use the *nlst* method and store the information in the files property.

To access each one of the files a for counter is performed, where *filename* registers each one of the files available in the remote folder of the FTP server. The download is performed by first creating a string with the full filename, including the local folder path where the file will be available on the local machine. Furthermore, with the open method, we create a new binary file. With the help of the method of retrbinary from the *ftp* class with the filename in the server and the method written to the *local_file* class where it is stored the new binary file, the download is performed. During this process, from the start, where it is identified a video file a console print information is done using the print method with the filename being downloaded, and when the writing of the new binary file is already completed, a console print informs us of the completion of the download. Afterward, the file is deleted from the remote FTP server using the delete method from the FTP class, and again, when the operation is finished, a console print is shown.

Once all the new files have been downloaded to the local machine and erased from the server to protect the privacy of the patient face image, the *quit* method from the *ftp* class is used to log in from the FTP server. Finally, errors are printed similarly on the console if an error occurs during the script with try-except statements. To enable the execution of this Python script that can be accessed by the Task Scheduler, a bat file is created from the script.

To create the bat file, a new editor text file is created, and a simple command line is used to perform the execution of the Python script using Python software, in our case version number 3.12 (161)In the .bat file, the full address of the Python software is written jointly separated with a space, as is the full address of the Python script. Note that for the Task Scheduler software, the action tab will include the full path of the batch file (.bat) and not the Python script.

```python
def download_and_delete_mp4_files():
    try:
        ftp = FTP(FTP_SERVER)
        ftp.login(FTP_USER, FTP_PASS)
        ftp.cwd(REMOTE_DIR)

        files = ftp.nlst()  # Get the list of files in the remote directory

        for filename in files:
            if filename.lower().endswith(".mp4"):
                print(f"Descargando archivo: {filename}")
                local_path = LOCAL_DIR + "\\" + filename
                with open(local_path, "wb") as local_file:
                    ftp.retrbinary("RETR " + filename, local_file.write)
                print(f"Archivo {filename} descargado.")

                # Delete the file from the server
                ftp.delete(filename)
                print(f"Archivo {filename} eliminado del servidor.")

        ftp.quit()
        print("Descarga y eliminación completada.")
    except Exception as e:
        print("Error:", e)

if __name__ == "__main__":
    download_and_delete_mp4_files()
```

Figure 3-10: Python script of the download and delete mp4 file method.

3.1.2.3. Explanation of the cropping Matlab algorithm to extract the eye frames from the image face.

In this step, 2 methods have been used for the cropping process. First, the Viola-Jones algorithm is performed due to the fast and less time-consuming during the obtention of the eye region and cropping method using multi-scale object detection where more information is available in the documentation of the computer vision toolbox related to *vision.CascadeObjectDetector* method. The other method used for the "problematic" video files is used for this purpose because it requires a higher time consumption, considering that the Dlib algorithm for face landmark detection (162) needed to work in the Matlab environment, which is a C++-developed code.

The process for cropping the video file is the same for both methods, and the script follows the next steps.

First, the video folder must be identified. All the videos downloaded from the recollection application server are stored in the folder. The next step consists of accessing the FTP server, where the cropped video files will be uploaded to keep track of the videos recorded by the patients. All the files successfully cropped and sent to the FTP server are registered in a file

text. This is done to keep track of the non-cropped videos or the ones where the Viola Jones have not successfully cropped the eye region for the analysis.

To perform access to the server, the *ftp* method is used by Matlab software, getting inputs of the server address, username, and password. With the *cd* function, we set the connection to a remote server directory where all the patient eye videos will be uploaded. Again, using the *dir* method, we get the information of all the files located in the remote server directory where the cropped eye videos are located. This is performed to check the files already uploaded and the ones that were unsuccessfully cropped due to patient head movement or corrupted files. An *if-else* statement is used to check if the remote server folder exists, to stop the algorithm and print an error message to specify a new directory address, or to check if the connection to the ftp server has been established correctly.

A *for* counter is set to convert the full path video filename of the local machine before cropping of the structure where the information of the non-analyzed videos is already present in the local machine and is converted to chars to compare furthermore the files that will be analyzed and the information of the full path of the analyzed file list. To get all the information present in the file list with the paths of the already analyzed and present eye videos on the remote server, *importdata* function is used to store the file paths.

We have to note that the *local_files* variable will store all the files already available in the list (cropped videos in the remote server) and the *remote_files* variable stores all the information of the videos in the local machine but with the full path of the remote directory folder where the cropped eye videos will end (non-cropped videos).

To get the files that need to be cropped, a setdiff function is used to check the files in the analysis machine and those in the remote server where the eyes were already cropped. The filenames of the files to be analyzed are stored in the new_files variable. The cd command is used to change the current directory, and the path where the code folder is located is inserted as input. This enables access to all the functions developed on separate matlab files.

Additionally, with *addpath* command, the FFMPEG executable is added to the Matlab script folder. The FFMPEG toolbox converts the videos and adds the codecs so Matlab can use them. The toolbox was created by Takeshi Ikuma. The BSD License is provided jointly with the programmed code. The need of using the FFMPEG toolbox is because the WEBM extension is not compatible with Matlab so all videos taken by a laptop or mobile camera using a web browser are not compatible.

Then the loop starts defined by a *for* counter reading each filename available in the *new_files* variable and the video name is read and stored as a string in the *VideoName* variable. With *mkdir* command, a folder is created if it does not exist where all cropped videos will be stored keeping a local copy before sending them to the remote server.

The next step consists of defining the *detector* variable with the help of the function "vision.CascadeObjectDetector", from the algorithm Viola Jones to select the feature to crop from a face frame. In our case 'EyePairBig', it crops both eyes. After the detection method is set, the loading of the original video is done with the help of *VideoReader* function, taking the name of the video stored on variable *VideoName* to access the properties and each frame of it.

If the *VideoReader* function doesn't succeed in loading the video file, a new file will be created using *ffmpegtranscode* function considering a new filename stored in *VideoName_C* variable where it adds the following chars 'C_' to the original file name. With the function *ffmpegtranscode* the new video file created will have H.264 codecs and AAC (Advanced Audio Coding) format with a MP4 file extension. This codec transformation is done to be able to load the video file in the Matlab environment using the *VideoReader* function. Once the video codecs are converted into the new file, the previous one is deleted, and the name of the new file is changed to the original one to ensure same key structure between the video files that needs conversion and the other ones that are used by devices with an output file format compatible with the Matlab environment.

The framerate properties are extracted from the output variable of the *VideoReader* function. In addition, the number of frames is extracted if possible. This is a way to check if the format is compatible, as sometimes the VideoReader is able to load the video without error, but it is corrupted. A try-catch statement is coded to perform the codec conversion if the number of frames cannot be accessed.

The variables *FirstFrame* and *LastFrame* specify the frames to be analyzed, where the last frame is the number of frames available in the object video file loaded.

The *MiddleFrame* variable stores the frame number considering the first and last frames. The first one is analyzed to get the region of interest (left eye region). This is done to ensure that the first frame to be analyzed has a face and that the user is not still setting the camera while the recording has already started.

The use of *read()* function to extract frames from the video object is used with the middle frame value first. The function *detector()* obtained as an output from *vision.CascadeObjectDetector('EyePairBig')* from Viola Jones algorithm, detects the eyes in the image selected. In this case, *detector()* is the viola algorithm function where the eyes feature from the face are selected. The input to this function will be the middle frame image read from the video. The output of the detector bounding box (*BB)* variable stores a vector with 4 variables. Points *X* and *Y* of reference of the left bottom corner and the *width* and the *height* of the box where the eyes are located.

Following the *detector* function, if and else statements are used to make sure that the eye is detected. If only the eye is detected and not the nose the program will continue. Sometimes two or more features are detected due to the image quality. The eyes image is cropped from the face image using the function *imcrop()* with the *BB* and image face inputs. Additionally, *BB0* and *BB1* variables are created and given the value of *BB* to have furthermore the possibility to compare the box used in the previous iterations and, if the eyes are not detected, prevent the algorithm from stopping and detecting the optimum box where for sure the eyes are available for further analysis. The size of the bounding box *BB* is stored in the *SzEyes1* variable to crop the image of both eyes to get only the left eye. The relative size of the bounding box for the single eye is obtained by experience, and the value of the width is known as ⅜ times the width of both pairs of eyes considering the same reference corner values and height. *Eyeright1* variable is used to store the left eye frame image.

After setting up the current directory as the one where it is located, the cropped eye frame video folder with the cd command, with the use of the *VideoWriter* function, it is possible to create the new video file. MPEG-4 (Moving Picture Experts Group) file format is used for this latest video to get the best quality for further loading analysis in the Matlab environment or different image processing software. With the *Framerate* property from the video class loaded, we set the value stored in the variable *fr* extracted from the original video property file.

The left eye bounding box size is stored in SzEyeright1 to define an empty matrix with a relative size for creating the video file's fixed width and height. This is done to ensure that all bounding boxes will fit without resizing the pixel image in the video file. During the video recording, the patient can move closer and further from the camera.

A *for* counter is used to access all the frames of the video with the help of *read()* function. Then, the zeros matrix of fixed size during all the video is defined. The next is the same as

done for the middle frame, considering that if-else statements are coded to consider using the first middle frame bounding box or the bounding box from the previous frame if no bounding box is detected to prevent the script from not cropping the eyes during the entire video. If the new bounding box position is very different from the previous frame and more than bounding box a comparison of the position and size is done to select the optimum bounding box. To check the optimum case, please go to the annex chapter where all the MATLAB scripts are available (see Chapter 0).

Each bounding box is introduced after in the zero's matrix *Eyes_sz*, considering the size of the current Eyeright matrix, where the values of the eye frame are stored. The matrix is transformed to an unsigned 8-bit integer using the uint8 function and then written in the new video file using the function writeVideo(), having as input the variable of the object created with the *VideoWriter()* function.

Once all the new eye frames cropped from the face are written in the new video file and the scripted lecture is out of the for counter, the video is closed using the *close()* function. With the help of the current directory command and the *ftp* object, the script accesses the remote directory. The upload is performed using the put () function and the directory of the new video file. Additionally, the size of each bounding box of each eye frame stored in the *SizeEachEye* variable is exported as a matlab matrix file. The same is done for the original video properties to ensure that all the information of the real eye size frame is available as the original framerate of the video. At the end, the analyzed file name when all the steps of the algorithm are successfully done the *local_file_list.txt* is open with the *fopen()* function and the name of the analyzed file is printed. This is performed as the script will know furthermore and as explained before, automatically which videos have been already cropped, and the new ones successively will be cropped one by one.

3.1.2.4. Explanation of the first algorithm for blink analysis.

The first thing needed is to locate where the script is and secondly where the videos to analyze are. Using the try and catch, we have an example of folders in mac and in windows. If the folder where the videos are is in a remote server, we use the ftp function to connect and download all the videos. The only videos downloaded are the ones that have an mp4 extension as with the app this extension is given.

Once the folder is specified and the server is connected, we proceed to check all the videos currently in the folder. In our case we check all the files that have an .mp4 and .mov extension. Other extensions can be added easily.

The variable "I" is used in this case to check if there are new videos. This is done so we analyze only the new videos. As coded in the app (see web app explanation) all the video names collected are numbers. After the conversion and analysis, a C is added to the file name. When the script is running the next time, only the videos starting with a number will be considered in the variable *NVN*. A message before the analysis will tell us the number of new videos and those, will be in the first positions. This way, only new videos will be analyzed once the Matlab program is compiled again. *Ffile* and *Lfile* are variables that ask the user to introduce the first and last file to analyze.

Once the user accepts, the script starts running and the first thing to be done is adding the *ffmpeg* executable path to the Matlab script folder. The FFMPEG toolbox converts the videos and adds the codecs so Matlab can use them. Takeshi Ikuma created the toolbox. The BSD License is provided jointly with the programmed code. The need to use the FFMPEG toolbox is because the WEBM extension is not compatible with Matlab, so all videos taken by a laptop or mobile camera using a web browser are not compatible.

Once the first loop with all files starts, the filename of the current video analyzed is read from the variable *theFiles*. With function *mkdir*, the video analysis folder and the result folder for the current video is created.

The vision function, *CascadeObjectDetector*, is from the Viola Jones algorithm. It selects the feature to crop from a face frame. In our case, '*EyePairBig*' crops both eyes.

Ffmpegtranscode is a function from FFMPEG toolbox that adds mp4 codecs to the video. *VideoName_C* is the variable with the new name of the file. We added a C to separate new videos from converted/analyzed ones. The function *VideoReader* is a Matlab function that reads videos. The video is stored in variable obj. Features from the video are extracted; in the first case, the frame rate is stored in variable *fr*. The use of read to extract frames from the video object. As seen in the subchapter 3.1.2.3, the video of the single eye is loaded considering the information stored in the matrix Matlab file to get the correct size of the bounding box of the eye's cropping method.

Once the loop starts, the image color segmentation algorithm analyzes all the frames in the current video. This algorithm is based on the non-supervised clustering method K-means. Note that the machine learning toolbox was used, and a self-made script was made. The parameters needed are *nBins*, *winSize*, and *nClass*, respectively. Those are the numbers of colors to consider, the size of the window to analyze, and the number of colors of the output image. We use 100, 15, and 4 colors in the output; after testing in different videos, we found out it is a good way to detect the iris and to know exactly where the lower lid is. With the function *colImgSeg* we extract the output image using as input the parameters specified before and image of the single eye. The output image is saved using the *imwrite* function.

Using the segmented image by colors, the lower lid is detected. The First thing we do is detect the color given to the iris in the output image. This is done by selecting reference points and making sure that it is the central color. Going from the reference point to the edge of the color in the two directions, we are supposed to find the same color. If that is the case, the reference point is on the iris color. We can see an example in Figure 5.

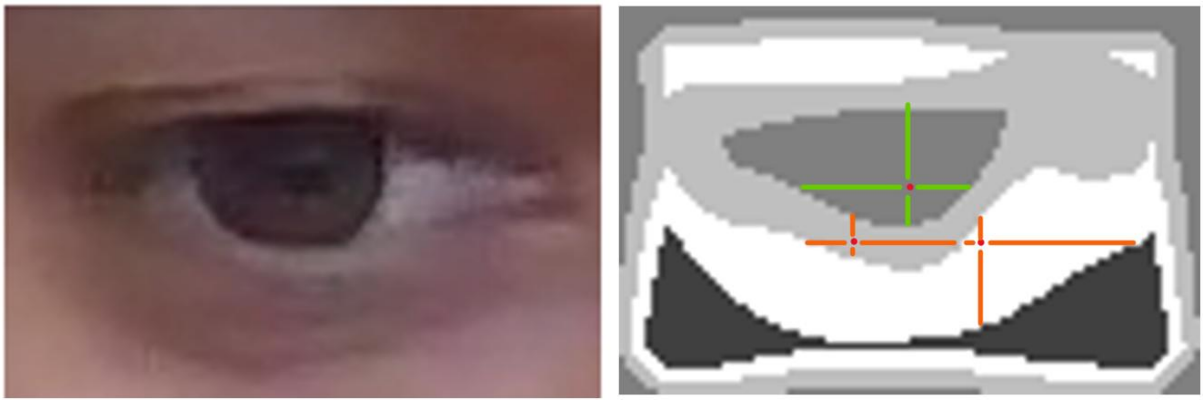

Figure 5: To the left, the eye image cropped from the detected pair of eyes using Viola Jones Algorithm. To the right, Image centering algorithm, criteria to find the iris color in the segmented image.

Reference points are tested following the next order. See in Figure 6.

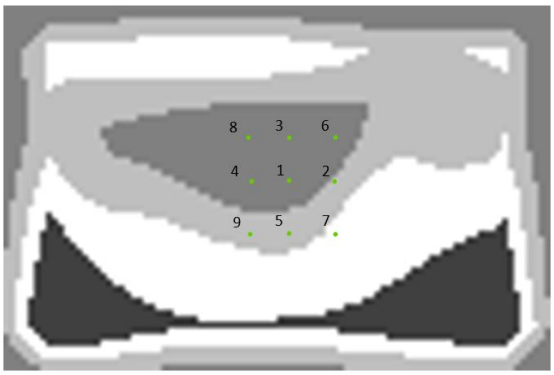

Figure 6: Reference points used to find the iris color in the segmented image. Image centering algorithm.

After the iris color is detected, the lower lid is located analyzing each row of the frame from the bottom to the top and stopping once a row has 10% of the pixels with the iris color. We must consider that the pixels analyzed in each row are 6.66% of the length of the image further from the edge. This is considered because the image can suffer a smoothing filter that disturbs the output image. A 5% height of the image is added to the row selected so the lower lid is completely visible.

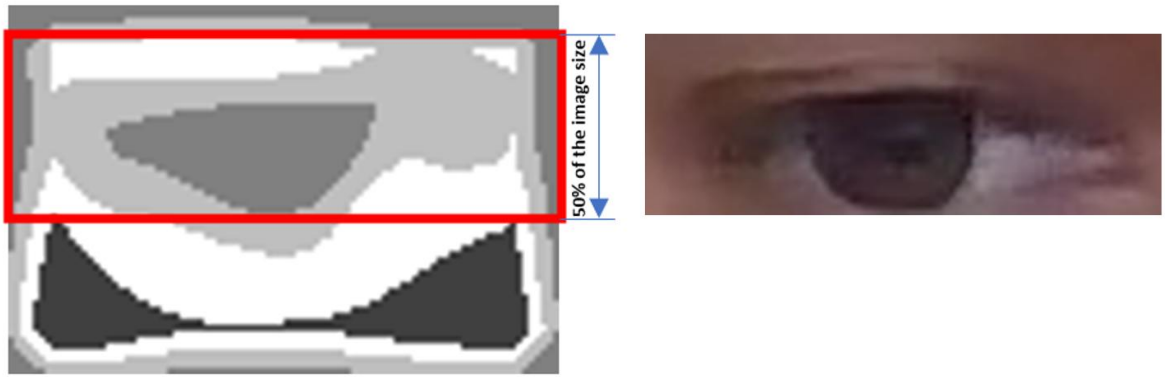

Figure 7: Result of the image centering algorithm.

Once the eye is centered (Figure 7), the image of the single eye cropped is turned into gray scale with the function rgb2gray. An average filter is applied to the image of size 12.5% of the size of the image. The filter is created with the function *fspecial* and applied to the image with the function *imfilter*. The cropped single eye centered image is saved.

Furthermore, the variable "*inr"* stores the mean value intensity of object in each row of each frame of the eye (163). The position of the lowest intensity row, corresponding to the upper lid position, is stored for all frames of the video in the variable *eyeOpen*. A correction is done to the value of the darkest row -1% of the value of the darkest row, to select a row that is in the image positioned slightly on top, to consider the eyelids (Figure 8 and Figure 9).

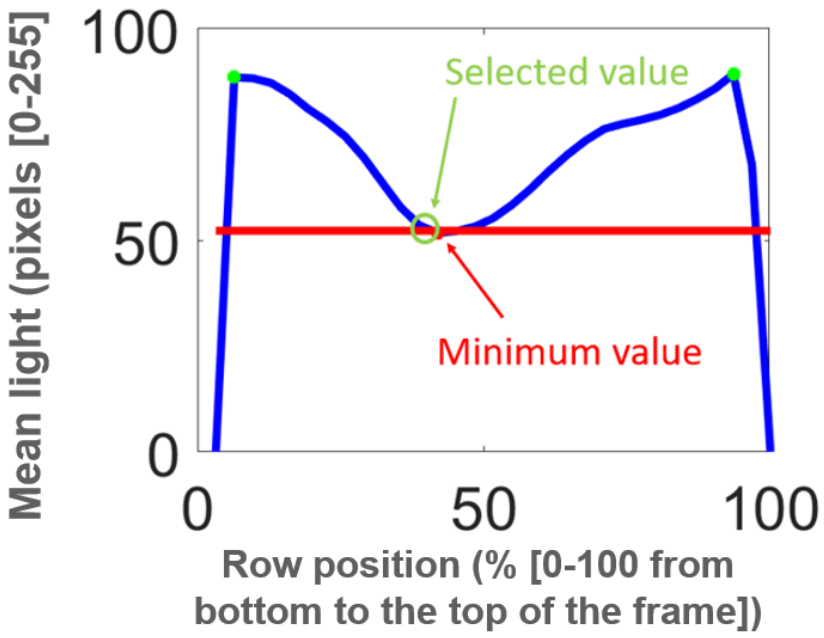

Figure 8: Matlab plot of one eye image of the position of each row of a frame [0% is the bottom of the frame and 100% is the top edge of the frame] vs the mean light intensity value of each row in pixels[0-255]. The darkest row is the minimum value and the selected one is 1% brighter, to consider eye lids.

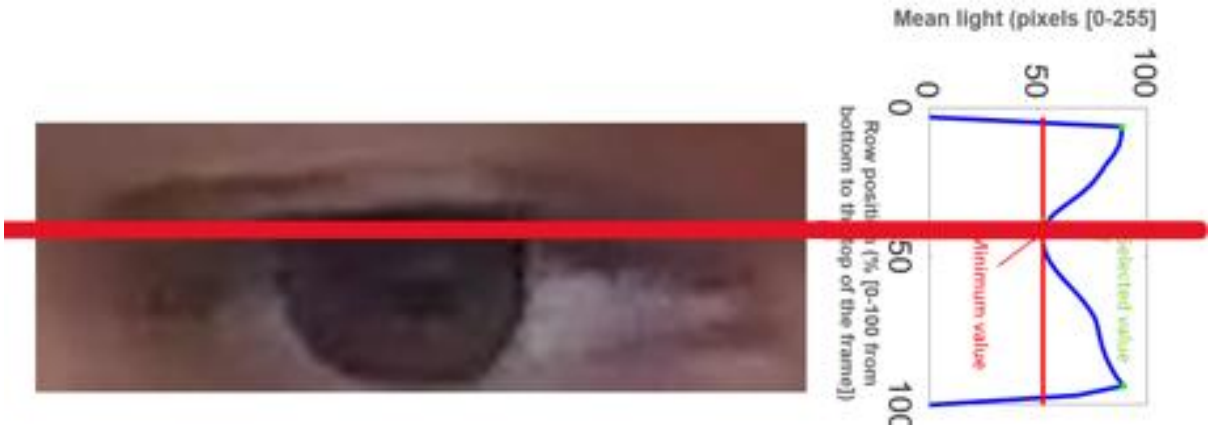

Figure 9: Result analysis of the technique used to detect the upper lid.

Once the video is totally analyzed, the eyeOpen variable is plotted versus the time of the video. A correction is applied to consider variations due to eye movement. The *polyfit* function is used to create a fitting curve of grade 3. The value of the fitted curve is extracted from the *eyeOpen* variable. The plot function is now used to plot:

- EyeOpen graph
- Gradient of the eyeOpen – time graph
- Probability density function.

- Cumulative density function.

Therefore, the probability function fits on the last half. This is done to calculate the criteria values for incomplete and complete blinks. The *polyfit* function is used to create a fit curve grade 9. The first minimum is the criteria for incomplete blinks, and the second one is for complete blinks.

To find the complete and incomplete blinks, we use the function deteccion_parpadeos2. The input variables are *tiempo*, *eyeOpen* and *localizaciones*. *localizaciones* is the same variable as *eyeOpen* but considering only the values higher than the criteria. For example, if the criteria is the position of the darkest row "50", *localizaciones* in our main script will be (*bincompletos* or *bcompletos*) will be a vector the same length and values as eyeOpen if it is higher than 50 but 0 if it is lower than the criteria. The function deteccion_parpadeos2 will return the duration of each blink. The initial time will be calculated taking out all the zero values from *localizaciones* using the function find. Then, we select each value of *localizaciones*. Each value will be treated as a blink or incomplete blink. The points that are near will be changed to zero.

From the reference point (the selected nonzero value from *localizaciones* or as told before, the value from *eyeOpen* that is higher than the criteria) we analyze the next and the previous values. See Figure 10.

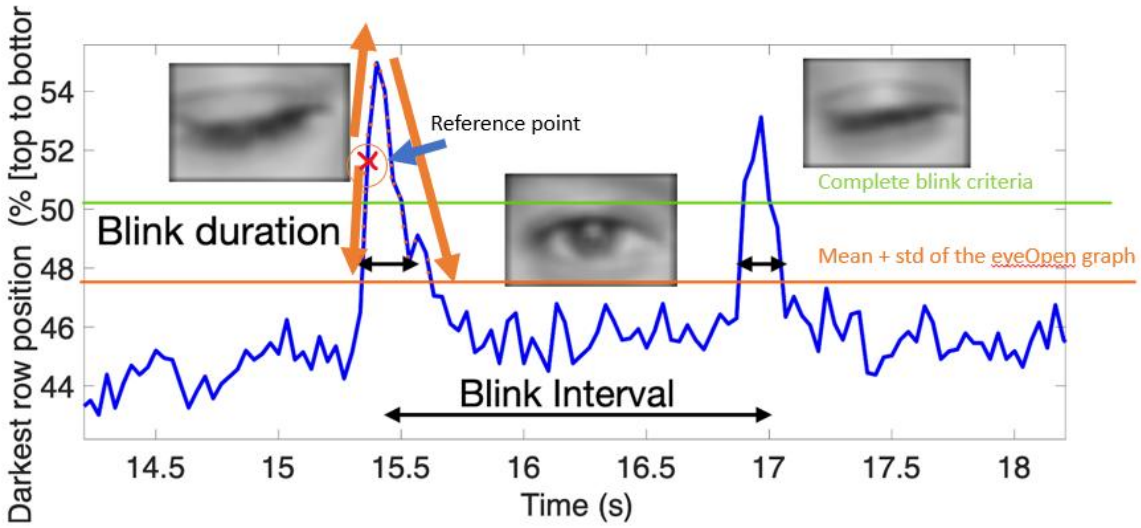

Figure 10: Plot of the upper lid position of each frame of the video.

As seen in the Figure 10, each point next to the reference point is analyzed. If the value of the point is above the mean + std, we jump to the next point, but if not, the loop stops, and we have

the bottom of the peak. That point, will represent as we can see in the figure above, the start or the end of the blink. The value of time between the 2 bottom points of the peak will be the blink duration. The blink interval is calculated as the distance between consecutive blink starts. Two loops are being used to analyze each value of the blink. A loop considering the previous points and another loop for the next points after the reference position.

Once the function *deteccion_parpadeos2* has been called for the complete and incomplete blinking criteria, we proceed to the creation of matrices with the duration of each blink.

The first thing we do after the creation of the matrices is to make sure the blink has more than one frame of duration; if this is the case, then data is removed. Secondly, because the incomplete blink criteria value is less than the full blink criteria, the array of the incomplete blink can have a complete blink. To take out the complete blinks from the incomplete blink matrix, we use the intersect function. A while loop is used to ensure the beginning and finishing values are cleaned from the incomplete blink matrix. Finally, If and else statements are also used to not calculate the interval and duration if there are no blinks. A matrix with all the duration of complete and incomplete blinks is also formed and stored in the variable *tiempo_parpadeos*.

The result is a table *T* with the number of blinks, the mean value of duration and intervals and their standard deviation of complete and incomplete blinks following the criteria calculated before.

Table 3-1: Example of output table "T" in the Matlab script.

	n	Blink_interv (s)	Blink_interv_std (s)	Blink_duration (s)	Blink_duration_std (s)
Incomplete Blinks	15	4.4655	10.401	0.28	0.47341
Complete Blinks	1	NaN	NaN	0.225	0

The *recoleccion_datos_MySQL* script is used to receive and send data to the MySQL database. With the function "*databaseConnectionOptions*", we create a connection to the localhost using jdbc driver of MySQL. We must consider before compiling the script, the use of putty program or the terminal for mac to create a tunnel connection to the remote database through the SSH ftp server to the localhost on port 3307. The inputs for the *databaseConnectionOptions* functions are the server's name, the port number, the jdbc driver location and the type of database in our case MySQL.

With the functions *testconnection* and database we test if all the data is correct and if the answer is 1 for the function *testconnection* the server is connected. Same thing for *isopen* function.

To get data from the database into a Matlab variable, I our case a table, we use the function fetch to select the data to extract. As all the data extracted from the MySQL database is stored as characters(char), we use the function str2double to store in Matlab the values as numbers of the variables needed.

To send data from Matlab to the MySQL database, we use the function *sqlwrite*. The input for the function is the table with data to send, the name of the table in MySQL and the connection data stored at the variable conn. The *sqlwrite* function is easy to use and will store the data and even create a table if there is none with the same name. We use a loop in this script version to read the data to send from .txt files.

With the help of a K-means clustering algorithm we perform image color segmentation (depending on the pixel color) to detect the lower lid to do a more accurate crop. This is done, to increase the precision of the algorithm, mostly the classification between half and full blinks, an image center technique is used to locate the center of the eye image in the centered position of the frame.

Once the eye area is visible, we turn into greyscale images, and an averaging filter size of 1/10 is applied to reduce the intensity variation between adjacent pixels (noise).

The mean value of each row is calculated. The upper lid position is detected as the darkest row of the frame.

Color segmentation helps the Viola-Jones algorithm detect the eye in each frame in the same centered position (list point number 2). Then, the rest of the algorithm's steps are applied. Now, two jumps are visible on the PDF: one corresponding to the threshold for complete blinks and the other for detecting incomplete blinks.

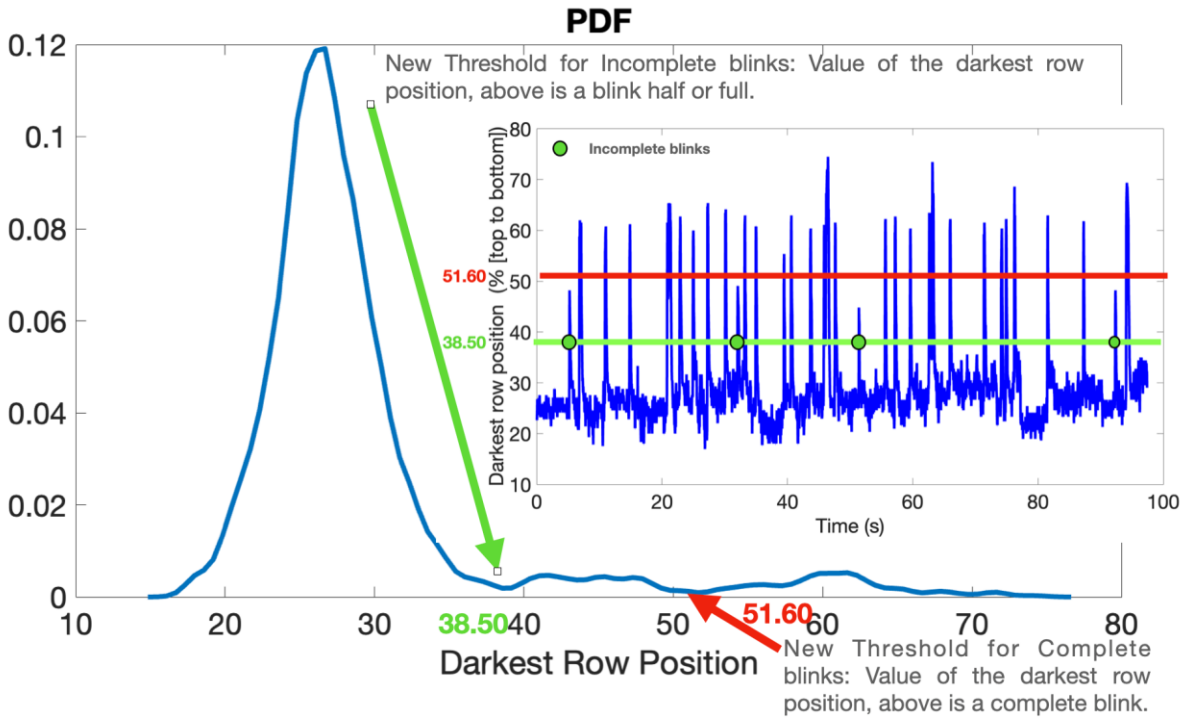

Figure 11: PDF with blink localization in time. Example Video 1.

Table 2: Results with centered corrections.

Video	Blink Class	Algorithm nº of Blinks	interv (s)	interv std (s)	duration (s)	duration std (s)	Visual count	Algorithm Error (%)
1	Incomplete Blinks	4	29.03	11.01	0.25	0.06	33	0.00
	Complete Blinks	29	3.12	2.01	0.39	0.12		
2	Incomplete Blinks	0	-	-	-	-	16	0.00
	Complete Blinks	16	2.34	2.12	0.34	0.03		
3	Incomplete Blinks	3	17.40	9.24	0.18		9	11.11
	Complete Blinks	5	3.71	2.54	0.21	0.04		
4	Incomplete Blinks	0	-	-	-	-	15	6.67
	Complete Blinks	16	3.06	2.95	0.34	0.09		

Finally, the display shows the number of blinks (peaks) in the video, the mean blink rate and its standard deviation and the mean speed of the blink and its standard deviation.

Once all the features of the video have been analyzed, a Matlab script is created to connect to a MySQL database to store all the video results. Those results will be combined with the air contamination data provided by the weather station for each video analyzed.

Finally, a more general characterization of the blink can be done by studying the spectrogram and the power spectrum of the signal given by the position in time of the darkest row (see previous figures). Given the complexity of the blinking process, the spectrum is usually of the

1/f type, characterized by two parameters $PSD = 10^b f^\alpha$ (90). Figure 12 shows the power spectrum density of the blinking dynamics and the fitting to 1/f power spectrum.

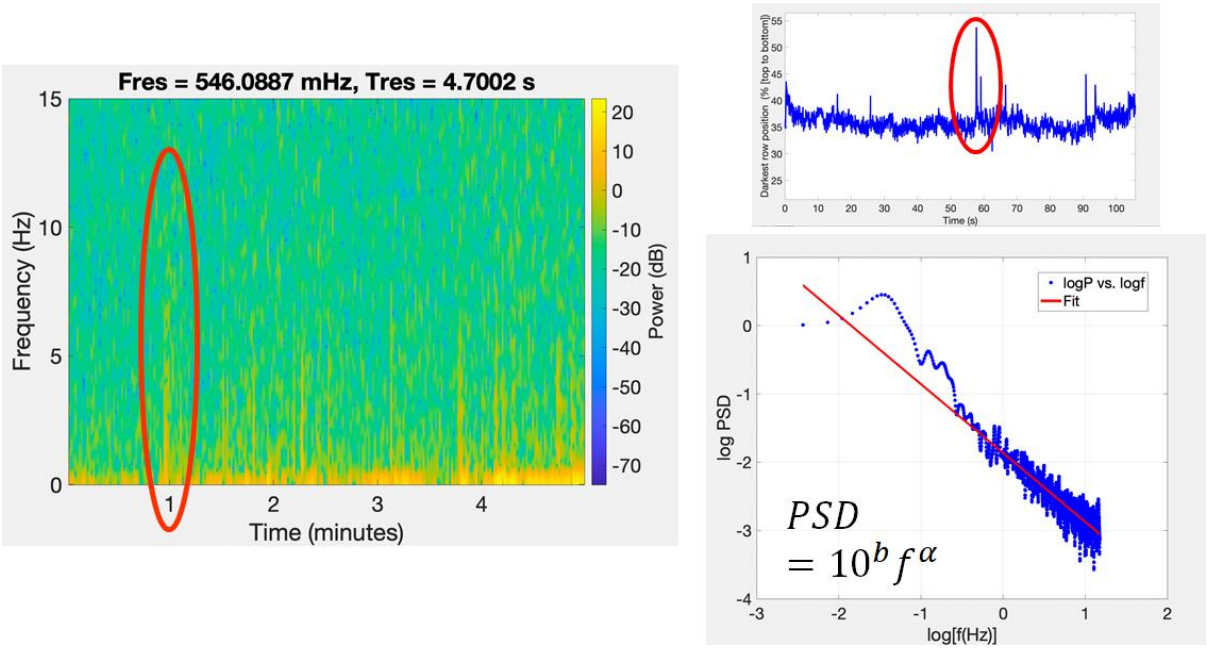

Figure 12: Power spectrum density of eyeOpen graph of a video example. Right up the EyeOpen graph (darkest row position vs time)

Table 3-2: Blink table in MySQL database.

blink_id ▲ 1	n_incom	interv_incom	stdinterv_incom	dur_incom	stddur_incom	n_com	interv_com	stdinterv_com	dur_com	stddur_com
	0	0	0	0	0	2	24.3333	0	0.16667	0.04714
	1	NULL	NULL	0.016667	0	12	5.5674	10.1456	0.031944	0.033679
	1	NULL	NULL	0.16667	0	12	4.9212	6.4091	0.51944	0.84906
	0	0	0	0	0	2	3.5667	0	0.10417	0.029463
	4	0.13333	0.060092	0.016667	0	8	8.7309	14.6109	0.015625	0.0029463
	0	0	0	0	0	64	1.0275	4.0904	0.082422	0.12839
	15	4.4655	10.4014	0.28	0.47341	1	NULL	NULL	0.225	0
	1	NULL	NULL	0.11667	0	43	1.4682	1.4365	0.16357	0.17369
	0	0	0	0	0	14	4.9365	8.1871	0.023809	0.029209
	49	1.1958	1.6657	0.082653	0.047533	6	10.1716	12.2189	0.055555	0.043355
	0	0	0	0	0	7	5.8111	13.7242	0.015476	0.0031497
	81	0.7776	2.0342	0.019239	0.014291	31	1.4467	2.3121	0.058064	0.18398
	16	3.9217	5.4822	0.017708	0.0041667	59	1.0743	1.4155	0.017797	0.0069992
	0	0	0	0	0	3	11.7833	15.5799	0.22778	0.28017
	0	0	0	0	0	12	4.095	8.422	0.25859	0.13356
	0	0	0	0	0	195	0.33462	0.34307	0.044102	0.088318
	0	0	0	0	0	25	2.6553	3.5249	0.24931	0.22567

The matlab script of this first blink detection algorithm can be found in the Annex.

3.1.3. Automatic eye blinking detection from videos recorded using ubiquitous cameras.

To get the video files of the eye being analyzed, a connection to the ftp server is made using functions from the FTP File Operations for Matlab environments to access and download the files. The results obtained are also posted on MongoDB from Matlab using Database Toolbox functions, since each video analyzed will have different sizes of results (depending on the blinks found) which makes the need for non-relational data storage. As the final purpose of this study is to show the quality and number of features and data extracted from the characterization of the blinking, we proceed to enlighten the techniques used.

3.1.3.1. Image preparation

The script performing the eye blinking detection starts with centered equal sized eye frames obtained after performing cropping and centering in a way that the lower eyelid matches the bottom row of the captured image.

First, the images cut from the eye are loaded. These images have been previously centered by the location of the iris so that the last row of the image coincides with the lower eyelid and the iris is centered horizontally (see Figure 3-11).

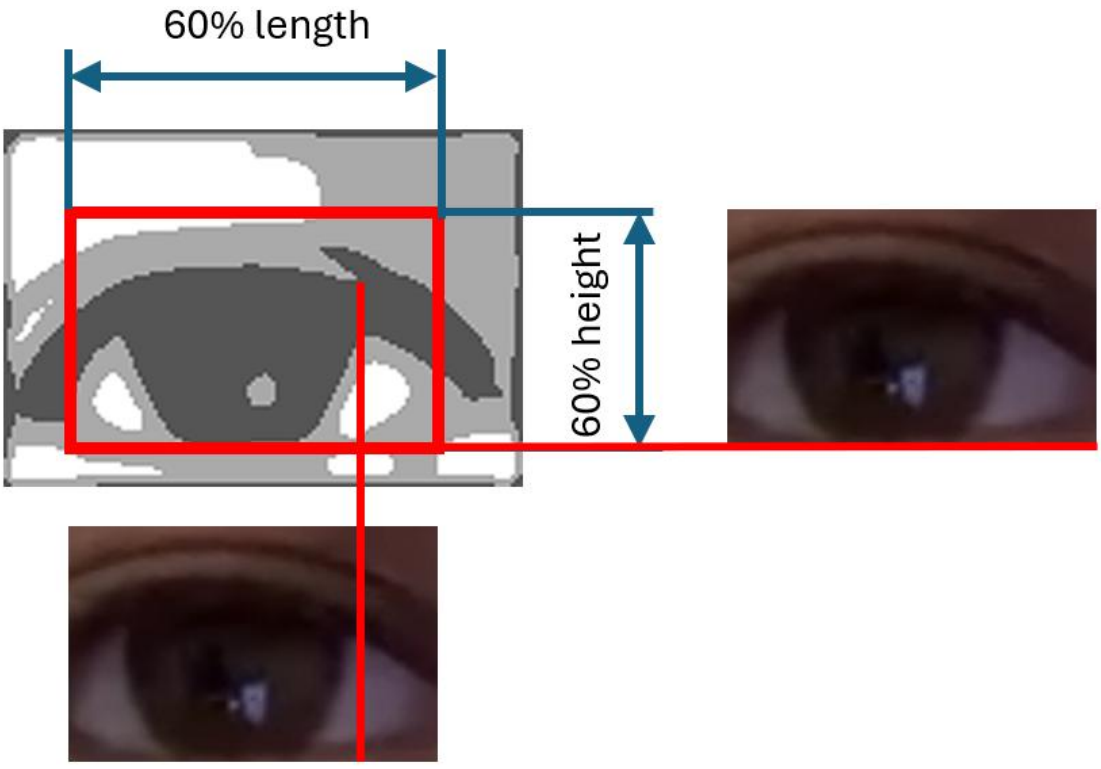

Figure 3-11: Preprocessing step to crop and center the eye frame image.

In the first load pass of each frame of the video of the first loop, the *media_local_polinomica* function is used to obtain the lighting of grade 3 to which it is adjusted from the intensity of the pixels of the rows and the columns of each of the frames of the video.

Figure 3-12: Grayscale eye image

The *media_local_polinomica* function fits the intensity values first of each one of the pixels belonging to a specific column to a polynomial curve, in our case to a third degree. When the polynomial curves are obtained for each column, the new values of intensity corresponding to these new curves are used as input to rows of the third-degree curve fit to get the mean local intensity image of each video frame. An example is shown in Figure 3-13 using as input the grayscale eye image shown in Figure 3-12.

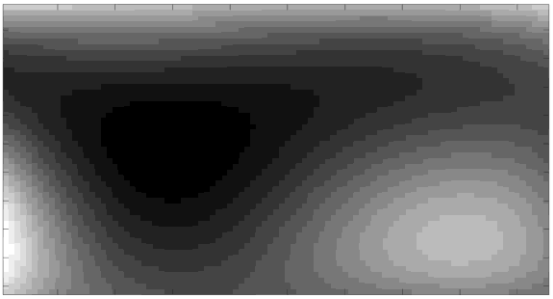

Figure 3-13: local intensity mean obtained using media_local_polinomica function of the eye frame under study in Figure 3-12.

Once each image of local mean intensity is obtained, the global mean intensity is calculated obtaining the mean value of each pixel position from the images of all the local mean intensity images of the video (an example is displayed in Figure 3-14). A correction is performed on each grayscale eye image of the video considering this global mean intensity image. An example of the result is displayed on Figure 3-15.

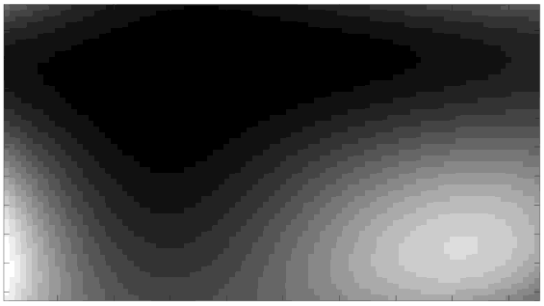

Figure 3-14: Global mean intensity considering it mean value of all the eye frames.

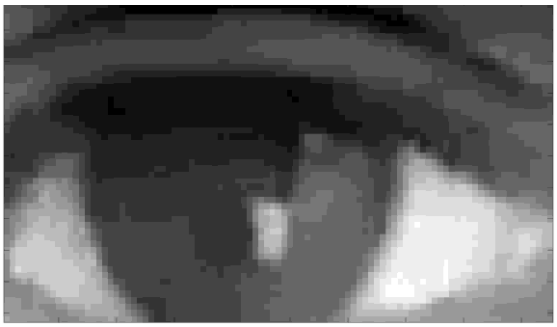

Figure 3-15: Grayscale eye image corrected using media_local_polinomica function

In summary, the illumination preprocessing step is done using a user-defined MATLAB function. This function is used to produce a constant illumination in time (video recordings are about on minute).

3.1.3.2. Signal PCA 1 Analysis and blinks classification method

After light illumination processing, Principal Component Analysis applied to image processing (164) is adapted to event camera based video images and performed to modify the video to eliminate the noise, and analyze only the first component that is storing the information of the closeness of the eye.

Principal component analysis allows to analyze the correlations and covariances between the different images present in the video $F(x,t)$, where "x" is the position in the image and "t" time. The covariance between the images is characterized by the covariance matrix $S(t,t')$ between two images taken at time t and t'. The principal components, Y_α (x), are defined by images that are a linear combination of the original frames, but exhibiting zero covariance between them. Mathematically:

$$Y_\alpha(x) = \sum_{t=1}^{N} e_\alpha(t)(F(x,t) - \langle F(t)\rangle_x) \tag{3-1}$$

where $e_\alpha(t)$ are the coefficients of the linear combination to be used and $\langle F(t)\rangle_x$ is the mean value of the image at time t. Equation 3-1 can be inverted so that we can view the original frames as a linear combination of the principal components by

$$F(x,t) = \langle F(t)\rangle_x + \sum_{\alpha=1}^{N} e_\alpha(t) Y_\alpha(x) \tag{3-2}$$

Equation 3-2 also allows to factor the spatiotemporal dependence of each frame into a spatial dependence (given by the corresponding principal component) and a temporal component. Mathematically it is shown that the coefficients of the linear combination $e_\alpha\,(t)$ are the eigenvectors of the covariance matrix S between the frames, associated to its corresponding eigenvalue λ_α. (165). These eigenvalues also represent the variance of each principal component, so the percentage of each one within the total variance of the data can be calculated by the value $\Omega_\alpha = 100\lambda_\alpha / \sum_\alpha \lambda_\alpha$. In this way, equation 3-2 can be used to select only those components that represent a large proportion of the data, which can serve to reduce the dimensionality of the data, as well as to separate a signal from its noise components (references applied optics).

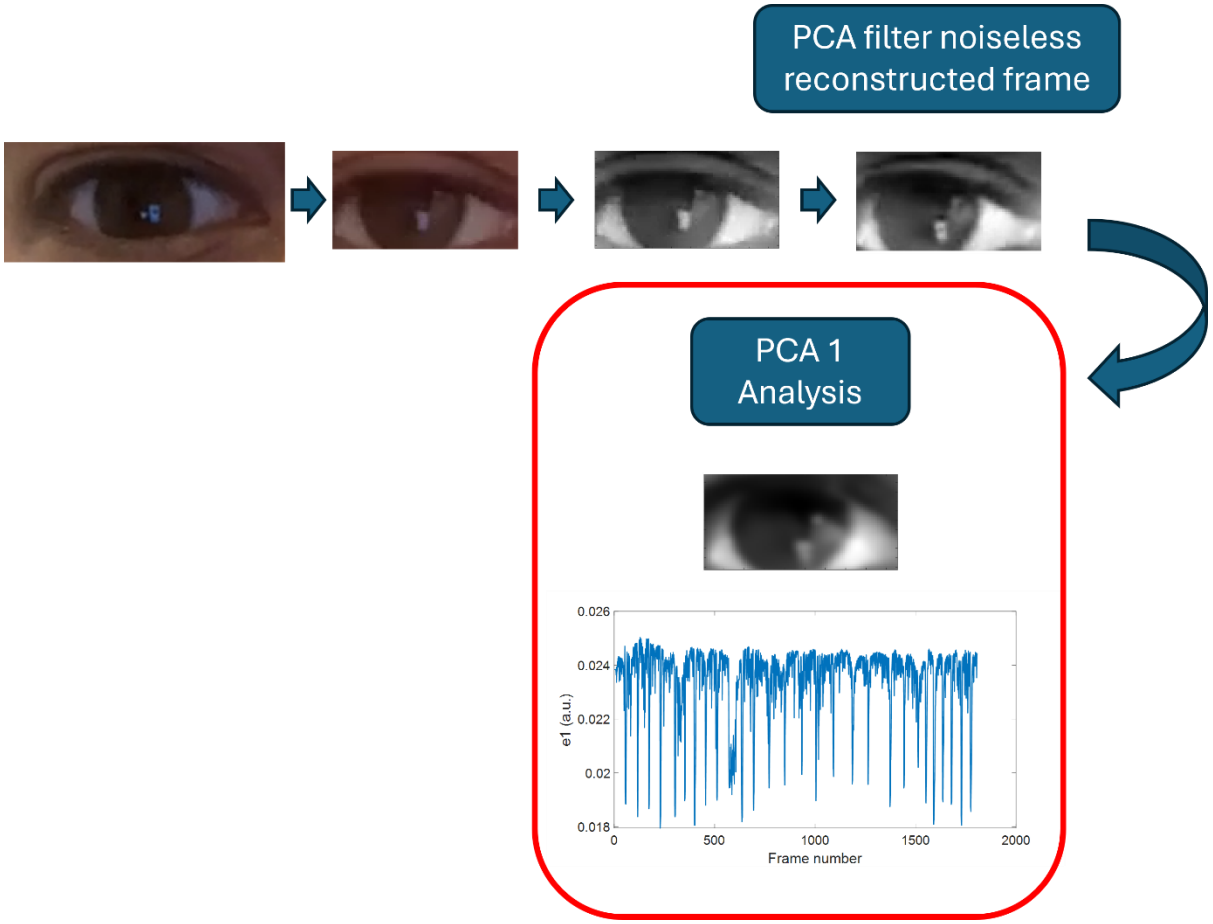

Figure 3-16: Preprocessing of eye frames and video PCA 1 analysis to locate the frames potentially involved in blinking sequence.

To eliminate the noise, a reconstruction of the video is performed using only the principal components photograms that capture the eye movement. To acknowledge which components, have to be used, the Figure 3-17 curve is plotted using logarithmic scale in y axis and the knee of this curve will be the limit principal component where noise of the camera is captured. To automatically measure this "knee point" a MATLAB code function is used (166,167). The selected principal component is shown in Figure 3-18. It is possible to see how the morphological features of the eye are slightly visible, while its structure is affected by noise. In the example figure the total amount of variance for the reconstructed video (taken the first 177 principal components) is 99.72%.

If we reconstruct the video with all the previous components, most of the noise from the camera is deleted and the photograms obtained will be used to detect the eyelid distance to detect the degree closeness of the eye to furthermore classify the quality of the blink in complete, incomplete and twitch.

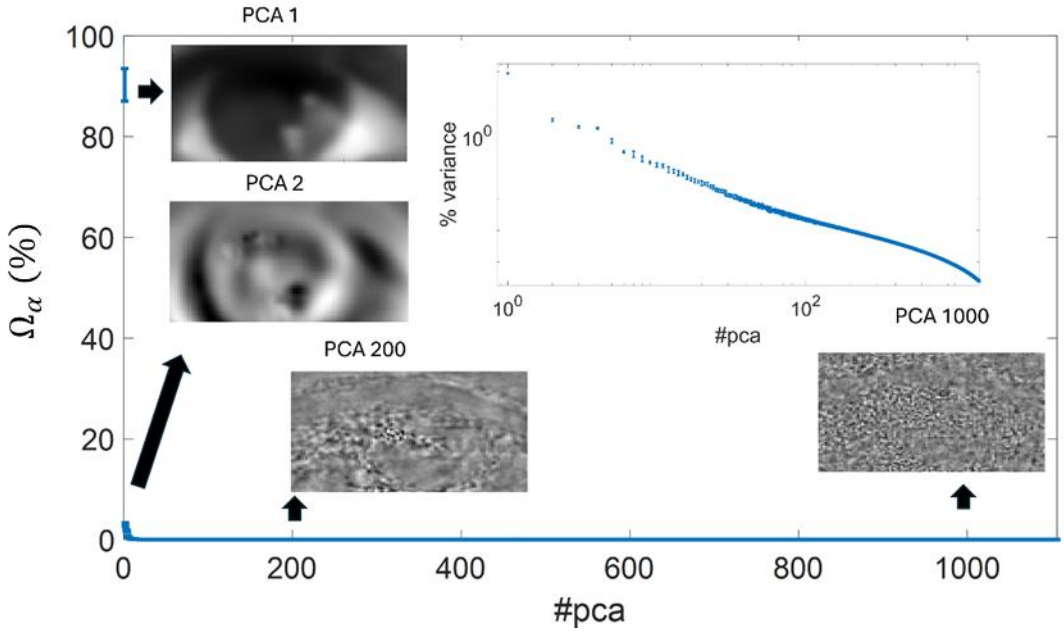

Figure 3-17: Comparison of each one of the PCA images considering their weight in the total variance of data.

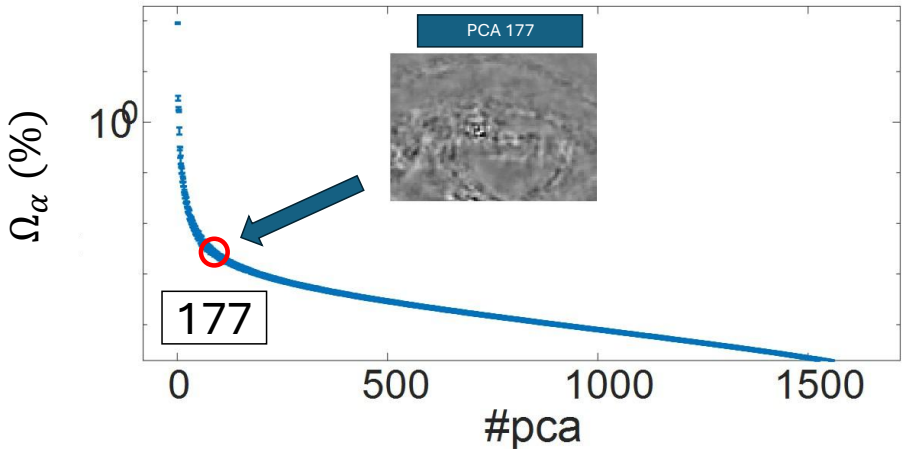

Figure 3-18: Weight of each Principal Component and its variance. Logarithmic scale is used for the y-axis. Component 177 is found to be the limit given by the knee point criteria.

Another interesting property of principal component decomposition is that components are selected in order of decreasing variance. Since the video consists of an eye blink, most of the images will consist of an open eye, with some moments when it closes and reopens. Hence the first component will have to be related to the structure of this open eye and $e_1(t)$ will reflect the temporal variation of this structure in the data (see Figure 3-16). Then, when analyzing only the first component, the huge variations applies when the eyelid closes, and the iris and sclera is not seen. The principal component images are built using all the images of the video and the coefficient (loadings) of each one of the original images to make the first principal component is represented in Figure 3-19. When there is a huge difference in between the principal

component image (the one that its pixel intensity will be the most similar considering all the images of the original video), a peak in the signal is displayed showing that this eye image is closed. For additional information considering the PCA function, as the code environment used is Matlab, refer to the documentation (168).

The value of $e_1(t)$ shown in Figure 3-16 shows a constant value of the first principal component (eye open) with "peaks" denoting a fall of this component. This is interpreted as a blink moment. Thus, temporal analysis of $e_1(t)$ can be used to characterize the process of the various blinks that occur. A new parameter is obtained in order to characterize more clearly the amplitude of various blinks. Then, a new variable Fe1 is defined as

$$Fe1 = \left|e_1(t) - Me\left(e_1(t)\right)\right| \tag{3-3}$$

Where $Me(e_1(t))$ is the median value of $e_1(t)$. This procedure is followed to produce a baseline close to zero. In this sense, the median is taken instead of mean value in order not to be so affected by the outlier points of blinking peaks. An example is given in Figure 3-19.

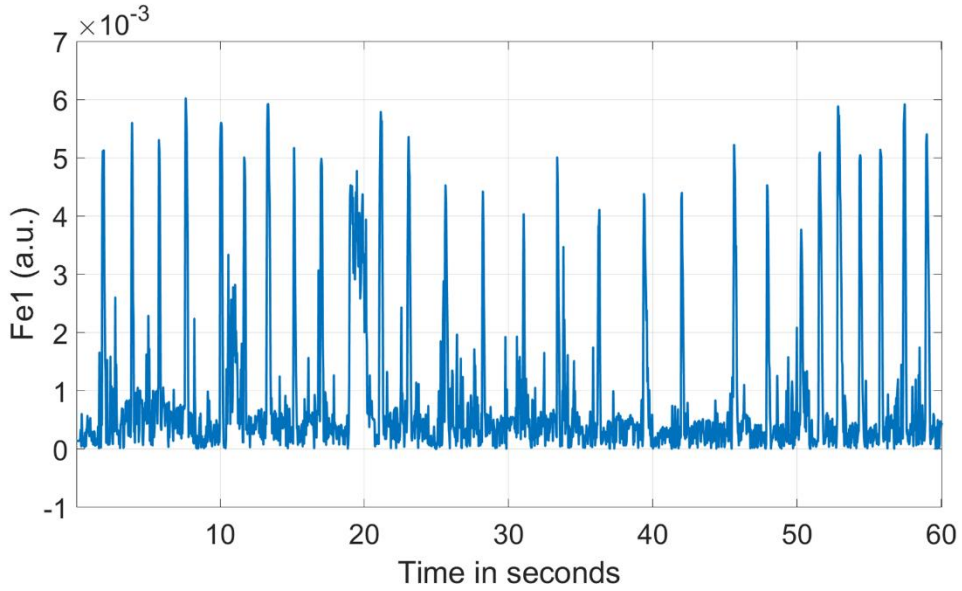

Figure 3-19: Fe1 for an example video of frame rate 30Hz, total number of frames 1807, video duration 60 seconds and 233 milliseconds.

Each of the above peaks of the Fe1 (see Figure 3-19) signal corresponds to a flicker. This flicker is characterized by certain parameters such as its duration, degree of closure, etc. With the help of the MATLAB “findpeaks” function, the values of the location of the peaks that indicate the blinking point are obtained. This function locates the local maxima (peaks), and each of them gives a prominence value as the difference between these local maxima and the local minima

located to the left and right of each of these local maxima (peaks). The function also provides a width at half height.

To detect only blinks and not eye movements or "twitch" (slight movement of the eyelid that indicates a closure of less than 30%), the minimum prominence of 2 times the standard deviation of Fe1 signal is taken to consider that point as a blink (see Figure 3-20).

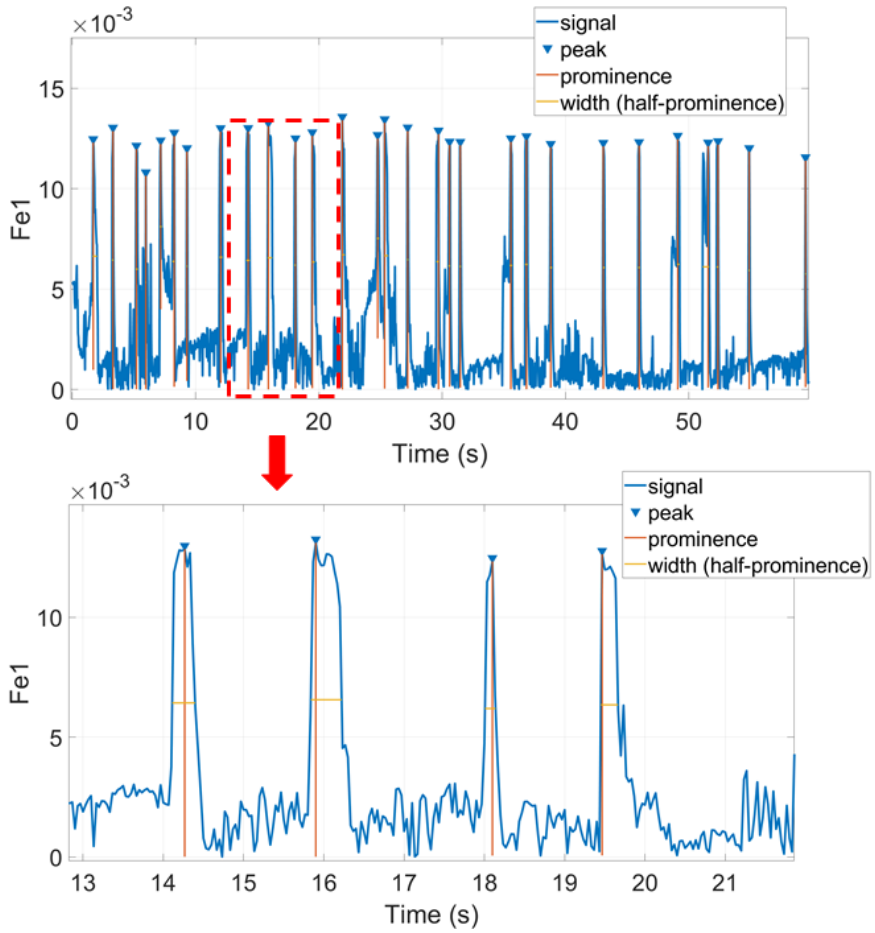

Figure 3-20: The value of "peaks" and prominence were detected in the example video blinking.

3.1.3.2.1. First study

The first study consisted of the check of the blink closeness degree by factorizing the PCA 1 signal using image processing techniques to identify the closeness variation of the eyelid distance with the characteristic frame of each peak detected on the signal displayed on Figure 3-20.

The degree of closeness of each blink frame is studied, obtained from the time value frame from the signal value of each one of the peaks frames Figure 3-20, and compared to the characteristic open eye image of the video (reconstructed image from first principal component) considered as the zero signal value. This is done to do a calibration and have data

on the amount of eye closure first in pixels and subsequently in percentage to be able to know and differentiate types of blinks. First, the image of the first principal component is reconstructed. This image is the one that is repeated the most throughout the video and will correspond to an open eye frame. In our case we choose this image as the open eye of which the eyelid distance in pixels will be checked for comparison purposes as eyelid distance for characteristic open eye.

Figure 3-21: Example of first principal component reconstructed frame.

To know the degree of opening of the open eye, we first identified the pca1 image, considering that all the frames have been cropped and centered considering the location of the iris. Having this information in hand, we can calculate the position of the upper eyelid by obtaining the minimum value, making the median of the gradient in the height direction of the image. The same methodology is applied for the upper eyelid detection in the analysis of a closed eye frame where the bottom eyelid doesn't always coincide with the last row see Figure 3-22 and Figure 3-23.

For the images detected in the PCA 1 signal (Figure 3-20) as blinks, the calculations related to the degree of opening in pixels have been carried out using the *blink_grade* function, see section 8.2. The *blink_grade* function calculates the degree of eye opening based on the distance between the upper and lower eyelids, obtaining the gradient of the pixel intensities on the x-axis and subsequently calculating the median (value of the central position). The resulting curve fits a polynomial of degree 5, the choice has been made for this value due to the behavior of this curve with different eye images since it correctly evaluates the position of the most significant rows (upper and lower eyelid). By locating the maximum and minimum value, the width of the palpebral slit of the noise-free image (reconstructed video) of the peak of the signal flicker location is obtained, representing the coefficients of the first principal component.

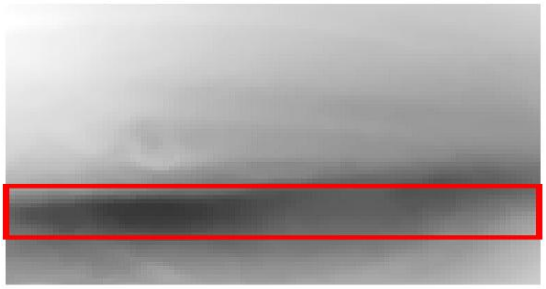

Figure 3-22: Example of image of a closed eye from the reconstructed video without noise. Peak of the blink sequence in video PCA 1 signal. The red square represents the eye region. See Figure 3-23 for calculations.

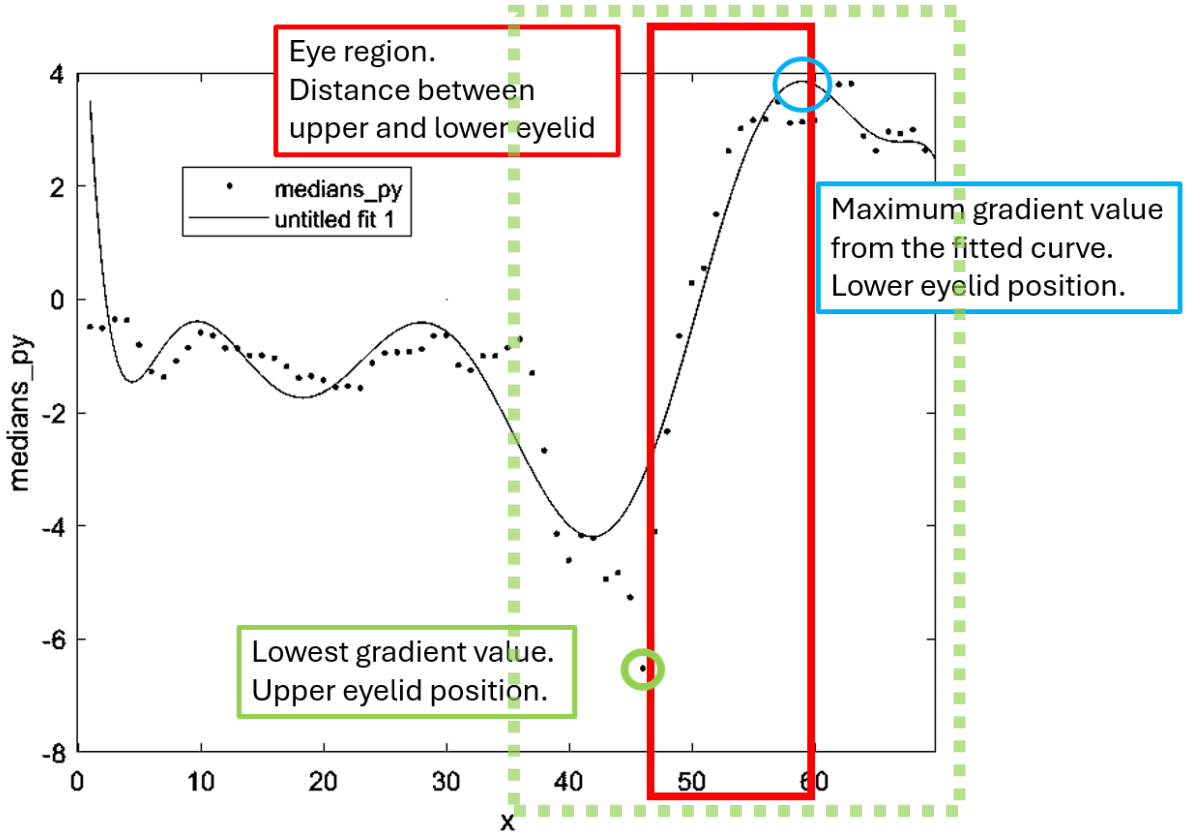

Figure 3-23: Gradient of the median of each row intensity of closed eye image from reconstructed video without noise, x represents the number of rows being zero the top row of the image and 70 the bottom row.

By performing this calculation for all detected blinks (peaks of the Fe1 signal) and obtaining the same values for the open eye (most probable image Fe1), a degree in percentage for each one of the blinks can be obtained.

3.1.3.2.2. Second study

The second study consists of obtaining the degree of closeness from the Fe1 signal (Figure 3-20) after visually checking that there is a complete blink in each video sequence studied. The maximum value of the signal of each video sequence is considered as the degree of closeness

of 100% and the Fe1 signal (Figure 3-19) from the maximum value of each one of the peaks it is possible to determine the degree of closeness and the quality of each one of the blink. The Fe1 zero value is considered as 0% of closeness.

3.1.3.2.3. Third study

The third study consists of obtaining a cross-correlation value of each one of the considered blink frames. The frames cross-correlated is a transformation from the ones that are characteristic of each one of the blink sequences and from the reconstructed PCA 1 frame considered as characteristic open eye frame of the video. This technique, besides obtaining the closeness percentage of the eye blink, will be used firstly as a threshold to classify if the blink if the peak obtained in the pca1 signal (Figure 3-20) is truly an eye blink or the peak is due to a different movement (non-centered eye frame, blur, …). In the next paragraphs an explanation of the technique used and the results obtained by comparing the percentage of closeness of the 3 studies is shown.

Cross-correlation study to determine the correctness of blinking sequence detection

As seen in the introduction part, involving the different algorithms for blink detection, the main result is the accuracy and correctness of the number of detected blinks (complete or incomplete) that happen during the video. Hence, not only the number of blinks but also the moment the blink happens in the video sequence. As it is difficult to classify between complete and incomplete during a video sequence by performing visual counting, the detection and differentiation by the algorithm itself make it even harder. In fact, to perform this task successfully, the eye detection algorithm that results in the cropped eye image from the original face video deleted from the server had to be done successfully. Furthermore, following the methodology from the algorithm of blink detection using the signal of the first principal component from the video sequence, a peak needs to be detected. When the frame (corresponding to the peak) is obtained and displayed, the image processing technique involving the detection of eyelid distance needs to correctly obtain the value in pixels. Comparing this eyelid distance with the value obtained from the photogram of the first principal component where the eye is fully open will give a degree of eye closeness that should be more than 25%, so we can consider that frame from a valid blink sequence. Even though this value is not actually the threshold in the literature review to consider the frame as part of a blink sequence, it is used to consider the sequences where the classification as an incomplete blink can be confusing. To make this step more efficient and accurate, as the eyelid distance

algorithm is not that accurate when the eye is fully open, we proceed to study the cross-correlation matrix between each one of the images detected as a blink (noiseless eye frame) from the principal component signal pca1. In this way, all the frames due to movement of the eye (change of target fixed) that are not due to blinking motion (eyelid closeness) that appear as peaks in the signal of the first principal component are identified as non-blinks and the results obtained will not be considered for the analysis of the blinking. This is identified as the last step, which makes the principal component analysis efficient, effective, and accurate for the blinking detection.

To obtain the degree of closure (closeness) of the blink, the image of the open eye will be used as a reference. Given the nature of the principal component analysis, we can take this image as the principal component 1 that represents most of the video (the closed eye is a very small fraction of all the frames in the video). Hence, we can measure the degree of closure by comparing the images of the open eye with those of the closed eye. This comparison will be made through cross-correlation. The cross correlation between two images, X and H, is defined by:

$$C_{X,H}(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X(m,n)\,\bar{H}(m-k,n-l) \tag{3-4}$$

Cross-correlation between images was calculated by means of the MATLAB function xcorr2. To facilitate the comparison between the shape of the images and the location of the eyelids in them (which tells us whether the eye is closed or not), the images are taken with zero means, and the gradient modulus of the images is then calculated to better locate their edges. In this way, the image of principal component 1 with zero mean is used as the comparison image X and the gradient modulus calculation (G_{pca1_0}) is applied to it. The image of the closed eye considered in each blink is introduced as the H image, also with zero measure and its gradient $\left(G_{blink_0}\right)$. The final comparison is made through a relative correlation measure C_{maxR}, MC_{maxR} defined as:

$$C_{maxR}(k,l) = \frac{C_{G_{pca1_0},G_{blink_0}}(k,l)}{\max C_{G_{pca1_0},G_{pca1_0}}(k,l)} \tag{3-5}$$

$$MC_{maxR} = \frac{\max C_{G_{pca1_0},G_{blink_0}}(k,l)}{\max C_{G_{pca1_0},G_{pca1_0}}(k,l)} \tag{3-6}$$

An example of the above calculations can be seen in Figure 3-24 for the case of one eye open and one eye closed.

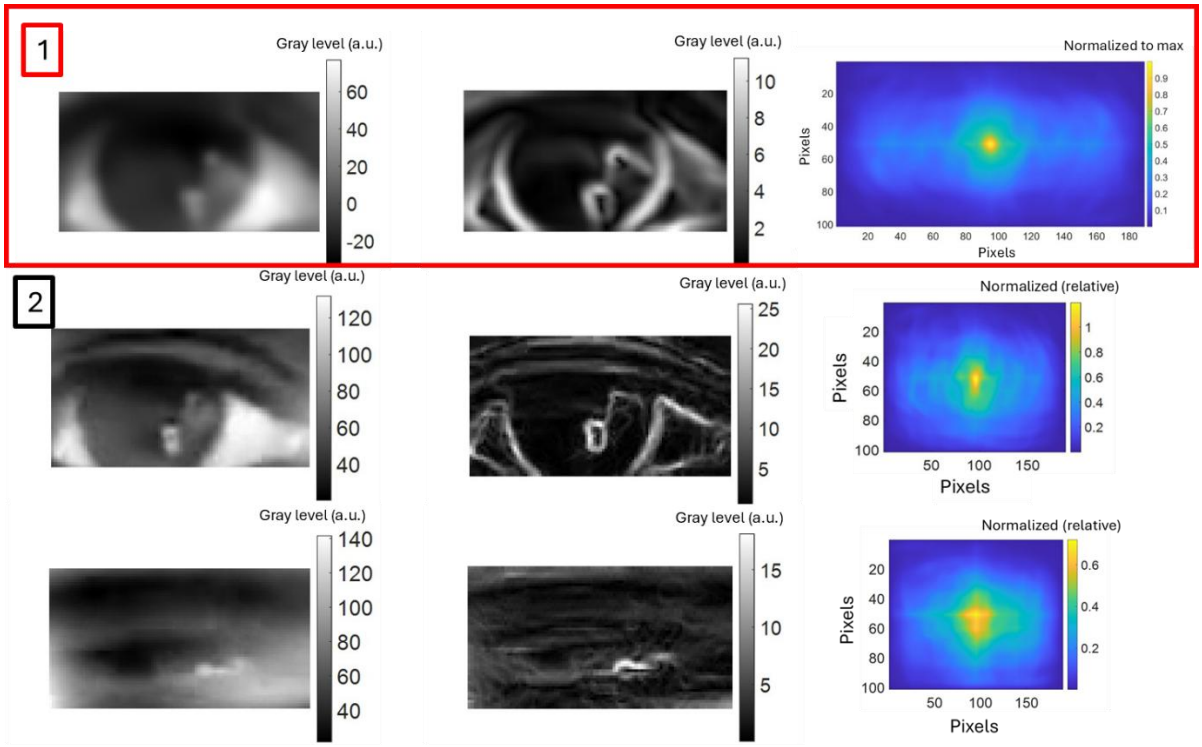

Figure 3-24: Sequence for the obtention of an estimated eye closeness. In the first row analysis for the PCA1 image (from left to right, original image with zero mean, gradient modulus, and $C_{G_{pca1_0},G_{pca1_0}}$). The same structure is used for an open and closed eye obtained from the reconstructed noiseless video in the second and third rows.

The value of MC_{maxR} is a measure of closeness since a low value means the image does not look like the open eye (it will be closed). Conversely, a high value indicates that the eye in that image will be open.

To ensure that the image is taken within the blink period with the highest degree of closeness, the MC_{maxR} is calculated for all images above the half prominence calculated according to the previous sections (see Figure 3-25). To do this, you must first ensure that the photos are of sufficient quality. This is because, since the videos last one minute and are made by the user-patient himself, movements may occur. After a visual inspection of several videos in which this effect occurs, it is concluded that the greatest effect is the introduction of a loss of resolution or "blur" due to this movement (see Figure 3-24).

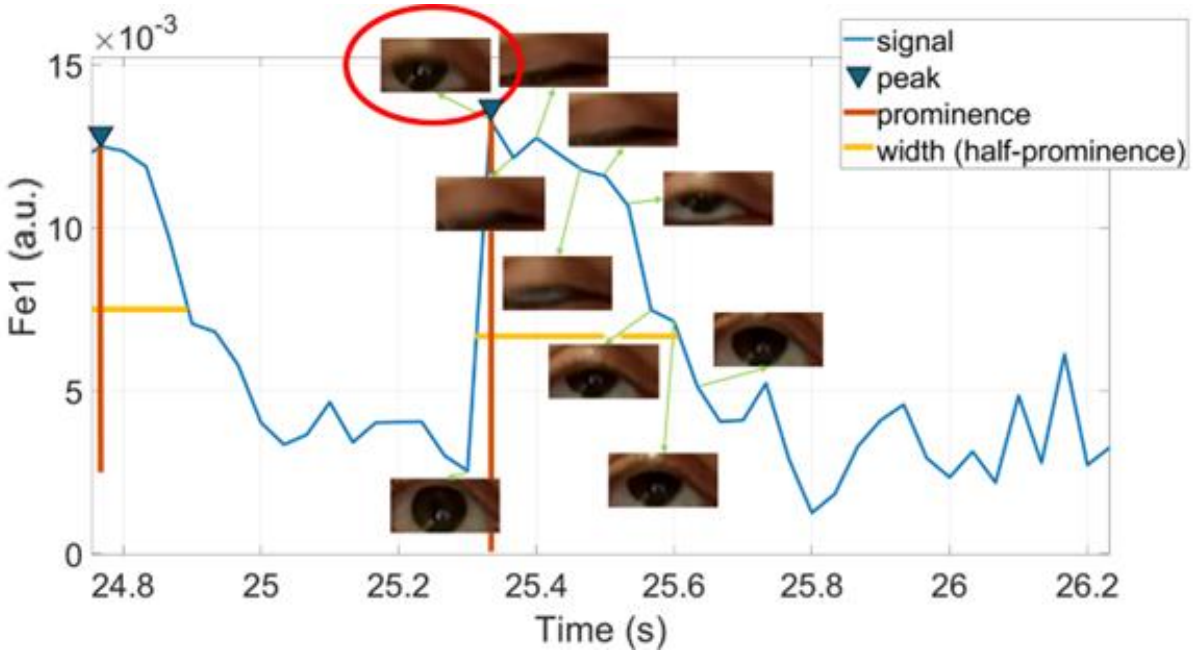

Figure 3-25: Example of a blink sequence where the highest value does not correspond with the most closed eye frame.

To study this phenomenon, we proceed to plot the 3D function of the normalized cross correlation function obtained for each analyzed image. In Figure 3-26 we observe that if we study the area covered by the function at half-height, an estimation of the blur can be obtained. This is because the cross-correlation measurement is performed on the gradients of the images. These images are compared with those of the first principal component. If any image has a greater blur than the others, its edges will be wider and its cross-correlation matrix will be wider. Hence, the width of this feature can serve as an indicator of the degree of blur, see Figure 3-26.

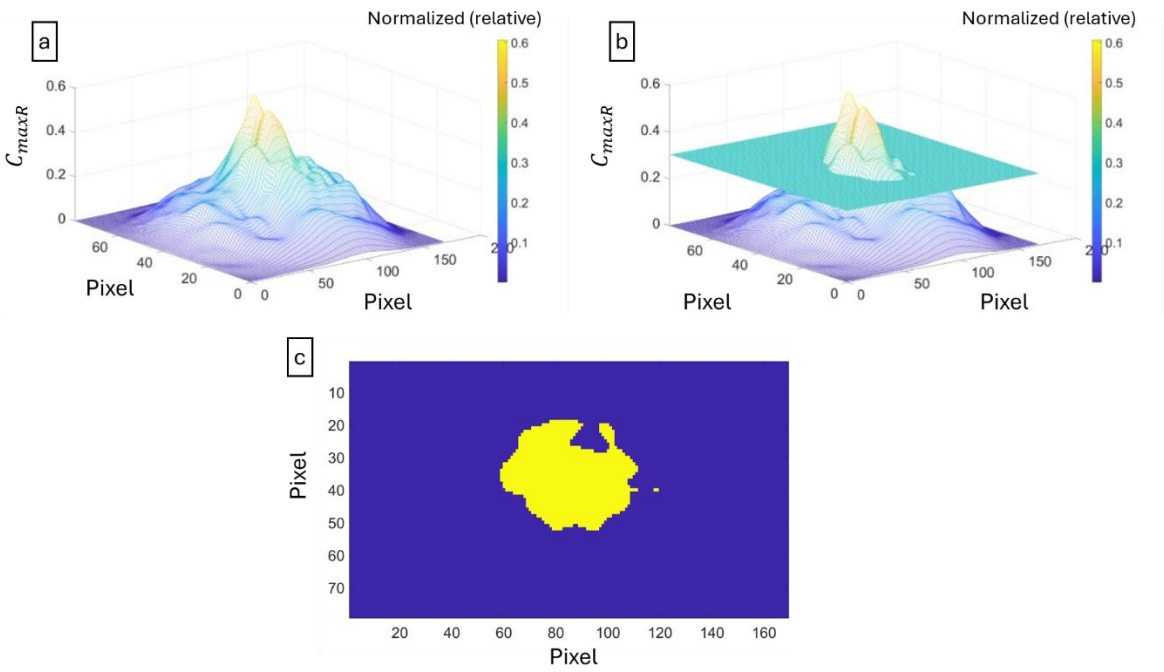

Figure 3-26: Obtention of the blur indicator, a) 3D normalized cross-correlation function, b) intersection at half prominence, c) resulting area from the intersection.

To characterize the width at half height of the cross correlations, a radius is calculated for each of them as $r_i = \sqrt{A_i/\pi}$. Those images that have a r_i greater than normal (outlier) are interpreted as a "blur" event and the frame in question is not considered for the calculations.

To consider a value as an outlier, the scaled median absolute deviation (MAD) is used (169). MAD is a robust estimate of the variability of a variable that is commonly used in the detection of outliers instead of using the standard deviation, precisely because of its character as a robust statistical estimator. Mathematically, in our case, it can be calculated using equation 3-7.

$$MAD = median(|r_i - median(r_i)|) \tag{3-7}$$

According to the reference (169) a threshold value can be defined given by equation 3-8 and 3-9. ($erfcinv$ is the inverse complementary error function (170)) Those images whose r_i are greater than that threshold will be considered as "outliers" and those images will not be considered in the calculations.

$$c = \left(-\frac{1}{\sqrt{2}\, erfcinv\left(\frac{3}{2}\right)} \right) \tag{3-8}$$

$$Threshold = median(A_i) + 3 * c * MAD \tag{3-9}$$

Another effect of manual image taken by the user in one minute can be the patient's voluntary or involuntary direction movement. This can cause the position of the iris to change on some images, producing abnormal PCA1 cross-correlation values (although this rarely occurs as the patient is instructed to look directly at the camera for the entire minute). of the video). However, another criterion for rejecting these frames is introduced to detect this anomalous situation.

To do this, a new threshold value is introduced for the maximum cross-correlation of the blinks MC_{maxR}. Figure 3-27 represents the probability distribution function (PDF) value of MC_{maxR} for all blinks detected in a sample of 20 videos from 20 different patients, together with the cumulative distribution function (CDF). The threshold for which the MC_{maxR} is considered is taken as below 0.95 of the CDF.

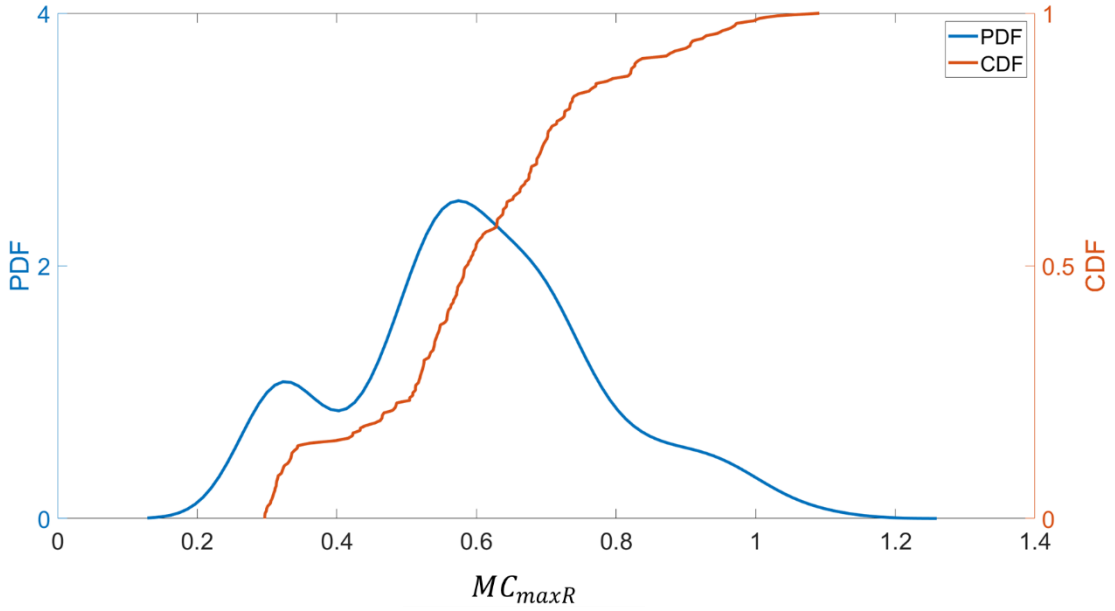

Figure 3-27: The probability density function and cumulative distributive function of cross-correlation values from frames are determined visually as part of a blink.

Finally, for each blink, the MC_{maxR} value is taken as the lowest of all the images considered. The frames considered are above the half-prominence, which is not an "outlier" due to their blur or the cross-correlation value above the threshold. The value of the degree of closure is estimated using equation 3-10, where the value of each blink is normalized to the highest degree of closure detected in the entire video of that user. (Given the length of the video, it is expected to be at least one complete or "closed" blink that will give the maximum scale of closeness for that patient).

$$Estimated\ closeness\ (\%) = 100\frac{1 - MC_{maxR}}{1 - \min\left(MC_{,maxN}\right)_{blinks}} \tag{3-10}$$

The summarized data concerning the 20 different videos from different patients can be found in Table 3-3. To get the maximum information about the video analyzed and to check additionally if the calculated ratio of eye closeness corresponds to the visual eye amplitude observed, we proceed to check both methodologies used degree obtained detecting upper and lower eyelid using image processing techniques and using a factorized first principal component coefficient for each one of the frames of the video sequence and checking the maximum amplitude obtained in each blink sequence. In the summary Table 3-3, the results of the R square parameter from fitting the degree obtained from both methodologies to a self-made eye visual closeness for each one of the videos, are also presented.

Table 3-3: Summary results from the analysis of 20 different patients' videos. From first to last column, the video number, the frame rate of the video camera used, the final single eye resolution frame,the visual count performed by the specialist, the detected peaks in the Fe1 signal of each video, the detected blinks after performing the steps below, the error in comparing the peaks in the Fe1 signal and the visual count, the error in the detection after performing the steps below.

Video number	Video frame rate (Hz)	Eye resolution (pixels)	Visual count of blinks	First detected peaks	Detected blinks	Error considering the peaks detection (%)	Error (%)
1	30	99x71	28	30	28	11.11	0.00
2	60	95x51	25	27	18	8.00	-28.00
3	30	61x29	15	15	14	0.00	-6.67
4	15	59x34	4	8	4	100.00	25.00
5	25	85x40	14	14	14	0.00	0.00
6	24	65x28	23	29	23	26.09	0.00
7	30	111x59	18	22	19	22.22	5.56
8	20	51x24	33	33	33	0.00	0.00
9	60	129x67	6	13	6	116.67	0.00
10	60	78x49	30	30	30	0.00	0.00
11	60	55x34	36	36	36	0.00	0.00
12	30	83x42	10	39	10	290.00	0.00
13	30	71x32	14	14	14	0.00	0.00
14	25	106x51	23	23	23	0.00	0.00
15	30	77x48	22	32	20	45.45	-9.09
16	15	33x21	39	42	38	7.69	-2.56
17	30	111x50	19	22	19	15.79	0.00
18	60	53x28	20	20	20	0.00	0.00
19	30	13x24	19	20	19	5.26	0.00
20	60	107x53	21	29	18	38.10	-14.29

From the data obtained, we can observe that the precision of the blinking count has increased significantly when the cross-correlation threshold is calculated to check the validity of the blinking. The overall precision, considering the total number of blinks and the ones finally detected as true blinks, is less than 5%, meaning that the results are higher than most of the literature-reviewed algorithms. But it clearly shows that in some cases the counting performed is not the same when each video is analyzed separately. The results of two videos (highlighted in bold and with red fonts) with an absolute error of higher than 10% are presented to check what is happening. From the observation of the Figure 3-28 and Figure 3-29 we can see that the videos considered where the algorithm eye blink detection is wrongly performing the task,

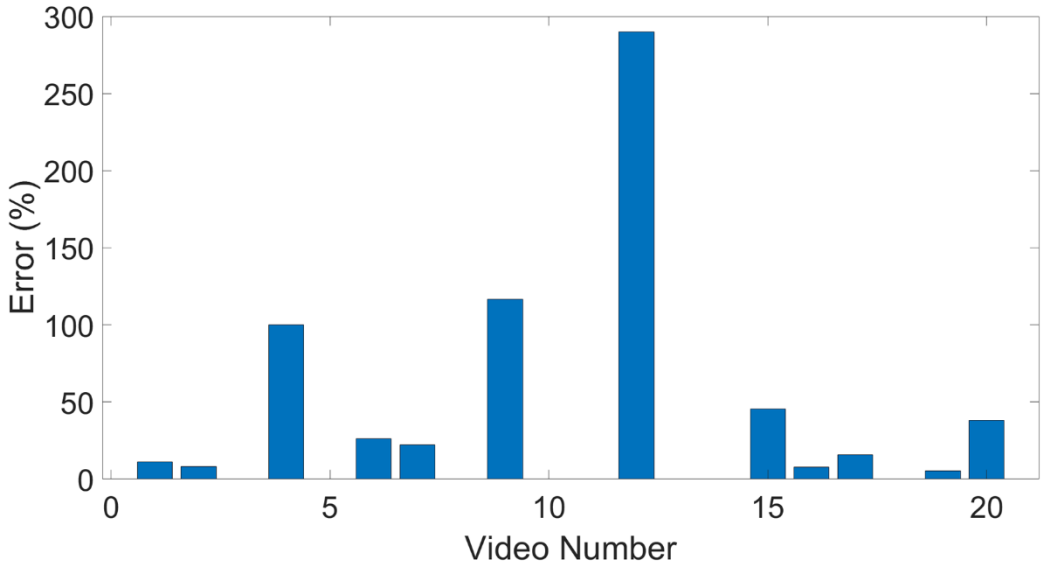

Figure 3-28: Error count before performing the normalization cross correlation threshold to analyze only the blinks.

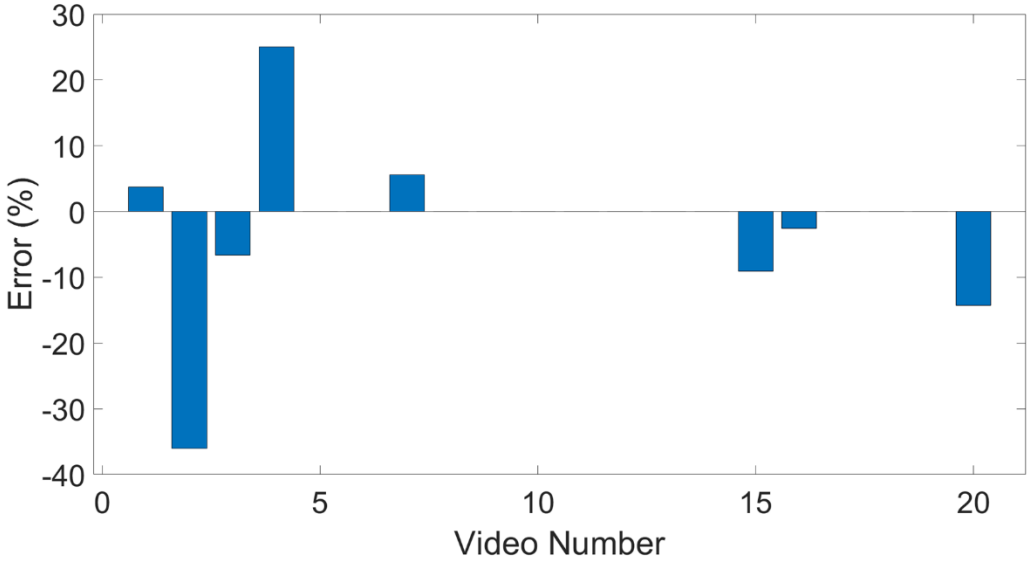

Figure 3-29: Error on blink detection after performing the normalized cross correlation threshold.

From the Figure 3-29 We can observe that videos 2 and 4 have a higher error on blink detection. To go further and check why these videos have that error, we can tell where the improvement on the current blink detection technique is used. Analyzing video number 2, we can observe the results obtained of the percentage of closeness obtained by a manual visual measurement using the most closed eye frame of each blink motion found in the video (Figure 3-30) by performing a simplified eyelid distance detection using the gradient of the half-closed or closed noiseless eye frame (Figure 3-31), or by factorizing the first principal component coefficient obtained for each one of the original frames of the video (Figure 3-32).

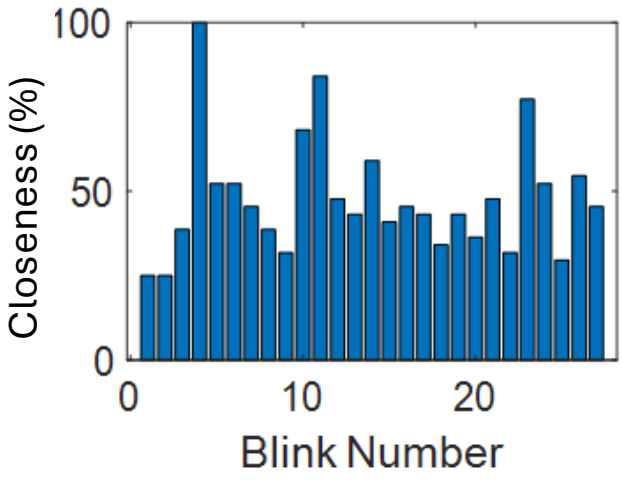

Figure 3-30: Visual measurement of the degree of eye closeness of video number 2.

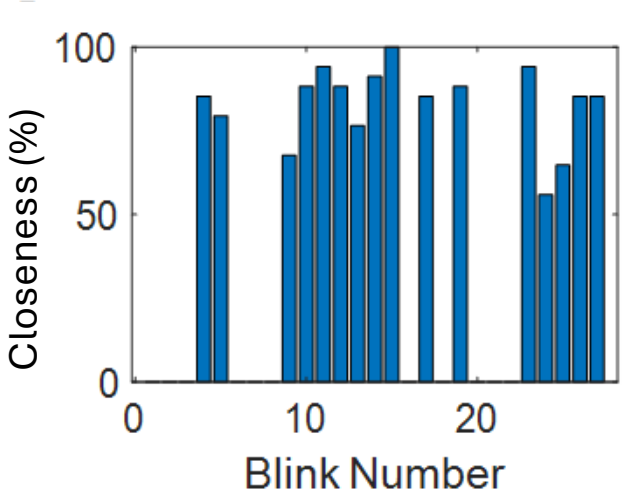

Figure 3-31: Percentage of closeness using Image processing to find the eyelid distances of video number 2.

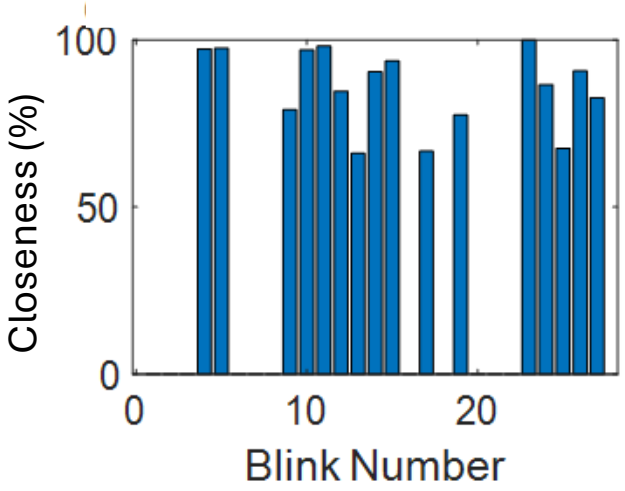

Figure 3-32: Percentage of closeness by factorizing the first component coefficient obtained for each one of the significant blink frames of the video, considering the heigh of the blink peaks of video number 2.

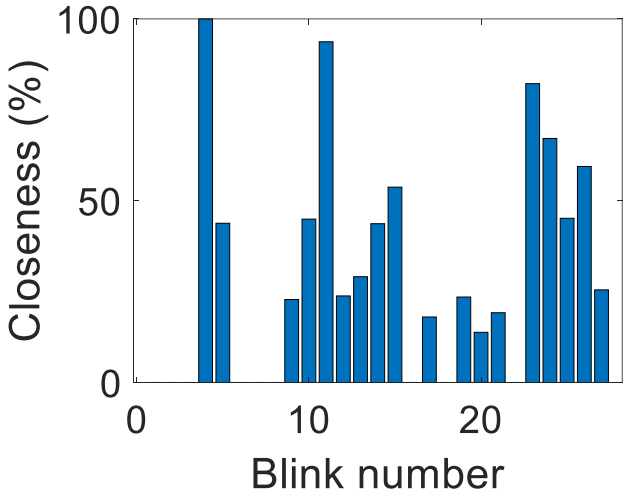

Figure 3-33: Percentage of closeness by studying the cross correlation values of each one of the characteristic blink frames of video number 2.

From the Figure 3-30, Figure 3-31, Figure 3-32 and Figure 3-33 we can conclude that the problem related to this video is caused by the number of half blinks and their amplitude, as most incomplete blinks have a degree of closeness (eyelid distance) on the most closed eye frame of the corresponding blink sequence lower than 50 %. Due to the number of incomplete blinks, the threshold chosen with the contrast criteria for this video is on the limit to include incomplete blinks. The video in question analyzed suffers from poor classification of the blinks detected since more than 80 % of the blinks observed have a ratio of closeness less than 60 %.

3.1.3.2.4. Selection of the best blink classification method and closeness classification.

The user performs the obtention of the true closeness value by manually analyzing each of the closest frames of each blink detected by the “findpeaks” Matlab function on the first principal coefficients for each of the original video frames. It is done by measuring the number of pixels in between the upper and lower eyelids, and to ensure that the full closed eye frame corresponds to a closeness in percentage of 100 %, we give the lowest distance measured considering all the blinks from the same video a value of 100% of closeness it can be observed.

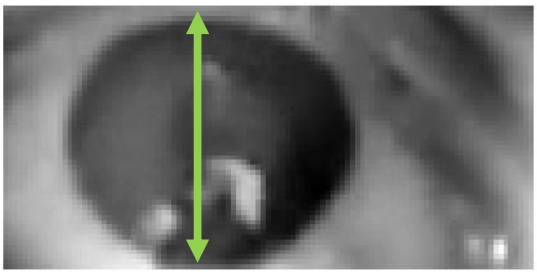

Figure 3-34: Manually measured the number of pixels between upper and lower eyelid.

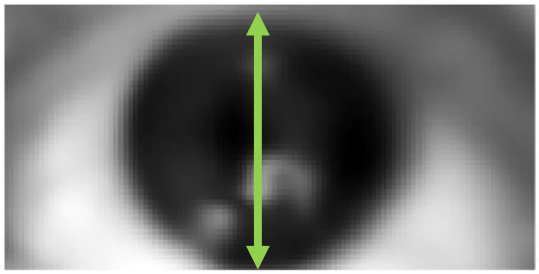

Figure 3-35: Eyelid distance measured in pixels on the first principal component image. The distance in question is used to calculate the closeness of the blink sequence.

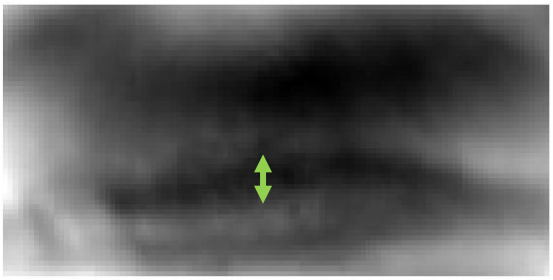

Figure 3-36: Eyelid distance measured in pixels on a closed eye frame.

$$Closeness\ (\%) = \frac{D_I - D_{pca}}{D_{pca}} \tag{3-11}$$

In equation 3-11, we observe how the measured closeness is obtained. D_I is the distance measured on the most significant eye blink frame sequence (Figure 3-32 or Figure 3-34) and the D_{pca} is the distance in pixels measured on the reconstructed first principal component image obtained after performing the PC Analysis; see Figure 3-35. The final closeness is adjusted considering that the minimum eyelid distance is usually set at 100% closeness if the frame shows a fully closed eye.

As the number of blinks obtained in video 4 was not very accurate, it is good to check the ratio of closeness and the number of blinks available in the video to see if it was correctly analyzed.

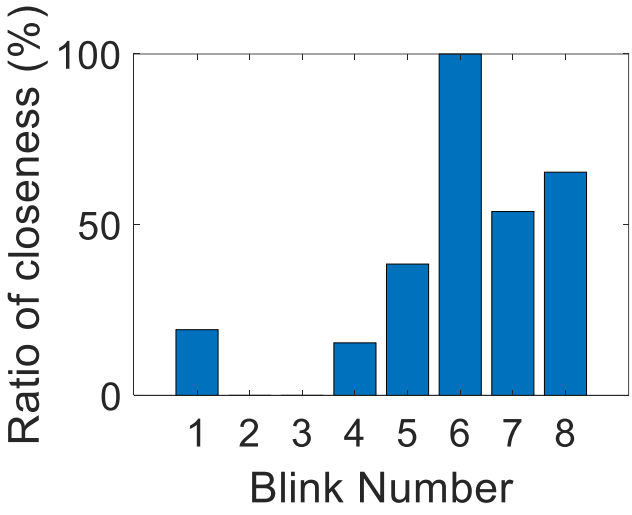

Figure 3-37: Visual measurement of the degree of eye closeness of video number 4.

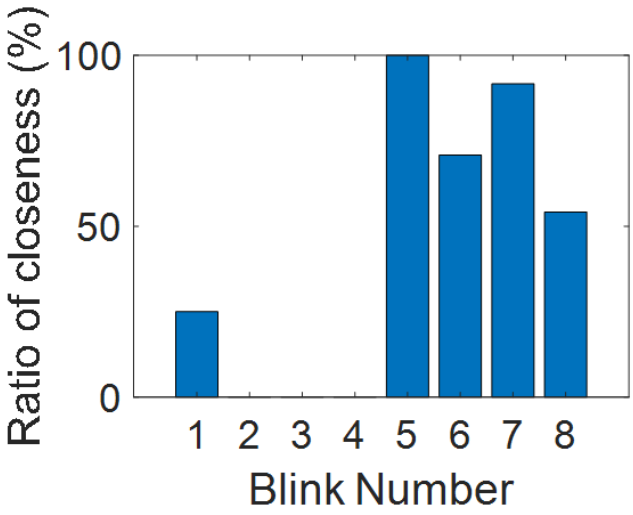

Figure 3-38: Degree of closeness using Image processing to find the eyelid distances of video number 4.

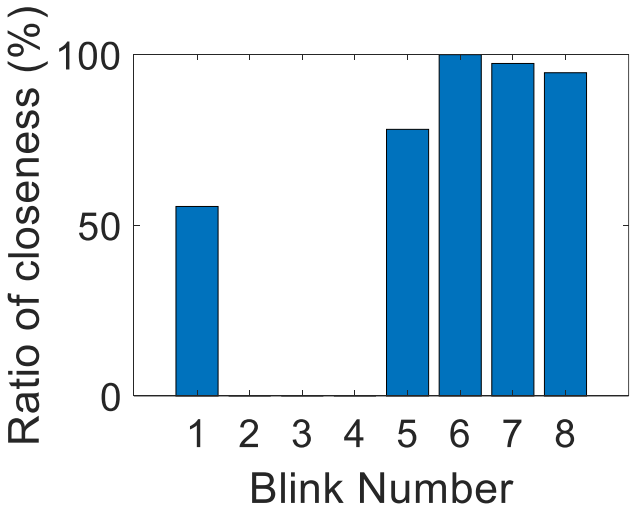

Figure 3-39: Degree of closeness by factorizing the first component coefficient obtained for each one of the significant blink frames of the video, considering the heigh of the blink peaks of video number 4.

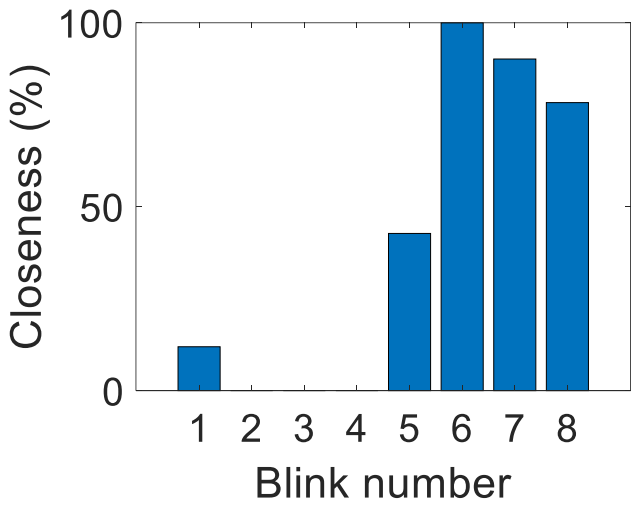

Figure 3-40: Percentage of closeness by studying the cross correlation values of each one of the characteristic blink frames of video number 4.

From the Figure 3-37, we observe that only four blinks should be considered in the analysis as it is considered a blink in the literature (82), when the ratio of closeness of the most significant blink frame is higher than 30%. As in this video, the first blink frame analyzed has a ratio of amplitude close to this cutting value, with the contrast blink image study appearing as considered as blink when the maximum amplitude doesn't reach that theoretical value. Additionally, the number of blinks available in that singular video is small, so the wrong classification of the first sequence as an incomplete blink when it is only a twitch (blink sequence ratio of closeness less than 30%) is translated into a higher percentage of error. When completing the checks of videos with a high percentage error, we can conclude by analyzing each one of the available and detected blink sequences, we can corroborate that the overall final error of 2.86 % found in Table 3-3 is a good indicator of the precision of the algorithm concerning 20 videos from the different patient eye, with distinct shape and size.

Finally, when the first 10 videos are analyzed, their characteristics are extracted, and the eyelid distance is measured for each observed blink using the three different techniques besides the manual obtention of closeness in percentage on each one of the videos, a final corroboration is performed to check if it is possible to accurately estimate the actual closeness of each one of the blink sequence in all videos with one of the techniques shown. It was found that the best fit, considering the manual detection of closeness as the output, is performed using the cross-correlation study with a third-degree polynomial fit where the independent term is set to zero

(see

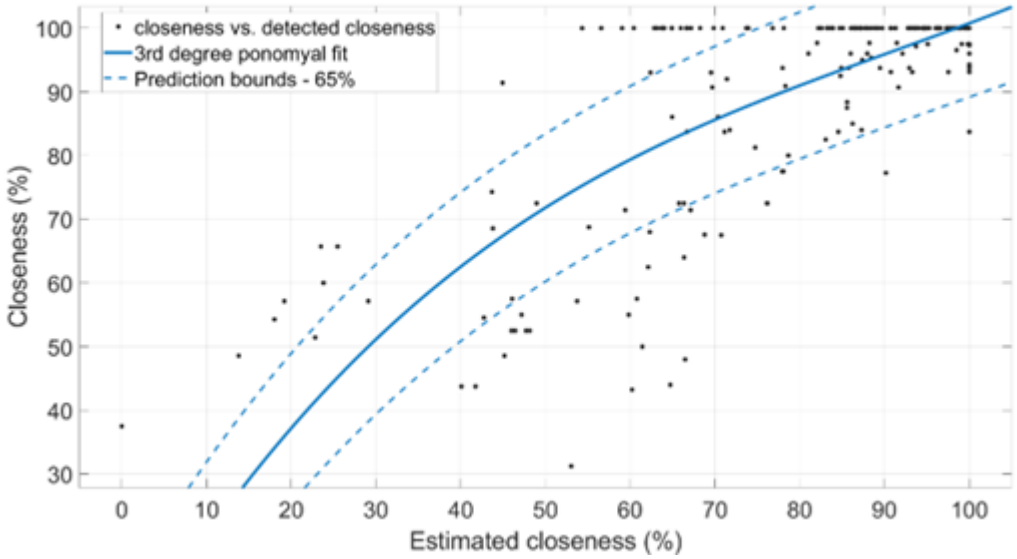

Figure 3-42).

Table 3-4: Pearson and Spearman correlation comparing the values of 200 blinks from the first 10 videos of detected closeness using each one of the explained techniques and the manually measured closeness.

Study Type	Pearson Correlation	Spearman Correlation
Study 1: IP (Image Processing)	0.38	0.30
Study 2: PCA	0.59	0.39
Study 3: Cross-correlation without blur correction	0.69	0.57
Study 3: Cross-correlation with blur correction	0.75	0.57

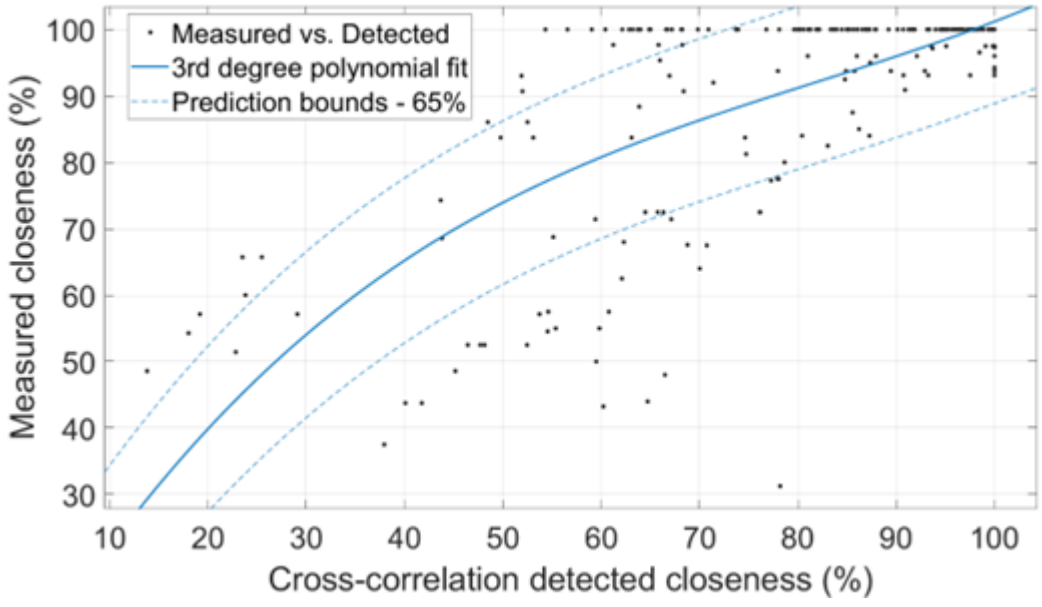

Figure 3-41: Comparison of the measured closeness and the detected closeness using cross-correlation normalized values of 200 blinks detected on the first 10 videos analyzed without the correction of blur. $R2 = 0.45, f(x) = p\_1 * x\hat{}3 + p\_2 * x\hat{}2 + p\_3 * x, where\ p\_1 = 9.7794e-05, p\_2 = -0.0240\ and\ p\_3 = 2.4344.$

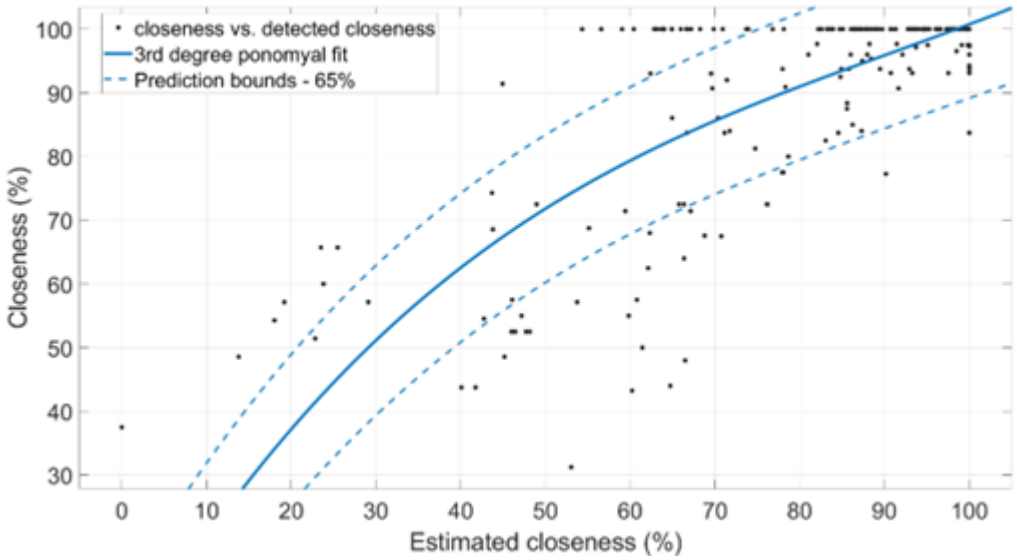

Figure 3-42: Comparison of the measured and detected closeness using cross-correlation normalized values of 200 blinks detected on the first 10 videos analyzed considering the blur correction. $R2 = 0.51, f(x) = p\_1 * x\hat{}3 + p\_2 * x\hat{}2 + p\_3 * x, where\ p\_1 = 6.9576e-05,\ p\_2 = -0.0190\ and\ p\_3 = 2.2129.$

After comparing the three techniques using 10 videos from different patients who were using or not using contact lenses, it was found that the degree estimation was better using the cross-correlation method. From the curve in Figure 3-52, we obtain the coefficients used to detect automatically an estimation of real closeness. Furthermore, the only value used to get closeness and the classification of the blinks depending on quality is done using the cross-correlation method.

The final closeness value in percentage is given by equation 9:

$$Final\ closeness\ (\%) = 6.9576e-05 * x^3 - 0.0190 * x^2 + 2.2129 * x, \qquad (3\text{-}12)$$

Where x is the estimated closeness in percentage obtained from equation 10

3.1.3.3. Blink characterization—features extraction from blink sequence.

The temporal evolution of individual blinks has been studied in the literature (171–173), both from an experimental and theoretical point of view, where it has been proven that an exponentially modified Gaussian (EMG) function can model the blinking amplitude (174), with an equation that describes it as:

$$f(x|\mu,\sigma,\tau) = \frac{1}{\tau}\exp\left(\frac{\mu}{\tau} + \frac{\sigma^2}{2\tau^2} - \frac{x}{\tau}\right)\Phi\left(\frac{x-\mu-{\sigma^2}/{\tau}}{\sigma}\right) \quad (3\text{-}13)$$

The resulting function is dependent on three different variables. σ and μ are the two parameters of the Gaussian distribution, where μ is the mean value, and σ is the standard deviation of the Gaussian distribution. The third parameter τ is the variable of the exponential distribution and gives the information related to the mean value of the exponential component. Consequently, the Exponentially Modified Gaussian function is identified as a convolution of 2 additive processes: a Gaussian function and an exponential function (174).

Speed and acceleration can be calculated from the blinking amplitude curve and the video frame rate. From these curves, many features can be extracted that characterize that specific flicker. This process has been described in the literature (171–173). In this work, the procedure described by J. Espinosa et al. will be followed based on the graphs and features described in Annex I.

Every signal blinking event will be adjusted to an exponentially modified Gaussian function as indicated by the previous equation 3-13, in order to extract the features. This involves selecting the relevant sections of the signal that correspond to each blink. The following procedure is then implemented:

1.- The probability distribution function (PDF) and the Cumulative Distribution Function (CDF) of the Fe1 signal are performed to locate the median and the value of Fe1 for a $\mathrm{CDF} > 0.85$ (threshold). A threshold value (u) is the difference between the threshold and the median.

2.- Each of the peaks of the Fe1 signal is chosen. The frame corresponding to this maximum will be the "0" point of the blink (peak). From this frame, the consecutive difference of the signal to the left and right of said point is calculated iteratively from a frame "i" starting with the zero value, which is the maximum located in each blink.

$$\Delta 1 = Fe1(i) - Fe1(i-1) \quad (3\text{-}14)$$

$$\Delta 2 = Fe1(i) - Fe1(i-2) \quad (3\text{-}15)$$

$$\Delta 3 = Fe1(i) - Fe1(i-3) \quad (3\text{-}16)$$

3.- For each iteration, the following conditions and the procedure for when the stopping condition given by equation 3-21 is met are calculated.

$$Condition1 = (\Delta 1 < 0 \ \wedge \ \Delta 2 < 0\ ) \tag{3-17}$$

$$Condition2 = \ (|\Delta 3| > u) \tag{3-18}$$

$$Condition3 = \ |Fe1(i-1) - Fe1(peak)| < u \tag{3-19}$$

$$Condition4 = |(i-1) - peak| < 2 * width \tag{3-20}$$

$$Stopping = Condition1 \bigvee Condition2 \bigvee Condition3 \bigwedge Condition4 \tag{3-21}$$

This procedure has been optimized for situations where several non-complete blinks can be grouped into one. Also, the start and end frames should be delimited to see the rise and fall of the Fe1 signal clearly. The localization is done by identifying whether the signal has dropped in two adjacent frames or multiple markers shifted over three frames more than three times the calculated deviation. Also, voluntary blinks of longer duration are considered by testing the next frame against some conditions. Lastly, blinks overlap or blink very next to each other, which is not considered based on the full width at half height. This method can also differentiate between spontaneous and voluntary blinks and isn't confounded by events that happen closely together. Figure 3-43 summarizes the previous process.

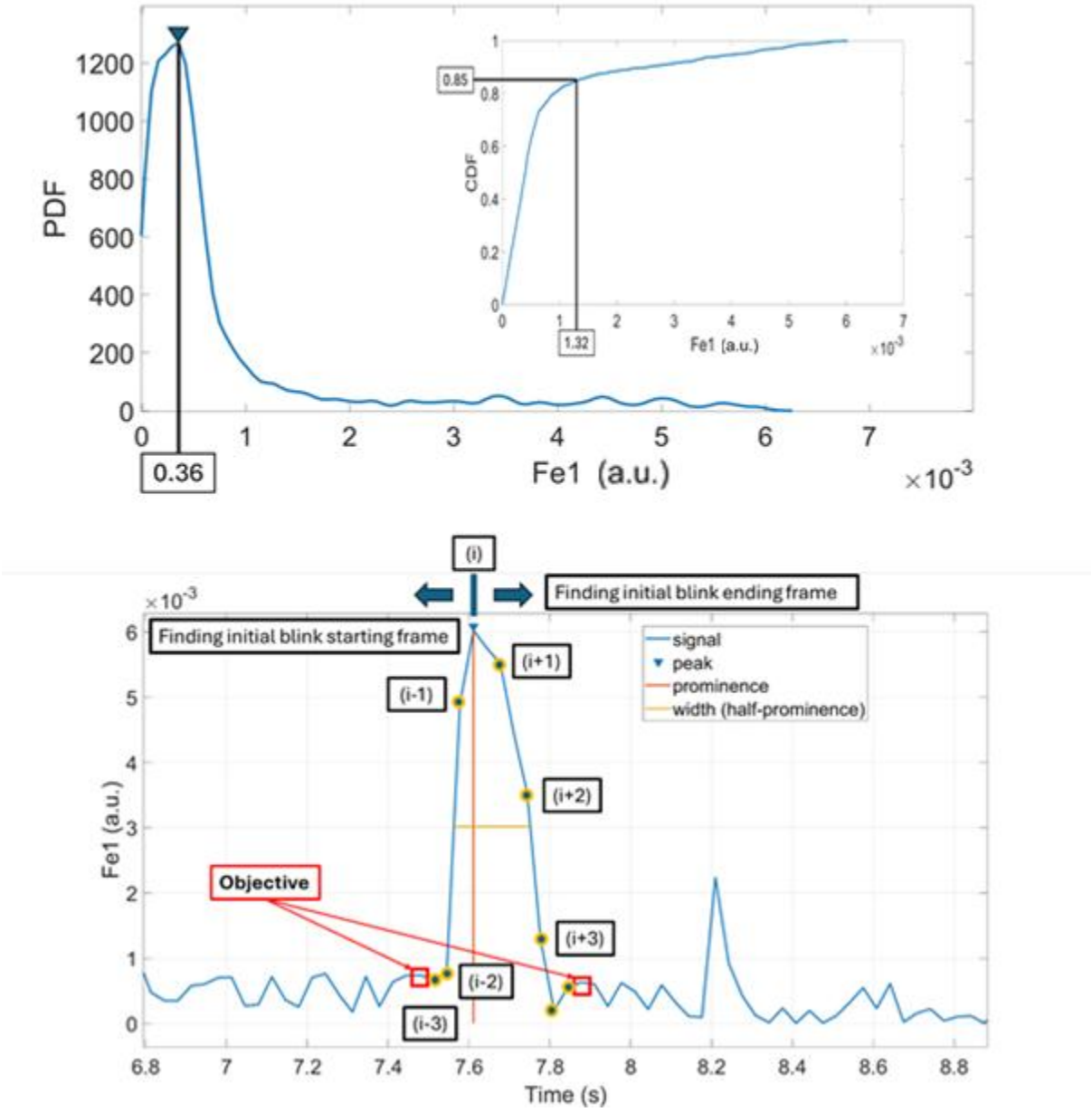

Figure 3-43: PDF and CDF value of the Fe1 signal (top) along with the threshold value and median of the calculated signal. Below is the search procedure for the start and end points of a blink.

To characterize the blinking curve and consider that the signal is generated from a video that will have a maximum of 60 frames per second, we use the interpolation function to obtain more information regarding the speed and displacement of the upper eyelid. The speed curve identifies the initial and end points, knowing that the speed is zero in the interpolated curve. As explained in the introduction, for the correct feature extraction, the resulting curve is fitted to an exponentially modified Gaussian function (Figure 15).

The fitting consists of an iterative process to obtain the minimum root square error between the isolated final signal in pixels for each blink event in the video. The transformation from Fe1

signal values to pixels is performed independently in each blink event using the value of closeness obtained from equation 3-12. First, the isolated signal in pixels is normalized using the cumulative signal value (norm). Then, an initial value is given to the three parameters needed (σ, μ and τ) to obtain the fitted EMG distribution (equation 3-13). The norm and starting values are explained below.

- **norm:** the cumulative value of the blink signal in pixels to perform a normalization of the curve before performing the fitting to the exponentially modified Gaussian.
- $\boldsymbol{\sigma}$**:** square root of the variance of the normalized curve minus the square value of $\boldsymbol{\tau}$ defined on (3-22), see equation 3-21, (174).

$$\sigma = \sqrt{\sigma_x^2 - \tau^2} \tag{3-22}$$

- $\boldsymbol{\mu}$**:** mean value of the normalized blink curve minus $\boldsymbol{\tau}$ parameter defined on (3-24), see equation 3-23, (174).

$$\mu = \bar{x} - \tau \tag{3-23}$$

- $\boldsymbol{\tau}$**:** standard deviation of the normalized blink curve multiplied by 0.8, see equation 3-24, (174).

$$\tau = \sigma_x * 0.8 \tag{3-24}$$

When the iteration is finally done, the curve is fitted, and the resultant curve R-squared value is registered.

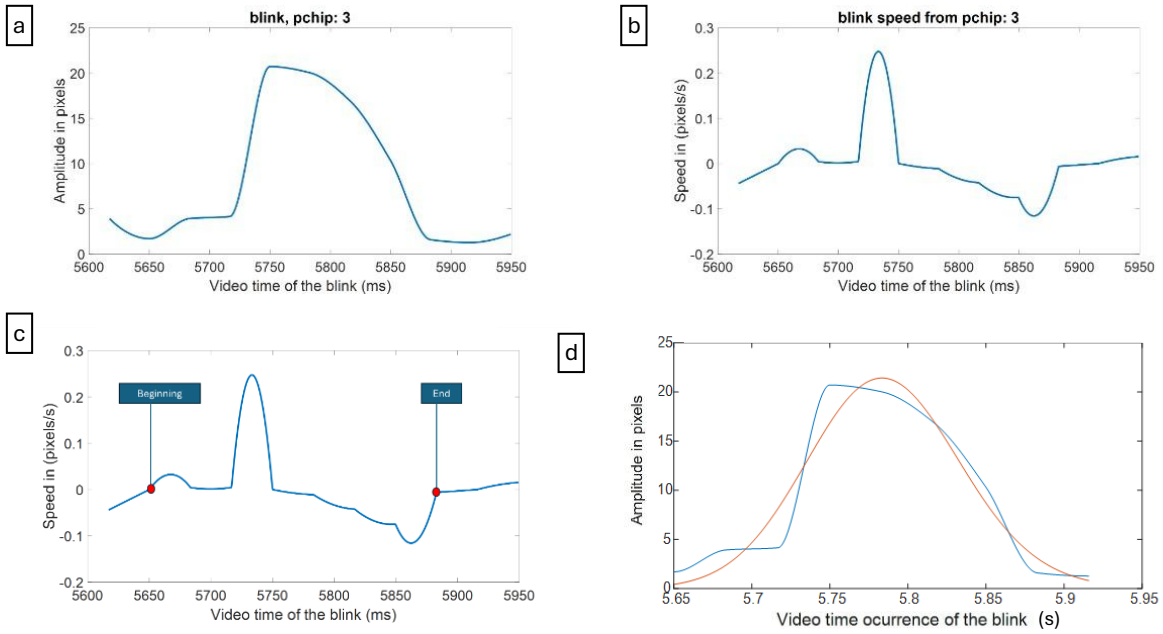

Figure 3-44: Isolating and fitting a blink sequence to an Exponentially Modified Gaussian function. a) Interpolated and transformed the Fe1 signal to pixels using the closeness (%) and the distance between the eyelids of the pca1 reconstructed image. b) blink speed, gradient of function a). c) Detection of the beginning and end of the blink sequence. d) fitting to an exponentially modified Gaussian function.

Once the fit to the Gaussian curve has been obtained, each blink and different video characteristics can be calculated. The features obtained from each isolated blink event besides the ones from the J. Espinosa et al. study are as follows:

- **half_width_left_ms_:** Duration of the upper eyelid lowering movement in milliseconds, taking as reference the half height.
- **half_width_right_ms_:** Duration of the upper eyelid raising movement in milliseconds, taking as reference the half height.
- **half_width_ms_:** Blink duration in milliseconds at half height.
- **width_left_ms_:** Duration of the total eye closing movement in milliseconds.
- **width_right_ms_:** Duration of the total eye-opening movement in milliseconds.
- **closeness__:** Percentage of eye closeness.

Additionally, two control variables are registered to check the validity of the fitting and that the fitted curve corresponds to an actual blink (closeness higher than 30%).

The parameters obtained for each video are the IBI (Inter-Blink interval) and duration mean and standard deviation values, in addition to the first, second, third, and fourth statistical order of all the feature distributions obtained for each blink event. Lastly, for each patient video, the blinking dynamics are characterized by transforming the Fe1 signal to mm of the blinking amplitude in the frequency domain.

The initial signal Fe1 is transformed to account for the degree of closure of the blink-in-length units. To achieve this, the Fe1 value of the frames involved in each localized blink is adjusted according to the procedures outlined in the previous sections. The Fe1 value for each frame is multiplied by a factor that correlates the maximum amplitude of the blink to the degree of closure estimated from the preceding Equation 3-12. A conversion to millimeters is performed by calculating and comparing the pixel length of the palpebral cleft with the user input, see subsection 3.1.3.3.1. The result of this process is represented in Figure 3-45

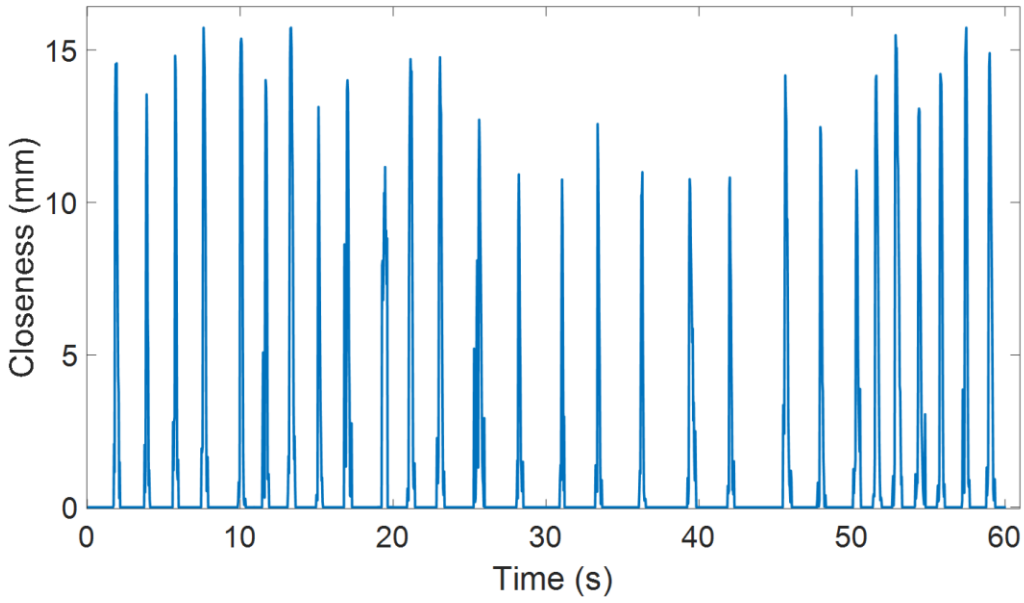

Figure 3-45: Transformed Fe1 signal (Figure 3-19) where the frames considered part of a blink sequence correspond in millimeters to the detected closeness of the eyelids.

The dynamics of the blinks in the video are characterized by calculating the Power Spectrum Density (PSD) of the degree of closure in mm with respect to time. This PSD is modeled as type $1/f^{-\alpha}$, with f being the frequency in Hz. This type of PSD commonly appears in complex and biological processes where different processes with different relaxation periods may be involved. (175–182). This type of noise also appears directly related to the dynamics of blinking in various cognitive processes (90). Specifically, the formulation for the PSD is used as

$$PSD = 10^b f^\alpha \tag{3-25}$$

with f being the frequency in Hz, b a parameter related to the amplitude, and α the parameter associated with the complexity of the blinking process (183). The figure shows the results of this adjustment, just with the signal spectrogram made with the "pspectum" function of the MATLAB "Signal Processing Toolbox" library.

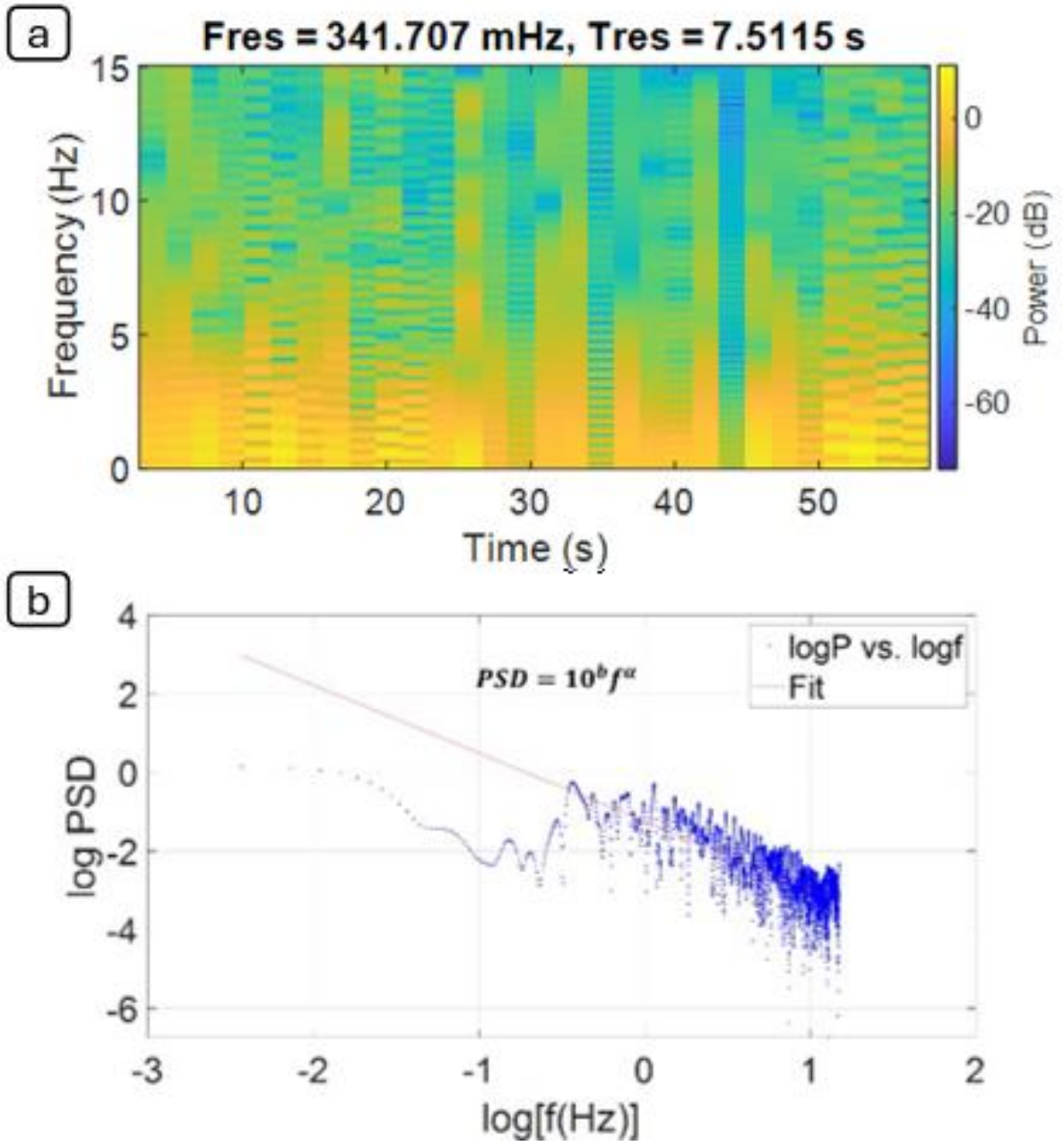

Figure 3-46: a) Spectrogram of the signal in millimeters shown in Figure 3-45, b) Characterization of the PSD to 1/f using parameters alpha, and b. PSD $= 10^b f^\alpha$.

Once the fit to the Gaussian curve has been obtained, the calculation of each blink characteristic, reflected in the introduction, chapter 2.5.1, is now possible. Obtaining the times

when the speed is zero corresponds to the beginning t0, end t8, and maximum amplitude t4, the points of maximum t2 and minimum speed t6 correspond to the points of zero power when the forces carried out by the muscles of OO and LP are null. At these points, too, the power becomes null. The points at which the normalized power becomes maximum correspond to those at which the force exerted by the OO muscle in the closing part of the blink and the maximum force of the LP muscle corresponding to the opening of the blink, in the descent correspond to t1 and on the rise correspond to t5 of the upper eyelid to complete the closing or opening of the eye. The power is minimal in the part of the descent t3 and the part of the ascent at point t7 since the effort this time is maximum and is made by the muscle that moves in the opposite direction. At these points, the effort made by the LP muscle is maximum, but when the eyelid is closing, t3 stops the closure, while at point t7, the maximum effort is made by the OO muscle to prevent the opening movement.

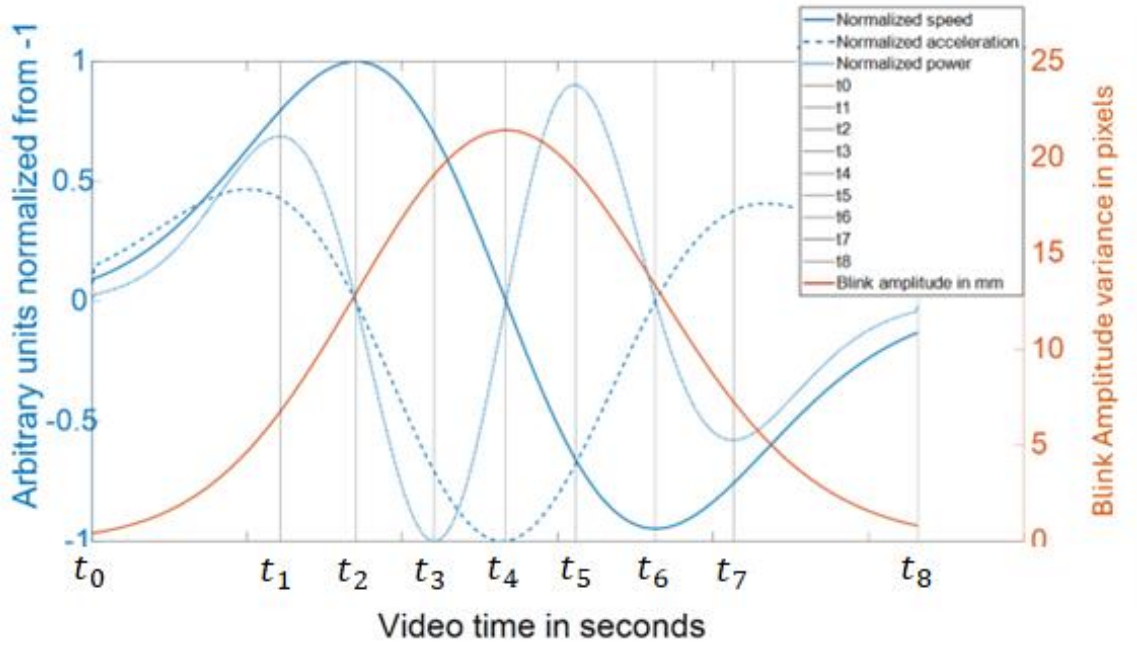

Figure 3-47: In red, amplitude in pixels of the fitted EMG function characterizing a blink event. In blue, functions are obtained from the EMG to display the normalized speed, acceleration, and power. In dark grey, the different characteristic times obtained by analyzing the different curves are shown.

3.1.3.3.1. Conversion from pixels to mm

For comparison purposes, as shown in Figure 3-45, the results and the signal are then transformed to millimeters. The calculation of the longitudinal palpebral cleft in pixels length is carried out from the Fe1 signal shown where an image of the open eye is identified, being the one with the lowest coefficient in the first 50 frames of the video. In this frame, the eye is completely open. To recognize the position of the eye's corners, we first eliminate the noise

due to lighting by adjusting the light intensity to a degree 3 in the x and y axes, giving rise to a that can be visualized with Matlab's mesh function. We are left with a noise-free image by subtracting the surface from the original image in gray levels and adding the total average value of the average light intensity surface.

The average signal of each column is calculated from the open-eye image without noise. The maximum values will correspond to areas where the sclera is very present. These two peaks have a corresponding base that coincides with the beginning with the iris area and the skin area. Considering the value of the prominence of these peaks, the points are located at a distance of 80% of the prominence concerning the maximum points; the column of the beginning and end of the length of the palpebral fissure is obtained. The following images show that in column 6 begins the sclera area, and in column 100, the sclera ends.

As the final purpose is to analyze the videos and data obtained from the patients using a self-made application, some information needed, such as the pixel-to-mm variation, is stored in a database using MySQL database manager. Access to the server to get information related to the size of the palpebral cleft introduced by the user is automatically done to have the actual size of the patient's eye and the distance between eyelids at the end. This connection is done using the Database Toolbox in Matlab.

Considering that the palpebral length longitude is given in millimeters by the user the first time accessing the application of data recollection, a conversion factor can be calculated to obtain all the results in millimeters instead of pixels. Additionally, To give the information in degrees, it has been proven in a study (184) For most subjects, a movement of the upper eyelid of 10 degrees translates, measured in millimeters, to 2.6 (+-0.1) mm. Additionally, it is assumed that the angular displacement of the eyelid is equal to 1.1 times the angular displacement of the eye.

Figure 3-48: Eye frame from a video sequence in gray level with the application of media_local_polinomica function.

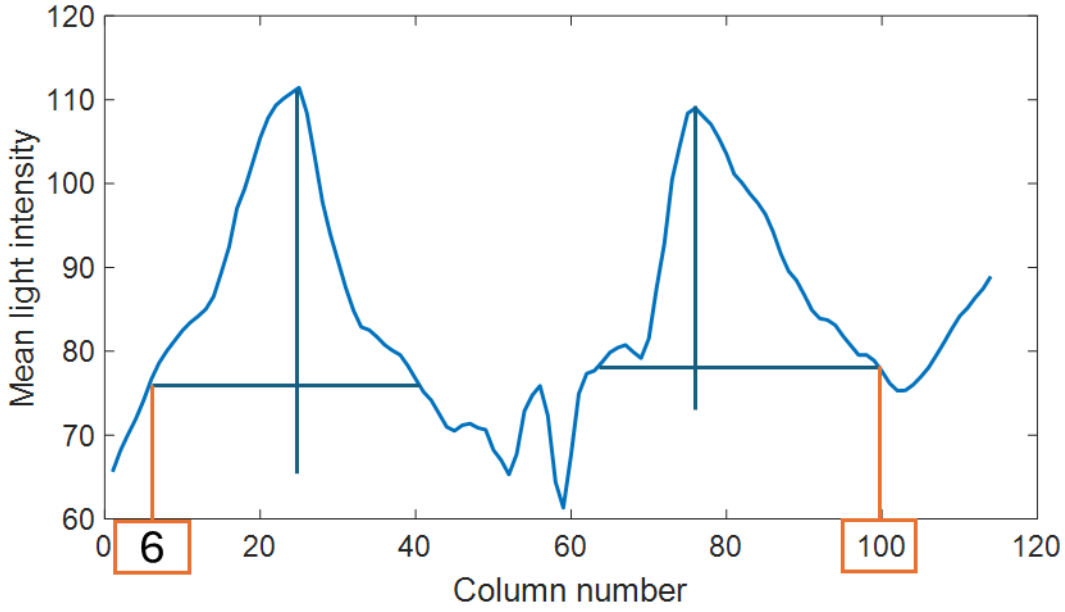

Figure 3-49: Mean pixel intensity of each column of the Open-eye frame; see

Figure 3-48.

3.1.3.4. Analysis of results obtained using the automatic blink detection algorithm

This section exposes some of the results obtained after performing all the steps shown in this section 3.1.3.

Table 1 presents some results of the algorithm used as an automatic blink counter in a one-minute video recorded by users. A blink is considered complete when the degree of closure exceeds 25%. Among these, a blink is classified as complete when the closure is greater than 75% and incomplete if it is less than that but greater than 25% (82). For this test, 20 recordings from various individuals were analyzed, each with or without contact lenses. For each video, a visual count of the number of blinks is conducted (based on the observation of each frame of the video, ensuring very little uncertainty). Finally, the percentage of error in the automatic detection of blinks is calculated. The results in Table 1 display two groups of records. One group has an accurate count (in this context, the average number of blinks counted across all videos is 21 plus or minus 9, consistent with the figures provided by the algorithm). In contrast, the other group exhibits a higher error rate. After reviewing the respective videos, this may be attributed to multiple blinks with degrees of closure close to 25%, which complicates their classification as blinks. The inherent error in this calculation causes fluctuations in the detected number of blinks. Furthermore, if the number of blinks is not substantial, these fluctuations can

result in significant changes in the error percentage. In any case, the average error is 5.85% across all videos. (100,101,105,110,122,126)

Table 2 shows the data for the degree of closure (closeness), Inter-Blink Interval (IBI), and blink duration for the videos considered, together with their standard deviations(185). To better estimate the duration of the blinks, this duration has been calculated from the adjustment of the blink signal to the EMG function (174) explained in the section on methodology.

For an understanding of the results, the probability density function (PDF) of these magnitudes, as well as the PDF of the Signal to noise ratio (SNR) of each of them, has been represented in Figure 3-51, Figure 3-52, and Figure 3-53.

Two distinct groups of records can be observed, more visible in the graphs on the mean blink duration (and within these, those related to the SNR, Figure 3-53). In general, these videos tend to have a shorter duration of the blink or a more significant variability in duration. To a lesser extent, these groups are also seen in Figure 3-52, Figure 3-53, and Figure 3-54, corresponding to the other magnitudes. If we compare the fundamental characteristics shared by these records, Table 1 shows that a large part of them are from presbyopic patients (186). Presbyopic patients present a more significant variability among themselves in these blink magnitudes, especially those associated with standard deviations.

In Table 3-7, the same parameters obtained in Table 3-6 are now calculated only considering the complete blinks and amplitude of eye closeness above 70 %. In Table 3-8 the results presented are related only to the incomplete blinks.

Finally, in Table 3-9 the results concerning the blink dynamics parameters considering the entire signal amplitude of eye closeness in millimeters are obtained. The characterization of this signal in the frequency domain by α and b parameters from equation 3-25. Figure 3-54 shows the probability density function for $|\alpha|$. The outliers to the left and right to the peak point correspond mainly to presbyopic patients.

Table 3-5: Summary results from the analysis of 20 different patients' videos (1/5). The second column identifies the presbyopic patients with "Y". Column 3 gives information about using contact lenses during the recording. Columns four and five give information related to the camera used (Frame rate and resolution), column six is the visual count of blinks, the seventh is the detected blink using the proposed method, and lastly, the eighth is the absolute error.

Video number	Presbyopic	Contact lens use	Video frame rate (Hz)	Eye image (pixels)	Visual count of blinks	Detected blinks	Error (%)	Mean R^2
1	N	N	30	99x71	28	28	0.00	0.92
2	N	N	60	95x51	25	18	28.00	0.89
3	N	N	30	61x29	15	14	6.67	0.64
4	N	N	15	59x34	4	4	0.00	0.95
5	N	N	25	85x40	14	14	0.00	0.89
6	Y	N	24	65x28	23	29	26.09	0.81
7	Y	N	30	111x59	18	20	11.11	0.7
8	Y	Y	20	51x24	33	33	0.00	0.91
9	Y	N	60	129x67	6	7	16.67	0.69
10	N	N	60	78x49	30	30	0.00	0.95
11	N	Y	60	55x34	36	36	0.00	0.71
12	Y	N	30	83x42	10	10	0.00	0.83
13	Y	N	30	71x32	14	14	0.00	0.76
14	Y	Y	25	106x51	23	23	0.00	0.93
15	N	N	30	77x48	22	20	9.09	0.87
16	Y	N	15	33x21	39	38	2.56	0.89
17	Y	Y	30	111x50	19	19	0.00	0.94
18	Y	N	60	53x28	20	19	5.00	0.95
19	Y	N	30	13x24	19	19	0.00	0.92
20	Y	N	60	107x53	21	23	9.52	0.89

Table 3-6: Summary results from the analysis of 20 different patients' videos (2/5). The results shown are, considering the number of all blinks detected in the video (column 2), the mean closeness in percentage of the blink events (column 3), and its standard deviation (column 4), the mean Inter-Blink Interval in seconds (column 5), and its standard deviation (column 6), the mean blink duration in milliseconds (column 7) and its standard deviation.

Video number	Detected blinks	closeness (%)	std(closeness)	IBI (s)	std(IBI)	Mean blink duration (ms)	std(duration)
1	28	85.44	10.85	1.97	0.71	350.50	96.08
2	18	61.90	22.29	2.17	1.97	201.70	53.11
3	14	87.59	12.37	3.74	3.30	324.56	55.94
4	4	75.07	28.33	7.68	6.23	476.75	122.38
5	14	90.40	6.00	3.78	2.34	610.88	202.43
6	29	86.95	6.97	2.00	0.76	363.58	72.83
7	20	89.06	11.83	2.66	1.57	557.39	431.89
8	33	94.97	4.83	1.80	1.36	354.82	82.55
9	7	79.73	21.18	4.64	3.92	229.16	134.65
10	30	99.16	0.95	1.97	0.26	370.63	64.34
11	36	95.38	4.05	1.61	1.38	273.73	64.39
12	10	81.68	13.55	1.47	1.57	404.34	131.88
13	14	89.40	9.38	3.84	2.34	315.65	62.05
14	23	92.58	5.48	2.54	0.55	386.68	96.63
15	20	90.29	12.94	1.82	1.52	441.96	112.38
16	38	90.11	8.57	1.40	0.48	458.88	233.94
17	19	84.65	7.53	2.68	1.02	380.59	59.48
18	19	83.13	13.93	2.83	2.01	256.65	67.32
19	19	84.65	10.31	2.62	1.84	387.44	66.81
20	23	88.98	10.85	2.06	1.08	376.06	69.01

Table 3-7: Summary results from the analysis of 20 different patients' videos (3/5). Same results are shown in Table 3-6 but considering only complete blinks.

Video number	Complete Blinks	closeness (%)	std(closeness)	IBI complete (s)	std(IBI_c)	duration complete (ms)	std(duration)
1	21	90.77	6.54	2.85	2.65	363.36	91.88
2	5	90.58	8.01	12.85	11.81	216.62	28.04
3	11	93.18	12.37	5.24	4.05	324.56	55.94
4	3	95.41	4.03	12.12	9.42	368.39	102.42
5	14	90.40	90.40	3.78	2.34	610.88	202.43
6	29	86.95	6.97	2.00	0.76	363.58	72.83
7	19	91.44	5.87	3.11	1.42	510.56	393.98
8	33	94.97	4.83	1.80	1.36	354.82	82.55
9	5	90.78	8.15	13.53	14.41	284.05	118.91
10	30	99.16	0.95	1.97	0.26	370.63	64.34
11	36	95.38	4.05	1.61	1.38	158.06	45.09
12	7	89.41	6.49	8.11	4.13	414.69	147.82
13	13	91.54	5.52	3.59	2.26	315.65	62.05
14	23	92.58	5.48	2.54	0.55	386.68	96.63
15	19	92.83	6.90	3.10	2.18	430.15	102.48
16	36	91.38	6.83	1.62	0.83	460.55	240.32
17	18	85.73	6.14	3.13	1.29	386.93	54.52
18	16	88.36	7.00	3.59	3.83	259.01	67.94
19	16	87.96	6.11	3.32	2.18	385.35	65.99
20	22	90.90	6.22	2.41	1.25	379.90	68.50

Figure 3-50

Table 3-8: Summary results from the analysis of 20 different patients' videos (4/5). Same results are shown in Table 3-6 but considering only incomplete blinks.

Video number	Incomplete blinks	Closeness (%)	std(closeness)	IBI incomplete (s)	std(IBI_i)	Duration incomplete (ms)	std(duration)
1	7	69.43	0.92	5.14	2.78	311.90	98.08
2	13	50.87	15.00	4.02	4.39	194.24	60.58
3	3	67.09	5.14	15.43	5.37	-	-
4	1	65.29	-	-	-	420.59	-
5	0	-	-	-	-	-	-
6	0	-	-	-	-	-	-
7	1	43.92	-	-	-	1400.33	-
8	0	-	-	-	-	-	-
9	2	52.11	18.35	14.83	0.00	91.93	41.71
10	0	-	-	-	-	-	-
11	0	-	-	-	-	-	-
12	3	63.62	6.93	13.26	1.45	380.17	78.44
13	1	62.00	-	-	-	-	-
14	0	-	-	-	-	-	-
15	1	42.09	-	-	-	666.38	-
16	2	67.23	1.82	27.27	-	429.67	29.67
17	1	65.18	-	-	-	266.54	-
18	3	55.24	6.59	4.67	3.32	244.06	62.42
19	3	67.03	10.28	20.82	27.84	397.88	69.84
20	1	46.81	-	-	-	299.33	-

Table 3-9: Summary results from the analysis of 20 different patients' videos (5/5). Features of blink dynamics.

Video number	α	std(α)	b	std(b)
1	-1.73	0.02	-1.22	0.02
2	-1.76	0.01	-1.48	0.01
3	-1.90	0.02	-3.44	0.02
4	-1.72	0.01	-1.33	0.01
5	-2.00	0.02	-1.25	0.01
6	-1.41	0.02	-2.21	0.01
7	-1.75	0.02	-1.07	0.02
8	-1.71	0.02	-1.10	0.01
9	-2.56	0.01	-1.22	0.01
10	-2.73	0.01	-0.15	0.01
11	-1.66	0.01	-0.59	0.02
12	-2.09	0.02	-1.53	0.01
13	-1.66	0.02	-1.94	0.01
14	-1.95	0.02	-1.19	0.01
15	-1.73	0.02	-0.86	0.02
16	-0.96	0.02	-1.05	0.01
17	-2.10	0.02	-1.15	0.01
18	-2.76	0.01	-0.31	0.01
19	-1.68	0.02	-1.22	0.02
20	-1.89	0.02	-1.15	0.02

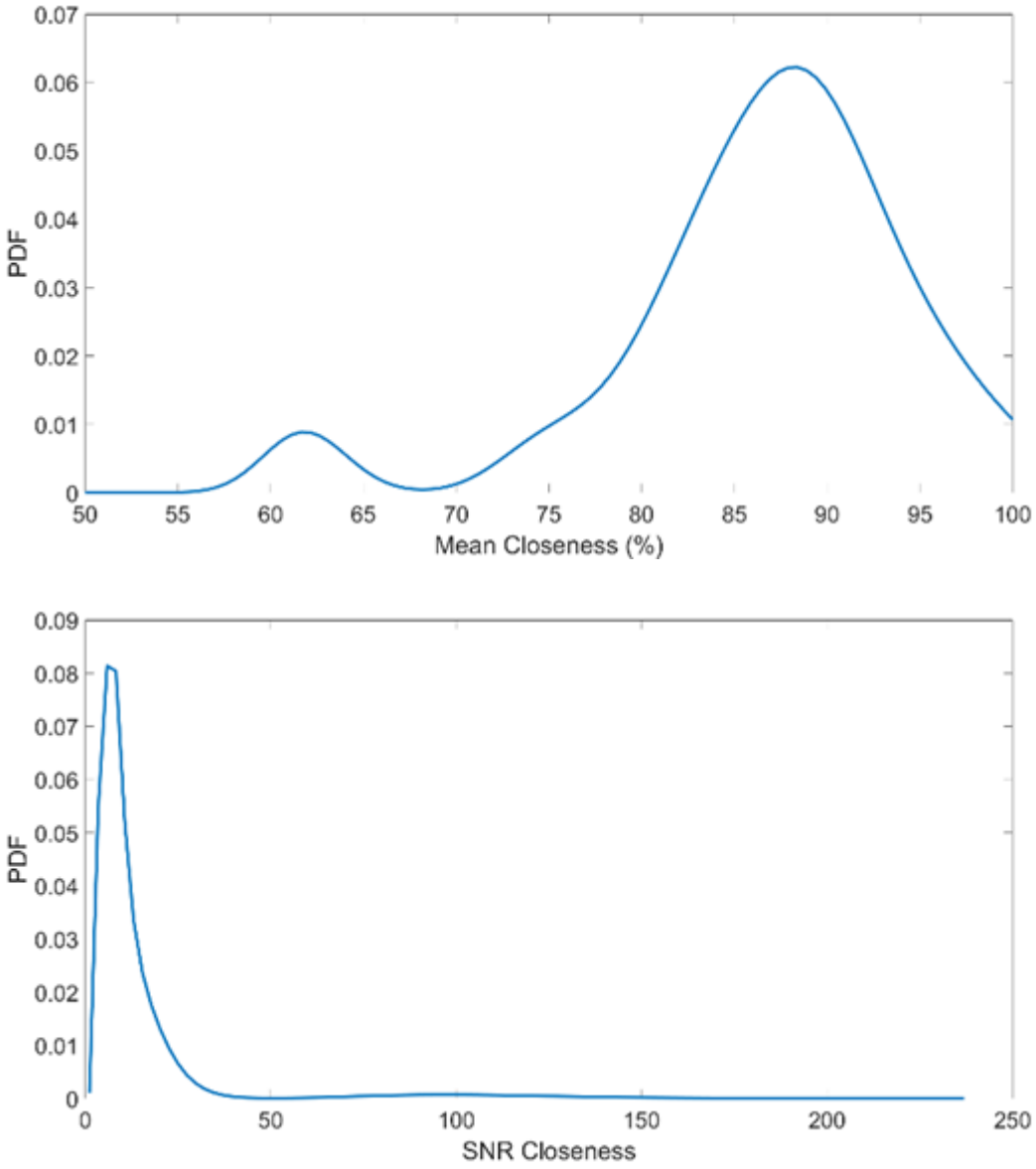

Figure 3-51: Probability density function (PDF) of the mean closeness values and the distribution of its SNR.

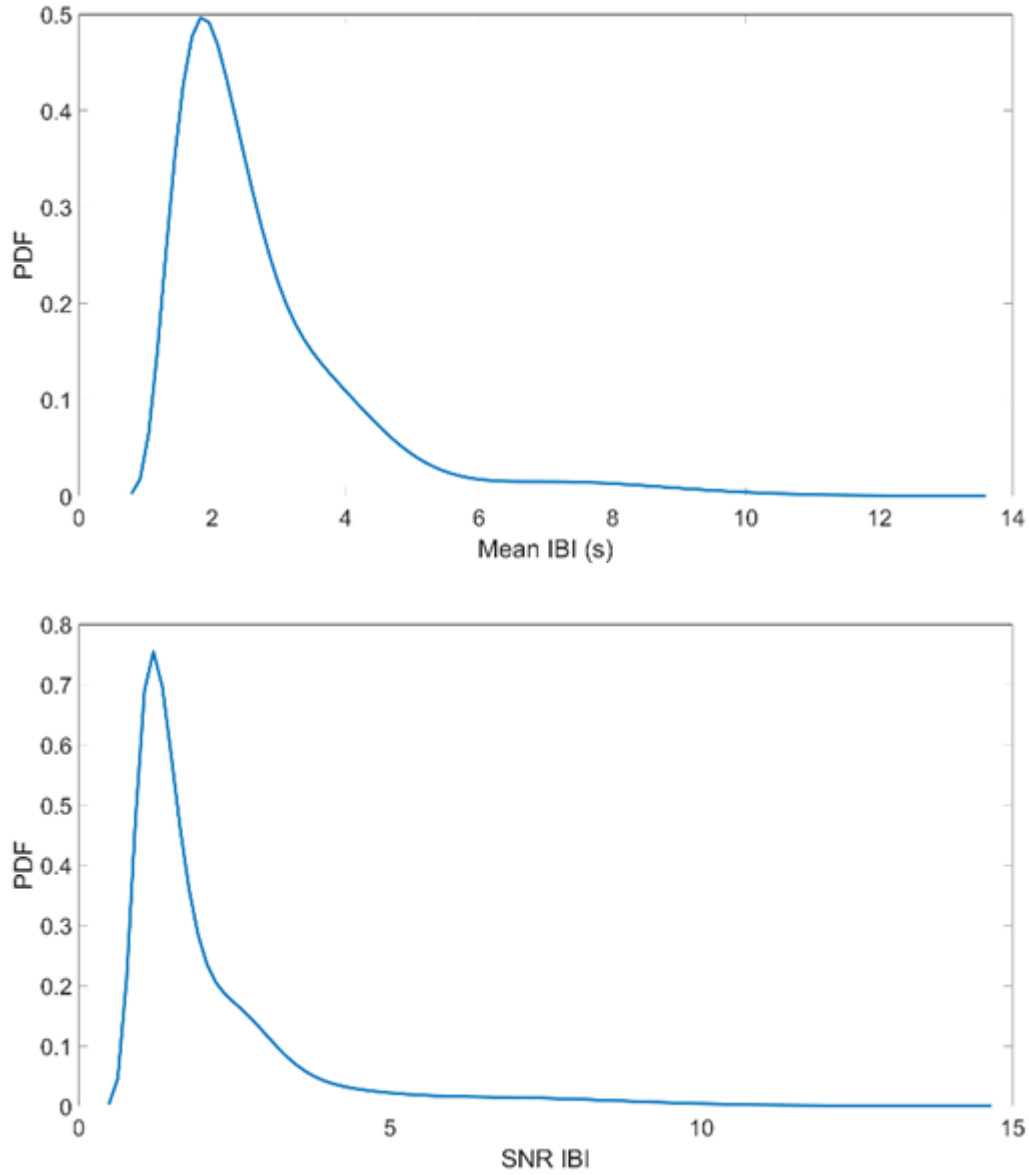

Figure 3-52: Probability density function (PDF) of the mean IBI values and the distribution of its SNR.

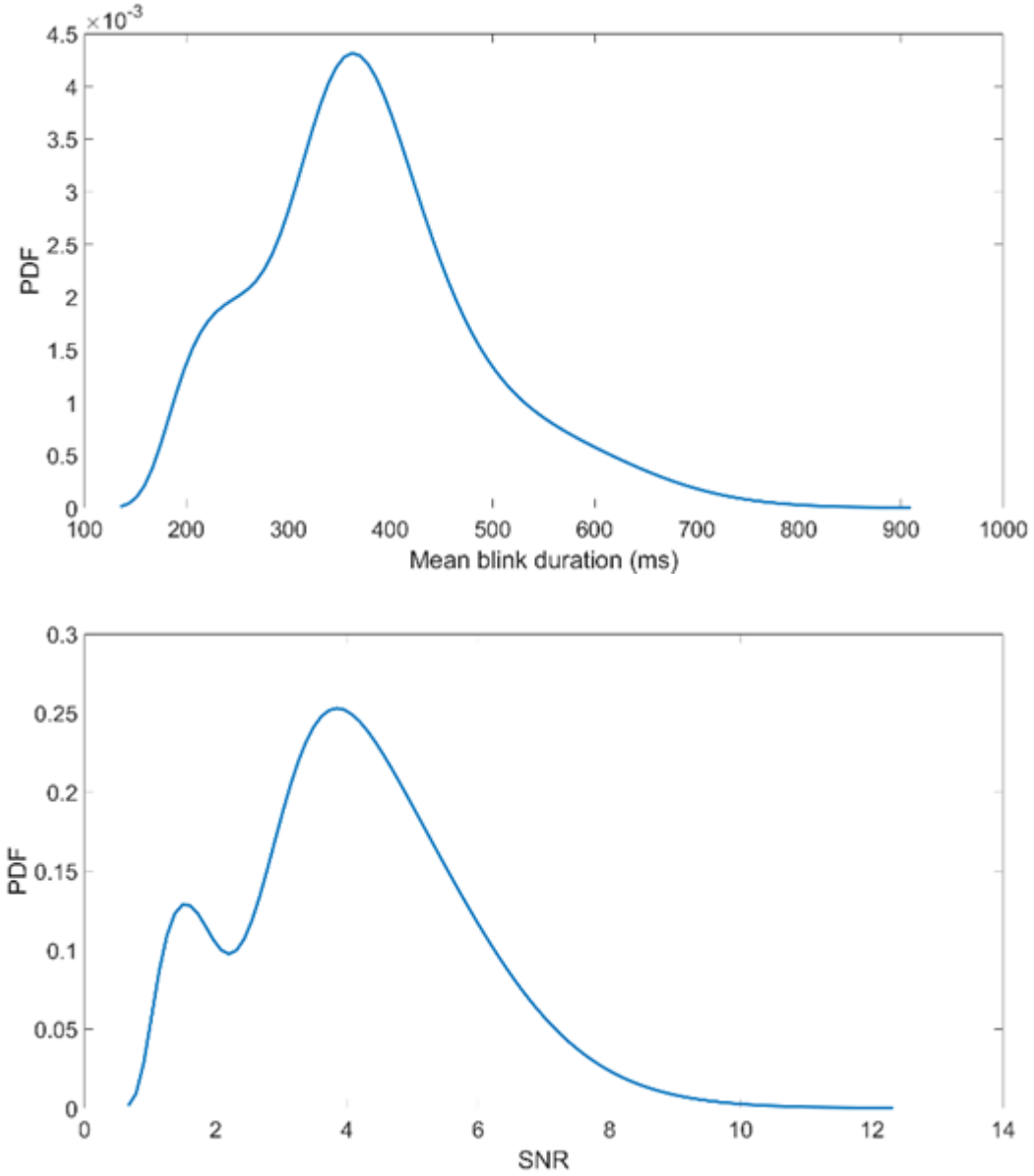

Figure 3-53: Probability density function (PDF) of the mean blink duration values and the distribution of its SNR.

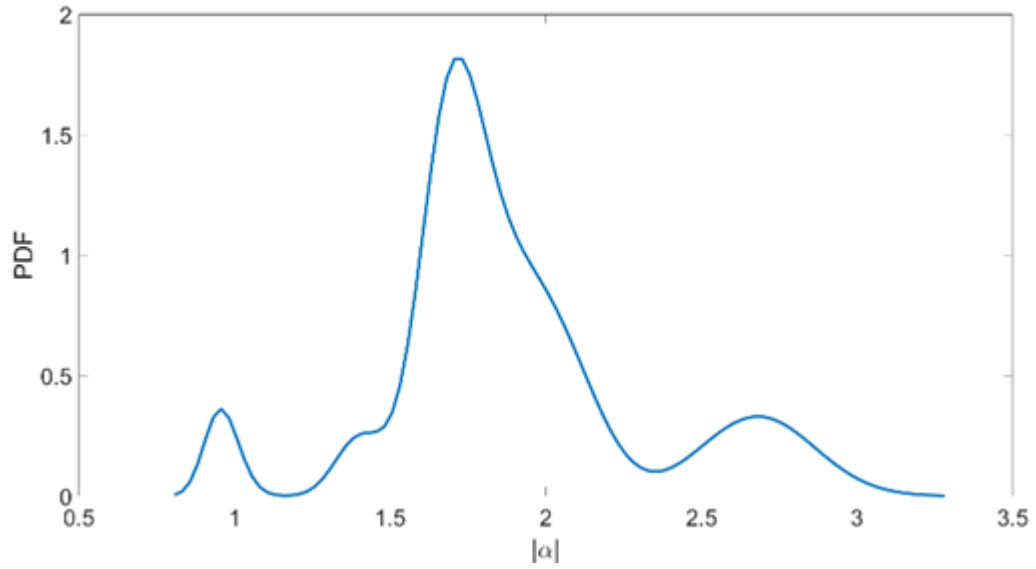

Figure 3-54: Probability density function (PDF) for |α| values.

3.1.4. Sclera segmentation algorithm.

The sclera segmentation algorithm takes the image of the eye as input. It is important to note that the input image is obtained using the Viola-Jones algorithm, which has been trained using a cascade object detector to detect faces and eyes. The Haar type feature has been used in these pre-trained classifiers offered by the Matlab computer vision Toolbox. The image output of the eye is obtained by returning a bounding box where both eyes are available first, as a matrix with four elements specifying the upper-left corner and size of the bounding box where the eyes are available in the original face image. This bounding box is used as an input image, where the number of skin pixels around the eye is high, and to extract the features needed, as in most of the samples, the color intensity between the sclera region and the skin one differs slightly. The color segmentation was not successful to be a method for the segmentation of the features of the input eye image, to obtain an automatic algorithm that can be used with any eye image.

In the first place, when the image is loaded, a reduction of the initial bounding box size is done to eliminate some region where for sure, the only pixels available belongs to the skin. If a different is used and the eye is cropped tightly, for example the Dlib Library (162), this step is not needed. Then, we proceed to turn into grayscale furthermore to a denoising of the image and to an equalization of the intensity considering that for each video of the database the light source used will be different and the incidence angle will vary depending on the relative position of user, camera and light source. A function is used to obtain an equalization of the

light intensity among the eye image frames. *Media_local_polinomica* is used to fit the intensity values first of each one of the pixels belonging to a specific column to a polynomial curve, in our case to a third degree. When the polynomial curves are obtained for each row, with the new values of intensity corresponding to these new curves, a fit to another third-degree curve is used but this time to fit each one of the columns. The result is a third-degree surface that shows the fluctuation of the light intensity across the image. If the mean value of this surface is obtained and used instead of the fluctuation of calculated using polynomial fitting, an equalization of the light intensity across the eye image is performed. In addition, a median filter using *medfilt2* function is used to reduce the noise using a default window size of 3 by 3.

Figure 3-55: Original eye image, obtained cropping the left eye from the output of the vision.CascadeObjectDetector

Figure 3-56: Cropped grayscale eye image to take out some skin region.

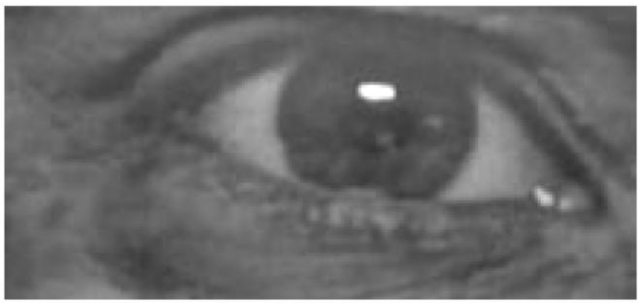

Figure 3-57: Intensity equalization using media_local_polinomica function

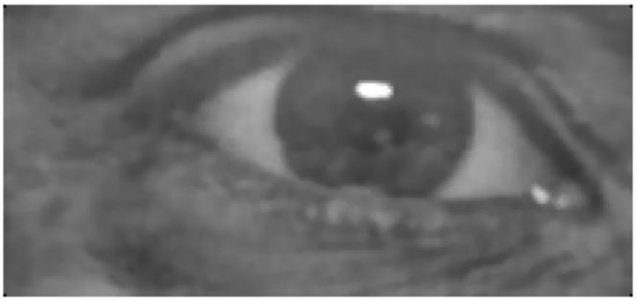

Figure 3-58: Median filter application with window size 3 by 3 pixels over equalized intensity cropped eye image.

Once the image is obtained the central zone is analyzed where the iris reference column will be found. This is done by cropping the first quarter width size and the last one and obtaining the signal of the median value of each one of the remaining columns.

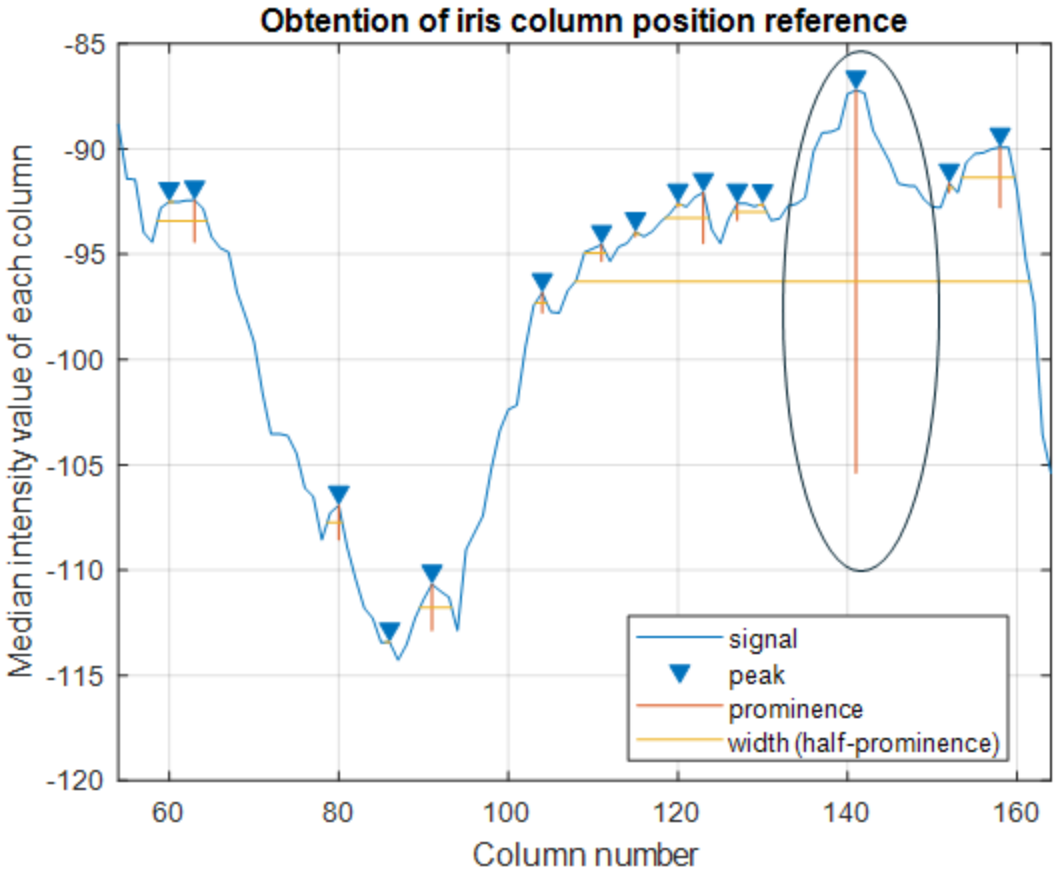

Figure 3-59: Median value of the intensity of each one of the central columns. The graph result is inverted to obtain the peak with the lowest value where we identify the iris/pupil region (columns 108-162).

The peak position will be se as the reference column of the iris/pupil and will be used to analyze each side of the image left and right separately to obtain the temporal and nasal sclera regions from the cropped eye image. This is done first with the temporal sclera obtaining the signal of median values of the columns of the left side of the image considering the last analysis column

the reference of the iris/pupil (column 141 in Figure 3-59). The temporal sclera region is identified as a peak using the function *findpeaks* of the Matlab environment and considering a minimum peak width (at half prominence) of 10% the size of the left side of the image and a minimum peak prominence of one standard deviation of the median signal. The last peak identified is considered the one representing the sclera region as is the one nearest to the iris/pupil region as the last column is the iris column reference. The results are shown in Figure 3-60.

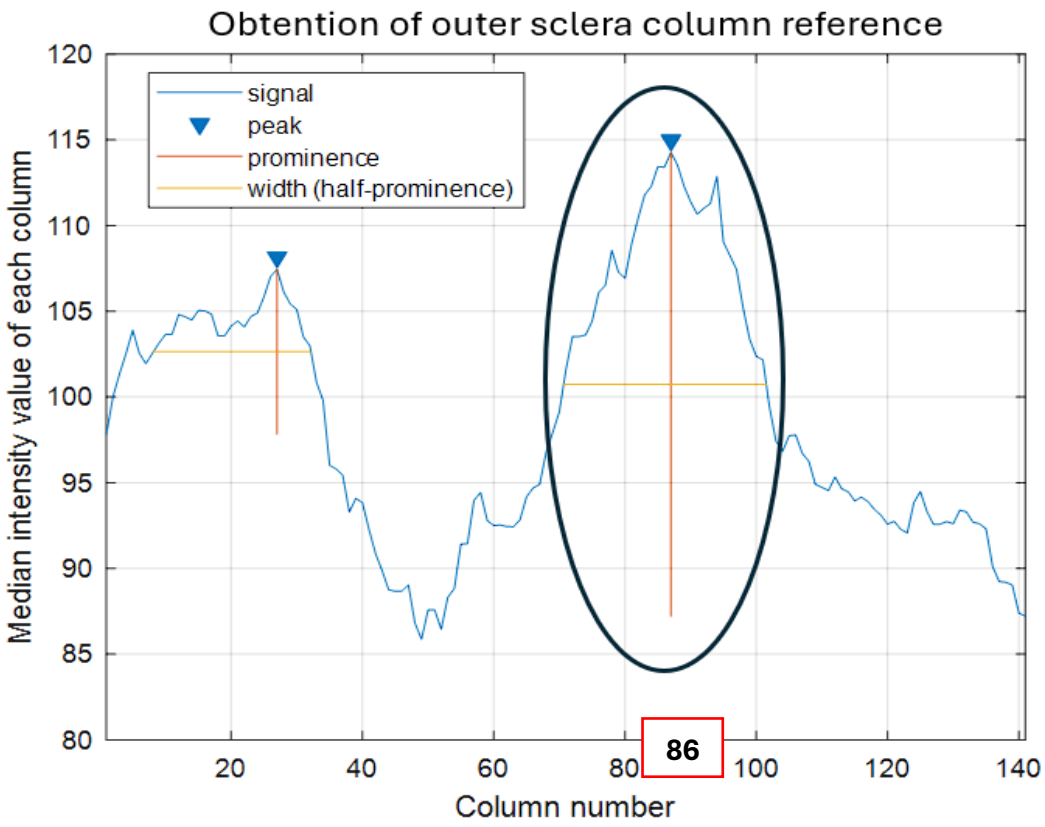

Figure 3-60: Signal with median values of each column of the left side of the image where the temporal sclera region is available.

The location of the peak (column 87) is stored in the variable *pos_os* (reference column for temporal sclera region) and will be the reference of the area of the temporal sclera. The halfwidth segment ends that are displayed in Figure 3-60 represents the beginning and end of the temporal sclera region for analysis purposes. These values are obtained by modifying the *findpeaks* Matlab function to extract these values (as an output, considering that they are calculated by linear interpolation (187).

To obtain the corner value of the temporal sclera, as this position will be darker than the clear area, it is wise to see that it will be the base of the peak of the temporal sclera region. From experience, analyzing different images of the open eye, the temporal sclera column border corresponds at the base of the peak. This value is obtained using the most conservative method of the two options. The first option is to use the peak's prominence to find the first base end. The second one is taking the minimum value considering 2 times the distance between the end at half width and the location of the peak. The option is chosen so that the distance to the peak is minimal to always consider the border as a column of sclera and to a skin. From the example shown in Figure 3-60, the column border is chosen as column 56, corresponding this time to the minimum value considering the signal before the peak with 2 times the left half width (~16).

Once the column border is obtained, we analyze the nearest columns of the temporal sclera to the iris region to find the border of the sclera with the eyelid skin to segment the temporal sclera. The most significant column of the temporal sclera is first analyzed because it will be the one with the most pixels of the sclera, and we can restrict the posterior column analysis. To perform the analysis, a variable is stored with the information of the column pixel intensities of the median filtered eye image; see Figure 3-58. Using the self-modified *findpeaks_2* Matlab function, we obtain the location of the highest peak that will correspond to the reference row of the sclera pixels. The half-width ends will be the row where a change from sclera to skin can be noticed. As the image is still a little noisy, this method is better than calculating the maximum gradient to find the exact pixel where a huge change in the intensity is produced.

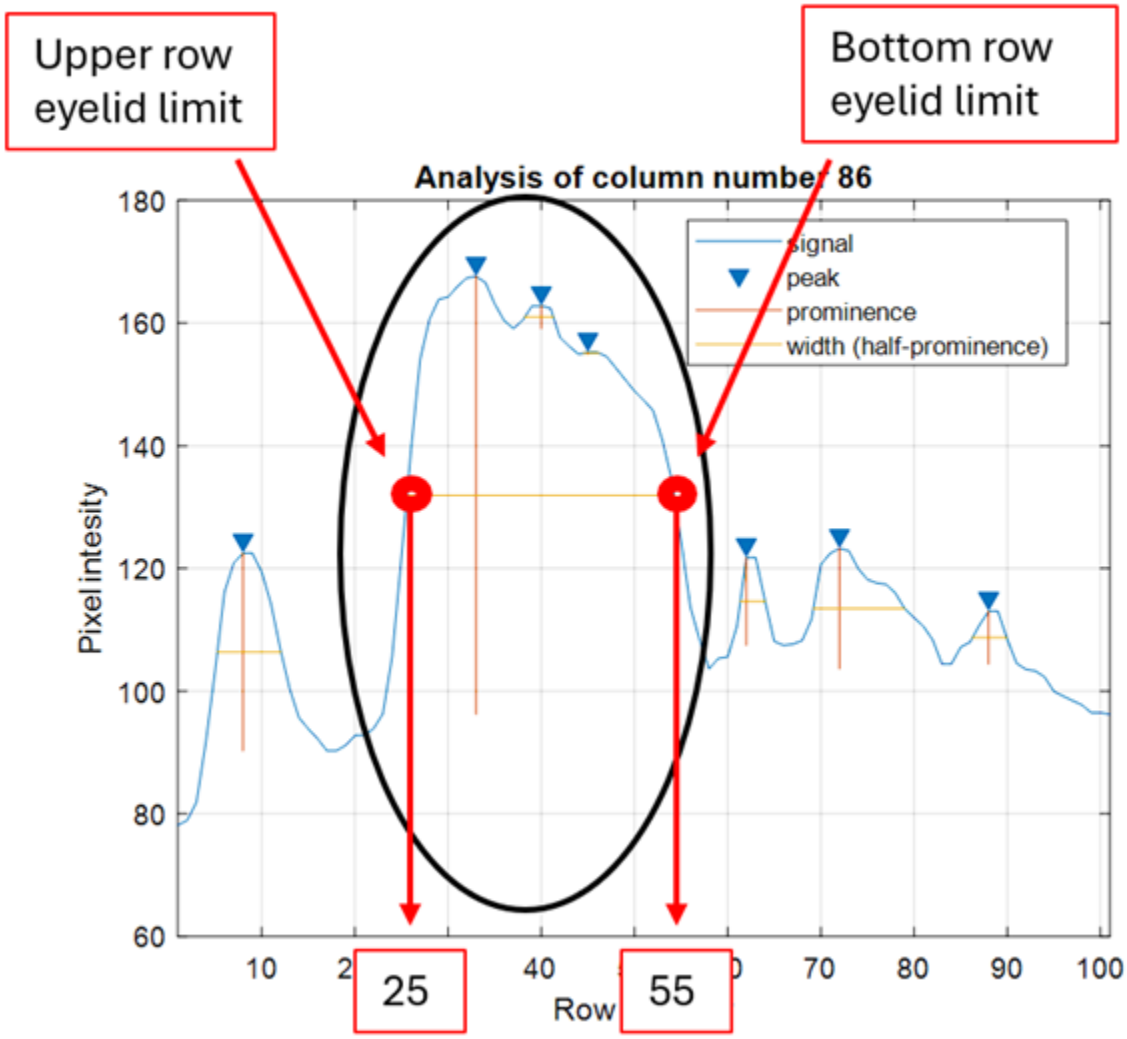

Figure 3-61:Example of analysis of the reference temporal sclera column to obtain the row of the upper eyelid (left end at the halfwidth segment) and the row limit of the lower eyelid (right end of the halfwidth segment).

Furthermore, each column will be analyzed in the same way using a for counter, from the reference column to half the distance to the column of the corner border. This area is selected considering the rows limit obtained for the reference column of the temporal sclera (column 86, see Figure 3-61). This way the restriction of the obtention of the peak where the sclera is available cannot be confused with a light skin that is reflecting high amount of light intensity. The row limits for the columns analyzed are obtained considering the location of the peak in the signal of the Figure 3-61 with a distance of two times the left side of the width at half prominence and the half of the total of the width at half prominence starting from right end, see Figure 3-61. In our case, the next figures of pixel intensity signals will start at row 18 and will end at row 69.

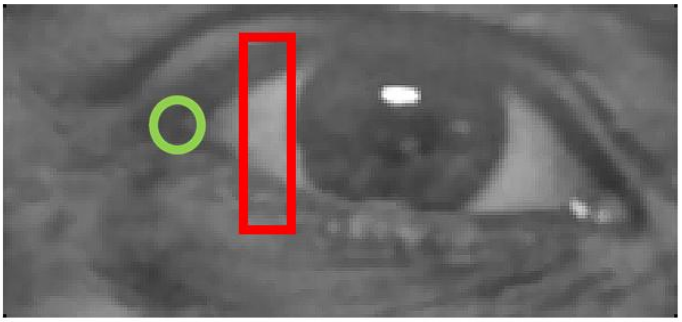

Figure 3-62: In red the columns to be analyzed, in green the location of the corner border.

The counter will start with the column reference of the temporal sclera until reaching the half distance to the left corner of the eye obtained previously. The further we are from the reference temporal sclera column and the closer we are to the corner, the lower the height of the sclera is. This can be observed in the Figure 3-62 and the results comparison can be seen in between two different analyzed columns in Figure 3-61 and Figure 3-63.

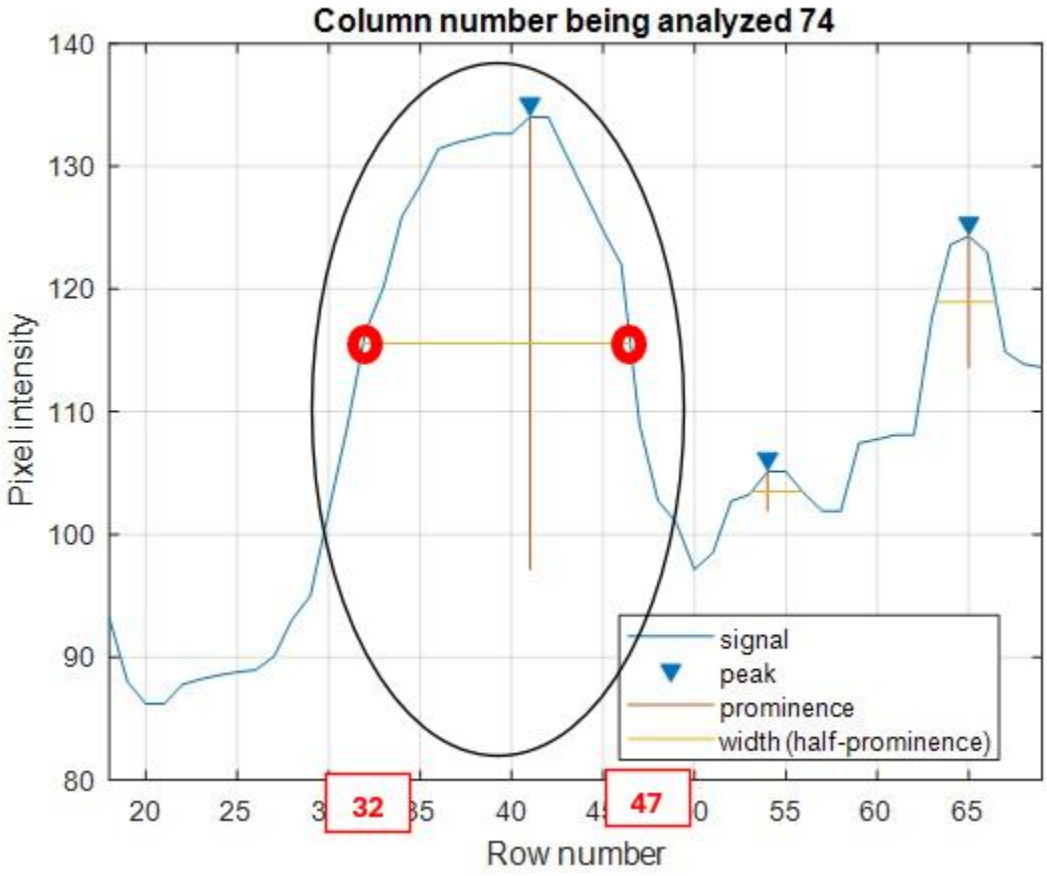

Figure 3-63: Column analysis number 74. The upper and lower eyelid row limit are shown in red. The peak in black corresponds to the pixels considered as belonging to the sclera.

The results of the column and row limits are stored in a matrix to further fitting to a second-degree polynomial curve as the palpebral cleft if divided in 4 areas, considering upper and lower zone over the external or nasal sclera in the eye image, see Figure 3-64.

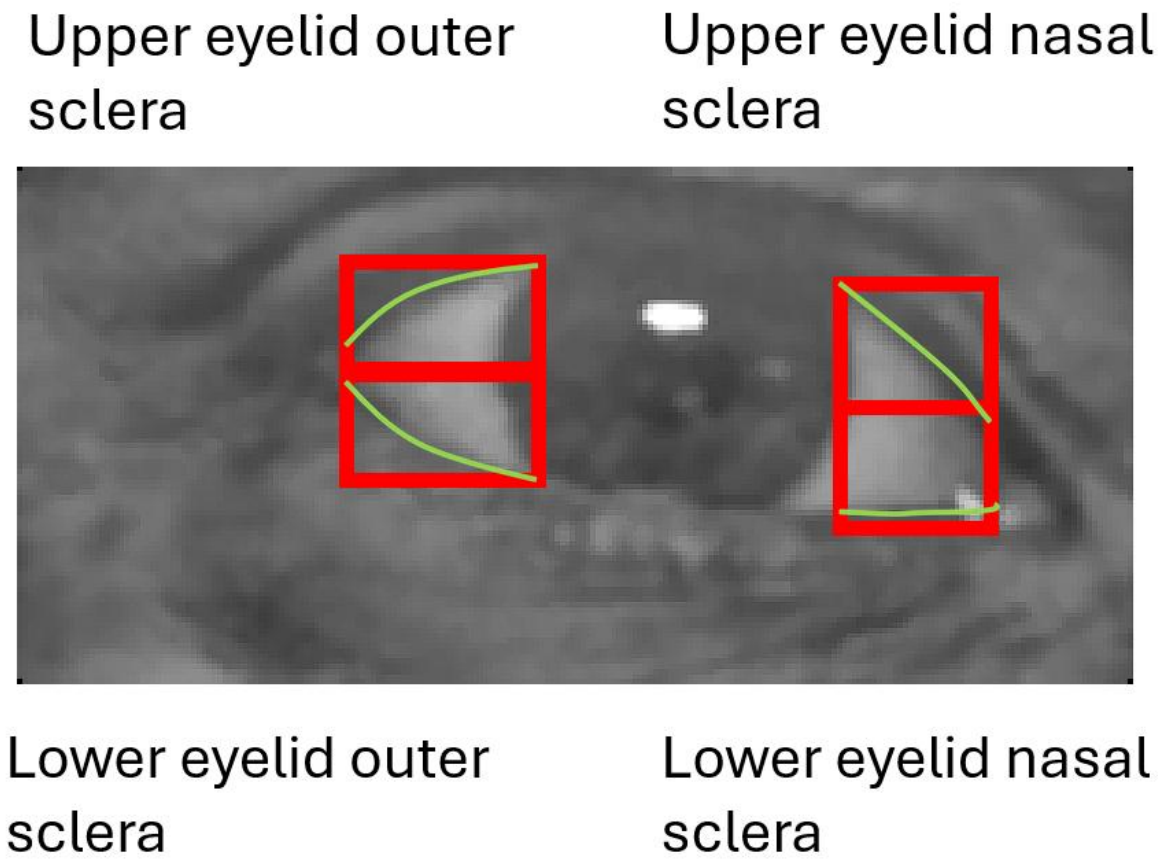

Figure 3-64: Criteria chosen to differentiate each curve of the palpebral cleft.

To proceed with the fitting of the temporal sclera limit curves, first we identify the concavity of the curve to be fitting, for example for the upper eyelid the fitting of the curve is limited to a negative first coefficient and a positive second coefficient to obtain the desired shape. Additionally, an exclusion criterion is used to not consider the wrong limits when the borders are not that clear due to illumination of the considered frame. The exclusion criteria for the temporal sclera curves are obtained from the reference temporal sclera column and for the nasal sclera, the reference nasal sclera column is used. The row limits can be seen in red in Figure 3-64. The upper exclusion limit for the upper eyelid curves is obtained from each one of the reference temporal and nasal column as the upper boundary, in the temporal sclera example in Figure 3-61, the upper boundary is row 25. The bottom exclusion limit is obtained as well from the bottom eyelid limits from the nasal and temporal sclera reference column respectively, in the temporal sclera example in Figure 3-61, the bottom boundary is row 55. The lower

exclusion limit for the upper eyelid is the mean value between the upper limit for the upper eyelid and the bottom limit for the lower eyelid for each one of sclera zones, nasal and temporal separately. The upper exclusion limit for the bottom eyelid is the mean value between the upper limit for the upper eyelid and the bottom limit for the lower eyelid again for each one of sclera zones, nasal and temporal separately. For the temporal sclera example in Figure 3-61, the mean value that is used as the bottom limit row value as a criterion of exclusion of the upper eyelid temporal sclera region and used as the upper limit row value as a criterion of exclusion of the lower eyelid temporal sclera region is row 40.

The fitting of the curves of the eyelid is performed using the fit function in Matlab. The fitting curve app has been used to generate the code to select the correct options for the fitting, include the exclusion criteria and boundaries for the second-degree coefficient so the curves for the upper eyelid should be concave and convex for the bottom eyelid.

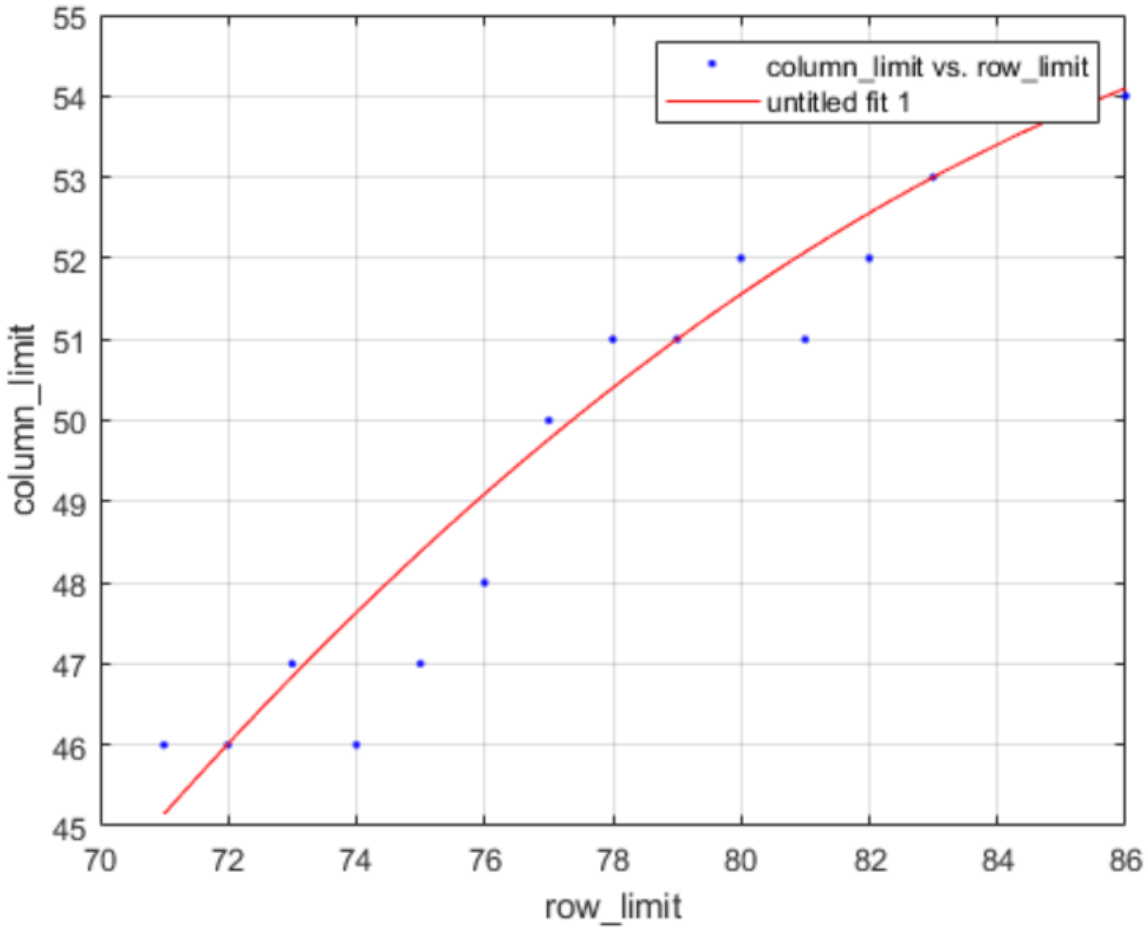

Figure 3-65: Fitting curve of the bottom eye lid limit considering the temporal sclera region.

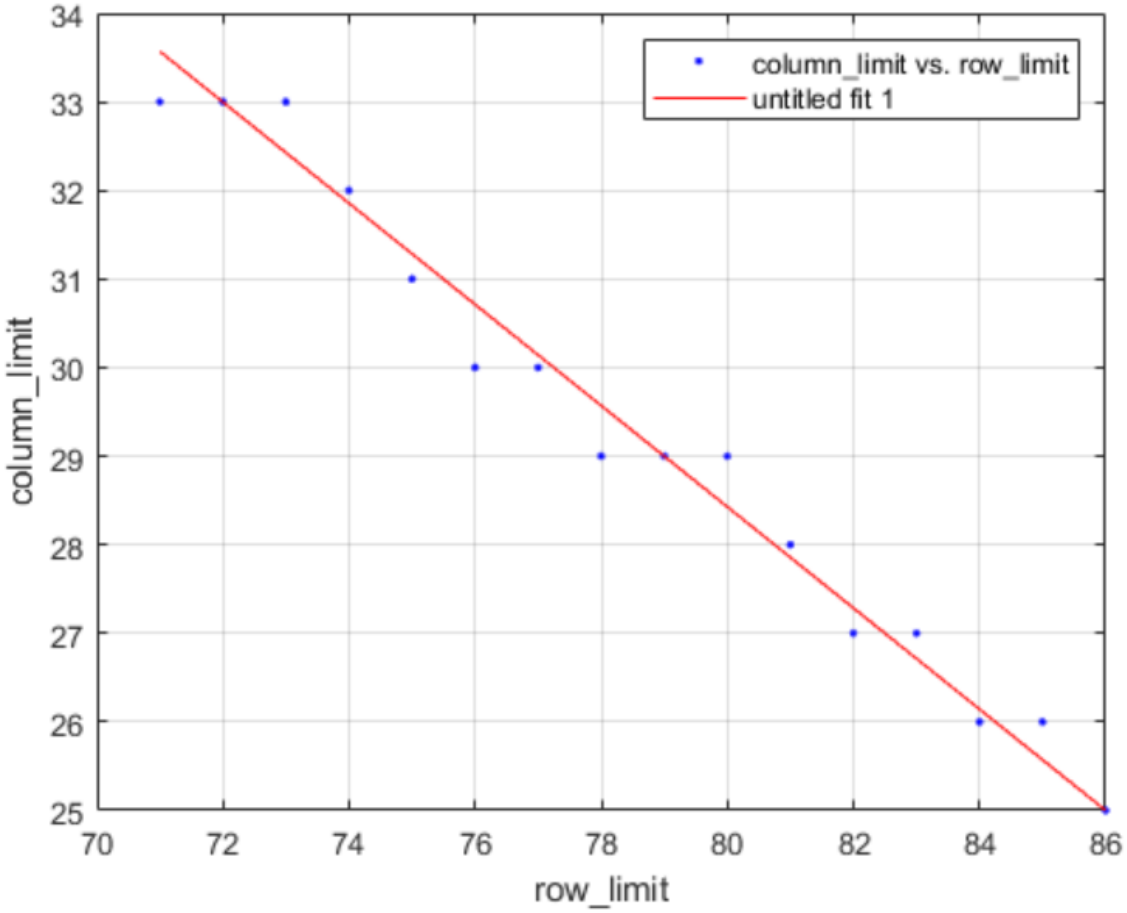

Figure 3-66: Fitting curve of the upper eye lid limit considering the temporal sclera region.

To connect the upper eyelid with the lower eyelid curves, an arc circumference generated using three points is generated. The three reference points are the last point of the upper and lower eyelid curve and the corner of the temporal sclera considering the mean row between the last point of the upper and bottom eyelid curves. If the radius of the arc is higher than the distance between the last column analyzed (last point of the curves of upper and lower eyelid of the temporal sclera region) the corner of the sclera is identified closer to the end of the curves of the eyelids. The distances are identified by experience to include conservatively only sclera pixels and obtain a smoother variation between the curves of the upper and lower eyelids to corner of the palpebral cleft.

Once the eyelids curves for the temporal sclera region are obtained and connected near the corner of the palpebral cleft, the same process is done for the obtention of the curves of the nasal sclera region. A unique difference in between the process of the nasal sclera region and the temporal sclera one is that for the calculation of median values of each column of the right side of the image where the nasal sclera is available, as all the eyes are more or less have a zero

degree roll angle rotation, the median values will be calculating considering the row limit values obtained in Figure 3-61 considering an increment of 15% of the size height of the eye frame in the bottom and upper limits. This is done to be conservative and consider small roll rotation angles in some of the cases and movements of the iris where sometimes the nasal region will have more height than the exterior sclera region.

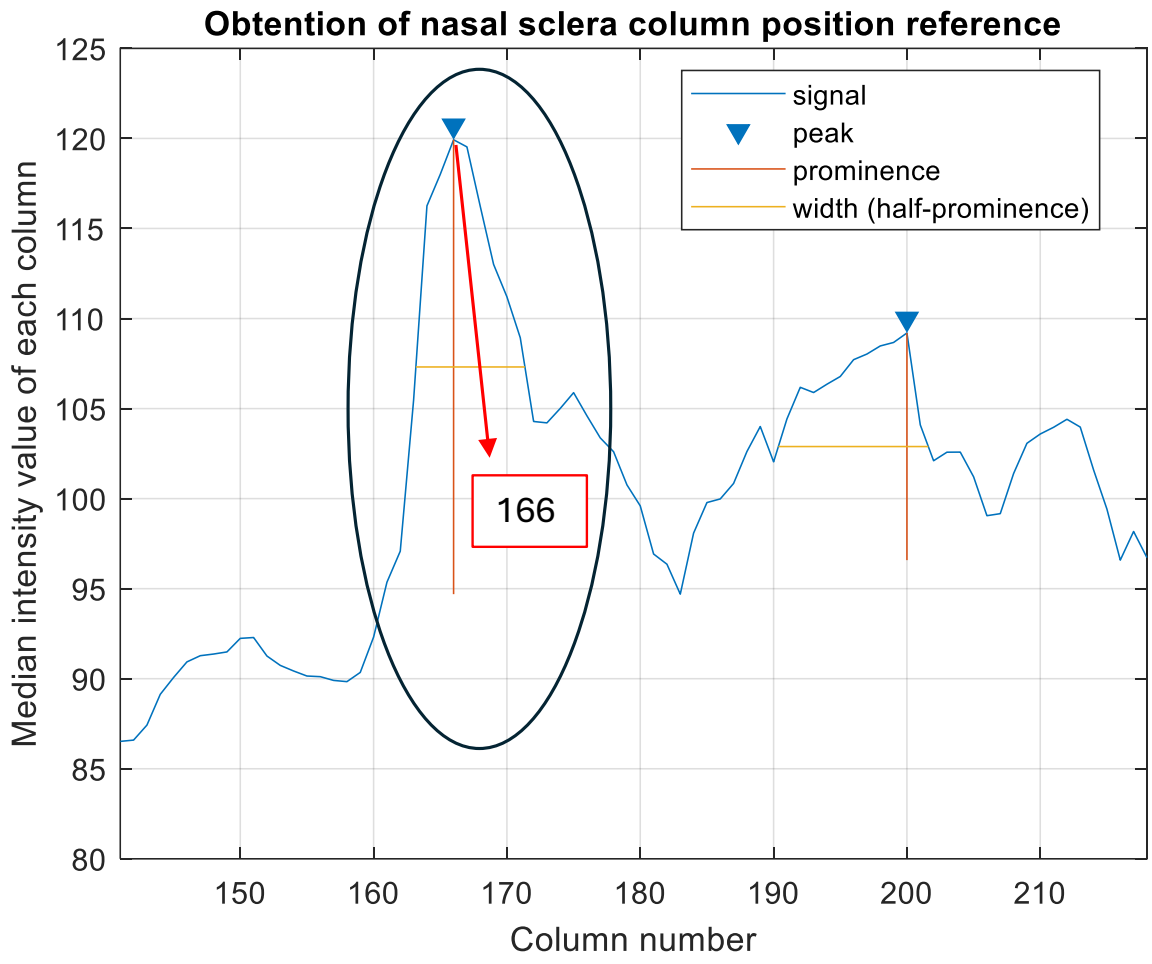

Figure 3-67: Location of the reference column of the nasal region sclera.

Once the column reference is localized, we proceed with the detection of the upper and lowest eyelid border limit of the nasal sclera region. The pixel intensities of the column 166 can be seen in the Figure 3-68, and the code development in section 8.3 where the script of the sclera segmentation algorithm is present.

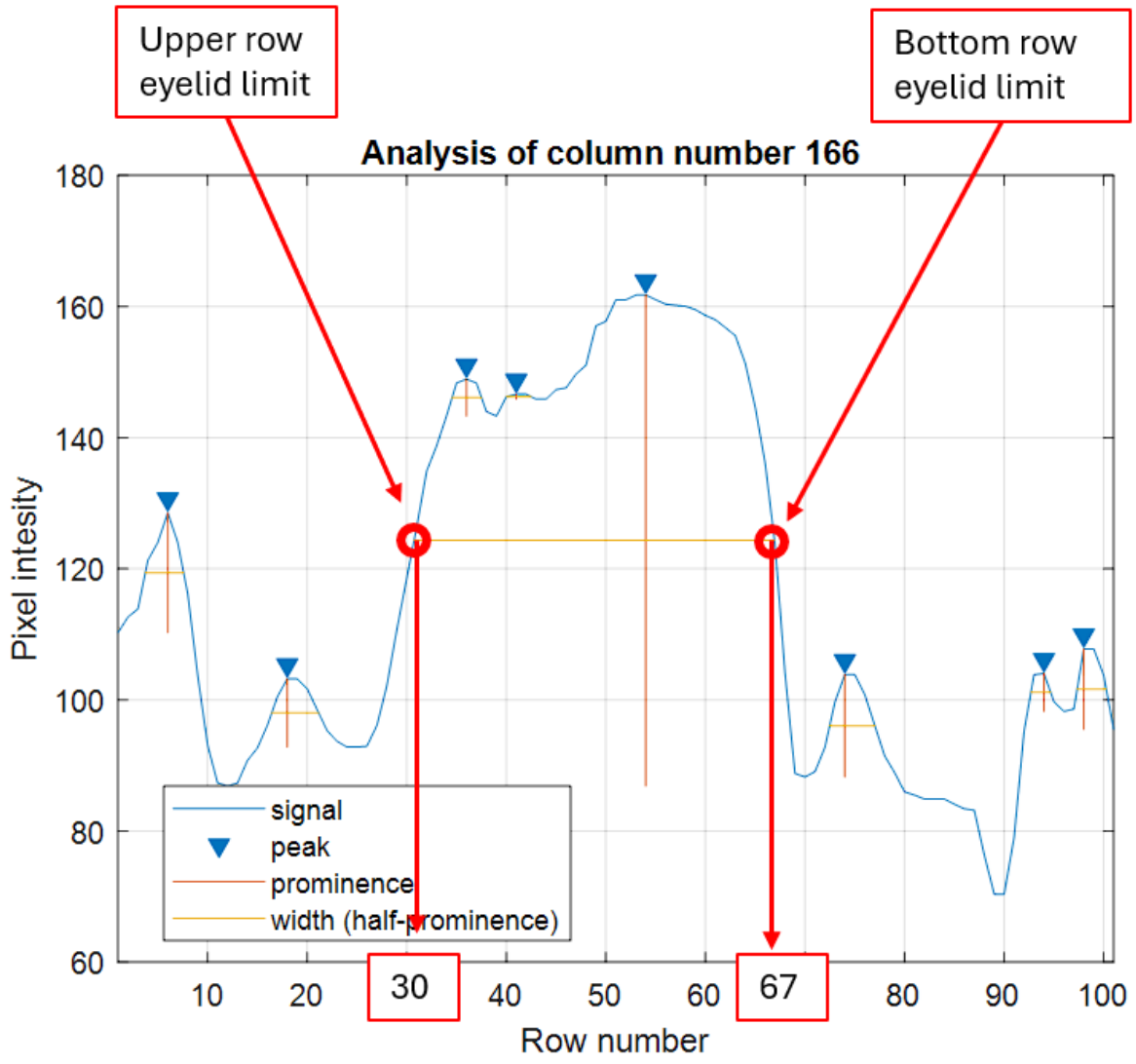

Figure 3-68: Obtention of the upper and bottom limit of the nasal sclera region column reference. These same values will be used as exclusion criteria for eyelid curve fitting.

The same methodology used for the temporal sclera region is again performed for the nasal sclera region. The obtention of the nasal sclera boundaries is performed using the for counter. Additional information of the code developed can be found in section 8.3 where the Sclera Segmentation Algorithm Script is available for consultation.

Furthermore, a fitting curve for the upper and bottom eyelid is performed to obtain the curve representing the palpebral cleft of the nasal sclera region. This is performed using the function fit and the generated code from the curve fitting Matlab App.

To connect the upper eyelid with the lower eyelid curves, an arc circumference using three points is generated. The three reference points are the last point of the upper and lower eyelid

curve and the corner of the temporal sclera considering the mean row between the last point of the upper and bottom eyelid curves. The same logic used for the temporal sclera region is reproduced for the nasal region.

In the following part of the program, we will represent in the black and white image with the same size as the cropped eye frame, the upper and bottom eyelid limits of both sclera regions.

The next important step is to find the limit of the sclera region border with the iris. This is performed using the gradient of the curve of the rows from the column reference of the temporal or nasal sclera to the mean column between the reference columns of the temporal and nasal sclera. The rows to be analyzed must be selected considering only the ones where the sclera and iris pixels are available. The first row to be analyzed for example, of the limit of the temporal sclera region and the iris is selected considering the reference temporal sclera column (number 86, see Figure 3-60) is the limits obtained in Figure 3-61 and for the nasal sclera region column (number 166, Figure 3-67) limits are obtained in Figure 3-68. As the column borders for each row are the first thing calculated in this step, we proceed to initialize the for counter to analyze each one of the iris border pixels first for the temporal sclera.

The process to obtain the column limit for each row considering the first limits explained deeply in the next figures with examples. First the obtention of a new column limit considering the median value of the signal of each row is done, see Figure 3-69.

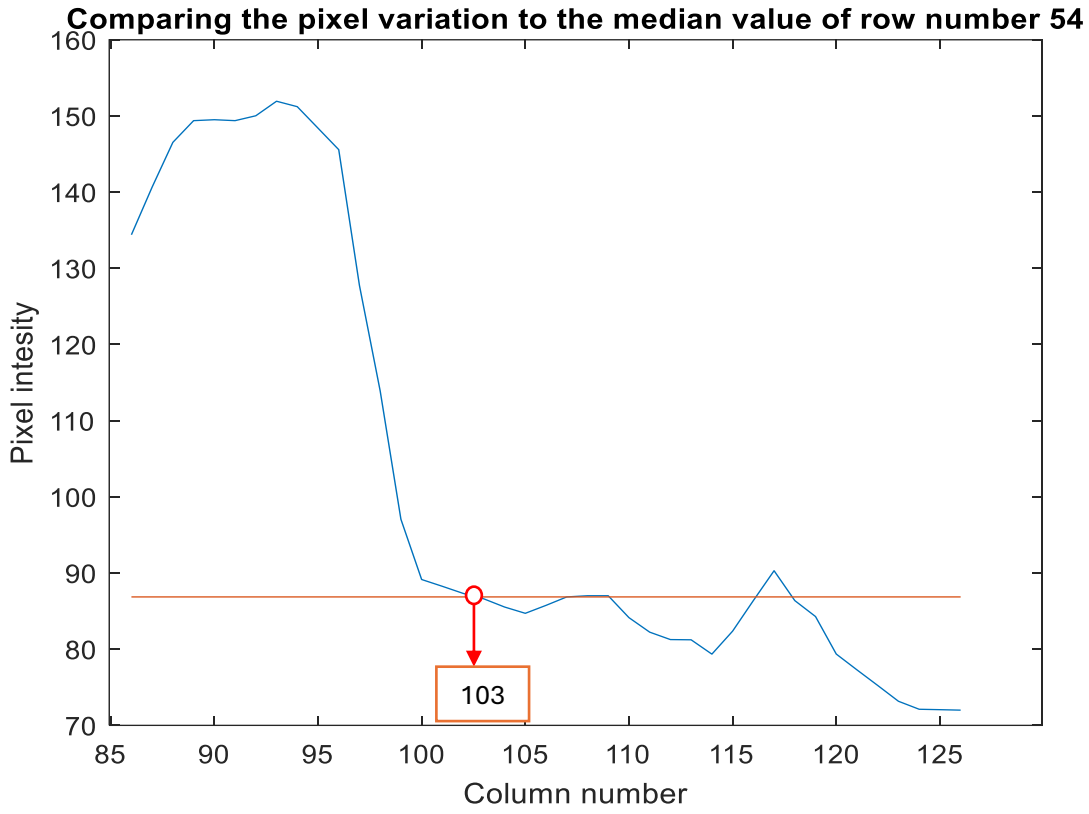

Figure 3-69: Pixel values for row number 54 from the temporal sclera column reference to the mean iris column reference obtained as the mean column between the nasal and the temporal sclera region column references. The value highlighted will be used as the limit column number for the calculation of precise sclera-iris border.

In the Figure 3-69, we can observe that on the left side of the median limit value pixels intensity is higher than the median value corresponding to those pixels with sclera region. To obtain a precise limit border, the gradient of the signal of Figure 3-69 is calculated until reaching column 103 (considered as already an iris pixel). The obtention of the minimum global value of the signal gradient will be the limit of the sclera region as a huge variation in pixels intensities happens as the normal sclera intensity is higher than the border of the iris region, Figure 3-70.

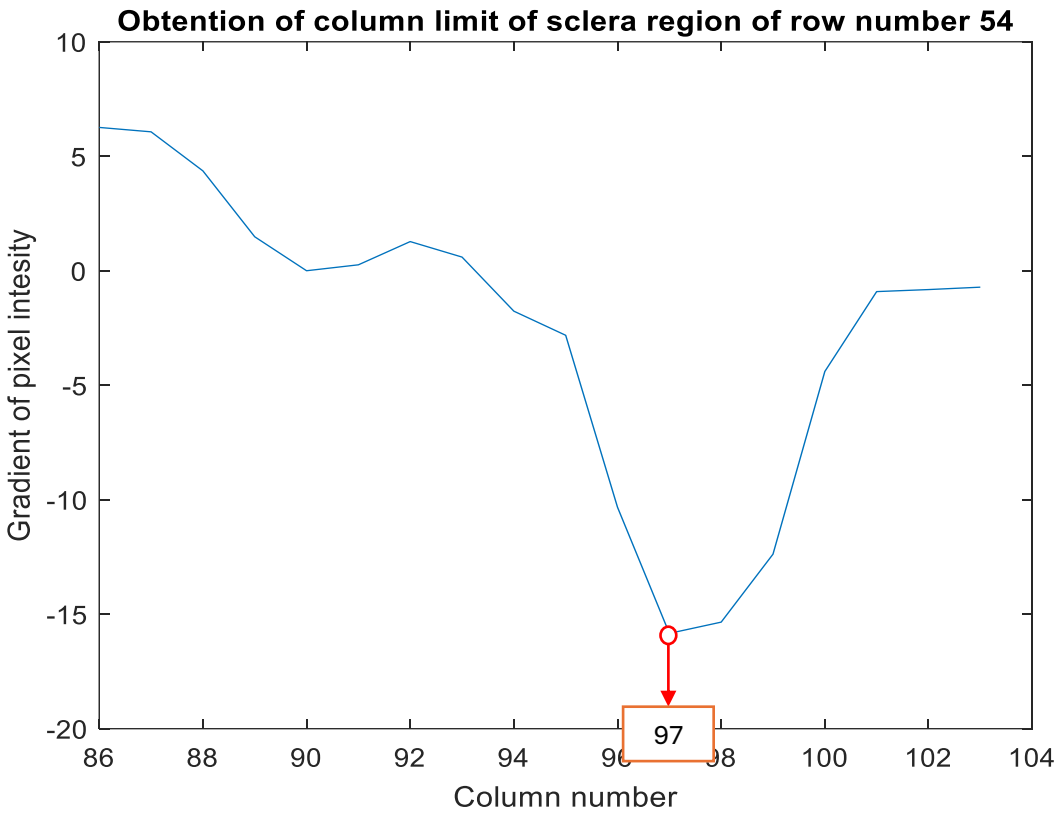

Figure 3-70: Obtention of limit border of sclera region for row number 54. The signal gradient is obtained until the limit calculated in Figure 3-69.

Each one of the rows in between the upper and lower eyelids is analyzed in the same way. The rows and columns of each position pixel is stored in the variable's *column_limit_outer* and *row_limit_outer*. The STOP_VALUES variable stores the information about the pixel value where the signal is minimum. In Figure 3-69 the intensity value is 72 and is obtained in column 125.

When storing all the minimum values for all the columns in between the upper and lower eyelid of one of the sclera regions, we obtain a curve with the variation of this minimum pixel intensity. Considering that the iris/pupil is much darker than the sclera region all the rows with iris pixels will have a very low value of intensity while the rows without iris region will have a much higher minimum signal value. If the gradient of the minimum values of each row signal has a high variation, it means we reached the area where both regions of sclera are connected and probably the column limit obtained is due only to the variation of pixels inside the sclera region, see Figure 3-71.

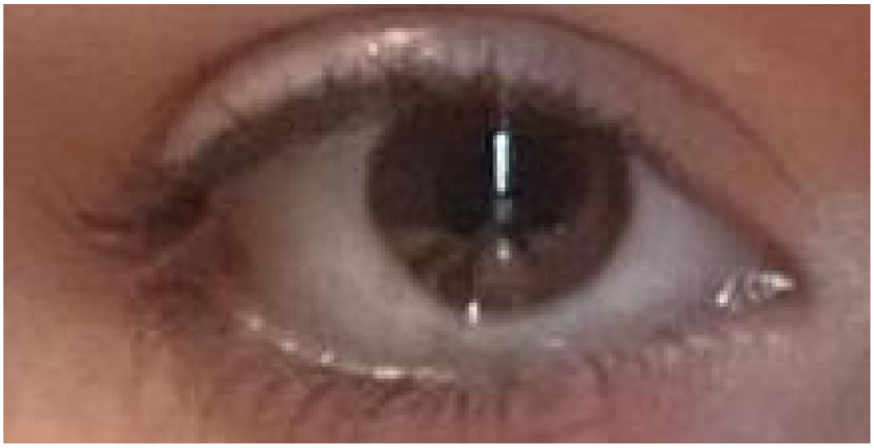

Figure 3-71: example of an open eye where the sclera regions are connected on the bottom.

If we analyze two rows of temporal sclera from the eye image in Figure 3-71, we obtain the following signals considering that one row is in the middle of the eye region and the other on the bottom where there is no pixels belonging to the iris region in that row. The differences between both row signals is the minimum intensity value and this criteria is used to not consider the rows without iris pixels for the obtention of a second-degree polynomial curve see Figure 3-72.

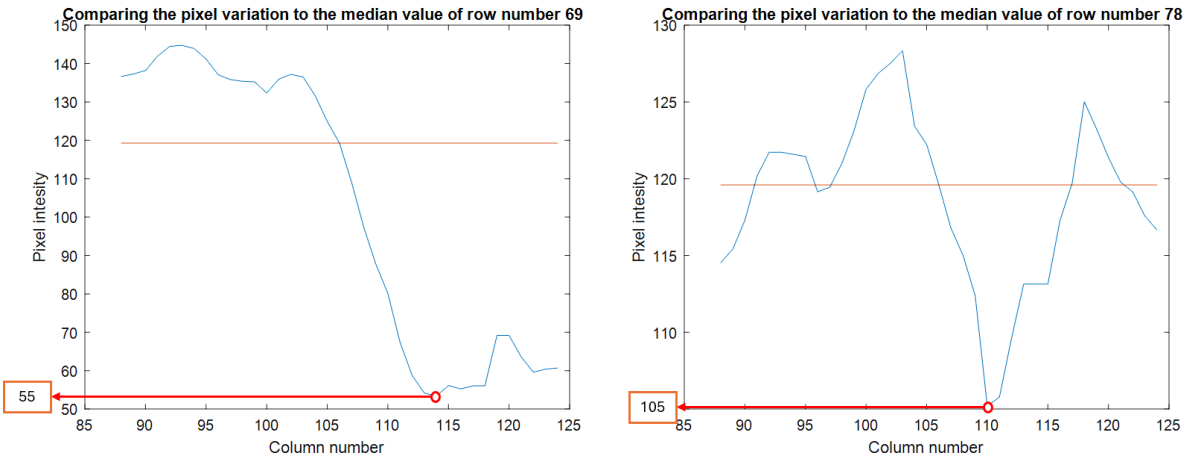

Figure 3-72: In the left figure, the minimum value of the row is 55 corresponding to an iris region pixel, while for the row 78, the minimum intensity is 105 corresponding to a sclera pixel.

Considering all the rows analyzed from upper to bottom eyelid temporal sclera region, considering the columns from the temporal sclera reference column to the mean iris column reference, the minimum value of the row signal is obtained and plotted in Figure 3-73.

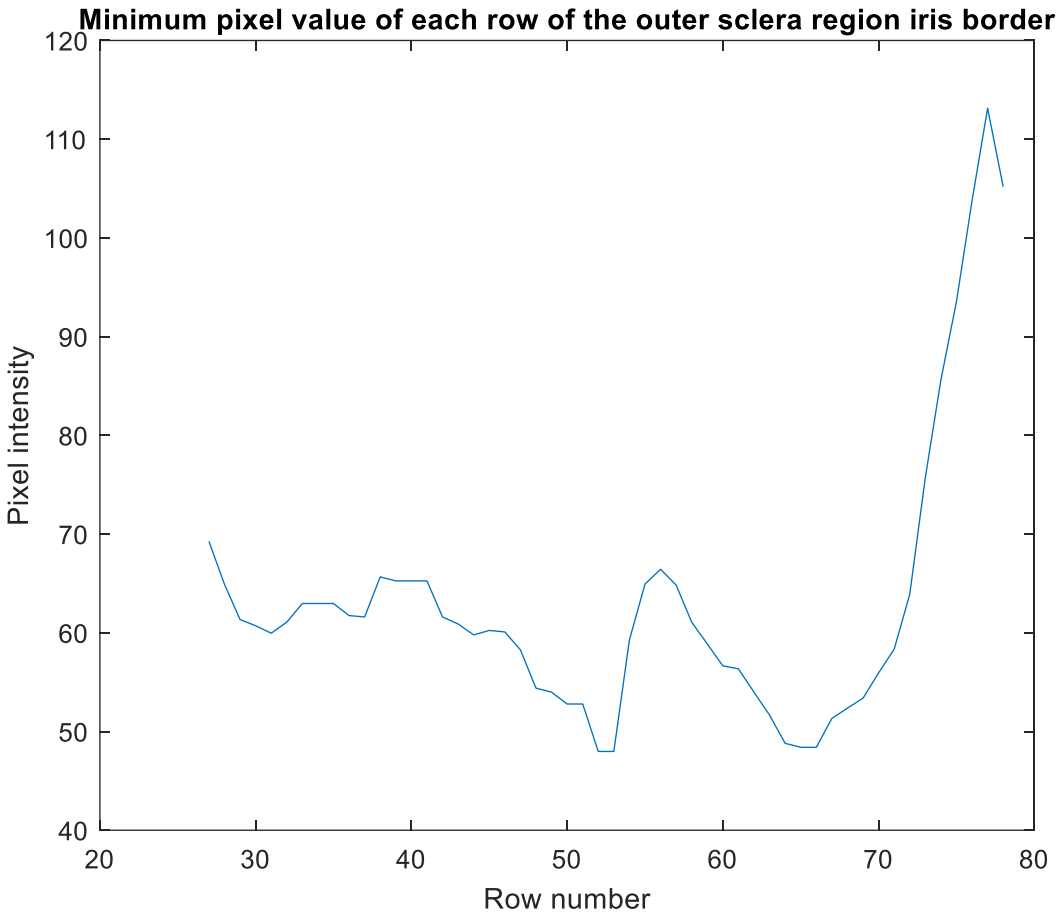

Figure 3-73: Minimum pixel intensity of each row of the temporal sclera iris region border.

The gradient of the signal in Figure 3-73 is used to exclude the pixel border calculated in the loop coded (Annex section 8.3) to eliminate the rows without iris region. The function *findpeaks_2* is used to locate the peak of the gradient that has a prominence above 3 times the standard deviation, where due to the change of the minimum value and a huge increase in the intensity when no sclera is available we can locate the row limit from the images where, as the example of Figure 3-71, we can see clearly a region of sclera below the iris.

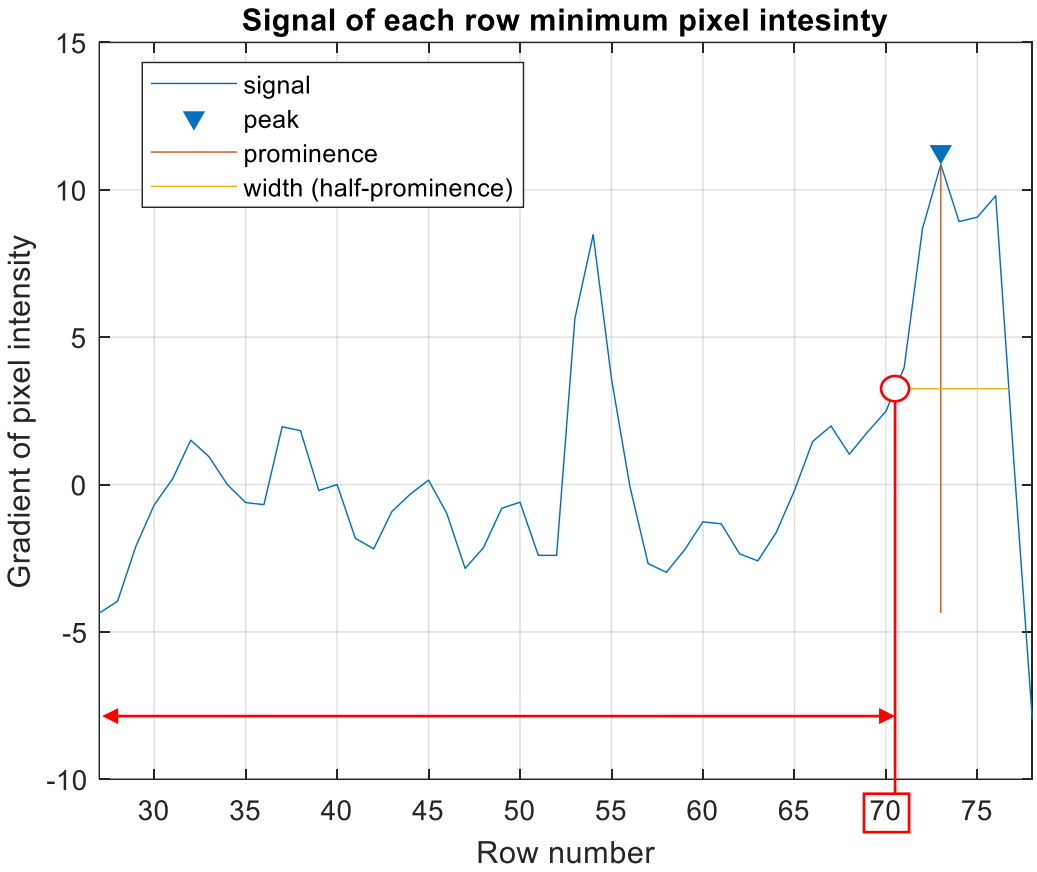

Figure 3-74: Gradient of the signal of minimum pixel intensity of each row of the temporal sclera iris region border.

The code developed to fit the resulting pixel border considering only the rows where iris and sclera region is available in the Annex section 8.3). A second-degree fitting curve is also used to characterize the curve of the iris border.

Once all the limit border curves are obtained for an open eye image, we proceed to create a mask to segment the sclera region for further analysis. The connection between the eyelid borders and the iris one is done considering the end point of each one of the fitted curves. A conservative procedure is performed to connect the curve with straight lines to, furthermore, with the *imfill* Matlab function, create the final mask.

The image obtained of the connected curves fitted of the eyelids and iris border for the example of Figure 3-58 is displayed on Figure 3-75. Figure 3-76 and Figure 3-77 shows the final masks automatically obtained for eye frames displayed on Figure 3-55 and Figure 3-71.

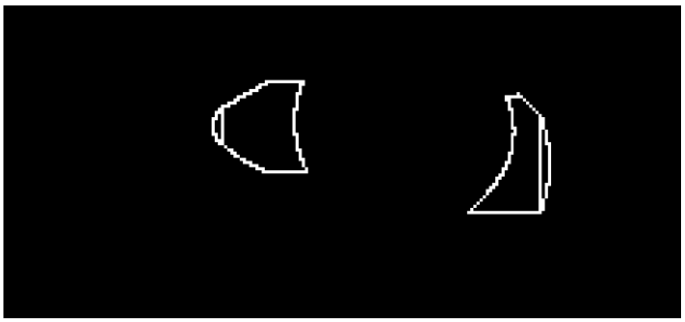

Figure 3-75: Curve limits obtained and displayed on a black and white eye sized image.

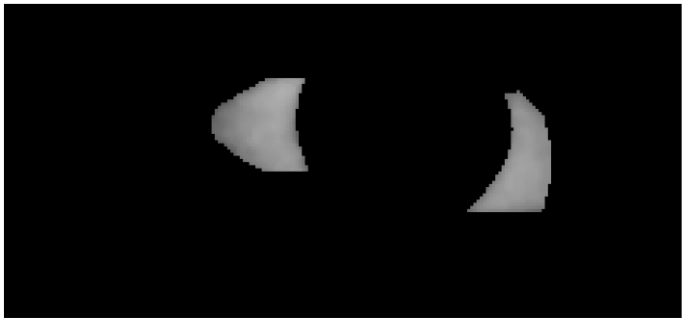

Figure 3-76: Sclera segmented for eye image from Figure 3-55.

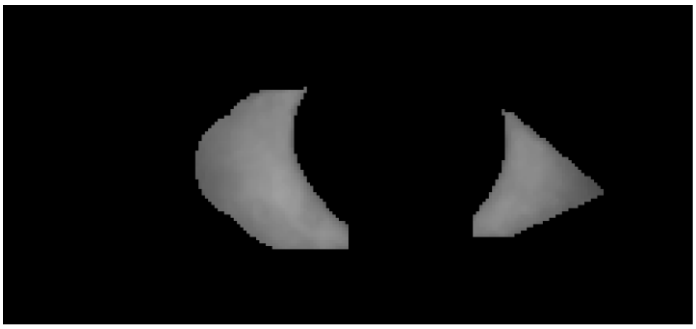

Figure 3-77: Sclera segmented for eye image from Figure 3-71.

3.1.4.1. Analysis of the color of the sclera region

When performing a color analysis, the first important thing is determining the appropriate color system to use for the job. The objective is to find noticeable color differences that will show meaningful variation between a healthy sclera, and one affected with redness. To do that, the normal color of a healthy sclera needs to be defined first. But even healthy sclera has some amount of natural statistical variation in color. To ensure high sensitivity in detecting changes, the color of healthy sclera should, theoretically, be relatively invariant, allowing more noticeable contrasts when compared with sclera affected by redness.

One way to study these diagrams numerically is by calculating the covariance matrix of the set of colors of each eye in the color spaces studied. If the vector is defined as (equation 3-26):

$$V(p) = [I_{c1}(p), I_{c2}(p), I_{c3}(p)] \tag{3-26}$$

where $I_{c1}(p), I_{c2}(p), I_{c3}(p)$ are the color coordinates of the point (p) of the image in the corresponding color system used in each case. From this vector, we can define a covariance matrix as (equation 3-27):

$$S_{ijcolor} = < \left(I_{ci}(p) - \overline{I_{ci}(p)}\right) * \left(I_{cj}(p) - \overline{I_{cj}(p)}\right) > \tag{3-27}$$

where $\overline{I_{ci}(p)}$ is the average of the channel "i" over the image pixels and the final average value $< >_p$ is also obtained on the image pixels.

The covariance matrix summarizes in it all the variability of the color of the pixels of a sclera region in the color space being considered, in such a way that the total variance of the data is given by the trace of the matrix $tr(Scolor)$. In effect, the trace of a matrix is independent of the basis used to calculate the matrix. Since a covariance matrix (symmetric and positive definite) is used, we could always express it in a base where it was diagonal, that is, a base where the covariance between the color channels was zero. Hence, the total variance of the data would be the sum of the diagonal, its trace. And since this is independent of the chosen base, the original $tr(Scolor)$ has the meaning of the total variance of the colors of that sclera region in the color system being studied.

The color stability of a set of healthy eyes in each color system can then be studied using the following figure of merit:

$$\left(\frac{S}{N}\right)_c = \frac{tr(\overline{S_{color}})}{tr\left(\sigma(S_{color})\right)} \tag{3-28}$$

where $\overline{S_{color}}$ is the mean covariance matrix of all eyes considered healthy and $\sigma(S_{color})$ is a matrix whose elements are the standard deviation of each of the elements of $\overline{S_{color}}$ through eyes considered healthy.

Equation 3-28 gives us the average variance of the colors of considering the 20 healthy sclera regions obtained from patients that weren't using contact lenses in color system *C* divided by a measure of the variability of said variance. It is, therefore, a kind of noise to signal ratio. A high value indicates that the set of colors of the contact lens nonuser's sclera regions remains

stable; while a low value would indicate that the variability of the colors of this same set of sclera region in this color system would be very large and, therefore, the sensitivity (or ability) to detect an anomalous feature would be small.

Therefore, a criterion for selecting the appropriate color system would be one that presents a higher S/N for the group of healthy eyes.

Table 3-10: Color systems study to check in which sensitivy is higher to detect an anomalous feature (eye redness).

Color system	$\overline{\boldsymbol{S_{color}}}$	$\boldsymbol{\sigma(S_{color})}$	$\left(\frac{\boldsymbol{S}}{\boldsymbol{N}}\right)_c$
RGB	$\begin{bmatrix} 477.92 & 486.08 & 482.28 \\ 486.07 & 556.27 & 561.96 \\ 482.28 & 561.95 & 582.76 \end{bmatrix}$	$\begin{bmatrix} 180.81 & 169.99 & 174.27 \\ 169.99 & 190.98 & 207.97 \\ 174.27 & 207.97 & 238.69 \end{bmatrix}$	2.65
HSV	$\begin{bmatrix} 295.54 & 249.72 & 241.15 \\ 249.72 & 286.62 & 278.89 \\ 241.15 & 278.89 & 283.92 \end{bmatrix}$	$\begin{bmatrix} 254.83 & 292.80 & 282.84 \\ 292.80 & 303.68 & 305.09 \\ 282.84 & 305.09 & 317.80 \end{bmatrix}$	0.88
CIELab	$\begin{bmatrix} 254.08 & 255.68 & 246.25 \\ 255.68 & 280.69 & 276.13 \\ 246.25 & 276.13 & 280.10 \end{bmatrix}$	$\begin{bmatrix} 291.26 & 287.14 & 278.01 \\ 287.14 & 309.31 & 307.67 \\ 278.01 & 307.67 & 321.27 \end{bmatrix}$	0.99

When analyzed the stability of the color systems (188), it was found that the RGB color system will give us a better sensitivity to detect anomalous pixels where a redness can be identified on sclera regions where the patient is using contact lens or has started suffering from contact lens induce dryness.

Even though the RGB system is the one proposing a higher S/N factor, the characteristic of the HSV and CIELab system makes them appropriate to perform a comparison on the same patient (where the same camera device is being used) if there is a variation in the hue signal for the HSV (color differentiation) on the sclera regions and on channels a^* and b^* as, if they are transformed to polar coordinates, the Channel C_{ab} interpretation is similar to the hue channel for HSV, see Figure 3-62.

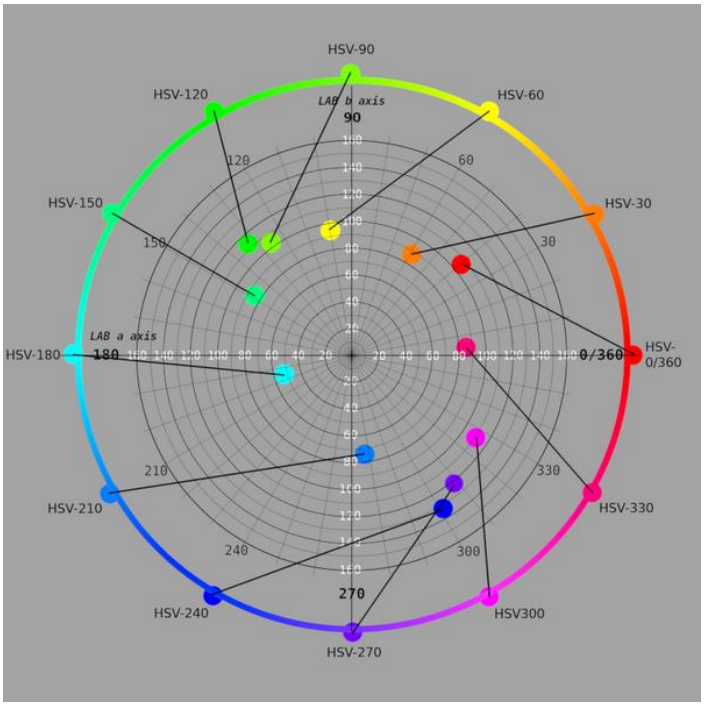

Figure 3-78: A HSV color wheel mapped onto Lab (a,b) space, showing the lack of uniformity in hue and saturation. Elle Stone © CC BY-SA 4.0.

First a visual comparison of the distribution is done to corroborate or to check if there is a change in some channels between contact lens use eye image and the one where the user is not wearing contact lenses.

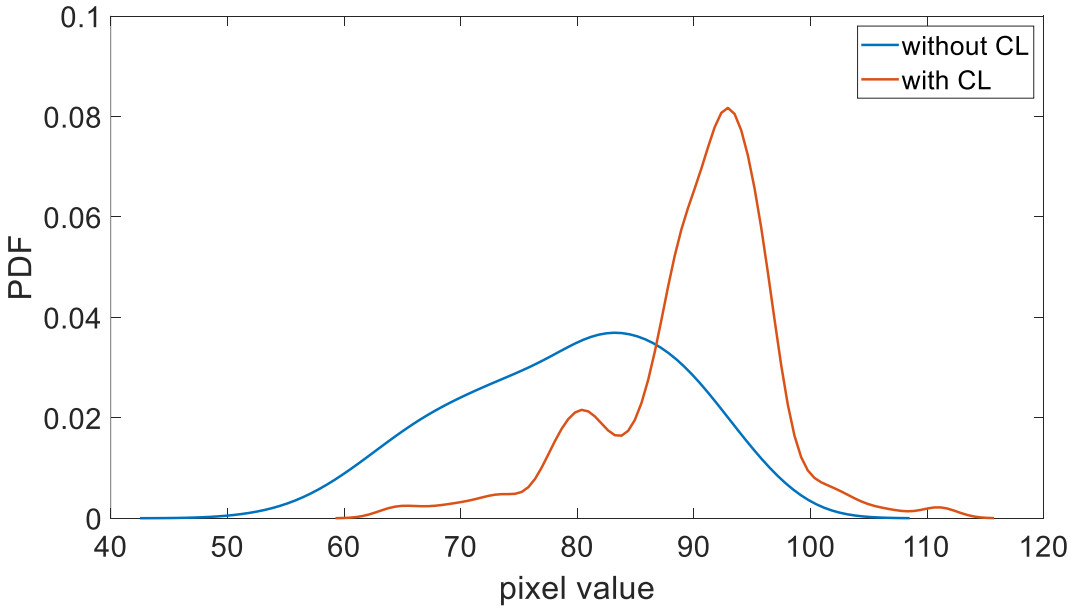

Figure 3-79: PDF of the R channel from the RGB color system. In blue the probability density function of the sclera region when no Contact Lens is being used and in red the user is wearing CL.

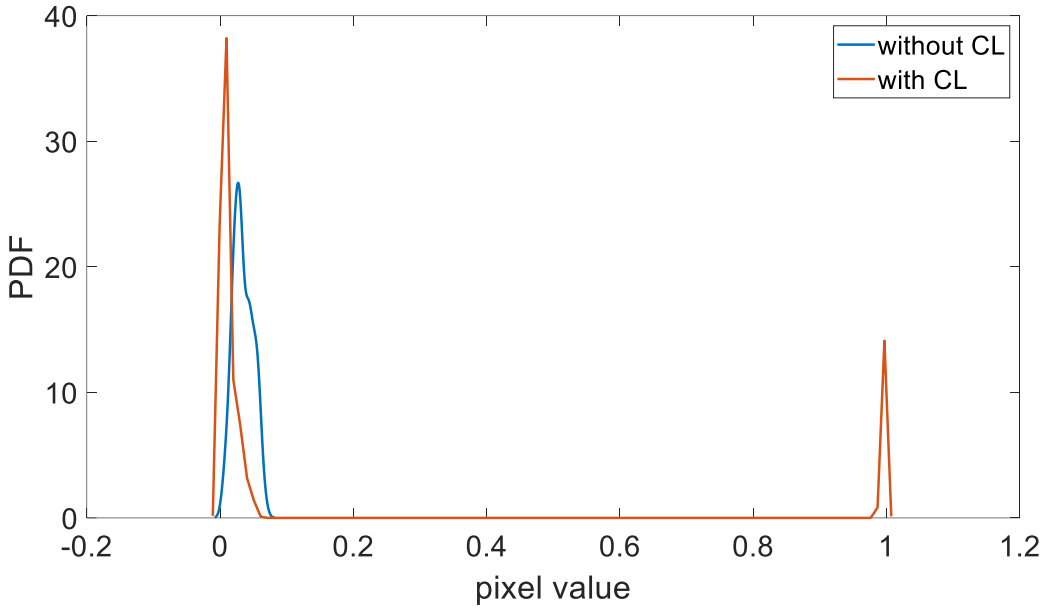

Figure 3-80: PDF of the H channel from the HSV color system. In blue the probability density function of the sclera region when no Contact Lens is being used and in red the user is wearing CL.

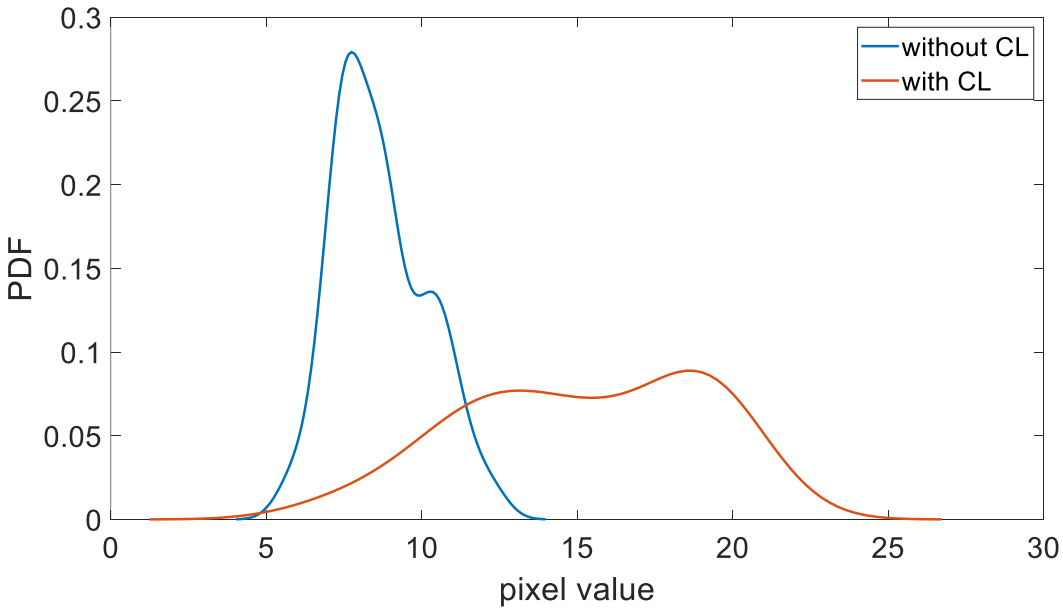

Figure 3-81: PDF of the a channel from the CIELab color system. In blue the probability density function of the sclera region when no Contact Lens is being used and in red the user is wearing CL.

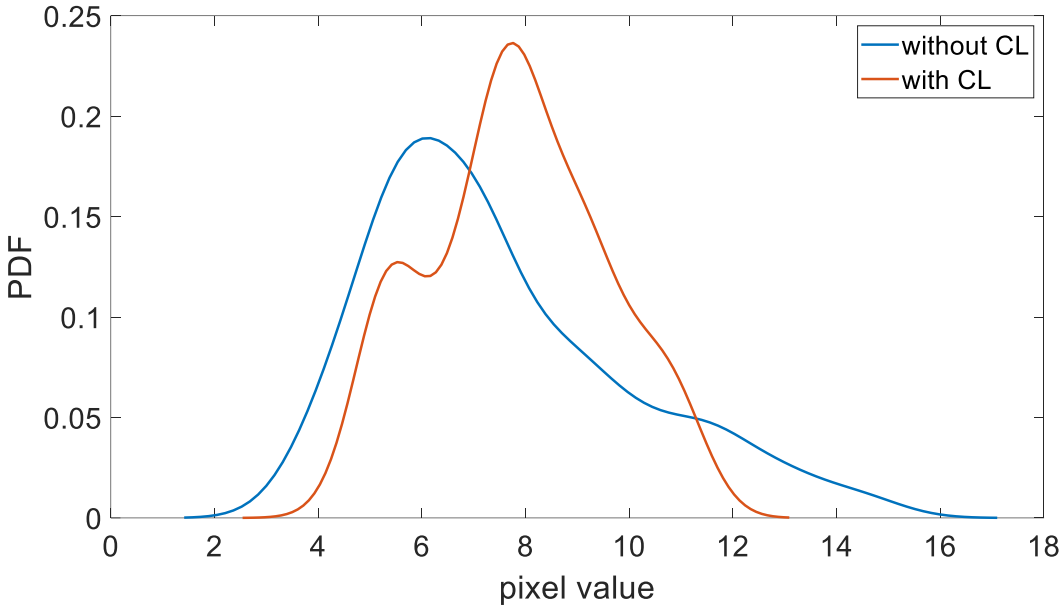

Figure 3-82: PDF of the b channel from the CIELab color system. In blue the probability density function of the sclera region when no Contact Lens is being used and in red the user is wearing CL.

3.1.5. Eyeblink and eye redness database.

To show our database to the public with all the different features extracted from each one of the videos and eye frames using both the automatic eye blink detection on section 3.1.3 and the sclera segmentation algorithm on section 3.1.4 on the METHODS chapter in an easy manner, the information structured in different online and offline databases using structured and non-structured databases (Link to the Annex chapter to see the different studies involving databases) is displayed for each register in an independent excel file. A script in Matlab is coded to read the information and structure the results contained in different excel sheets

3.1.5.1. Excel register file

Each register is constituted from the analysis of 1 min of the left eye where the patient is wearing or not contact lenses. In the first excel sheet, the summary of the eyeblink algorithm is available. The name of the excel is defined considering all the users registered in the application for the study completion. Each user is identified by the P and subsequently as numbers going from 1 to 57. The second number value delimited by a low bar (_), corresponds to a logical value where 1 is that the user is wearing contact lens and 0 if he is not wearing it. The third number separated by the same low bar (_) corresponds to the number of contact lens videos registered by the patient. The fourth number corresponds to the number of videos without contact lenses registered by the same patient. The fifth number will indicate which

video is being analyzed. The last code separated by a low bar (_) identifies the questionnaire type answered in that specific register. To note Q1 will identify an early self-made questionnaire while identifies CLDEQ8 questionnaire. For example, an excel filename corresponding to 'P2_0_8_5_1_Q1.xls' identify the results for patient 2 in the first register without contact lens, knowing that patient 2 has registered 8 videos using contact lenses and 5 videos without using contact lenses, that answered the initial self-made questionnaire.

3.1.5.1.1. Excel sheet 1:

In this excel sheet first we find all the features obtained for each one of the blinks (this parameters are going to be explained on the second excel sheet) and subsequently de blink interval, the mean duration the number of blinks, the power spectrum density fitting to $1/f$ dynamics characterizing the total blinking video time (parameters α, b and $R2$ of the fitting).

Each one of the parameters appears in columns, and the rows values correspond to the mean value (3-29), standard deviation (3-30), kurtosis (3-31) and skewness (3-32) considering both blinks' qualities, complete, and incomplete, only complete blinks and only incomplete blinks. The mean, std, kurtosis and skewness are obtained for each case. The formulation and explanations can be retrieved on (189).

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3-29}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{3-30}$$

$$k = \frac{E(x-\mu)^4}{\sigma^4} \tag{3-31}$$

$$s = \frac{E(x-\mu)^3}{\sigma^3} \tag{3-32}$$

3.1.5.1.2. Excel sheet 2:

In the second excel sheet, the parameters obtained for each blink appears on columns. Each row corresponds to each one of the blinks that appear on the current video analyzed.

The parameters obtained are:

- **t0p_s_:** Time 0 of the start of blinking. With a value of 0 seconds indicating the start.
- **t1p_s_:** Time 1, in which the maximum acceleration of the upper eyelid blinking occurs on the way down. This moment coincides with the maximum power developed by the OOP muscle.
- **t2p_s_:** Time 2, in which the maximum speed of the upper eyelid lowering is obtained (closing the eye). The acceleration is zero at this point since the efforts made by the OOP and LP muscles are counterbalanced.
- **t3p_s_:** Time 3 corresponds to a minimum power, because the effort is the maximum made by the LP muscle, which has the function of opening the eye. At this point the power is opposite to the movement of the upper eyelid.
- **t4p_s_:** Time 4 corresponds to zero speed; at this point the maximum amplitude occurs.
- **t5p_s_:** Time 5, in which the maximum acceleration of the upper eyelid blinking occurs on the way up. This moment coincides with the maximum power developed by the LP muscle.
- **t6p_s_:** Time 6, in which the maximum speed of the upper eyelid rising is obtained (opening the eye). Acceleration is zero at this point because the efforts made by the OOP and LP muscles are counterbalanced.
- **t7p_s_:** Time 7 corresponds to a minimum power, because the effort is the maximum made by the OOP muscle, which has the function of closing the eye. At this point, the power is opposite to the movement of the upper eyelid.
- **t8p_s_:** Time 8 coincides with the end of blinking. At this point, the speed is zero and the amplitude is minimum because the eye has fully opened again.
- **P1:** Normalized power 1, maximum power developed to produce eye closure.
- **P3:** Normalized power 3, maximum power developed by the muscle antagonistic to the blinking movement to stop the eye from closing.
- **P5:** Normalized power 5, corresponding to the maximum power produced during eye opening.
- **P7:** Normalized power 7, maximum power developed by the muscle antagonistic to the blinking movement to stop the eye from opening.
- **W02:** Total work 02 performed to close the eye (OOP muscle).
- **W04:** Total work 24 to stop the eye from closing (LP muscle).
- **W06:** Total work 46 performed to open the eye (LP muscle).
- **W08:** Total work 68 to slow down the opening of the eye (OOP muscle).
- **t1a_s_:** Time 1 of maximum acceleration. Occurs during the eye closing movement.

- **t2a_s_:** Time 2 of minimum acceleration. Occurs just before the eye closes.
- **t3a_s_:** Time 3 of maximum acceleration. Occurs during the eye-opening movement.
- **a1a_s_:** Normalized acceleration 1.
- **a2a_s_:** Normalized acceleration 2.
- **a3a_s_:** Normalized acceleration 3.
- **J02:** Impulse in arbitrary units from point 0 to point 2.
- **J26:** Impulse in arbitrary units from point 2 to point 6.
- **J68:** Impulse in arbitrary units from point 6 to point 8.
- **v2p:** Normalized maximum closing speed, point 2.
- **v6p:** Normalized maximum opening speed, point 6.
- **S:** Ratio between the maximum opening and closing speeds of the eye.
- **R2_ajuste:** R square value of the fitting of the blink signal in mm to the exponentially modified gaussian.
- **norma:** cumulative value of the blink signal in pixels to perform a normalization of the curve before performing the fitting to the exponentially modified gaussian.
- **sigma:** square root of the variance of the normalized curve minus square value of tau defined on (3-35), see equation (3-33), (174).

$$sig = \sqrt{\sigma^2 - tau^2} \tag{3-33}$$

- **mu:** mean value of the normalized blink curve minus tau parameter defined on (3-34), see equation (3-34) , (174).

$$mu = \mu - tau \tag{3-34}$$

- **tau:** standard deviation of the normalized blink curve multiplied by 0.8, see equation (3-35) , (174).

$$tau = \sigma * 0.8 \tag{3-35}$$

- **base:** Value of the base signal taken as fully open eye at the corresponding blinking sequence studied in pixels units.
- **Prom:** Signal amplitude variation between the base and the point of maximum amplitude in pixels units.
- **tiempo_blink_ms_:** Blink duration in seconds.
- **half_width_left_ms_:** Duration of the upper eyelid lowering movement in milliseconds taking as reference the half height.
- **half_width_right_ms_:** Duration of the upper eyelid raising movement in milliseconds taking as reference the half height.

- **half_width_ms_:** Blink duration in milliseconds at half height.
- **width_left_ms_:** Duration of the total eye closing movement in milliseconds.
- **width_right_ms_:** Duration of the total eye-opening movement in milliseconds.
- **width_ms_:** Total blink duration in ms.
- **a1a_sn:** Maximum acceleration during eye closure without normalizing.
- **a2a_sn:** Minimum acceleration during blinking without normalizing.
- **a3a_sn:** Maximum acceleration during eye opening without normalizing.
- **v2p_sn:** Maximum speed of the upper eyelid lowering.
- **v6p_sn:** Maximum speed of the upper eyelid opening.
- **locs_fp_s_:** Time in seconds where the maximum blink amplitude occurs in the video.
- **Hwidth_fp_ms_:** Width at half height of the peak corresponding to the blink under study.
- **validity:** Control variable to check if all the steps concerning the fitting to the exponentially modified gaussian have been performed smoothly from the location of the peak from the pca1 signal of the video. 1 for valid and 0 for non-valid.
- **isblink:** control variable, to check if the data corresponds to a blink (complete or incomplete). 1 for blink 0 for non-blink.
- **closeness__:** Percentage of eye closeness.

3.1.5.1.3. Excel sheet 3:

In excel sheet 3, the information available is related to the complete signal pca1 autovectors used to locate all the most probable blinks before performing the cross-correlation check and the modified PCA 1 autovector signal after performing the conversion to mm of the considered blink sequences.

The column variables are:

- **Frame_number:** Frame number of the video sequence starting from 1 to the maximum number of photograms available in the video.
- **Time_s_:** Time in seconds where corresponding frame is captured by the video camera.
- **Fe1_signal:** Factorized autovectors of first principal component to correct baseline value for each one of photogram of the video. The signal is used to locate the most probable blink locations.
- **signal_mm:** Transformed Fe1_signal where the frames considered part of a blink sequence corresponds in millimeters to the detected closeness of the eyelids.

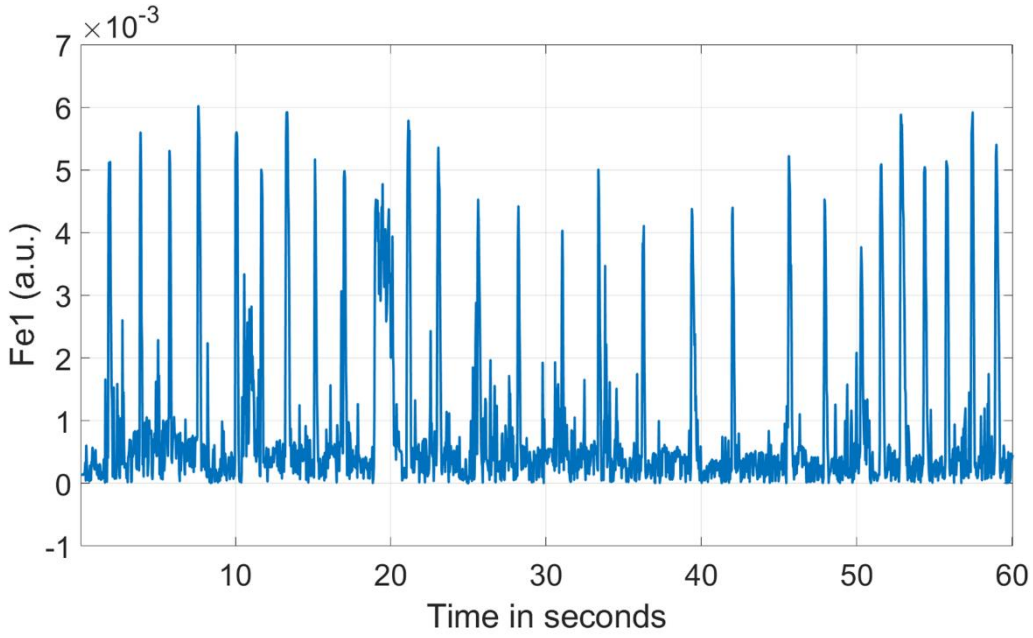

Figure 3-83: Example of a PCA 1 autovectors signal of a video.

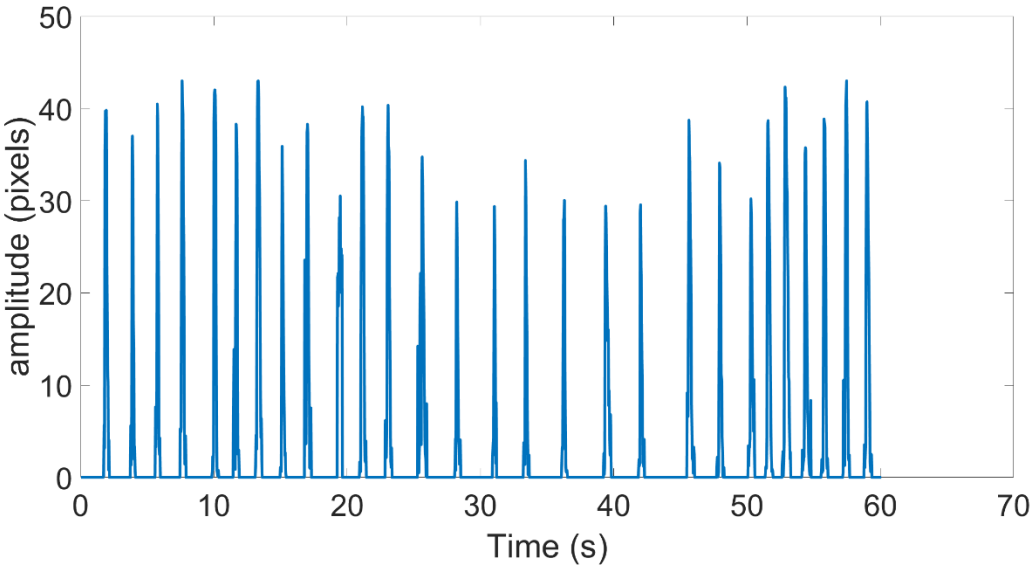

Figure 3-84: Example of a signal in pixels showing the closeness of the eyelids during the entire video sequence obtained from the PCA 1 autovectors signal.

3.1.5.1.4. Excel Sheet 4:

In excel sheet 4, the video properties are shown. The columns are:

- **FrameRate:** Frame rate of the video.
- **VideoHeight_pixels_:** Height in pixels of each one of the eye photograms.
- **VideoWidth_pixels_:** Width in pixels of each one of the eye photograms.

- **LongitudePalpebralCleft_mm_:** Longitudinal palpebral cleft in mm of the patient's eye.
- **FactorPixels2mm_mm_Pixels_:** Conversion factor value in mm/pixels used to get the eyelid distance in millimeters from the eye photogram.
- **CL_brand:** Contact lens brand used by the patient.

3.1.5.1.5. Excel Sheet 5:

In excel sheet 5, the info available is related to the study of images of the open eye, where the sclera region is detected automatically using the algorithm of sclera detection explained on the chapter method concerning the sclera detection on section 3.1.4.

From the detected sclera region of several open-eye images from the same video (maximum 5 images), the pixel values are separated in different channels from 3 different color systems (HSV, LAB, and RGB), each with its own special characteristics for specifying a redness color degree.

The variables or columns are each one of the channels:

- **channel_h:** Channel H from HSV system. The range of values goes from 0 to 1.
- **channel_s:** Channel S from HSV system. The range of values goes from 0 to 1.
- **channel_v:** Channel V from HSV system. The range of values goes from 0 to 1.
- **channel_l:** Channel L from CIELAB system. The range of values goes from 0 to 100.
- **channel_a:** Channel a from CIELAB system. No defined range but most data are between $[-100,100]$.
- **channel_b_lab:** Channel b from CIELAB system. No defined range but most data are between $[-100,100]$.
- **channel_r:** Channel R from RGB system. The range of values goes from 0 to 255.
- **channel_g:** Channel G from RGB system. The range of values goes from 0 to 255.
- **channel_b_rgb:** Channel B from RGB system. The range of values goes from 0 to 255.

The rows correspond to:

- **mean_nasal (1):** Mean value considering all the data corresponding to nasal sclera region for each one of the channel variables.
- **std_nasal (2):** Standard deviation value considering all the data corresponding to nasal sclera region for each one of the channel variables.

- **kurtosis_nasal (3):** kurtosis value considering all the data corresponding to nasal sclera region for each one of the channel variables.
- **skewness_nasal (4):** skewness value considering all the data corresponding to nasal sclera region for each one of the channel variables.
- **mean_temporal (1):** Mean value considering all the data corresponding to temporal sclera region for each one of the channel variables.
- **std_temporal (2):** Standard deviation value considering all the data corresponding to temporal sclera region for each one of the channel variables.
- **kurtosis_temporal (3):** kurtosis value considering all the data corresponding to temporal sclera region for each one of the channel variables.
- **skewness_temporal (4):** Skewness value considering all the data corresponding to temporal sclera region for each one of the channel variables.

3.1.5.1.6. Excel sheet nasal_color_data:

In excel sheet nasal_color_data the information available are all the pixels value of nasal sclera region of all the images of open eye analyzed for this specific video.

The variables corresponds to each one of the channels in the 3 different color systems (HSV, CIELab and RGB).

The variables or columns are each one of the channels:

- **channel_h:** Channel H from HSV system. The range of values goes from 0 to 1.
- **channel_s:** Channel S from HSV system. The range of values goes from 0 to 1.
- **channel_v:** Channel V from HSV system. The range of values goes from 0 to 1.
- **channel_l:** Channel L from CIELAB system. The range of values goes from 0 to 100.
- **channel_a:** Channel a from CIELAB system. No defined range but most data are between $[-100,100]$.
- **channel_b_lab:** Channel b from CIELAB system. No defined range but most data are between $[-100,100]$.
- **channel_r:** Channel R from RGB system. The range of values goes from 0 to 255.
- **channel_g:** Channel G from RGB system. The range of values goes from 0 to 255.
- **channel_b_rgb:** Channel B from RGB system. The range of values goes from 0 to 255.

Each row corresponds to a different pixel position of nasal sclera region and the value is specific to each one of the channels.

3.1.5.1.7. Excel sheet temporal_color_data:

In the excel sheet temporal_color_data the information available is all the pixels value of temporal sclera region of all the images of open eye analyzed for this specific video.

The variables correspond to each one of the channels in the 3 different color systems (HSV, CIELab and RGB).

The variables or columns are each one of the channels:

- **channel_h:** Channel H from HSV system. The range of values goes from 0 to 1.
- **channel_s:** Channel S from HSV system. The range of values goes from 0 to 1.
- **channel_v:** Channel V from HSV system. The range of values goes from 0 to 1.
- **channel_l:** Channel L from CIELAB system. The range of values goes from 0 to 100.
- **channel_a:** Channel a from CIELAB system. No defined range but most data are between $[-100,100]$.
- **channel_b_lab:** Channel b from CIELAB system. No defined range but most data are between $[-100,100]$.
- **channel_r:** Channel R from RGB system. The range of values goes from 0 to 255.
- **channel_g:** Channel G from RGB system. The range of values goes from 0 to 255.
- **channel_b_rgb:** Channel B from RGB system. The range of values goes from 0 to 255.

Each row corresponds to a different pixel position of temporal sclera region and the value is specific to each one of the channels.

3.1.5.1.8. Excel sheet Environment_data:

In excel sheet Environment_data we can find the environmental data. This data is obtained from the weather stations but also from air contamination stations. Additionally, the values entered by the user/patient related to the Interior contamination can also be found. In total the Parameters that can be entered are as follows.

- **no2:** Nitrogen dioxide measured in the previous hour in $\mu g/m^3$.
- **o3:** Ozone measured in the previous hour in $\mu g/m^3$.
- **pm10:** 10 microns particles measured in the previous hour in $\mu g/m^3$.
- **pm25:** 2.5 microns particles measured in the previous hour in $\mu g/m^3$.
- **so2:** Sulfur dioxide measured in the previous hour in $\mu g/m^3$.
- **b:** benzene measured in the previous hour in $\mu g/m^3$.

- **ta:** Exterior Temperature measured in the previous hour in $°C$
- **vmax:** Maximum wind speed, maximum wind value maintained for 3 seconds and recorded in 60 minutes prior to the time indicated by the observation period (m/s).
- **vv:** Average wind speed, scalar average of the samples acquired every 0.25 or 1 second in the 10-minute period (m/s).
- **dv:** Average wind direction, in the 10-minute period prior to the date indicated.
- **prec:** Accumulated precipitation, measured by the rain gauge, during the 60 minutes prior to the time indicated by the observation period (mm, equivalent to l/m2)
- **pres:** Instantaneous pressure at the level at which the wind direction is measured which the barometer is installed and corresponding to the date given (hPa).
- **hr:** Instantaneous relative humidity of the air corresponding to the date given (%).
- **T_int:** Temperature registered by the user in the interior
- **hr_int:** Relative humidity registered by the user in the interior.

3.1.5.1.9. Excel sheet Questionnaire_type_#1;2:

In the questionnaire excel sheet appears all the answers related to the contact lens comfort. Two types of questionnaires have been answered related to the comfort using contact lenses. All the questions and their identifications are available in a separate excel file to check the differences. In the name of the excel file a code identifying each questionnaire has been answered in that specific register.

In the variables or columns, we can find the identification number of the question of the questionnaire type identified in the excel filename, the question and the answer given by the user.

3.1.5.2. Questionnaires excel file

As stated in the previous section with the explanation of the register excel file, all the questions and possible answers have been saved in an excel file. Two types of questionnaires have been answered by the patients.

As in both excel questionnaire files the information is divided into excel sheets. In the first excel sheet all the demographic questions that are being answered only at the first register of each patient as the data is not being changed across the study. In the second excel sheet appears the control questions that identify whether the patient is using contact lenses or not and if they have been using them during the day. If both answers are negative, no comfort questionnaire

is going to be answered by the user nor be shown on the excel register. Lastly, on the third excel sheet, all the questions related to the comfort of the patient while using contact lenses will appear.

The structure of each excel sheet is:

- **Id:** identifies the question in the app and in the excel register.
- **Langue:** Language of the question.
- **Questions:** Question to be answered.
- **reponse_1:** Possible first response.
- **reponse_2:** Possible second response.
- **reponse_3:** Possible third response.
- **reponse_4:** Possible forth response.
- **reponse_5:** Possible fifth response.
- **reponse_6:** Possible sixth response.
- **num:** number of the question.

4 RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

First, once the recollection tool was used with different contact lens users, it was possible to analyze the data using the various algorithms explained in the methods chapter. Initially, we have data related to blinking and the color of the sclera, which are included as characteristics of the patient's eye, measured objectively from the analysis of the one-minute videos recorded by each patient involved in the study. Secondly, in addition to the objective eye data, we used the atmospheric stations available in the region of the Community of Madrid and the air pollution stations in the municipality of Madrid. During the study, a thermometer and hygrometer sensor were provided to register the humidity and temperature values recorded by the patients themselves if the environment is being controlled. In summary, we have objective data about the environment in which patients go about their daily lives. Thirdly, we have objective data on the characteristics of the video, which depend on the camera used. Finally, while collecting patient responses about the comfort of wearing contact lenses, we also gathered subjective data.

4.1 **First study:** Pearson Correlation to evaluate the relationship between different feature groups.

The first analysis studies the correlation between each objective characteristic in the database. For the color and blink analysis, we have 208 records from 35 patients, of which 12 are contact lens users whose records have finally been analyzed. In addition to the environmental and air quality, a correlation between the extracted features was performed to check for a global relationship between the different features. Parameters studied in the Pearson correlation results are shown on Figure 4-1 and Figure 4-2 are shown in Table 4-2and

Table 4-3 where the variables related to the blinking motion go from 1 to 180 and parameters related to the sclera color go from 181-252. The correlations between the same group of variables can be identified inside a square in Figure 4-1 and Figure 4-2. The Pearson correlation parameters are obtained using the Statistic and Machine Learning Toolbox of the Matlab Environment by performing the "corr" function on the database data related to the different groups of features.

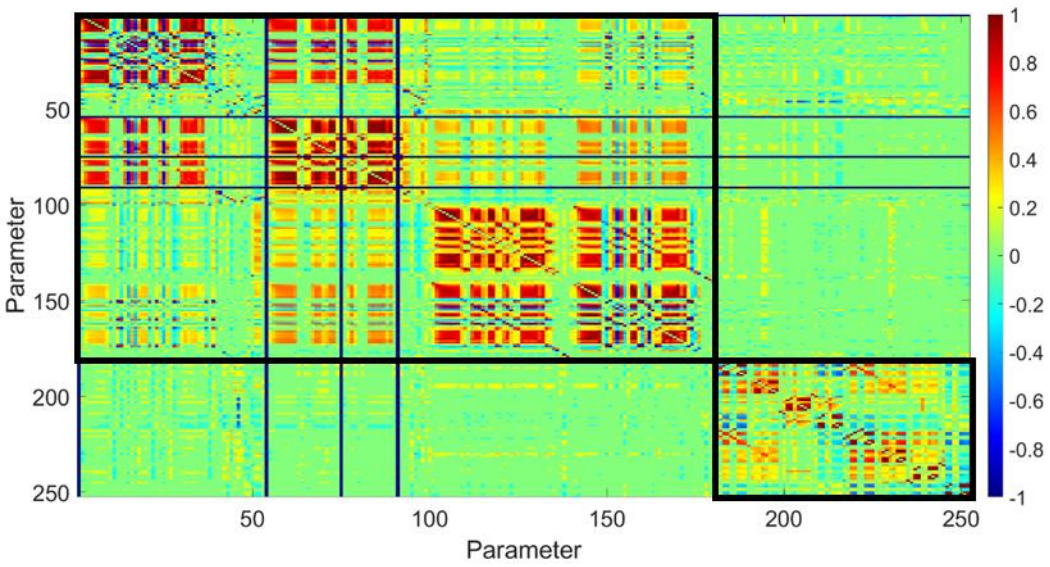

Figure 4-1: Correlation value between parameters. The first group corresponds to blinking values (1-180) and the second group (181-252) to the sclera color features see.

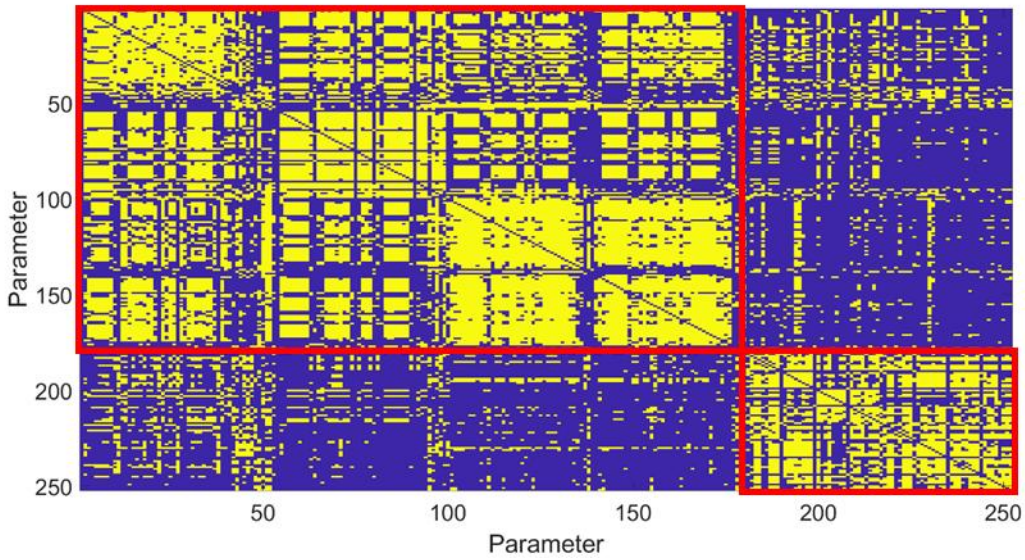

Figure 4-2: Map of the significantly correlated parameters (p-value<0.05, in yellow).

Based on the observed results (Figure 4-1 and Figure 4-2), and considering that the analyzed data comprises a mixed group of records in which the patient has either worn or not worn contact lenses, we see that the various parameter groups, specifically blinking features and sclera color characteristics, show a significant correlation. The blinking variables have significant correlations between them, as do the color variables. However, between these two groups (blinking-color correlations), significant correlations are only established between several of them. It is found that the standard deviation of the channel a* from the CIELAB color system for both sclera regions (nasal and temporal) is correlated to most mean, skewness, and kurtosis values from all blinking parameters. The correlation values, however, are of the order of 0.2 and are not considered that high. This result justifies that having both groups of features is relevant for a deeper analysis. These first results show only the parameters for two different groups of features, blinking and sclera color. This first study was conducted because all those registers have the same data features and information available. This is because of the 208 registers; not all of them have the same air contamination data and weather conditions registered. This is because the information available from different weather and air contamination stations is not the same for all.

Additionally, in some of those 208 registers, the user uses contact lenses; in others, the user does not. Due to this, and the fact that we are curious to observe how the different groups of features will be correlated to each other when using or not using contact lenses, we proceed to make the same Pearson correlation analysis, but this time including weather and air contamination data and separating the registers in the contact lens users group from registers where no contact lenses have been used.

An algorithm is used for this step to perform data cleaning and maximize the registers and data features from the different groups of parameters, such as blinking, sclera color, and weather and air contamination data. Considering the different groups of features, the registers where typical weather and air contamination data were available were prioritized. Furthermore, the optimization was performed without considering registers or variables depending on the lack of information. The deletion of a variable is done because it is not obtained in most of the registers. In contrast, the deletion of a register is due to poor quality when not enough blinks were detected to obtain kurtosis or skewness values).

The Pearson correlation between the features in the video groups with and without lenses is analyzed to compare the internal correlation of the features with and without lenses. Figure 4-3

shows the value of the correlation for the group of videos without a lens and Figure 4-4 indicates when the correlation is statistically significant (value 1) or not (value 0), taking as a criterion a p-value less than 0.05 ($p<0.05$). The calculations have been carried out through the MATLAB toolbox. Analogous results for the lens video group are shown in Figure 4-5 and Figure 4-6. The parameters group ranges from 1 to 101, corresponding to eye blinking features, 102 to 173 to sclera color, and 174 to 180 to environmental and contamination data. To check each one of the features, please refer to Table 4-6 and Table 4-7. The Pearson correlation analysis considering weather and air contamination data with features extracted from eye videos shows 25 registers with contact lenses and 52 registers without contact lenses.

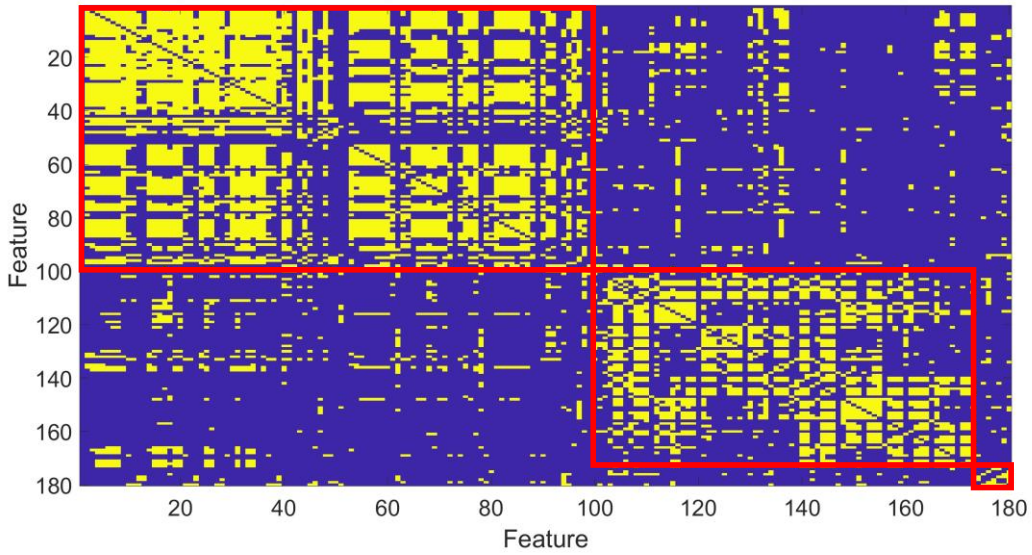

Figure 4-3: Pearson correlation has significant correlations, with a p-value lower than 0.05 in yellow for the group without contact lenses. In the red boxes, we can see the significant correlations between features of the same group (blinking features, sclera color features, and weather and air contamination parameters).

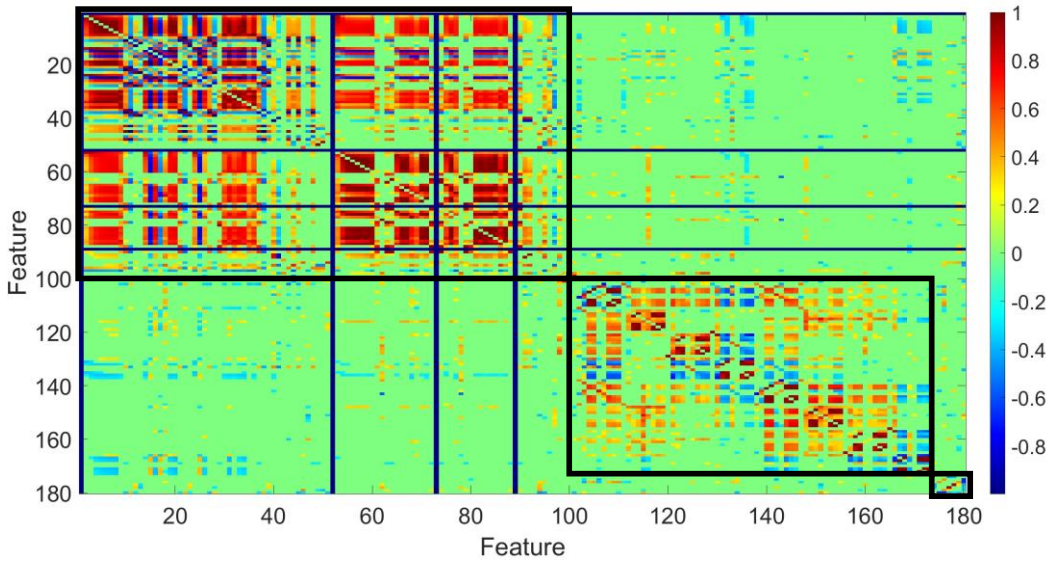

Figure 4-4: Pearson Correlation value for the group without contact lenses. In the black boxes, we can see the correlation values between features of the same group (blinking features, sclera color features, and weather and air contamination parameters).

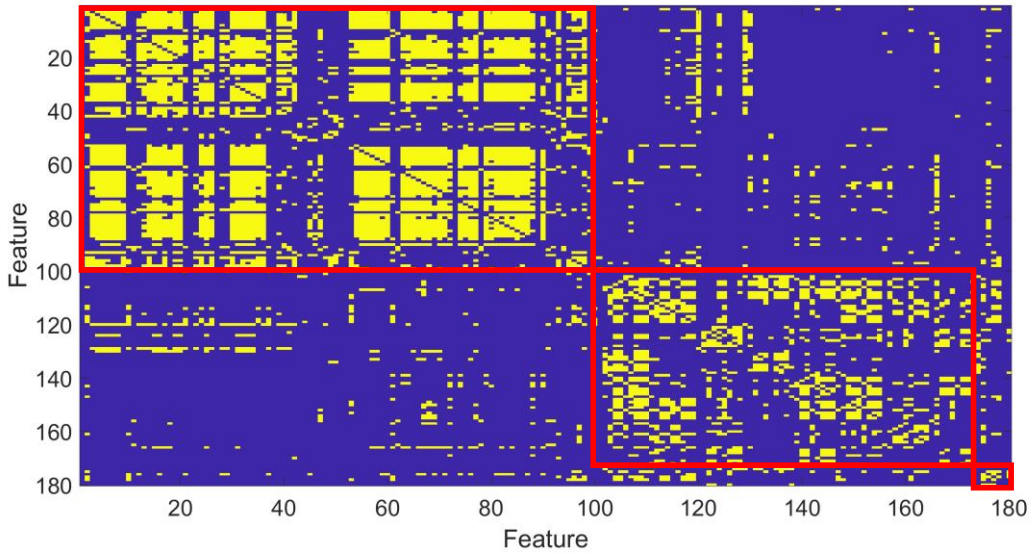

Figure 4-5: Pearson correlation significant correlations p-value lower than 0.05 in yellow for the group with contact lenses. In the red boxes, we can see considerable correlations between features of the same group (blinking features, sclera color features, and weather and air contamination parameters).

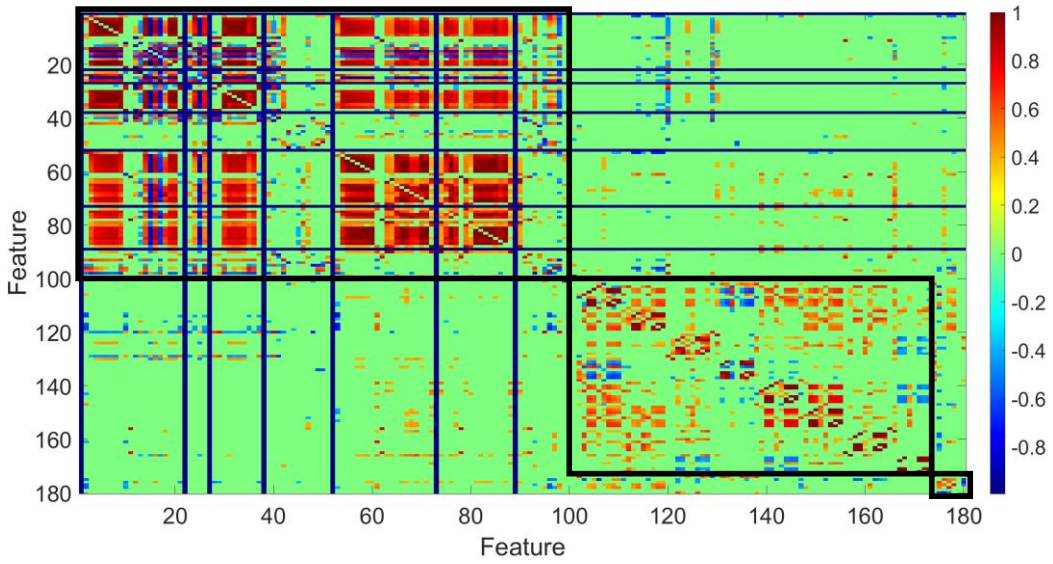

Figure 4-6: Pearson Correlation value for the group with contact lenses. In the black boxes, we can see the correlation values between features of the same group (blinking features, sclera color features, and weather and air contamination parameters).

From all the figures on data corresponding to the group without contact lenses and with contact lenses, we can see that some color features are correlated with nearly all the blinking features. In addition, some weather and air contamination parameters are also associated with sclera color and blinking features.

The results of the Pearson correlation are shown in Figure 4-3 and Figure 4-4, the color sclera correlated features with more blinking features when no contact lenses are being used are related to channel h of the HSV color system by the mean and std value of the nasal sclera pixels. This variable is one of the most important ones considering the color sclera features as it is defined as the "attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them." (190). Other sclera color-correlated features with many blinking features are the standard deviation value of the b* channel from the CIELAB color system in the nasal sclera region, the same value of the b* channel but for the temporal sclera region, and the skewness value of channel r of the temporal sclera region, among others. The relationship between features of sclera color and blinking features of both group registers (registers with contact lenses and not) will be studied furthermore considering more registers.

From the results of Pearson correlation shown in Figure 4-5 and Figure 4-6, the color sclera correlated features with more blinking features, when contact lenses are being used, are related to channel h from the HSV color system and, more specifically, to the value of kurtosis and

skewness related to the nasal sclera region. It's importance is significant as the hue value from HSV color system is and indicative of the color change.

When studying the correlation between the different groups of blinking features and those of sclera color, we observe a higher density of yellow regions when analyzing the recordings where no contact lenses were used. This indicates that blinking features are more related to sclera color features when no contact lenses are used. Additionally, when contact lenses are used, we observe a higher density of significant correlations between weather and air contamination data with the objective features obtained from the sclera color regions compared to the results from the group of registers without contact lenses.

Once we have checked all the features from eye blinking and sclera redness and seen the changes in the variables that happen when patients wear contact lenses, we proceed to zoom into the analysis to check the relationship between the environmental conditions and air contamination data with features obtained from eye blinking and sclera color (see Figure 4-7 and Figure 4-8).

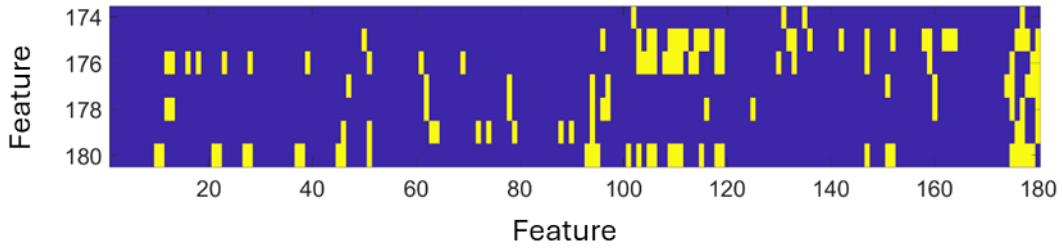

Figure 4-7: Significant Pearson Correlation $(p < 0.05)$ *of air contamination data and environmental conditions, where parameters 174, 175, 176, 177, 178, 179, and 180 correspond to* NO_2, O_3, *Temperature, maximum wind speed, wind speed, wind direction, and humidity, respectively. Poner feature en el label de la figura.*

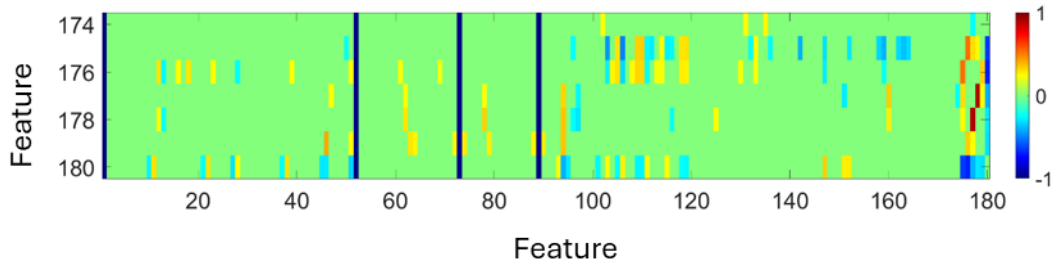

Figure 4-8: Value of Pearson Correlation when a significant correlation is found $(p < 0.05)$ *of air contamination data and environmental conditions with eye blink and sclera color features, where parameters 174, 175, 176, 177, 178, 179, 180 correspond to* NO_2, O_3, *Temperature, maximum wind speed, wind speed, wind direction, and humidity, respectively.*

As expected, the humidity (feature 180, see Table 4-7) correlation is significant with most blinking features (with positive and negative values), including parameters such as the inter-blink interval, duration of the blinking, degree of completeness of the blink, and many others that affect the dynamic and amplitude of the blinking motion (1/f features). Additionally, from the Figure 4-7 and the Figure 4-8, it was observed that humidity affects some parameters of eye redness, such as channel a* from the CIELAB system in the nasal and temporal regions.

Table 4-1: Significant Pearson correlations of features from blinking and sclera color with ambient relative humidity.

Positive Pearson correlation	**Negative Pearson correlation**
11: “P3”	10: “P1”
22: “a2a (-)”	21: “a1a (-)”
28: “v6p”	27: “v2p”
38: “a2a_sn (mm/s^2)”	37: “a1a (mm/s^2)”
93: “std_locs_fp (s)”	45: “m_cp_complete”
103: “channel_s”	46: “m_cp_incomplete”
106: “channel_a'”	51: “num_incomplete”
111: “nasal_std_channel_h”	94: “std_cp”
115: “nasal_std_channel_a”	95: “std_ibi”
147: “temporal_std_channel_h”	101: “skew_ibi”
151: “temporal_std_channel_a”	105: “channel_l”
152: “temporal_std_channel_b_lab”	109: “channel_g”
	110: “channel_b_rgb'
	118: “nasal_std_channel_g”
	119: “nasal_std_channel_b_rgb”

Furthermore, a study of the correlation factors of the other environmental conditions and the air contamination parameters was performed. On the one hand, air condition parameters NO_2 and O_3, from the observation of Figure 4-7 and Figure 4-8, are more correlated to eye redness parameters than blinking features. On the other hand, the wind parameters (speed, direction, and maximum speed) are correlated to more eye-blinking features than eye-redness parameters.

The analysis is conducted again to increase the number of videos in the sample, but this time, the environmental and pollution variables are removed from the study. This is because connection failures with the station often occur, resulting in data not being collected. Furthermore, the same subject may have environmental data from different stations due to patient mobility, which may produce varying parameters. Attempting to make all videos include the same variables can decrease the total number of videos. Therefore, we compare videos with and without a lens, but now without pollution and environmental "features." As a result, the sample size increased to 208 videos, of which 142 are without lenses and 66 have lenses. The results for the correlations between features in this group of videos are shown in Figure 4-9, Figure 4-10, Figure 4-11 and Figure 4-12.

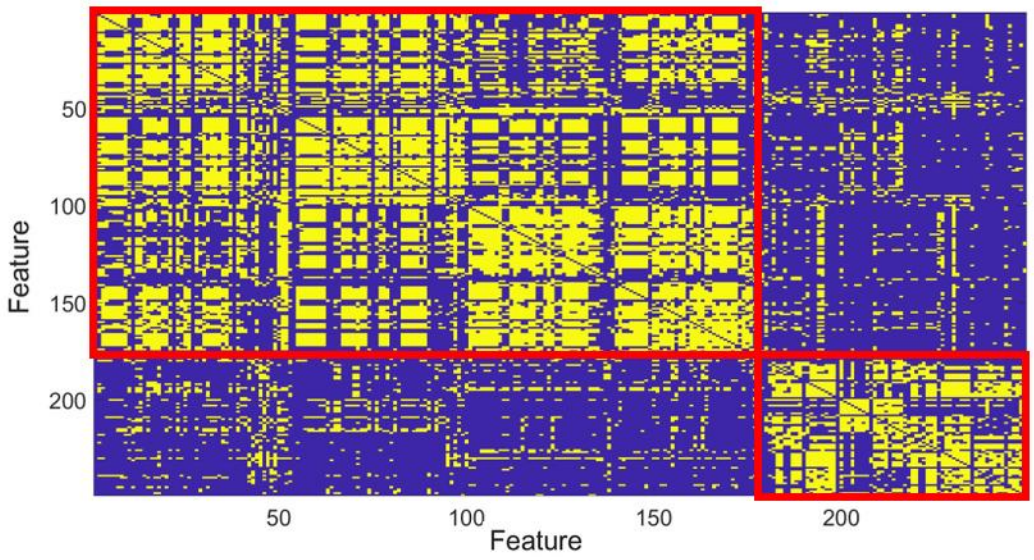

Figure 4-9: Pearson correlation significant correlations p-value lower than 0.05 in yellow for the group without contact lenses. Considering only features from eye videos.

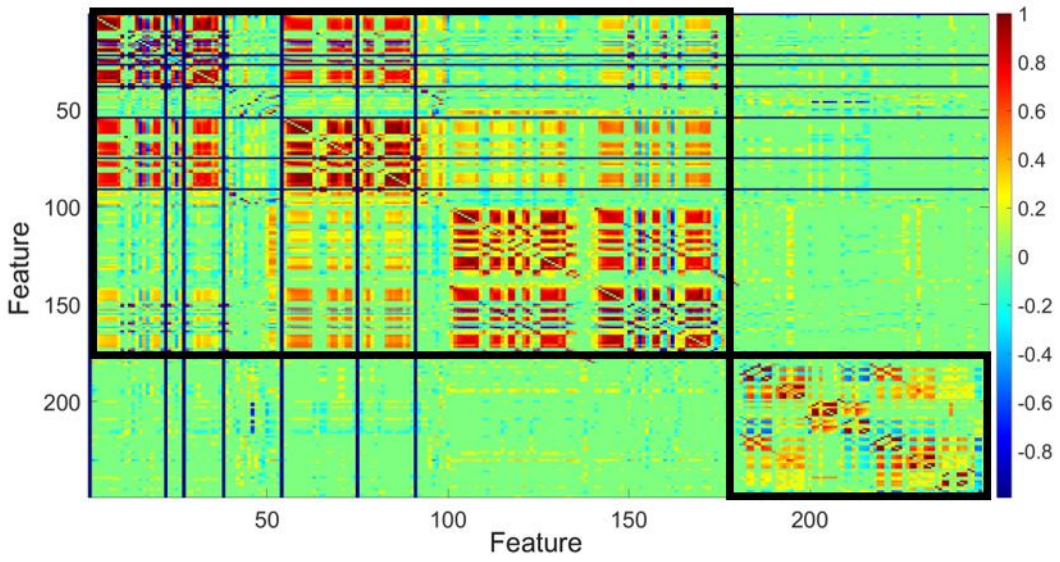

Figure 4-10: Pearson Correlation significant for the group without contact lenses. Considering only features from eye videos.

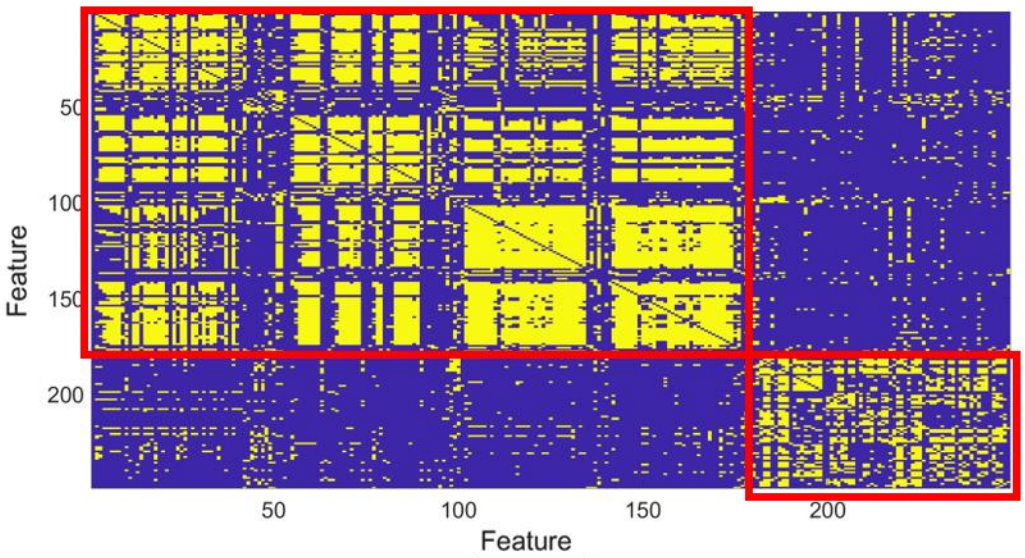

Figure 4-11: Pearson correlation significant correlations p-value lower than 0.05 in yellow for the group with contact lenses. Considering only features from eye videos.

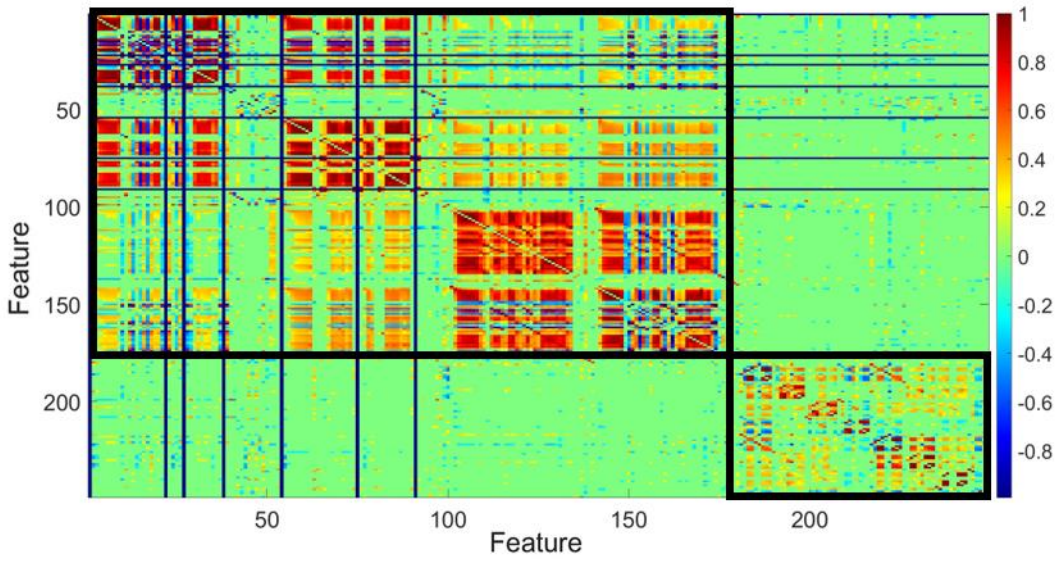

Figure 4-12: Pearson Correlation significant for the group with contact lenses. Considering only features from eye videos

When the increase in the number of registers is prioritized instead of the weather and air conditions features variable while performing the data cleaning, the results don't differ that much because the group of registers without contact lenses shows more features from different groups significantly correlated than the group of registers where contact lenses are being used.

In the group without contact lenses, we observed that when using more registers, a low correlation value is observed between most of the blinking features with a standard deviation value of the channel a* from CIELAB color in both sclera regions, and the standard value is

observed to change in the b* channel of the CIELAB color system. Both a* and b* channels indicate a color change, and their use is similar to the hue value in the HSV color system. A change in the a* channel suggests a variation in color from red to green and can be used as an indicator of the redness of the sclera regions.

When using contact lenses, the Figure 4-11 and Figure 4-12 shows fewer features correlated between sclera color region and blinking features than when analyzing registers without contact lenses. The results don’t differ much when using more registers.

In summary, from this first preliminary study, it is observed that there is a significant correlation between the different groups of features obtained from the patient's eye videos and the weather and air contamination data. Additionally, we observe that this relationship between features changes when the patients are using or not contact lenses, and it is interesting to observe which parameters specifically are altered and alter the blinking motion and redness on the sclera regions.

Table 4-2: Pearson correlation characteristics mixed group (1/2).

1	't0p (s)'	31	'half_width_left (ms)'	61	'std_t7p (s)'	91	'std_a2a_sn'	121	'kur_J02'
2	't1p (s)'	32	'half_width_right (ms)'	62	'std_t8p (s)'	92	'std_a3a_sn'	122	'kur_J26'
3	't2p (s)'	33	'half_width (ms)'	63	'std_P1'	93	'std_v2p_sn'	123	'kur_J68'
4	't3p (s)'	34	'width_left (ms)'	64	'std_P3'	94	'std_v6p_sn'	124	'kur_v6p'
5	't4p (s)'	35	'width_right (ms)'	65	'std_P5'	95	'std_locs (s)'	125	'kur_S'
6	't5p (s)'	36	'width (ms)'	66	'std_P7'	96	'std_cp'	126	'kur_tiempo_blink (ms)'
7	't6p (s)'	37	'a1a_sn'	67	'std_W02'	97	'std_ibi'	127	'kur_half_width_left (ms)'
8	't7p (s)'	38	'a2a_sn'	68	'std_W24'	98	'std_dur'	128	'kur_half_width_right (ms)'
9	't8p (s)'	39	'a3a_sn'	69	'std_W46'	99	'std_alfa_mm'	129	'kur_half_width (ms)'
10	'P1'	40	'v2p_sn'	70	'std_W68'	100	'std_beta_mm'	130	'kur_width_left (ms)'
11	'P3'	41	'v6p_sn'	71	'std_t1a (s)'	101	'kur_t1p (s)'	131	'kur_width_right (ms)'
12	'P5'	42	'locs_fp (s)'	72	'std_t2a (s)'	102	'kur_t2p (s)'	132	'kur_width (ms)'
13	'P7'	43	'm_cp'	73	'std_t3a (s)'	103	'kur_t3p (s)'	133	'kur_a1a_sn'
14	'W02'	44	'm_ibi'	74	'std_a1a (s)'	104	'kur_t4p (s)'	134	'kur_a3a_sn'
15	'W24'	45	'm_dur'	75	'std_a2a (s)'	105	'kur_t5p (s)'	135	'kur_v2p_sn'
16	'W46'	46	'm_cp_complete'	76	'std_a3a (s)'	106	'kur_t6p (s)'	136	'kur_v6p_sn'
17	'W68'	47	'm_dur_complete'	77	'std_J02'	107	'kur_t7p (s)'	137	'kur_locs_fp (s)'
18	't1a (s)'	48	'm_cp_incomplete'	78	'std_J26'	108	'kur_t8p (s)'	138	'kurt_cp'
19	't2a (s)'	49	'alpha_mm'	79	'std_J68'	109	'kur_P1'	139	'kurt_ibi'
20	't3a (s)'	50	'b_mm'	80	'std_v2p'	110	'kur_P5'	140	'kurt_dur'
21	'a1a (s)'	51	'num'	81	'std_v6p'	111	'kur_P7'	141	'skew_t1p (s)'
22	'a2a (s)'	52	'num_complet'	82	'std_S'	112	'kur_W02'	142	'skew_t2p (s)'
23	'a3a (s)'	53	'num_incomplet'	83	'std_tiempo_blink (ms)'	113	'kur_W24'	143	'skew_t3p (s)'
24	'J02'	54	'std_t0p (s)'	84	'std_half_width_left (ms)'	114	'kur_W46'	144	'skew_t4p (s)'
25	'J26'	55	'std_t1p (s)'	85	'std_half_width_right (ms)'	115	'kur_W68'	145	'skew_t5p (s)'
26	'J68'	56	'std_t2p (s)'	86	'std_half_width (ms)'	116	'kur_t1a (s)'	146	'skew_t6p (s)'
27	'v2p'	57	'std_t3p (s)'	87	'std_width_left (ms)'	117	'kur_t2a (s)'	147	'skew_t7p (s)'
28	'v6p'	58	'std_t4p (s)'	88	'std_width_right (ms)'	118	'kur_t3a (s)'	148	'skew_t8p (s)'
29	'S'	59	'std_t5p (s)'	89	'std_width (ms)'	119	'kur_a1a (s)'	149	'skew_P1'
30	'tiempo_blink (ms)'	60	'std_t6p (s)'	90	'std_a1a_sn'	120	'kur_a3a (s)'	150	'skew_P5'

Table 4-3: Pearson correlation characteristics mixed group (2/2).

151	'skew_P7'	181	'channel_h'	211	'nasal_skew_channel_l'	241	'temporal_kurt_channel_r'
152	'skew_W02'	182	'channel_s'	212	'nasal_skew_channel_a'	242	'temporal_kurt_channel_g'
153	'skew_W24'	183	'channel_v'	213	'nasal_skew_channel_b_lab'	243	'temporal_kurt_channel_b_rgb'
154	'skew_W46'	184	'channel_l'	214	'nasal_skew_channel_r'	244	'temporal_skew_channel_h'
155	'skew_W68'	185	'channel_a'	215	'nasal_skew_channel_g'	245	'temporal_skew_channel_s'
156	'skew_t1a (s)'	186	'channel_b_lab'	216	'nasal_skew_channel_b_rgb'	246	'temporal_skew_channel_v'
157	'skew_t2a (s)'	187	'channel_r'	217	'temporal_mean_channel_h'	247	'temporal_skew_channel_l'
158	'skew_t3a (s)'	188	'channel_g'	218	'temporal_mean_channel_s'	248	'temporal_skew_channel_a'
159	'skew_a1a (s)'	189	'channel_b_rgb'	219	'temporal_mean_channel_v'	249	'temporal_skew_channel_b_lab'
160	'skew_a3a (s)'	190	'nasal_std_channel_h'	220	'temporal_mean_channel_l'	250	'temporal_skew_channel_r'
161	'skew_J02'	191	'nasal_std_channel_s'	221	'temporal_mean_channel_a'	251	'temporal_skew_channel_g'
162	'skew_J26'	192	'nasal_std_channel_v'	222	'temporal_mean_channel_b_lab'	252	'temporal_skew_channel_b_rgb'
163	'skew_J68'	193	'nasal_std_channel_l'	223	'temporal_mean_channel_r'		
164	'skew_v6p'	194	'nasal_std_channel_a'	224	'temporal_mean_channel_g'		
165	'skew_S'	195	'nasal_std_channel_b_lab'	225	'temporal_mean_channel_b_rgb'		
166	'skew_tiempo_blink (ms)'	196	'nasal_std_channel_r'	226	'temporal_std_channel_h'		
167	'skew_half_width_left (ms)'	197	'nasal_std_channel_g'	227	'temporal_std_channel_s'		
168	'skew_half_width_right (ms)'	198	'nasal_std_channel_b_rgb'	228	'temporal_std_channel_v'		
169	'skew_half_width (ms)'	199	'nasal_kurt_channel_h'	229	'temporal_std_channel_l'		
170	'skew_width_left (ms)'	200	'nasal_kurt_channel_s'	230	'temporal_std_channel_a'		
171	'skew_width_right (ms)'	201	'nasal_kurt_channel_v'	231	'temporal_std_channel_b_lab'		
172	'skew_width (ms)'	202	'nasal_kurt_channel_l'	232	'temporal_std_channel_r'		
173	'skew_a1a_sn'	203	'nasal_kurt_channel_a'	233	'temporal_std_channel_g'		
174	'skew_a3a_sn'	204	'nasal_kurt_channel_b_lab'	234	'temporal_std_channel_b_rgb'		
175	'skew_v2p_sn'	205	'nasal_kurt_channel_r'	235	'temporal_kurt_channel_h'		
176	'skew_v6p_sn'	206	'nasal_kurt_channel_g'	236	'temporal_kurt_channel_s'		
177	'skew_locs_fp (s)'	207	'nasal_kurt_channel_b_rgb'	237	'temporal_kurt_channel_v'		
178	'skew_cp'	208	'nasal_skew_channel_h'	238	'temporal_kurt_channel_l'		
179	'skew_ibi'	209	'nasal_skew_channel_s'	239	'temporal_kurt_channel_a'		
180	'skew_dur'	210	'nasal_skew_channel_v'	240	'temporal_kurt_channel_b_lab'		

Table 4-4: Pearson correlation characteristics when separating the register with and without contact lenses (1/2).

1	't0p (s)'	31	'half_width_left (ms)'	61	'std_t7p (s)'	91	'std_a2a_sn'	121	'kur_J02'
2	't1p (s)'	32	'half_width_right (ms)'	62	'std_t8p (s)'	92	'std_a3a_sn'	122	'kur_J26'
3	't2p (s)'	33	'half_width (ms)'	63	'std_P1'	93	'std_v2p_sn'	123	'kur_J68'
4	't3p (s)'	34	'width_left (ms)'	64	'std_P3'	94	'std_v6p_sn'	124	'kur_v6p'
5	't4p (s)'	35	'width_right (ms)'	65	'std_P5'	95	'std_locs_fp (s)'	125	'kur_S'
6	't5p (s)'	36	'width (ms)'	66	'std_P7'	96	'std_cp'	126	'kur_tiempo_blink (ms)'
7	't6p (s)'	37	'a1a_sn'	67	'std_W02'	97	'std_ibi'	127	'kur_half_width_left (ms)'
8	't7p (s)'	38	'a2a_sn'	68	'std_W24'	98	'std_dur'	128	'kur_half_width_right (ms)'
9	't8p (s)'	39	'a3a_sn'	69	'std_W46'	99	'std_alfa_mm'	129	'kur_half_width (ms)'
10	'P1'	40	'v2p_sn'	70	'std_W68'	100	'std_beta_mm'	130	'kur_width_left (ms)'
11	'P3'	41	'v6p_sn'	71	'std_t1a (s)'	101	'kur_t1p (s)'	131	'kur_width_right (ms)'
12	'P5'	42	'locs_fp (s)'	72	'std_t2a (s)'	102	'kur_t2p (s)'	132	'kur_width (ms)'
13	'P7'	43	'm_cp'	73	'std_t3a (s)'	103	'kur_t3p (s)'	133	'kur_a1a_sn'
14	'W02'	44	'm_ibi'	74	'std_a1a (s)'	104	'kur_t4p (s)'	134	'kur_a3a_sn'
15	'W24'	45	'm_dur'	75	'std_a2a (s)'	105	'kur_t5p (s)'	135	'kur_v2p_sn'
16	'W46'	46	'm_cp_complete'	76	'std_a3a (s)'	106	'kur_t6p (s)'	136	'kur_v6p_sn'
17	'W68'	47	'm_dur_complete'	77	'std_J02'	107	'kur_t7p (s)'	137	'kur_locs_fp (s)'
18	't1a (s)'	48	'm_cp_incomplete'	78	'std_J26'	108	'kur_t8p (s)'	138	'kurt_cp'
19	't2a (s)'	49	'alpha_mm'	79	'std_J68'	109	'kur_P1'	139	'kurt_ibi'
20	't3a (s)'	50	'b_mm'	80	'std_v2p'	110	'kur_P5'	140	'kurt_dur'
21	'a1a (s)'	51	'num'	81	'std_v6p'	111	'kur_P7'	141	'skew_t1p (s)'
22	'a2a (s)'	52	'num_complet'	82	'std_S'	112	'kur_W02'	142	'skew_t2p (s)'
23	'a3a (s)'	53	'num_incomplet'	83	'std_tiempo_blink (ms)'	113	'kur_W24'	143	'skew_t3p (s)'
24	'J02'	54	'std_t0p (s)'	84	'std_half_width_left (ms)'	114	'kur_W46'	144	'skew_t4p (s)'
25	'J26'	55	'std_t1p (s)'	85	'std_half_width_right (ms)'	115	'kur_W68'	145	'skew_t5p (s)'
26	'J68'	56	'std_t2p (s)'	86	'std_half_width (ms)'	116	'kur_t1a (s)'	146	'skew_t6p (s)'
27	'v2p'	57	'std_t3p (s)'	87	'std_width_left (ms)'	117	'kur_t2a (s)'	147	'skew_t7p (s)'
28	'v6p'	58	'std_t4p (s)'	88	'std_width_right (ms)'	118	'kur_t3a (s)'	148	'skew_t8p (s)'
29	'S'	59	'std_t5p (s)'	89	'std_width (ms)'	119	'kur_a1a (s)'	149	'skew_P1'
30	'tiempo_blink (ms)'	60	'std_t6p (s)'	90	'std_a1a_sn'	120	'kur_a3a (s)'	150	'skew_P5'

Table 4-5: Pearson correlation characteristics when separating the register with and without contact lenses (2/2).

151	'skew_P7'	181	'channel_h'	211	'nasal_skew_channel_l'	241	'temporal_kurt_channel_r'
152	'skew_W02'	182	'channel_s'	212	'nasal_skew_channel_a'	242	'temporal_kurt_channel_g'
153	'skew_W24'	183	'channel_v'	213	'nasal_skew_channel_b_lab'	243	'temporal_kurt_channel_b_rgb'
154	'skew_W46'	184	'channel_l'	214	'nasal_skew_channel_r'	244	'temporal_skew_channel_h'
155	'skew_W68'	185	'channel_a'	215	'nasal_skew_channel_g'	245	'temporal_skew_channel_s'
156	'skew_t1a (s)'	186	'channel_b_lab'	216	'nasal_skew_channel_b_rgb'	246	'temporal_skew_channel_v'
157	'skew_t2a (s)'	187	'channel_r'	217	'temporal_mean_channel_h'	247	'temporal_skew_channel_l'
158	'skew_t3a (s)'	188	'channel_g'	218	'temporal_mean_channel_s'	248	'temporal_skew_channel_a'
159	'skew_a1a (s)'	189	'channel_b_rgb'	219	'temporal_mean_channel_v'	249	'temporal_skew_channel_b_lab'
160	'skew_a3a (s)'	190	'nasal_std_channel_h'	220	'temporal_mean_channel_l'		
161	'skew_J02'	191	'nasal_std_channel_s'	221	'temporal_mean_channel_a'		
162	'skew_J26'	192	'nasal_std_channel_v'	222	'temporal_mean_channel_b_lab'		
163	'skew_J68'	193	'nasal_std_channel_l'	223	'temporal_mean_channel_r'		
164	'skew_v6p'	194	'nasal_std_channel_a'	224	'temporal_mean_channel_g'		
165	'skew_S'	195	'nasal_std_channel_b_lab'	225	'temporal_mean_channel_b_rgb'		
166	'skew_tiempo_blink (ms)'	196	'nasal_std_channel_r'	226	'temporal_std_channel_h'		
167	'skew_half_width_left (ms)'	197	'nasal_std_channel_g'	227	'temporal_std_channel_s'		
168	'skew_half_width_right (ms)'	198	'nasal_std_channel_b_rgb'	228	'temporal_std_channel_v'		
169	'skew_half_width (ms)'	199	'nasal_kurt_channel_h'	229	'temporal_std_channel_l'		
170	'skew_width_left (ms)'	200	'nasal_kurt_channel_s'	230	'temporal_std_channel_a'		
171	'skew_width_right (ms)'	201	'nasal_kurt_channel_v'	231	'temporal_std_channel_b_lab'		
172	'skew_width (ms)'	202	'nasal_kurt_channel_l'	232	'temporal_std_channel_r'		
173	'skew_a1a_sn'	203	'nasal_kurt_channel_a'	233	'temporal_std_channel_g'		
174	'skew_a3a_sn'	204	'nasal_kurt_channel_b_lab'	234	'temporal_std_channel_b_rgb'		
175	'skew_v2p_sn'	205	'nasal_kurt_channel_r'	235	'temporal_kurt_channel_h'		
176	'skew_v6p_sn'	206	'nasal_kurt_channel_g'	236	'temporal_kurt_channel_s'		
177	'skew_locs_fp (s)'	207	'nasal_kurt_channel_b_rgb'	237	'temporal_kurt_channel_v'		
178	'skew_cp'	208	'nasal_skew_channel_h'	238	'temporal_kurt_channel_l'		
179	'skew_ibi'	209	'nasal_skew_channel_s'	239	'temporal_kurt_channel_a'		
180	'skew_dur'	210	'nasal_skew_channel_v'	240	'temporal_kurt_channel_b_lab'		

Table 4-6: Pearson correlation characteristics when analyzing weather and air conditions (1/2).

1	't0p (s)'	31	'half_width_left (ms)'	61	'std_P1'
2	't1p (s)'	32	'half_width_right (ms)'	62	'std_P3'
3	't2p (s)'	33	'half_width (ms)'	63	'std_P5'
4	't3p (s)'	34	'width_left (ms)'	64	'std_P7'
5	't4p (s)'	35	'width_right (ms)'	65	'std_W02'
6	't5p (s)'	36	'width (ms)'	66	'std_W24'
7	't6p (s)'	37	'a1a_sn (mm/s^2)'	67	'std_W46'
8	't7p (s)'	38	'a2a_sn (mm/s^2)'	68	'std_W68'
9	't8p (s)'	39	'a3a_sn (mm/s^2)'	69	'std_t1a (s)'
10	'P1'	40	'v2p_sn (mm/s)'	70	'std_t2a (s)'
11	'P3'	41	'v6p_sn (mm/s)'	71	'std_t3a (s)'
12	'P5'	42	'locs_fp (s)'	72	'std_a1a (s)'
13	'P7'	43	'm_cp'	73	'std_a2a (s)'
14	'W02'	44	'm_ibi'	74	'std_a3a (s)'
15	'W24'	45	'm_cp_complete'	75	'std_J02'
16	'W46'	46	'm_cp_incomplete'	76	'std_J26'
17	'W68'	47	'alpha_mm'	77	'std_J68'
18	't1a (s)'	48	'b_mm'	78	'std_v2p'
19	't2a (s)'	49	'num'	79	'std_v6p'
20	't3a (s)'	50	'num_complete'	80	'std_S'
21	'a1a (-)'	51	'num_incomplete'	81	'std_tiempo_blink (ms)'
22	'a2a (-)'	52	'std_t0p (s)'	82	'std_half_width_left (ms)'
23	'a3a (-)'	53	'std_t1p (s)'	83	'std_half_width_right (ms)'
24	'J02'	54	'std_t2p (s)'	84	'std_half_width (ms)'
25	'J26'	55	'std_t3p (s)'	85	'std_width_left (ms)'
26	'J68'	56	'std_t4p (s)'	86	'std_width_right (ms)'
27	'v2p'	57	'std_t5p (s)'	87	'std_width (ms)'
28	'v6p'	58	'std_t6p (s)'	88	'std_a1a_sn'
29	'S'	59	'std_t7p (s)'	89	'std_a2a_sn'
30	'tiempo_blink (ms)'	60	'std_t8p (s)'	90	'std_a3a_sn'

Table 4-7: Pearson correlation characteristics when analyzing weather and air conditions (2/2).

91	'std_v2p_sn'	121	'nasal_kurt_channel_s'	151	'temporal_std_channel_a'
92	'std_v6p_sn'	122	'nasal_kurt_channel_v'	152	'temporal_std_channel_b_lab'
93	'std_locs_fp (s)'	123	'nasal_kurt_channel_l'	153	'temporal_std_channel_r'
94	'std_cp'	124	'nasal_kurt_channel_a'	154	'temporal_std_channel_g'
95	'std_ibi'	125	'nasal_kurt_channel_b_lab'	155	'temporal_std_channel_b_rgb'
96	'std_alfa_mm'	126	'nasal_kurt_channel_r'	156	'temporal_kurt_channel_h'
97	'std_beta_mm'	127	'nasal_kurt_channel_g'	157	'temporal_kurt_channel_s'
98	'kur_locs_fp (s)'	128	'nasal_kurt_channel_b_rgb'	158	'temporal_kurt_channel_v'
99	'kurt_ibi'	129	'nasal_skew_channel_h'	159	'temporal_kurt_channel_l'
100	'skew_locs_fp (s)'	130	'nasal_skew_channel_s'	160	'temporal_kurt_channel_a'
101	'skew_ibi'	131	'nasal_skew_channel_v'	161	'temporal_kurt_channel_b_lab'
102	'channel_h'	132	'nasal_skew_channel_l'	162	'temporal_kurt_channel_r'
103	'channel_s'	133	'nasal_skew_channel_a'	163	'temporal_kurt_channel_g'
104	'channel_v'	134	'nasal_skew_channel_b_lab'	164	'temporal_kurt_channel_b_rgb'
105	'channel_l'	135	'nasal_skew_channel_r'	165	'temporal_skew_channel_h'
106	'channel_a'	136	'nasal_skew_channel_g'	166	'temporal_skew_channel_s'
107	'channel_b_lab'	137	'nasal_skew_channel_b_rgb'	167	'temporal_skew_channel_v'
108	'channel_r'	138	'temporal_mean_channel_h'	168	'temporal_skew_channel_l'
109	'channel_g'	139	'temporal_mean_channel_s'	169	'temporal_skew_channel_a'
110	'channel_b_rgb'	140	'temporal_mean_channel_v'	170	'temporal_skew_channel_b_lab'
111	'nasal_std_channel_h'	141	'temporal_mean_channel_l'	171	'temporal_skew_channel_r'
112	'nasal_std_channel_s'	142	'temporal_mean_channel_a'	172	'temporal_skew_channel_g'
113	'nasal_std_channel_v'	143	'temporal_mean_channel_b_lab'	173	'temporal_skew_channel_b_rgb'
114	'nasal_std_channel_l'	144	'temporal_mean_channel_r'	174	'no2'
115	'nasal_std_channel_a'	145	'temporal_mean_channel_g'	175	'o3'
116	'nasal_std_channel_b_lab'	146	'temporal_mean_channel_b_rgb'	176	'ta'
117	'nasal_std_channel_r'	147	'temporal_std_channel_h'	177	'vmax'
118	'nasal_std_channel_g'	148	'temporal_std_channel_s'	178	'vv'
119	'nasal_std_channel_b_rgb'	149	'temporal_std_channel_v'	179	'dv'
120	'nasal_kurt_channel_h'	150	'temporal_std_channel_l'	180	'hr'

4.2 **Second study**: Calculating the differences between the use and non-use of contact lenses. Kolmogorov-Smirnov and Wilcoxon test.

One of the most interesting uses of the compiled database is to be able to study which features present variations in the case of using contact lenses or not. To do this, the differences between the features for patients with contact lenses and without them are studied. The study focuses on selecting only the features extracted from the eye video (blinking dynamics and color variables). The number of patients selected is 12 patients each with between 6 and 10 videos each with a lens and the same number without a lens.

Each selected feature is analyzed, forming two groups for each patient: one group with the features of that patient without a lens and another group with the same features but with a lens. Two types of statistical tests implemented through the MATLAB library on "AI, Data Science and Statistics" are applied to these two groups (for each feature).

- Wilcoxon test (191): Tests the null hypothesis ($h=0$, $p>0.05$) that the two data groups have the same median regardless of the continuous probability distributions that follow (non-parametric statistical test). If $p<0.05$, the test is not passed, and the null hypothesis is refuted ($h=1$). The MATLAB function used for this test is the "ranksum" function. The fact that it is a non-parametric test is helpful since the probability distribution function that the features follow is not known a priori.
- Kolmogorv-Smirnov test (192): It is a non-parametric statistical hypothesis contrast test. Tests the null hypothesis ($h=0$, $p>0.05$) that the two groups of data come from the same continuous probability distribution against the hypothesis that they do not meet this condition ($h=1$, $p<0.05$).

These two tests provide complementary information since two features could have similar mean or median values with and without lenses but differ in higher-order statistics such as standard deviation, skewness, kurtosis, etc. That is, the use of the contact lens influences the variability of the variable more than the difference in its value when going from not using a contact lens to using it.

On the other hand, these analyses are performed for each patient separately but can then be grouped to give mean values and standard deviations of the results when averaged over the different patients.

The results of the Wilcoxon test on the equality of the medians of the characteristics with and without contact lenses allow us to separate the group of patients into two. A first group in which at least some characteristic changes in a statistically significant way (group 1) from the rest of patients where no significant variation occurs. Group 1 is made up of 50% of all patients. Within it, 50% are under thirty years old and the remaining 50% are between 45 and 60 years old.

To better analyze these results, group 1 of patients in whom statistically significant variations in their characteristics occur when wearing contact lenses is studied in more depth. Specifically, the mean value of the h statistic and the mean value and standard deviation of the p statistic are studied. The results appear in Figure 4-13, Figure 4-14, and Figure 4-15.

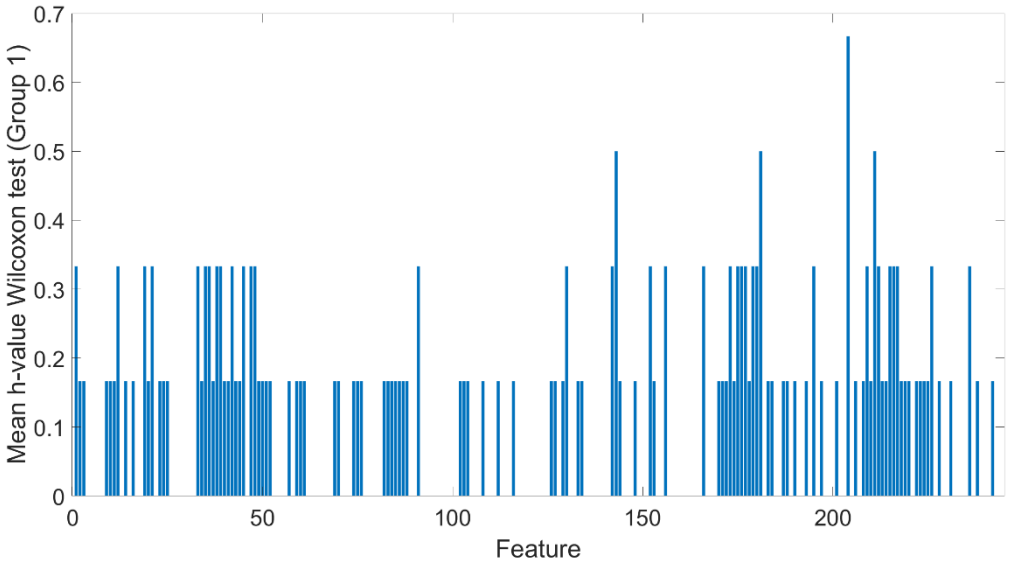

Figure 4-13: Mean h-value from Wilcoxon test considering patients of group 1, where at least one parameter h value is 1.

As seen in Figure 4-13, the features in Group 1 can be classified into three subgroups. The first group, where they change in almost all the patients in the group, is another group of medium features, and finally, a group where the change only occurs in some. Among the first would be the features number 143 (Skewness of the normalized blink power 7, Table 4-8 and Table 4-9), 181 (half color channel B, RGB color system, nasal zone), 204 (skewness channel A, CIELAB1931, nasal zone), and 211 (half channel V, HSV color, temporal zone). The first of these features corresponds to the asymmetry (between the different blinks of the same patient) in the probability distribution of the blink opening power (it is found that in three of them, it increases, and in three, it decreases). The rest of the features have to do with alterations in the

color of the eye, which I agree with friction processes and possible inflammation caused by contact lens use (12–14,92,93).

Figure 4-14 represents the same mean value of the h statistic of group 1, but now, together with the standard deviation of these values. If we consider the standard deviation, there is a high probability that there are more features where the changes can become significant. Obviously, the previous ones continue to appear, but now we could consider more change features associated with color (features 173, 175-180, 195, 209, 212, 215, 216-217, 226, see table 11) and flicker dynamics (features 1, 12, 19, 21, 33, 35-36, 38-39, 42, 45, 47-48, 91,130, 142-143, 152, 156, 166, see table 10). Within the latter are the variables associated with the speed and acceleration of the "Downstroke" (eye closure) and the "Upstroke" (eye-opening). Also noteworthy are those related to the degree of closure of the blink, the interval between them (especially for complete blinks), the number of blinks, and the parameter α that governs the dynamics of the PSD of the blink. Compared with the previous graph Figure 4-13, it is observed that these variations would occur in fewer patients, the most important being those that have been named previously (changes in color and eye-opening power, as a summary).

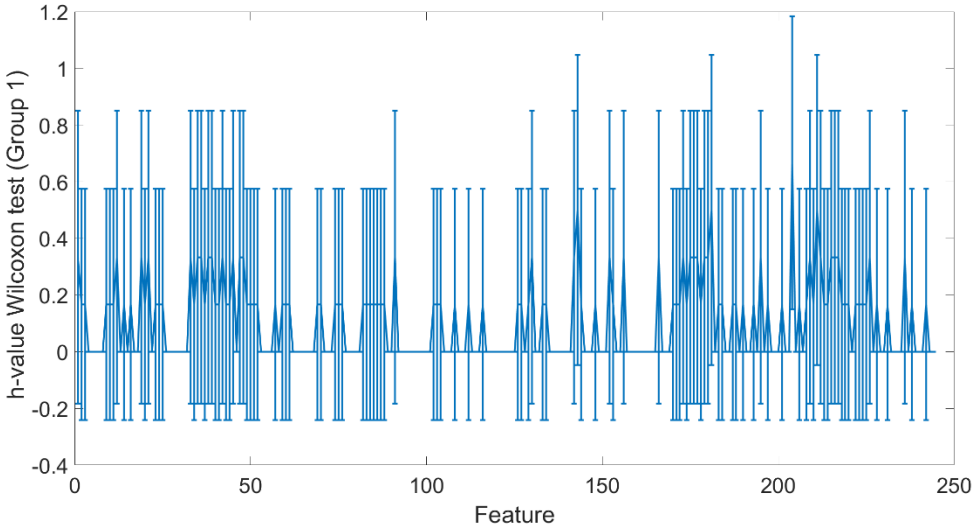

Figure 4-14: Mean h-value (and standard deviation) from Wilcoxon test considering patients of group 1, where at least one parameter "h" value is 1.

As a way of supporting the above results, Figure 4-15 shows the results for the mean value of the p statistic for the features of patient group 1, along with its standard deviation in this patient group. As can be seen, again, considering the standard deviation, those we have seen previously could appear as change features.

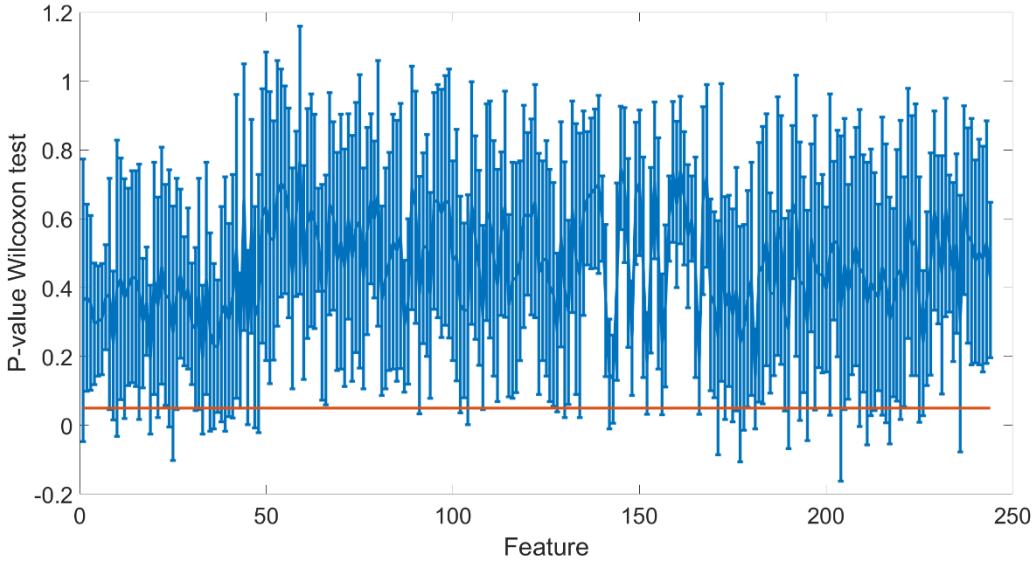

Figure 4-15: mean probability value and its standard deviation of the Wilcoxon test, considering all patients where at least one characteristic has been rejected, the null hypothesis. The red line marks the threshold of p<0.05.

The previous results show that the selection of features that may present changes when wearing contact lenses may be affected by the sample size. The sample size is understood not as the number of videos analyzed but as the number of patients who will be part of the study. In this sense, it is essential to remember that the standard deviations and means of the statistical parameters h and p of the previous graphs (indicate what graphs they are) are made on a small number of patients. Hence, it is interesting to explore the data trend when it can be analyzed with more videos of patients participating in the App's data collection.

To estimate this situation, bootstrap statistical techniques are used below. The Bootstrap resampling technique consists of choosing random samples with replacements from a data set and analyzing each sample similarly. This technique allows the estimation of the sampling distribution of almost any statistic using random sampling methods (mean, deviation, variance, among others) (193). In this case, the mean value is estimated for each feature using 200 samples to check if these new distributions can be considered the same medians of both distributions are considered similar (Wilcoxon test). For the bootstrap techniques, the Statistics and Machine Learning Toolbox has been used, through the bootstrp function.

As expected, more patients show differences between the use or not of contact lenses, going from 50% to 67% (new group 1). Within the group of patients who have changed, the group of presbyopes goes from 50% of this group to 62%. This shows that increasing the precision of the estimates (which can also be done by taking more videos) allows us to detect more

significant differences in presbyopic people compared to non-presbyopic people. This also justifies the data collection developments presented in the present thesis since presbyopic patients seem to have unique characteristics that make them more sensitive to contact lenses' changes in the eye.

The new results for the statistical parameters p of the Wilcoxon test on this new group 1 of patients that show differences in their features by increasing the precision of the bootstrap estimates appear in

Figure 4-16 and Figure 4-17. As seen in the results, the increase in the estimation's precision means that almost all the features now show differences. However, the features previously detected as significantly different and relevant continue to appear with a much lower p-value than the others.

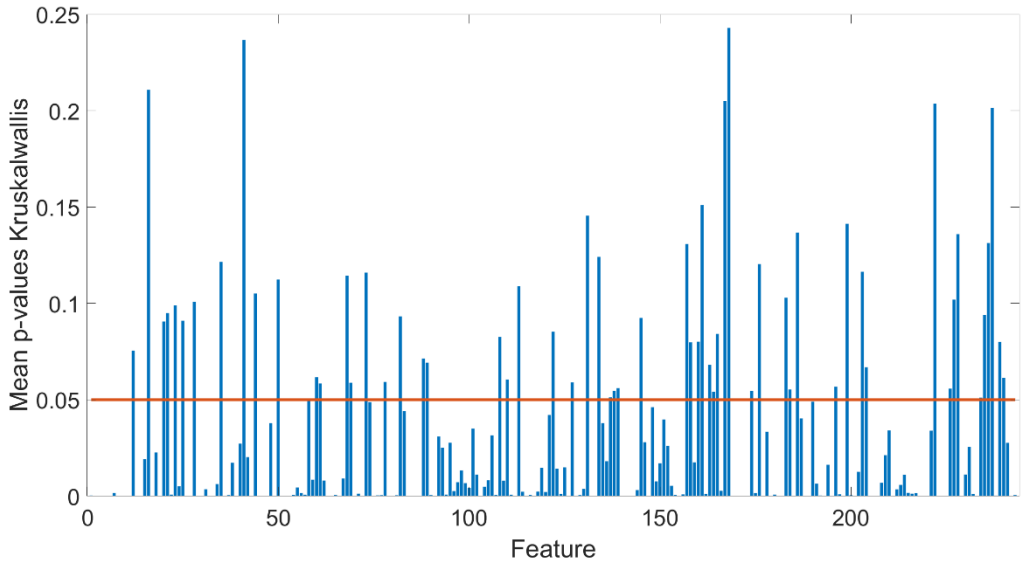

Figure 4-16: Mean value of the p statistic for comparison between features in the group of patients with some statistically significant feature (h=1). The red line marks the threshold value of statistical significance (p<0.05).

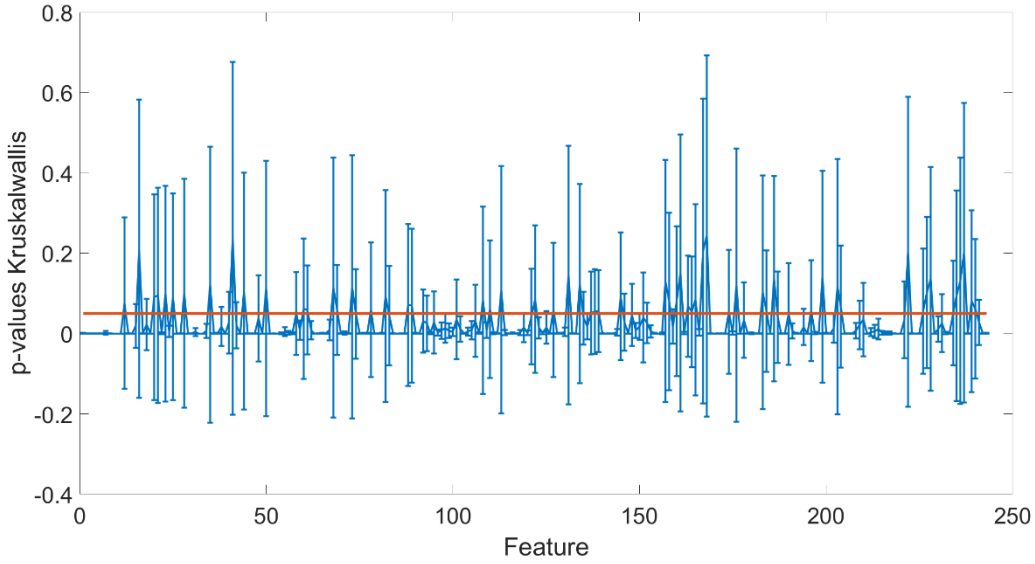

Figure 4-17: Mean value of the p statistic and its standard deviation for comparison between features in the group of patients with some statistically significant feature (h=1). The red line marks the threshold value of statistical significance ($p<0.05$).

Table 4-8: List of study 2 parameters, Wilcoxon test (1/2).

1	t1p (s)	31	width_right (ms)	61	std_P7	91	std_alfa_mm	121	kur_half_width (ms)
2	t2p (s)	32	width (ms)	62	std_W02	92	std_beta_mm	122	kur_width_left (ms)
3	t3p (s)	33	a1a_sn	63	std_W24	93	kur_t1p (s)	123	kur_width_right (ms)
4	t4p (s)	34	a3a_sn	64	std_W46	94	kur_t2p (s)	124	kur_width (ms)
5	t5p (s)	35	v2p_sn	65	std_W68	95	kur_t3p (s)	125	kur_a1a_sn
6	t6p (s)	36	v6p_sn	66	std_t1a (s)	96	kur_t4p (s)	126	kur_a3a_sn
7	t7p (s)	37	locs_fp (s)	67	std_t2a (s)	97	kur_t5p (s)	127	kur_v2p_sn
8	t8p (s)	38	m_cp	68	std_t3a (s)	98	kur_t6p (s)	128	kur_v6p_sn
9	P1	39	m_ibi	69	std_a1a (s)	99	kur_t7p (s)	129	kur_locs_fp (s)
10	P5	40	m_dur	70	std_a3a (s)	100	kur_t8p (s)	130	kurt_cp
11	P7	41	m_cp_complete	71	std_J02	101	kur_P1	131	kurt_ibi
12	W02	42	m_ibi_complete	72	std_J26	102	kur_P5	132	kurt_dur
13	W24	43	m_dur_complete	73	std_J68	103	kur_P7	133	skew_t1p (s)
14	W46	44	m_cp_incomplete	74	std_v6p	104	kur_W02	134	skew_t2p (s)
15	W68	45	alpha_mm	75	std_S	105	kur_W24	135	skew_t3p (s)
16	t1a (s)	46	b_mm	76	std_tiempo_blink (ms)	106	kur_W46	136	skew_t4p (s)
17	t2a (s)	47	num	77	std_half_width_left (ms)	107	kur_W68	137	skew_t5p (s)
18	t3a (s)	48	num_complet	78	std_half_width_right (ms)	108	kur_t1a (s)	138	skew_t6p (s)
19	a1a (s)	49	num_incomplet	79	std_half_width (ms)	109	kur_t2a (s)	139	skew_t7p (s)
20	a3a (s)	50	std_t1p (s)	80	std_width_left (ms)	110	kur_t3a (s)	140	skew_t8p (s)
21	J02	51	std_t2p (s)	81	std_width_right (ms)	111	kur_a1a (s)	141	skew_P1
22	J26	52	std_t3p (s)	82	std_width (ms)	112	kur_a3a (s)	142	skew_P5
23	J68	53	std_t4p (s)	83	std_a1a_sn	113	kur_J02	143	skew_P7
24	v6p	54	std_t5p (s)	84	std_a3a_sn	114	kur_J26	144	skew_W02
25	S	55	std_t6p (s)	85	std_v2p_sn	115	kur_J68	145	skew_W24
26	tiempo_blink (ms)	56	std_t7p (s)	86	std_v6p_sn	116	kur_v6p	146	skew_W46
27	half_width_left (ms)	57	std_t8p (s)	87	std_locs_fp (s)	117	kur_S	147	skew_W68
28	half_width_right (ms)	58	std_P1	88	std_cp	118	kur_tiempo_blink (ms)	148	skew_t1a (s)
29	half_width (ms)	59	std_P3	89	std_ibi	119	kur_half_width_left (ms)	149	skew_t2a (s)
30	width_left (ms)	60	std_P5	90	std_dur	120	kur_half_width_right (ms)	150	skew_t3a (s)

Table 4-9: List of study 2 parameters, Wilcoxon test (2/2).

151	skew_a1a (s)	181	channel_b_rgb	211	temporal_mean_channel_v	241	temporal_skew_channel_b_lab
152	skew_a3a (s)	182	nasal_std_channel_h	212	temporal_mean_channel_l	242	temporal_skew_channel_r
153	skew_J02	183	nasal_std_channel_s	213	temporal_mean_channel_a	243	temporal_skew_channel_g
154	skew_J26	184	nasal_std_channel_v	214	temporal_mean_channel_b_lab	244	temporal_skew_channel_b_rgb
155	skew_J68	185	nasal_std_channel_l	215	temporal_mean_channel_r		
156	skew_v6p	186	nasal_std_channel_a	216	temporal_mean_channel_g		
157	skew_S	187	nasal_std_channel_b_lab	217	temporal_mean_channel_b_rgb		
158	skew_tiempo_blink (ms)	188	nasal_std_channel_r	218	temporal_std_channel_h		
159	skew_half_width_left (ms)	189	nasal_std_channel_g	219	temporal_std_channel_s		
160	skew_half_width_right (ms)	190	nasal_std_channel_b_rgb	220	temporal_std_channel_v		
161	skew_half_width (ms)	191	nasal_kurt_channel_h	221	temporal_std_channel_l		
162	skew_width_left (ms)	192	nasal_kurt_channel_s	222	temporal_std_channel_a		
163	skew_width_right (ms)	193	nasal_kurt_channel_v	223	temporal_std_channel_b_lab		
164	skew_width (ms)	194	nasal_kurt_channel_l	224	temporal_std_channel_r		
165	skew_a1a_sn	195	nasal_kurt_channel_a	225	temporal_std_channel_g		
166	skew_a3a_sn	196	nasal_kurt_channel_b_lab	226	temporal_std_channel_b_rgb		
167	skew_v2p_sn	197	nasal_kurt_channel_r	227	temporal_kurt_channel_h		
168	skew_v6p_sn	198	nasal_kurt_channel_g	228	temporal_kurt_channel_s		
169	skew_locs_fp (s)	199	nasal_kurt_channel_b_rgb	229	temporal_kurt_channel_v		
170	skew_cp	200	nasal_skew_channel_h	230	temporal_kurt_channel_l		
171	skew_ibi	201	nasal_skew_channel_s	231	temporal_kurt_channel_a		
172	skew_dur	202	nasal_skew_channel_v	232	temporal_kurt_channel_b_lab		
173	channel_h	203	nasal_skew_channel_l	233	temporal_kurt_channel_r		
174	channel_s	204	nasal_skew_channel_a	234	temporal_kurt_channel_g		
175	channel_v	205	nasal_skew_channel_b_lab	235	temporal_kurt_channel_b_rgb		
176	channel_l	206	nasal_skew_channel_r	236	temporal_skew_channel_h		
177	channel_a	207	nasal_skew_channel_g	237	temporal_skew_channel_s		
178	channel_b_lab	208	nasal_skew_channel_b_rgb	238	temporal_skew_channel_v		
179	channel_r	209	temporal_mean_channel_h	239	temporal_skew_channel_l		
180	channel_g	210	temporal_mean_channel_s	240	temporal_skew_channel_a		

4.3 **Third study:** Mahalanobis distance and clustering analysis (194).

This third study focuses on studying the variability of the data or "features" mainly between the groups of videos with or without contact lenses. A metric known as Mahalanobis distance (194) is associated with each video (record) to study this variability. This metric is performed through the following steps:

- Each record $R_\epsilon, \epsilon = 1, \dots, N$, is characterized by a feature vector given by $R\_\epsilon = [f\_1, f\_2, \dots., f\_M\ ], i = 1, \dots M$, with N being the number of records (videos) and M the number of features analyzed.
- We can take the features f_i as the variables to analyze and each of the R_ϵ records as a "multidimensional observation" of them.
- From the set of observations, the covariance matrix between the features $Sf_{i,j}$ can be calculated as:

$$Sf_{ij} = \langle (R_\epsilon - \bar{R})^T (R_\epsilon - \bar{R}) \rangle_\epsilon \quad (4\text{-}1)$$

  where the average is calculated across all records and $\bar{R} = \left[(\overline{f_1}), (\overline{f_2}), \dots., (\overline{f_M})\right]$ is the vector of the average of each of the features.
- The Mahalanobis distance of any record to this mean value is calculated as (194):

$$D^2(\epsilon) = \left((R_\epsilon - \bar{R})\right) Sf^{-1}((R_\epsilon - \bar{R})^T) \quad (4\text{-}2)$$

  where D is the Mahalanonis distance. This distance measures the Euclidean distance of the selected vector to the mean vector, but in units of the "noise" (standard deviation) of the data in the direction that joins the mean vector with the considered vector.

This distance is commonly used in statistical classification and detecting anomalous events and outliers (165,195,196).

One of the possible drawbacks when using this type of distance to analyze vectors with a large number of features is that the covariance matrix may be "bad conditioned". In this case, the range of the matrix is smaller than the number of features, and the inverse calculation leads to errors because the matrix has eigenvalues very close to zero. Some algorithms that use the inversion of covariance matrices (such as the principal component decomposition algorithm implemented in MATLAB) increase this precision by calculating an inverse through the SVD (Singular value decomposition) of the covariance matrix (197–199). This decomposition allows us to know each matrix A as decomposed in the form $A = UEV^T$, where U and V are unitary matrices, and E is a diagonal rectangular matrix (in whose diagonal the eigenvalues of

matrix A are arranged). It is precisely the smallest eigenvalue close to zero that we can suppress from the expression for E, if we know the range N_r of the matrix. In this way, a new matrix A' is generated consisting only of the eigenvalues from 1 to N_r, with the rest being zero. This new matrix is well conditioned, and its inverse can be taken.

Since the number of variables ("features") analyzed is very large, the previous method is used when the number of videos is less than the number of variables. This same algorithm is implemented in, for example, the principal component decomposition in MATLAB (PCA function).

This metric distance is first used to characterize what is the distance of each of the videos (records) without contact lenses to the mean vector without contact lenses $\bar{R}_{wCL} = \left[\overline{f_1}, \overline{f_2}, \ldots., \overline{f_M}\right]_{wCL}$. To do this, follow the steps outlined above. Next, a threshold value of this distance $D_{th_{wCL}}$ is chosen as the distance for which the probability of appearance is greater than 95%, which marks the point from which a distance would be anomalous.

With this data, the distance of each of the records with contact lenses is calculated to the group of records without contact lenses as:

Since the number of variables ("features") analyzed is very large, the previous method is used when the number of videos is less than the number of variables. This same algorithm is implemented in, for example, the decomposition of principal components in MATLAB (function PCA).

This metric distance is first used to characterize the distance of each of the videos (records) without contact lenses to the mean vector without contact lenses

$$\frac{D(\epsilon)}{D_{th_{wCL}}} = \frac{(R_\epsilon - \bar{R}_{wCL}) Sf^{-1}((R_\epsilon - \bar{R}_{wCL})^T)^{\frac{1}{2}}}{D_{th_{wCL}}} \tag{4-3}$$

Thus, values of the above equation (4-3) greater than 1 mean that the registration with contact lenses would be considered anomalous in the contact lens group. The values of this distance, as well as the probability distribution function, appear in figures Figure 4-18 and Figure 4-19.

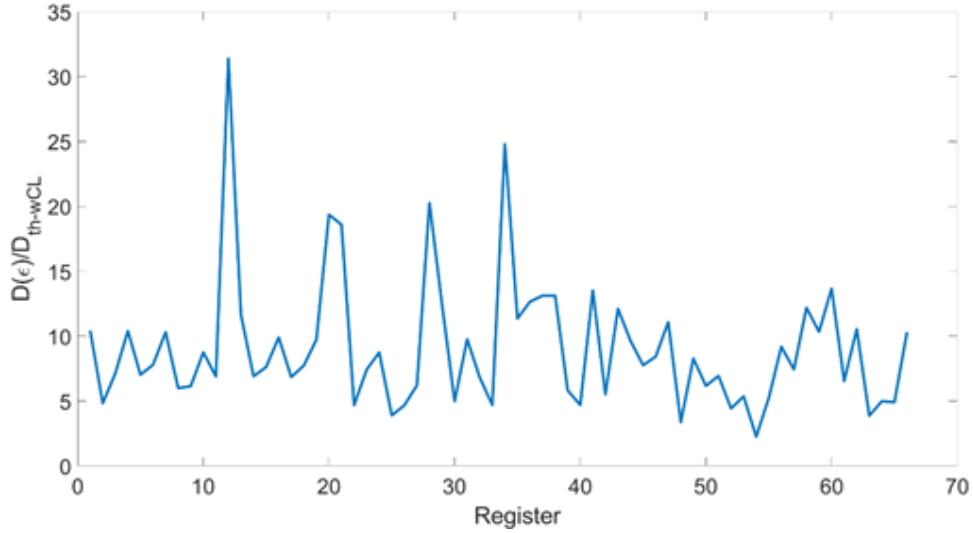

Figure 4-18:Mahalonobis distance values from each of the videos (records) with contact lenses, to the group of videos without contact lenses, normalized by the value of the distance admitted in the group without contact lenses.

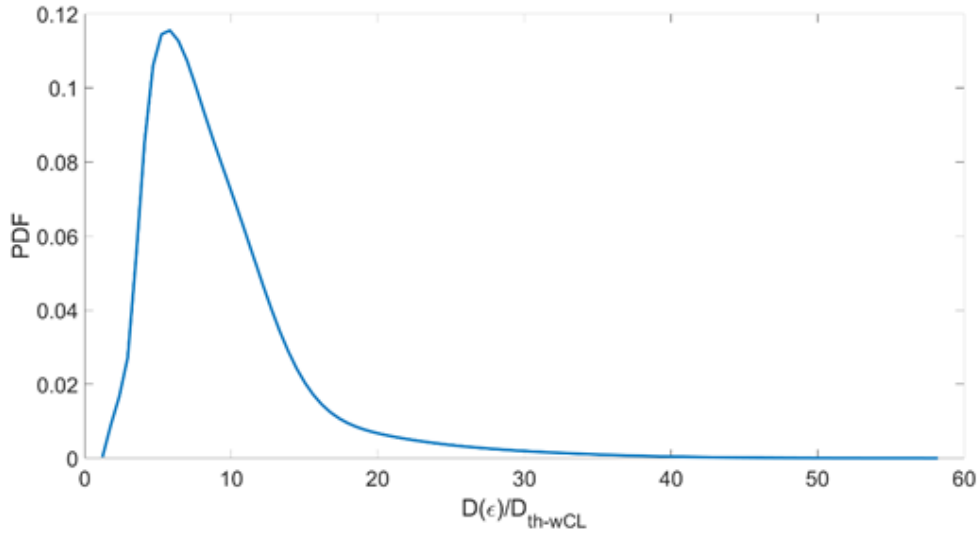

Figure 4-19: Probability distribution of the normalized Mahalanobis distance from the records with contact lenses to the group without contact lenses.

The above results show that all contact lens records have a normalized Mahalanobis distance greater than one. This means that they are distinguishable from the non-contact lens group. Furthermore, the probability distribution representation of the normalized Mahalanobis distance $((D(\epsilon)) \ / \ (D_{thwCL}))$ indicates an asymmetry of the distribution with long tails towards large distances. To further analyze this variability in the probability distribution, a clustering analysis is performed in the contact and non-contact lens groups.

To corroborate the results obtained and determine whether it is possible to identify clusters in the contact lens group, we proceeded to check using clustering algorithms.

For this purpose, the MATLAB script included in the Toolbox "AI Data Science and Statistic" in the live script "Cluster Data" is used. The input to this algorithm is a matrix with the observations (records) in rows and the features in columns. With this input configuration, the video records are classified into groups. Then, the type of clustering algorithm is selected (K-means or Hierarchical clustering, see introduction section 2.4.3), the type of distance or difference measure between records to be used, the range of the number of clusters in which the algorithm will try to optimize, the number of iterations of the algorithm and the starting point of the clustering. The default options of the algorithm have been used in the range of clusters to analyze (between 2 and 5), the number of iterations (100) and the starting point of the search ("plus" method of searching for the centroid of the data as a starting point). The K-means clustering method has been used for the rest of the parameters because it is the most commonly used in the literature (200–203). For the calculation of the distance, correlation-based distances have not been used (which would be the closest to the Mahalanobis distance above, which does take it into account), since this MATLAB algorithm does not have built-in the calculation of inverses of matrices with a number of observations less than the number of variables. The closest thing is, then, to use Euclidean distances. To see the optimal number of clusters, the Calinski-Harabasz index criterion is selected in the algorithm (204) which compares the weighted distances between clusters with the internal distances between members of the same cluster.

To provide more information about the constitution of the clusters, the algorithm performs a principal component decomposition (205) of the data. It chooses the first two principal components (those that represent the majority of the variance of the data) and represents in a "scatter plot" the part of each of the records in those components, but classified by the clusters found, together with an "x" that represents the centroid of each cluster.

With these parameters the results appear in figures Figure 4-20, Figure 4-21, Figure 4-22 and Figure 4-23.

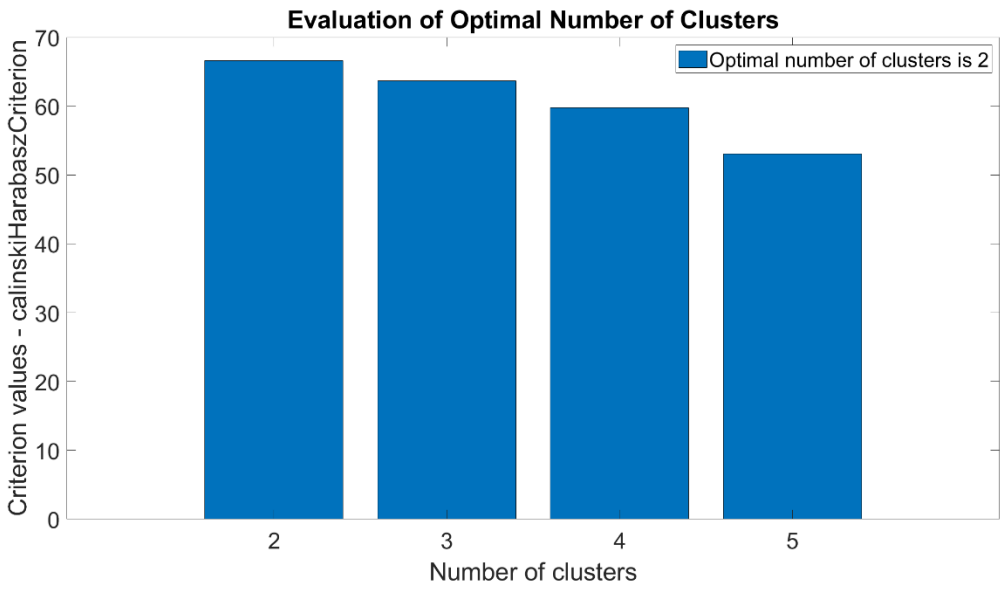

Figure 4-20: Optimal cluster number when analyzing the group of registers without CL use.

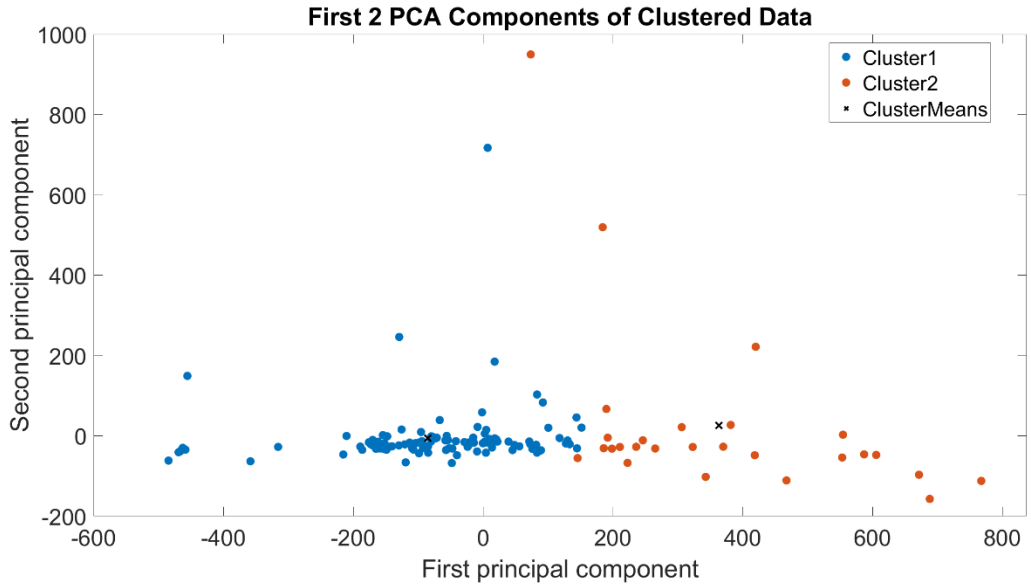

Figure 4-21: Clustered data of the registers without contact lens use.

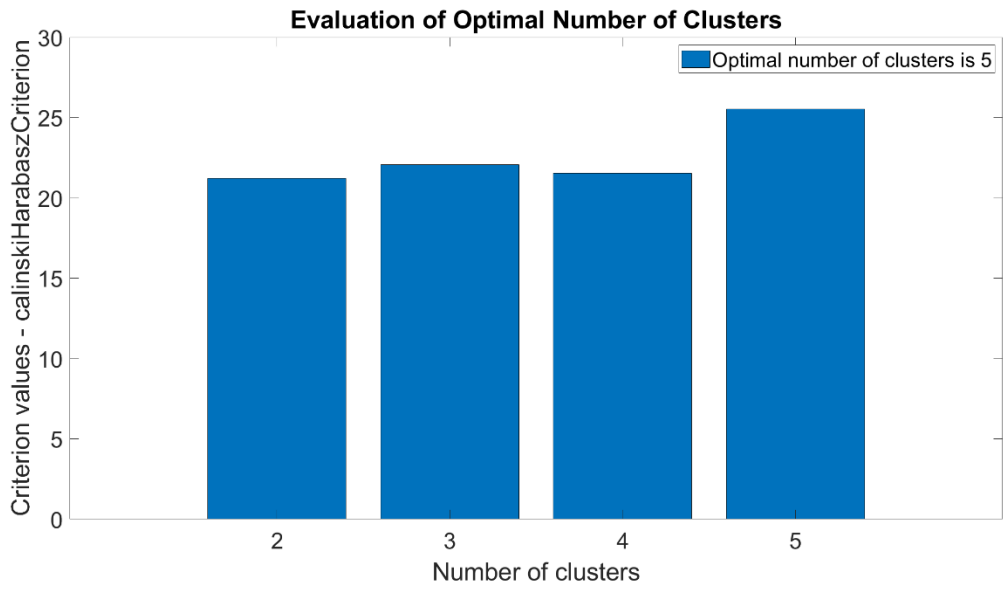

Figure 4-22: Clustered data of the registers with contact lens use.

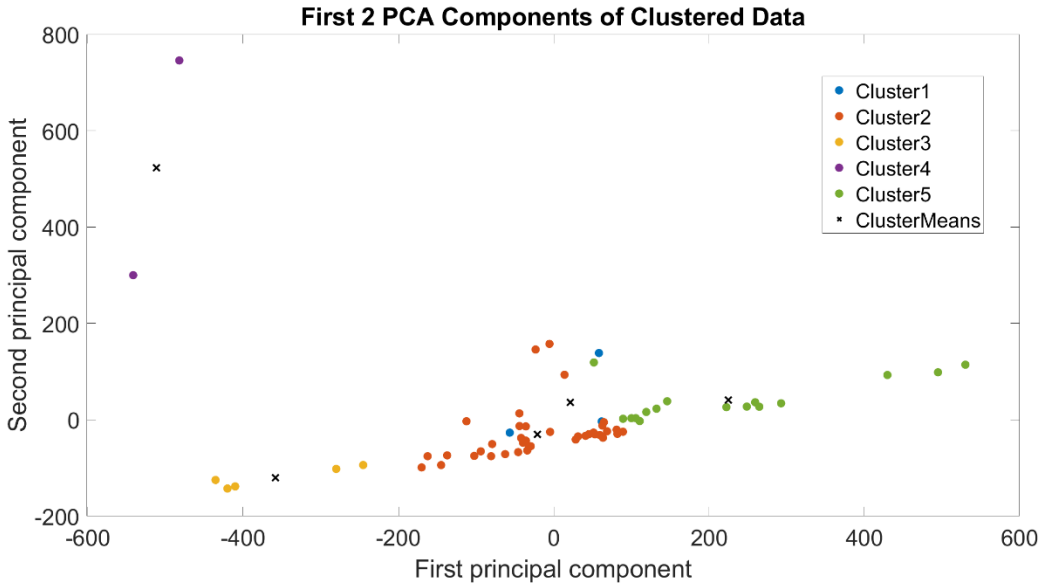

Figure 4-23: Clustered data of the registers with contact lens use.

The above results can be analyzed in several aspects.

- In the group of records without contact lenses, two clusters appear that are distinguished, fundamentally, by their weight in the main component (preferably positive in one case and negative in the other). Cluster 1 represents 85% of the records, and Cluster 2 14.78%. Within each of them, cluster 1 has 41.32% presbyopic records and 58.67% non-presbyopic records. To compare which type of features are different, the average value of the features in clusters 1 and 2 ($\overline{R}_{C1}, \overline{R}_{C2}$) has been compared using the metric.

$$Feature\ Variation\ \% = 100\frac{\overline{R}_{C2} - \overline{R}_{C1}}{\overline{R}_{C1}} \quad (4\text{-}4)$$

The results are shown in figure Figure 4-24. The changes are mainly in features 50-100, which correspond to the standard deviations of the features associated with the blink dynamics (see Table 4-10).

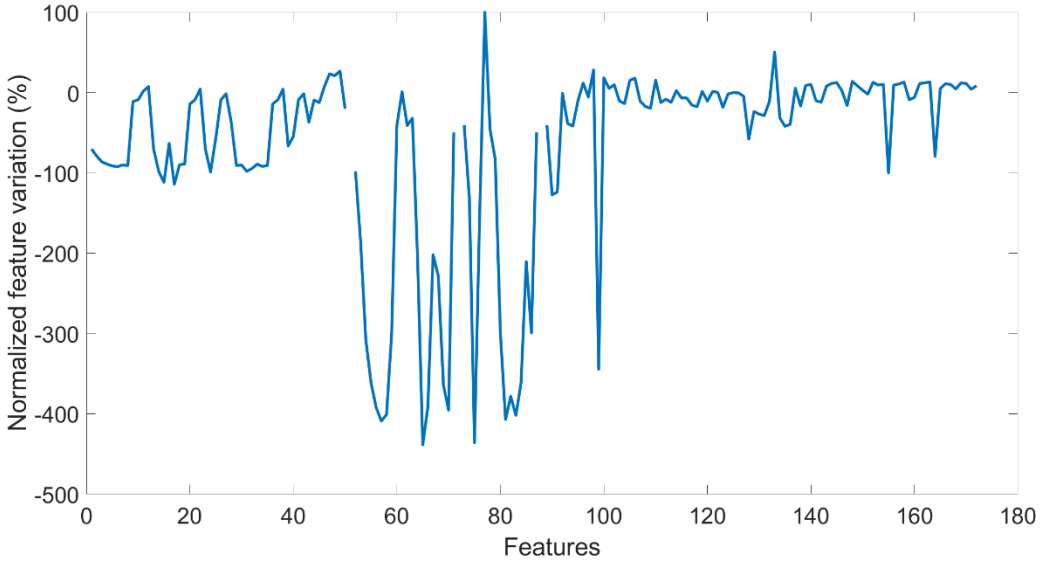

Figure 4-24: Average variation of the features of cluster 2 with respect to cluster 1 in the case of recordings without lens.

- In the group of records with contact lenses, more clusters appear. The decomposition of principal components (see Figure 4-23) continues to differentiate these by their importance in the first component, with cluster 1 in positive values (preferably), while with negative values of the first principal component, several clusters now appear. The distribution of records in each cluster and the presence of presbyopes and non-presbyopes in them appear in Table 4-14 and Table 4-15.

Table 4-10: Percentage of records associated with each cluster and internal distribution between presbyopes and non-presbyopes for the group of records with contact lenses.

	Registers (%)	Presbyopic in cluster (%)	Non presbyopic in cluster (%)
Cluster 1	10.6	100	0
Cluster 2	50.0	57.6	42.4
Cluster 3	25.7	47.0	53.0
Cluster 4	3.0	100	0
Cluster 5	10.6	100	0

From the results in Table 4-10 it can be seen that the use of contact lenses has caused a greater variation in the group, with a greater number of clusters. In addition, this alteration is greater in presbyopic registrants than in non-presbyopic registrants.

The average variations of the features between cluster 1 and the others (for the group of registries with contact lenses) appear in Figure 4-25, Figure 4-26, Figure 4-27, and Figure 4-28. In the Table 4-11, Table 4-12, and Table 4-13 , for each cluster, the ones that represent variations greater than 100% (positive or negative) with respect to cluster 1 have been compiled.

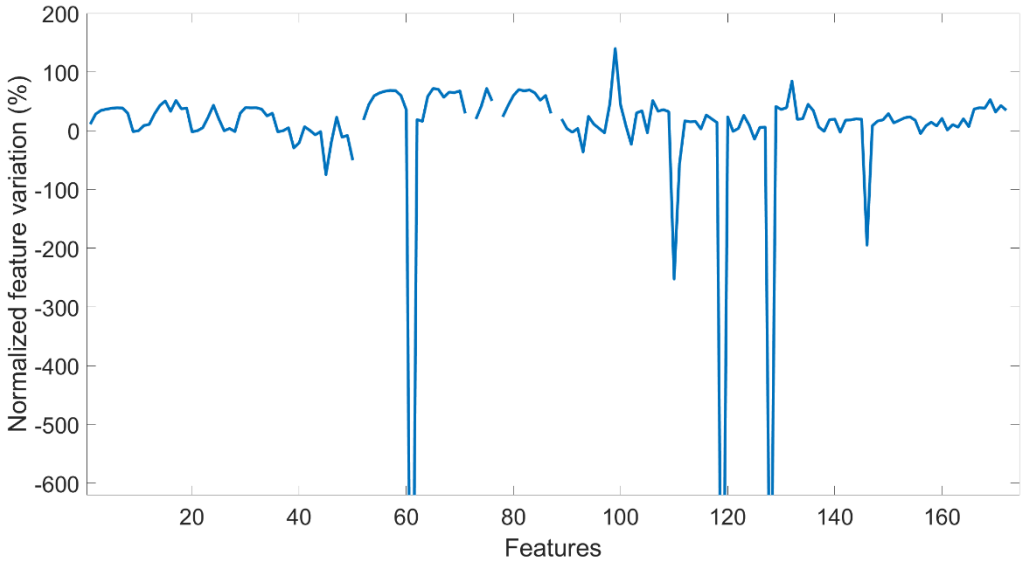

Figure 4-25: Normalized feature variation comparing cluster data 2 with cluster data 1 from contact lens registers.

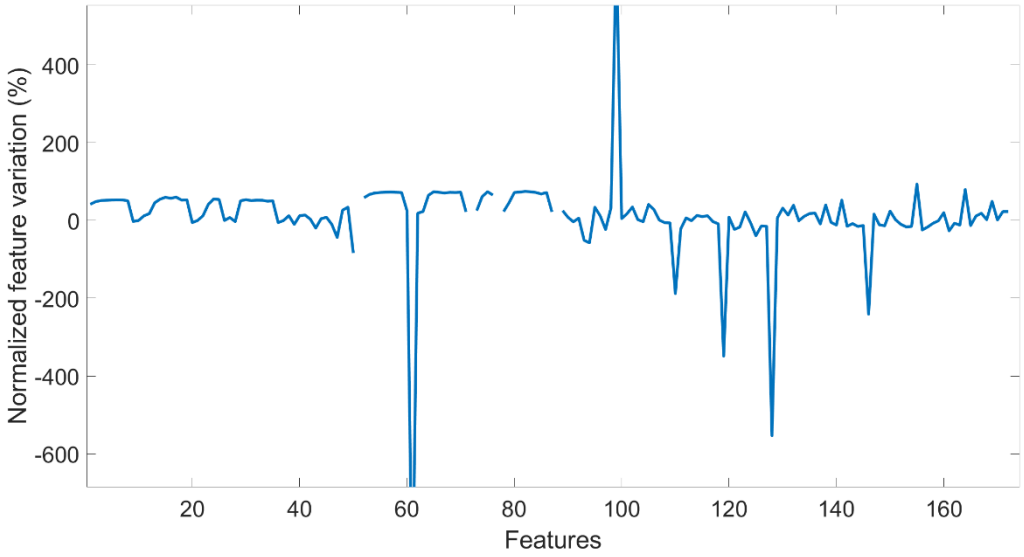

Figure 4-26: Normalized feature variation comparing cluster data number 3 with cluster data number 1 from contact lens registers.

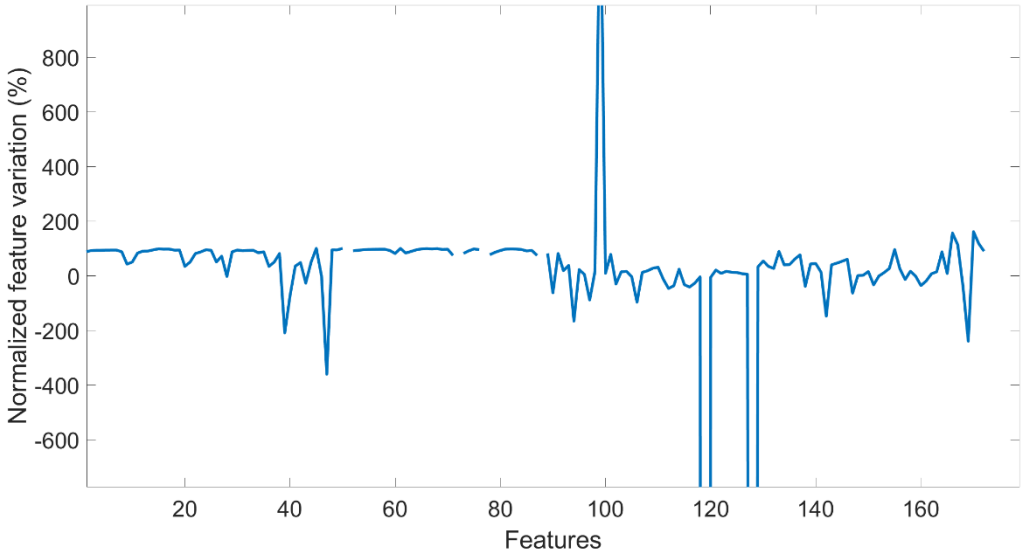

Figure 4-27: Normalized feature variation comparing cluster data number 4 with cluster data number 1from contact lens registers.

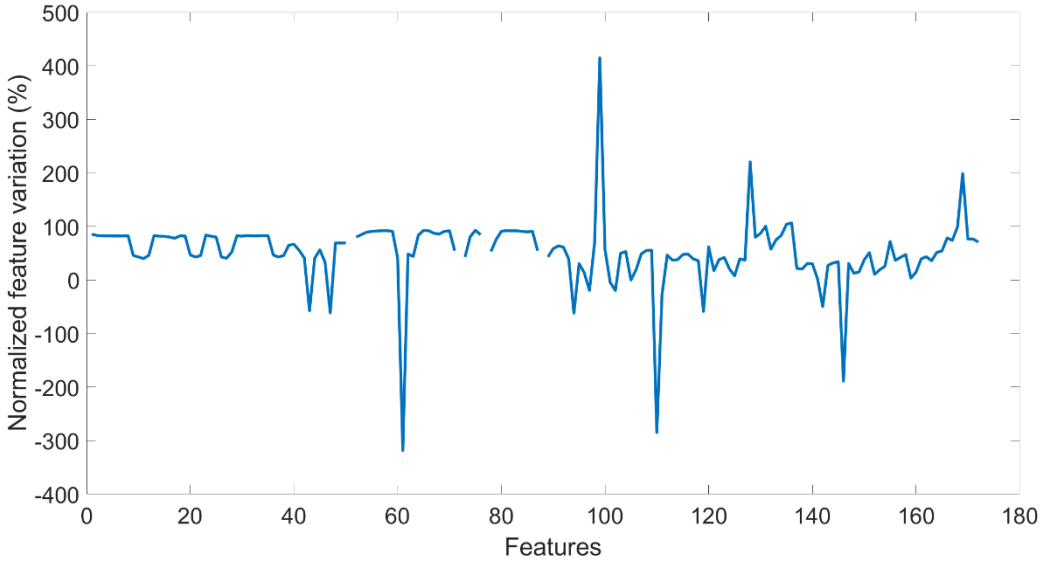

Figure 4-28: Normalized feature variation comparing cluster data number 5 with cluster data number 1 from contact lens registers.

Table 4-11: Features with a variation higher than 100% when comparing cluster 2 and 3, to cluster 1.

Number	Feature
61	'std_P3'
99	'skew_locs_fp (s)'
110	'nasal_std_channel_h'
119	'nasal_kurt_channel_h'
128	'nasal_skew_channel_h'
146	'temporal_std_channel_h'

Table 4-12: Features with a variation higher than 100% when comparing cluster 4 to cluster 1.

Number	Feature
39	'v2p_sn'
47	'b_mm'
94	'std_ibi'
99	'skew_locs_fp (s)'
119	'nasal_kurt_channel_h'
128	'nasal_skew_channel_h'
142	'temporal_mean_channel_b_lab'
166	'temporal_skew_channel_v'
167	'temporal_skew_channel_l'
169	'temporal_skew_channel_b_lab'
170	'temporal_skew_channel_r'
171	'temporal_skew_channel_g'

Table 4-13: Features with a variation higher than 100% when comparing cluster 5 to cluster 1.

Number	Feature
61	'std_P3'
99	'skew_locs_fp (s)'
110	'nasal_std_channel_h'
128	'nasal_skew_channel_h'
146	'temporal_std_channel_h'
169	'temporal_skew_channel_b_lab'

The above results show that, with regard to blinking, the variation in the closing braking power of the blink is mainly altered, as is the uniformity of the blinks over time. But, above all, what appears as a classifier now are the changes in the color variables, both in the nasal and temporal parts.

As a general conclusion of this study, we could say that the records without contact lenses show more excellent uniformity among themselves (a cluster appears with 85% of the records). In contrast, there is more significant heterogeneity in the group with contact lenses. This heterogeneity is focused to a greater extent on presbyopic subjects. The alterations that allow them to be classified into different groups are mainly characteristics of differences associated with the color of the nasal and temporal area of the eye. To a lesser extent, classification features appear associated with the eye braking power and the non-uniform distribution of blinks over the considered time. These results point to a different behavior of presbyopic subjects regarding the adaptation of contact lenses and the processes of damage, inflammation, friction, the different state of dry eye, etc., that occur in the eye (206,207). This tells us that developing evaluation techniques and personalization of adaptation are important, especially in presbyopic subjects.

Table 4-14: features involved in the Mahalanobis distance analysis (1/2).

1	't1p (s)'	31	'half_width_right (ms)'	61	'std_P3'
2	't2p (s)'	32	'half_width (ms)'	62	'std_P5'
3	't3p (s)'	33	'width_left (ms)'	63	'std_P7'
4	't4p (s)'	34	'width_right (ms)'	64	'std_W02'
5	't5p (s)'	35	'width (ms)'	65	'std_W24'
6	't6p (s)'	36	'a1a_sn'	66	'std_W46'
7	't7p (s)'	37	'a2a_sn'	67	'std_W68'
8	't8p (s)'	38	'a3a_sn'	68	'std_t1a (s)'
9	'P1'	39	'v2p_sn'	69	'std_t2a (s)'
10	'P3'	40	'v6p_sn'	70	'std_t3a (s)'
11	'P5'	41	'locs_fp (s)'	71	'std_a1a (s)'
12	'P7'	42	'm_cp'	72	'std_a2a (s)'
13	'W02'	43	'm_ibi'	73	'std_a3a (s)'
14	'W24'	44	'm_cp_complete'	74	'std_J02'
15	'W46'	45	'm_cp_incomplete'	75	'std_J26'
16	'W68'	46	'alpha_mm'	76	'std_J68'
17	't1a (s)'	47	'b_mm'	77	'std_v2p'
18	't2a (s)'	48	'num'	78	'std_v6p'
19	't3a (s)'	49	'num_complet'	79	'std_S'
20	'a1a (s)'	50	'num_incomplet'	80	'std_tiempo_blink (ms)'
21	'a2a (s)'	51	'std_t0p (s)'	81	'std_half_width_left (ms)'
22	'a3a (s)'	52	'std_t1p (s)'	82	'std_half_width_right (ms)'
23	'J02'	53	'std_t2p (s)'	83	'std_half_width (ms)'
24	'J26'	54	'std_t3p (s)'	84	'std_width_left (ms)'
25	'J68'	55	'std_t4p (s)'	85	'std_width_right (ms)'
26	'v2p'	56	'std_t5p (s)'	86	'std_width (ms)'
27	'v6p'	57	'std_t6p (s)'	87	'std_a1a_sn'
28	'S'	58	'std_t7p (s)'	88	'std_a2a_sn'
29	'tiempo_blink (ms)'	59	'std_t8p (s)'	89	'std_a3a_sn'
30	'half_width_left (ms)'	60	'std_P1'	90	'std_v2p_sn'

Table 4-15: features involved in the Mahalanobis distance analysis (2/2).

91	'std_v6p_sn'	121	'nasal_kurt_channel_v'	151	'temporal_std_channel_b_lab'
92	'std_locs_fp (s)'	122	'nasal_kurt_channel_l'	152	'temporal_std_channel_r'
93	'std_cp'	123	'nasal_kurt_channel_a'	153	'temporal_std_channel_g'
94	'std_ibi'	124	'nasal_kurt_channel_b_lab'	154	'temporal_std_channel_b_rgb'
95	'std_alfa_mm'	125	'nasal_kurt_channel_r'	155	'temporal_kurt_channel_h'
96	'std_beta_mm'	126	'nasal_kurt_channel_g'	156	'temporal_kurt_channel_s'
97	'kur_locs_fp (s)'	127	'nasal_kurt_channel_b_rgb'	157	'temporal_kurt_channel_v'
98	'kurt_ibi'	128	'nasal_skew_channel_h'	158	'temporal_kurt_channel_l'
99	'skew_locs_fp (s)'	129	'nasal_skew_channel_s'	159	'temporal_kurt_channel_a'
100	'skew_ibi'	130	'nasal_skew_channel_v'	160	'temporal_kurt_channel_b_lab'
101	'channel_h'	131	'nasal_skew_channel_l'	161	'temporal_kurt_channel_r'
102	'channel_s'	132	'nasal_skew_channel_a'	162	'temporal_kurt_channel_g'
103	'channel_v'	133	'nasal_skew_channel_b_lab'	163	'temporal_kurt_channel_b_rgb'
104	'channel_l'	134	'nasal_skew_channel_r'	164	'temporal_skew_channel_h'
105	'channel_a'	135	'nasal_skew_channel_g'	165	'temporal_skew_channel_s'
106	'channel_b_lab'	136	'nasal_skew_channel_b_rgb'	166	'temporal_skew_channel_v'
107	'channel_r'	137	'temporal_mean_channel_h'	167	'temporal_skew_channel_l'
108	'channel_g'	138	'temporal_mean_channel_s'	168	'temporal_skew_channel_a'
109	'channel_b_rgb'	139	'temporal_mean_channel_v'	169	'temporal_skew_channel_b_lab'
110	'nasal_std_channel_h'	140	'temporal_mean_channel_l'	170	'temporal_skew_channel_r'
111	'nasal_std_channel_s'	141	'temporal_mean_channel_a'	171	'temporal_skew_channel_g'
112	'nasal_std_channel_v'	142	'temporal_mean_channel_b_lab'	172	'temporal_skew_channel_b_rgb'
113	'nasal_std_channel_l'	143	'temporal_mean_channel_r'		
114	'nasal_std_channel_a'	144	'temporal_mean_channel_g'		
115	'nasal_std_channel_b_lab'	145	'temporal_mean_channel_b_rgb'		
116	'nasal_std_channel_r'	146	'temporal_std_channel_h'		
117	'nasal_std_channel_g'	147	'temporal_std_channel_s'		
118	'nasal_std_channel_b_rgb'	148	'temporal_std_channel_v'		
119	'nasal_kurt_channel_h'	149	'temporal_std_channel_l'		
120	'nasal_kurt_channel_s'	150	'temporal_std_channel_a'		

4.4 **Fourth study**: Regression learner and classification learner to estimate the results of the CLDEQ-8 questionnaire.

This study aims to evaluate whether the subjective responses given by patients to the questionnaire carried out by the application can be predicted and related to the objective "features" also obtained by the application through the analysis of the videos. The responses analyzed are those of the CLDEQ-8 questionnaire implemented in the App, which is a standard for this type of studies (208).

For this, the MATLAB Statistics and Machine Learning toolbox will be used. (209).

This function library contains various algorithms for hypothesis testing, regression and automatic classifiers. The library is mainly used to calculate two types of situations:

- As a classifier: In this sense, the patient's response to each question in the questionnaire can be “classified” into a discrete response (1-5 in some questions and 1-6 in others). For each question in the questionnaire, an attempt is made to “predict” this category based on the features of each patient's video. These features are what the different algorithms are trained with.
- The MATLAB library implements different classification algorithms (Tree, Discriminant, SVM, etc. ((65,69,210)). An overview of these algorithms has been described in the introductory chapter of this thesis. The reader is referred to said chapter for a reminder of them. In total, the library can analyze nine general classification models. With the variants associated with each of them, a total of thirty-three classifiers are implemented. All of them can be used through the MATLAB App “Classification Learner” (211). For this thesis, the version implemented in MATLAB 2023a has been used.
- As a predictor: In this sense, the patient's response to each question in the questionnaire is “predicted” from the “features” of the videos as a continuous variable based on regression and adjustment algorithms (linear regression, SVM regression, decision trees, Neural Networks, etc.). In total, the library implements seven types of regression models with variants in some of them, reaching about 26 total models. These algorithms are implemented in the MATLAB App “Regression Learner” in its MATLAB 2023a version.

The questions considered are:

- The first question, about eye discomfort, is: During a typical day in the past 2 weeks, how often did your eyes feel discomfort while wearing your contact lenses? (Answer:1-5).
- The second question, about eye discomfort, is: When your eyes felt discomfort with your contact lenses, how intense was this feeling of discomfort at the end of your wearing time? (Answer: 1-6).
- The third question, about eye dryness, is: During a typical day in the past 2 weeks, how often did your eyes feel dry? (Answer: 1-5).
- The fourth question, about eye dryness, is: When your eyes felt dry, how intense was this feeling of dryness at the end of your wearing time? (Answer: 1-6).
- The fifth question, about changeable, blurry vision, is: During a typical day in the past 2 weeks, how often did your vision change between clear and blurry or foggy while wearing your contact lenses? (Answer: 1-5).
- The sixth question, about changeable, blurry vision, is: When your vision was blurry, how noticeable was the changeable, blurry, or foggy vision at the end of your wearing time? (Answer: 1-6).
- The seventh question, about closing your eyes, is: During a typical day in the past 2 weeks, how often did your eyes bother you so much that you wanted to close them? (Answer: 1-5).
- The eighth question, about removing your lenses, is: How often during the past 2 weeks, did your eyes bother so much while wearing your contact lenses that you felt as if you needed to stop whatever you were doing and take out your contact lenses? (Answer: 1-5).

The different types of models used for classification learners are:

- Tree
- Discriminant
- Efficient logistic regression
- Efficient Linear SVM
- Naïve Bayes.
- SVM
- KNN
- Neuronal Network

- Kernel (65)

The different types of models used for regression learners are:

- Linear regression
- Stepwise Linear regression
- Tree
- SVM
- Gaussian Process Regression.
- Neural Network
- Kernel (65).

An intuitive explanation of each one of the models is available in the introduction chapter of this thesis (Poner hypervinculo a capitulo de introduccion). Each trained model provides an accuracy in percent of correctly classified responses. For adjustment or regression models, the application provides several control parameters, the one used transversally by all of them being the RMSE (root mean square error) between the predicted and trained values. For both types of algorithms, it is trained and validated using the K-fold cross validation model (212) using 80% of the data as training and the remaining 20% as validation, carrying out this process five times and in each of them the validation group is different.

The best models for each one of the questions are:

- For the first question, KNN and Neural Networks reached the maximum accuracy in the validation step regarding the classification Learner models (80.65%). Using Regression models, the best root mean square error during the validation step was obtained using a Support Vector Machine (RMSE=0.29).
- For the second question, the Bagged Trees model obtained the best accuracy on the validation step (87.10%). When using the Regression learner model, Gaussian Process Regression obtained the lowest RMSE (RMSE=0.39).
- For the third question, Efficient Logistic Regression and efficient linear SVM reached the maximum accuracy in the validation step regarding the classification Learner models (83.87%). Using Regression models, the best root mean square error during the validation step was obtained using a Support Vector Machine (RMSE=0.29).

- The 4th question tree model obtained the best accuracy in the validation step (83.87%). The kernel obtained the lowest RMSE when using the regression learner model (RMSE=0.37).
- For the 5th question, Bagged Trees reached the maximum accuracy in the validation step concerning the Classification Learner models (80.65%). Regression models obtained the best root means square error during the validation step using Gaussian Process Regression (RMSE=0.37).
- For the 6th question, the Bagged Trees model obtained the best accuracy on the validation step (77.2%). When using the Regression learner model, Gaussian Process Regression obtained the lowest RMSE (RMSE=0.38).
- For the 7th question, the Support Vector Machine reached the maximum accuracy in the validation step concerning the classification of Learner models (54.84%). Using Regression models, the best root mean square error during the validation step was obtained using a Support Vector Machine (RMSE=0.34).
- For the 8th question, the trees model obtained the best accuracy on the validation step for the classification learner (87.10%). For the regression learner, the lowest RMSE was obtained using the regression learner model, Gaussian process regression, and kernel. For these models, the RMSE on the validation step was 0. We chose a Neural Network with an RMSE of 0.2927 as the optimum model because the Kernel and Gaussian Process Regression seem overfitted.

Table 4-16: Best classification and regression models for each one of the predicted answer question.

Question CLDEQ-8	**Classification model (Accuracy)**	**Regression model (RMSE)**
1	KLN, Nerural Network (80.65%)	SVM (0.29)
2	Bagged Trees (87.10%)	Gaussian Process (0.39)
3	Efficient Linear Regression, SVM (83.87%)	SVM (0.29)
4	Tree (83.87%)	Kernel (0.37)
5	Bagged Trees (80.65%)	Gaussian Process (0.37)
6	Bagged Tress (77.2 %)	Grausian Process (0.38)
7	SVM (54.84%)	SVM (0.34)
8	Tree (87.10%)	Neural Network (0.29)

Several comments can be made from the previous results:

- Regarding the classification models, almost all have an accuracy above 80% for all questions with a lower percentage in questions 6 and 7. These are associated with more “internal” characteristics of the vision process, such as the quality of vision or

annoyance thresholds. These issues may not be reflected as well as the rest in the features calculated from the video images.

- Regression models perform somewhat worse than classifiers. This could be because the response the patient gives to the questions is, in reality, discreet and therefore closer to a classification process.
- In any case, an interesting conclusion is that the objective characteristics (features of the videos) seem to be related to the patients' subjective responses. And therefore, they could be used as objective variables in the characterization of comfort.
- Regarding the different classification and regression methods, those that give the best results are those associated with tree methods. Let us remember from the introductory chapter that these methods "look for" features that can be used to classify the results where each of these groups differ in that feature. Hence, it seems suggested that not all of them have the same importance in the classification process.

From the previous results, and in order to be able to estimate the patients' responses to the questionnaire, based only on the most relevant features, a GLM ("Generalized Linear Model") adjustment was carried out with a Stepwise adjustment method. (213). This model is implemented in the Stepwiselm function of the MATLAB Statisics and Machine Learning Toolbox library. In this type of models, the variable to be predicted (in this case the answers to the questionnaire) are calculated as combinations of the variables (in our case the objective features taken from the video images). These variables are added and subtracted iteratively, checking at each step the statistical significance of the adjustment when that variable is used ($p<0.05$), ending the process with the best combination (214). La función de MATLAB usada usa por defecto no sólo las variables en sí, sino también sus productos dos a dos.

A linear fitting to predict the patient response using the stepwise regression model is performed for the different questions related to the comfort of the eye. The linear regression model equation for each one of the different questions will have the structure of equation 4-4:

$$y_{resp1} \sim I + E_1 * x_1 + \cdots + E_{n-1} * x_{n-1} + E_n * x_n + \sum_{i,j=1}^{n} E_{ij} x_i x_j \qquad (4\text{-}5)$$

Where "I" is the value of the estimate of the Intercept, E_n is the value of the estimate for the variables taken into account in the regression model and x_n is the predictor value of the feature in question and E_{ij} the value for the coefficient of the corresponding features.

The results for the first response related to eye comfort are available on the Table 4-17

Table 4-17: Results of stepwise regression model to fit the response of the 1st question of the CLDEQ-8 questionnaire.Poner aquí la table donde se pueden mirar cuáles son las variables.

Variable	Estimate	pValue	Feature
(Intercept)	-0.94395	0	-
x42	5.50E-09	0	locs_fp (s)
x53	-0.002371	0	num_incomplet
x55	18.096	0	std_t1p (s)
x66	-1.5145	0	std_P7
x96	0.095809	0	std_cp
x98	0.2064	0	std_dur
x99	2.1894	0	std_alfa_mm
x101	0.00037398	0	kur_t1p (s)
x111	-0.026074	0	kur_P7
x119	0.16432	0	kur_a1a (s)
x120	-0.09105	0	kur_a3a (s)
x153	-0.00016932	0	skew_W24
x171	0.20759	0	skew_width_right (ms)
x185	-0.059653	0	channel_a
x194	0.24472	0	nasal_std_channel_a
x199	0.00044655	0	nasal_kurt_channel_h
x210	-0.058618	0	nasal_skew_channel_v
x212	-0.1458	0	nasal_skew_channel_a
x234	0.063776	0	temporal_std_channel_b_rgb
x55:x98	-1.2524	0	-
x55:x111	8.0436	0	-
x55:x234	0.040539	0	-
x96:x111	-0.008298	0	-
x96:x234	-0.0025109	0	-
x98:x111	-1.05E-05	0	-
x98:x119	-0.074607	0	-

The characteristics of the first question model are:

- Number of observations: 29, Error degrees of freedom: 2
- R-squared: 1, Adjusted R-Squared: 1
- F-statistic vs. constant model: Inf, p-value = 0

The results for the second question related also to eye comfort are available on the Table 4-18.

Table 4-18: Results of stepwise regression model to fit the response of the 2nd question of the CLDEQ-8 questionnaire.

Variable	Estimate	pValue	Feature
(Intercept)	1.8945	2.36E-07	-
x2	15.456	0.001506	t1p (s)
x49	0.48763	1.06E-05	alpha_mm
x80	1.35E+06	0.004943	std_v2p
x96	0.018943	0.021427	std_cp
x191	6.216	6.53E-05	nasal_std_channel_s
x210	-0.35926	5.69E-05	nasal_skew_channel_v
x221	-0.05117	0.000176	temporal_mean_channel_a

The characteristics of the second question model are:

- Number of observations: 29, Error degrees of freedom: 21
- Root Mean Squared Error: 0.196
- R-squared: 0.898, Adjusted R-Squared: 0.863
- F-statistic vs. constant model: 26.3, p-value = 5.02e-09

The results for the third response related to eye dryness are available on the Table 4-19.

Table 4-19: Results of stepwise regression model to fit the response of the 3rd question of the CLDEQ-8 questionnaire.

Variable	Estimate	pValue	Feature
(Intercept)	1.9891	0	-
x2	3.7379	0	t1p (s)
x41	-1.1952	0	v6p_sn
x52	0.0036638	0	num_complet
x77	0.63492	0	std_J02
x80	-8.38E+05	0	std_v2p
x95	-0.022828	0	std_locs_fp (s)
x97	0.014789	0	std_ibi
x102	0.024314	0	kur_t2p (s)
x135	0.00016262	0	kur_v2p_sn
x140	0.0024093	0	kurt_dur
x155	0.018348	0	skew_W68
x156	0.051394	0	skew_t1a (s)
x163	0.016435	0	skew_J68
x175	-1.1418	0	skew_v2p_sn
x180	0.1349	0	skew_dur
x185	-0.029232	0	channel_a
x189	0.00065928	0	channel_b_rgb
x197	-3.31E-08	0	nasal_std_channel_g
x211	0.24516	0	nasal_skew_channel_l
x214	-0.21636	0	nasal_skew_channel_r
x2:x175	34.078	0	-
x41:x214	-0.00097486	0	-
x52:x156	-4.85E-05	0	-
x102:x180	-0.024942	0	-
x156:x214	0.0041367	0	-
x175:x211	-0.30099	0	-

The characteristics of the third question model are:

- Number of observations: 29, Error degrees of freedom: 2
- R-squared: 1, Adjusted R-Squared: 1
- F-statistic vs. constant model: 5.72e+29, p-value = 1.75e-30

The results for the fourth response related to eye dryness are available on the Table 4-20.

Table 4-20: Results of stepwise regression model to fit the response of the 4th question of the CLDEQ-8 questionnaire.

Variable	Estimate	pValue	Feature
(Intercept)	2.5667	0	-
x12	8.15E-07	0	P5
x18	0.3191	0	t1a (s)
x44	-0.6927	0	m_ibi
x52	0.011024	0	num_complet
x65	-0.33829	0	std_P5
x98	0.03002	0	std_dur
x116	0.006635	0	kur_t1a (s)
x141	0.03581	0	skew_t1p (s)
x145	0.12391	0	skew_t5p (s)
x146	-0.00705	0	skew_t6p (s)
x151	-0.14569	0	skew_P7
x167	-0.00149	0	skew_half_width_left (ms)
x175	-0.2463	0	skew_v2p_sn
x182	5.4709	0	channel_s
x191	-4.5691	0	nasal_std_channel_s
x215	-0.09906	0	nasal_skew_channel_g
x44:x65	1.7857	0	-
x44:x151	0.007531	0	-
x44:x175	0.076793	0	-
x65:x98	-0.18705	0	-
x65:x116	-0.00015	0	-
x65:x146	0.000419	0	-
x65:x182	-22.029	0	-
x141:x182	0.023204	0	-
x145:x151	-0.14028	0	-
x191:x215	-1.5262	0	-

The characteristics of the fourth question model are:

- Number of observations: 29, Error degrees of freedom: 2
- R-squared: 1, Adjusted R-Squared: 1
- F-statistic vs. constant model: Inf, p-value = 0

The results for the fifth response related to changeable, blurry vision are available on the Table 4-21.

Table 4-21: Results of stepwise regression model to fit the response of the 5th question of the CLDEQ-8 questionnaire.

Variable	Estimate	pValue	Feature
(Intercept)	-0.07386	0.85642	-
x35	-0.00474	0.018414	width_right (ms)
x47	0.101	0.005334	m_dur_complete
x99	169.47	4.06E-05	std_alfa_mm
x244	-0.04451	0.041874	temporal_skew_channel_h

The characteristics of the fifth question model are:

- Number of observations: 29, Error degrees of freedom: 24
- Root Mean Squared Error: 0.421
- R-squared: 0.629, Adjusted R-Squared: 0.567
- F-statistic vs. constant model: 10.2, p-value = 5.79e-05

The results for the sixth response related to changeable, blurry vision are available on the Table 4-22.

Table 4-22: Results of stepwise regression model to fit the response of the 6th question of the CLDEQ-8 questionnaire.

	Estimate	pValue	Feature
(Intercept)	-3.2442	0	-
x48	4.06E-05	0	m_cp_incomplete
x99	201.11	0	std_alfa_mm
x115	0.11843	0	kur_W68
x137	-0.02236	0	kur_locs_fp (s)
x139	0.010943	0	kurt_ibi
x141	0.52922	0	skew_t1p (s)
x157	-0.59339	0	skew_t2a (s)
x158	-0.25239	0	skew_t3a (s)
x168	0.80648	0	skew_half_width_right (ms)
x175	-0.31649	0	skew_v2p_sn
x178	-0.33876	0	skew_cp
x182	-1.1716	0	channel_s
x194	0.76652	0	nasal_std_channel_a
x196	-2.74E-08	0	nasal_std_channel_r
x204	0.16215	0	nasal_kurt_channel_b_lab
x207	0.34093	0	nasal_kurt_channel_b_rgb
x227	-6.0796	0	temporal_std_channel_s
x241	-0.00045	0	temporal_kurt_channel_r
x243	-0.21381	0	temporal_kurt_channel_b_rgb
x99:x204	-3.7298	0	-
x99:x207	-15.825	0	-
x99:x227	3.7868	0	-
x139:x175	-0.01304	0	-
x158:x168	9.57E-06	0	-
x158:x243	-0.13657	0	-
x182:x194	0.021015	0	-

The characteristics of the sixth question model are:

- Number of observations: 29, Error degrees of freedom: 2
- R-squared: 1, Adjusted R-Squared: 1
- F-statistic vs. constant model: 9.84e+29, p-value = 1.02e-30

The results for the seventh response related to closing your eyes, are available on the Table 4-23.

Table 4-23: Results of stepwise regression model to fit the response of the 7th question of the CLDEQ-8 questionnaire.

Variable	Estimate	pValue	Feature
(Intercept)	0.22682	0.20806	-
x41	-3.836	2.22E-05	v6p_sn
x165	0.13704	0.008138	skew_S
x167	0.22219	2.05E-05	skew_half_width_left (ms)
x191	12.371	4.75E-09	nasal_std_channel_s
x202	0.12474	0.019427	nasal_kurt_channel_l
x221	-0.05108	0.001197	temporal_mean_channel_a
x227	-10.843	5.10E-06	temporal_std_channel_s
x245	0.37211	0.000677	temporal_skew_channel_s

The characteristics of the seventh question model are:

- Number of observations: 29, Error degrees of freedom: 20
- Root Mean Squared Error: 0.195
- R-squared: 0.931, Adjusted R-Squared: 0.904
- F-statistic vs. constant model: 33.9, p-value = 5.46e-10

The results for the eighth response related to contact lens removal are available on the Table 4-24.

Table 4-24: Results of stepwise regression model to fit the response of the 8th question of the CLDEQ-8 questionnaire.

Variable	Estimate	pValue	Feature
(Intercept)	1.121	1.83E-10	-
x40	8.1783	6.63E-06	v2p_sn
x93	-22.218	1.40E-07	std_v2p_sn
x151	-0.08514	0.001299	skew_P7
x197	0.030231	1.04E-11	nasal_std_channel_g
x216	0.071685	0.040596	nasal_skew_channel_b_rgb
x237	0.41786	3.42E-07	temporal_kurt_channel_v
x243	-0.68177	2.31E-10	temporal_kurt_channel_b_rgb
x40:x237	-1.7827	0.000926	-
x93:x243	6.2714	1.03E-06	-

The characteristics of the eighth question model are:

- Number of observations: 29, Error degrees of freedom: 19
- Root Mean Squared Error: 0.0751
- R-squared: 0.974, Adjusted R-Squared: 0.962
- F-statistic vs. constant model: 79.3, p-value = 3.7e-13

Again, from the results of the previous tables we can make several comments.

- There are certain questions in the questionnaire (2,5,7 and 8) that seem to depend linearly on fewer features, these being, above all, features associated with color. These questions represent issues more related to situations of high specific discomfort. (Hence, the variables that record color are the most involved since the strongest ocular discomfort due to the use of contact lenses is usually associated with processes of inflammation and redness of the eye).
- The rest of the questions, more associated with evaluating longer-term, more diffuse comfort and better image quality, are associated with a greater number of features with statistical significance (pvalue<0.5). Most of these have to do with variables characterizing the high-order statistics (std, skewness and kurtosis) of the flicker dynamics, which indicates a complex interaction between this dynamic and subjective comfort.
- In any case, the R-squared of the adjustments are high, which also indicates a strong relationship between subjective comfort and objective variables that can be measured. This points in the direction of validating the methodology used in this doctoral thesis, as well as the necessary collection of more data in the future.

The conclusion related to the most important feature for each topic related to symptoms induced due to contact lens wear from the CLDEQ-8 questionnaire can be extracted from the stepwise analysis and checking which variables are mostly involved in each question.

For the eye discomfort questions, we observe that the first and second questions have a common feature involved in the stepwise regression learner, and it is the standard deviation of the degree of eye closeness of the blinks. This variable is used to characterize the quality of the blinks, complete and incomplete. In addition, we observe that the first model has the standard deviation of the α parameter from the logarithmic PSD fitting of the eyelid blinking cinematic, which this value represents the dynamic of the blinking motion. For the second question, the same alpha parameter is used as a feature, but this time, it is used as the mean value. From the observation of the different variables involved in both question answers estimation, we see that essential features related to blinks are used for the analysis of the answers as the duration of the blinks, maximum speed on the downstroke motion, the power involved in the upstroke motion $P7$ or the number of incomplete blinks. Additionally, some sclera color features

estimate the answers to both eye discomfort questions. The most important feature is channel a* from the CIELAB system, which indicates the red-green axis.

Regarding eye dryness, we can observe that common features for both questions are the inter-blink interval, and the number of complete blinks recorded in each session. In addition, the duration is also related to both questions, although for the first question, the mean value of the duration is used as an estimate. In contrast, for the second question, skewness and kurtosis values of the duration are used as predictors for the answer. Concerning sclera color features, we note that the channel a* from the CIELAB system is again an important feature to estimate for the question of eye dryness.

For the topic of changeable, blurred vision, we observe that the standard deviation of the inter-blink interval is a common feature for both questions regarding eye blinking. When examining both questions, we can observe that key features such as the duration of complete blinks and the degree of proximity of incomplete blinks contribute to estimating patient responses related to changes in vision while wearing contact lenses. Concerning the color of the sclera, we again highlight the significance of channel a*, which measures the level of redness in the scleral region.

For the question related to closing the eye due to ocular symptoms using contact lenses, we highlight the important features from blinking at maximum speed at upstroke motion. Concerning the color sclera region, we observe again that the channel a* from the CIELAB system is again used as an estimate variable.

For the last question concerning the discontinuation of use of contact lens due to ocular symptoms, we highlight as estimate the maximum speed and it standard deviation value as estimates to check the answer of the patient.

It is relevant to refer to previous research that has addressed these aspects to strengthen the results obtained through the CLDEQ-8 questionnaire and establish a stronger relationship between ocular comfort and the factors identified in the study.

Various studies have pointed out the relationship between discomfort with contact lens use and factors such as ocular redness and color perception. For example, previous research has shown that dryness and the desire to remove contact lenses may be directly related to conjunctival hyperemia and other clinical signs of ocular discomfort. This aligns with studies such as that

of Meijome et al., who have analyzed the dynamics of the tear film and its impact on the comfort of contact lens users (206,207).

Likewise, some scientific papers have explored how objective measures of tear stability and alterations in the ocular surface can support the user's subjective perception of discomfort. Research such as that of Pult et al. has evaluated the association between the stability of the tear meniscus and the discomfort reported in questionnaires such as the CLDEQ-8, providing evidence on the validity of these instruments in the detection of relevant symptoms (215).

In summary, the results obtained in this fourth study show that various blinking and sclera color features, particularly the standard deviation of blink dynamics and the a* channel from the CIELAB system, are consistently crucial in estimating answers across different ocular symptom questions. These variables reflect the dynamic nature of blinking and the eye’s response to contact lens usage, providing valuable insights for discomfort, dryness, and vision changes.

5 CONCLUSION

5. CONCLUSION

The conclusions obtained in this research are as listed below:

1. A web application was successfully developed to collect objective and subjective data from patients, including eye videos, environmental conditions (temperature and humidity), and user-reported outcomes, ensuring real-time monitoring and data acquisition.
2. The application integrates geolocation services to connect with weather and air contamination stations, providing relevant environmental information without compromising patient privacy.
3. Image processing and artificial intelligence algorithms were implemented to extract eye physical parameters from videos, allowing for a non-invasive and efficient analysis of blinking dynamics and scleral color changes.
4. An extensive database was generated by automating the connection between the web application and the workstation used for video processing, facilitating large-scale data collection.
5. Big data techniques were applied to analyze the generated database, enabling classification, diagnosis, and prediction of contact lens adaptation outcomes, with a particular focus on presbyopic subjects who exhibited greater variability in blink dynamics and ocular surface characteristics.

6 SCIENTIFIC CONTRIBUTION

6. SCIENTIFIC CONTRIBUTION

During the development of this doctoral thesis, the following contributions were made under the funding grant stated below. This list is divided into articles in scientific journals and oral and poster presentations at conferences and congresses.

Scientific articles:

- **First article**

  DOI: Under review.

  Title: Development of an App to collect data at real time for optometric applications
  Authors: Youssef Marrakchi Chikri, Nuria Garzón, Mariano González Pérez, J.M. López-Alonso
  Scientific Journal: Heliyon
  It is currently under peer review. It is a public/private sector joint publication - Alain Afflelou company information has been included as the paper was developed while I was under contract with them.

- Second Article:
  DOI: In preparation
  Title: Review of Algorithms for Detecting Eye Blinks from Ubiquitous Video Cameras
  Authors: Youssef Marrakchi Chikri, J.M. López-Alonso
  The publication will be available in Open Access. It is currently in preparation and it is not a public/private sector joint publication

- Third Article:
  DOI: In preparation
  Title: Algorithm for Eye Blinks Feature Extraction from Ubiquitous Video Cameras
  Authors: Youssef Marrakchi Chikri, J.M. López-Alonso
  The publication will be available in Open Access. It is currently in preparation and it is not a joint public/private sector publication.

Congress and Conferences presentations:

- 'Development of e-Health instruments and technologies for Big Data analysis in contact lens for presbyopia'. Oral presentation at the 'PhDay congress VI edition' organized by the UCM, Madrid, Spain. 7 october 2022.

- ‘Development of e-Health technologies for Big Data Analysis in contact lens’. Posters presentations in the EVER international congress 2022, Valencia, Spain, From 13-15 October 2022. The abstracts of these posters were published in the Special Issue of Acta Ophthalmologica Journal: Abstracts from the 2022 European Association for Vision and Eye Research Festival, 13-15 October 2022, Valencia.
- ‘Development of an algorithm to track blinks from a face video’ poster presentation at the SIYO congress 2022 (online), organized by Universitat de Valencia, Spain. 14-28 November 2022. The poster presented in this congress was published in ‘Proceedings of the International Online Symposium of Young Optometrists. SIYO 2022’.
- EYE project representation in the fair ‘Science is wonderful’ held in Brussels on the 16-17th of March 2023. Jointly with ESR3 and ESR4 we proposed an explanation of the eye system to young students and conducted experiments and interactive games with them.
- ‘Development of e-Health technologies for Big Data Analysis: application to blink dynamics’. Oral presentation at the ‘PhDay congress VII edition’ organized by the UCM, Madrid, Spain. 6 October 2023.

FUNDING STATEMENT

This project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 956274.

7 REFERENCES

7. REFERENCES

1. Christensen K, Doblhammer G, Rau R, Vaupel JW. Ageing populations: the challenges ahead. The Lancet. 2009 Oct;374(9696):1196–208.

2. Bourne RRA, Flaxman SR, Braithwaite T, Cicinelli MV, Das A, Jonas JB, et al. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. Lancet Glob Health. 2017 Sep;5(9):e888–97.

3. Wolffsohn JS, Davies LN. Presbyopia: Effectiveness of correction strategies. Prog Retin Eye Res. 2019 Jan;68:124–43.

4. Lafosse E, Wolffsohn J, Talens-Estarelles C, García-Lázaro S. Presbyopia and the aging eye: Existing refractive approaches and their potential impact on dry eye signs and symptoms. Contact Lens Anterior Eye. 2020 Apr;43(2):103–14.

5. Bron AJ, Vrensen GFJM, Koretz J, Maraini G, Harding JJ. The Ageing Lens. Ophthalmologica. 2000;214(1):86–104.

6. Glasser A, Campbell MCW. Presbyopia and the optical changes in the human crystalline lens with age. Vision Res. 1998 Jan;38(2):209–29.

7. Sharma A, Hindman HB. Aging: A Predisposition to Dry Eyes. J Ophthalmol. 2014;2014:1–8.

8. Charman WN. Developments in the correction of presbyopia I: spectacle and contact lenses. Ophthalmic Physiol Opt. 2014 Jan;34(1):8–29.

9. Luo BP, Brown GC, Luo SC, Brown MM. The Quality of Life Associated with Presbyopia. Am J Ophthalmol. 2008 Apr;145(4):618-622.e1.

10. Rueff EM, Bailey MD. Presbyopic and non-presbyopic contact lens opinions and vision correction preferences. Contact Lens Anterior Eye. 2017 Oct;40(5):323–8.

11. Alipour F, Khaheshi S, Soleimanzadeh M, Heidarzadeh S. Contact Lens-related Complications: A Review. 2017;12(2).

12. Fonn D. Targeting Contact Lens Induced Dryness and Discomfort: What Properties Will Make Lenses More Comfortable. Optom Vis Sci. 2007 Apr;84(4):279–85.

13. Batlle-Ferrando S, Marín-Martínez S, Boniquet S, Sabater N. Complicaciones asociadas al uso de lentes de contacto blandas. Med Fam SEMERGEN. 2020 Apr;46(3):208–13.

14. Richdale K, Sinnott LT, Skadahl E, Nichols JJ. Frequency of and Factors Associated With Contact Lens Dissatisfaction and Discontinuation. Cornea. 2007 Feb;26(2):168–74.

15. Naroo SA, Nagra M, Retallic N. Exploring contact lens opportunities for patients above the age of 40 years. Contact Lens Anterior Eye. 2022 Dec;45(6):101599.

16. Pérez-prados R, Piñero DP, Pérez-cambrodí RJ, Madrid-costa D. Soft multifocal simultaneous image contact lenses: a review. Clin Exp Optom. 2017 Mar 1;100(2):107–27.

17. Nicolson PC, Vogt J. Soft contact lens polymers: an evolution. Biomaterials. 2001 Dec;22(24):3273–83.

18. Dimitrov DV. Medical Internet of Things and Big Data in Healthcare. Healthc Inform Res. 2016;22(3):156.

19. World Health Organization - Regional Office for the Eastern Mediterranean [Internet]. 2023 [cited 2023 Aug 24]. WHO EMRO | eHealth | Health topics. Available from: http://www.emro.who.int/health-topics/ehealth/

20. The top 10 causes of death [Internet]. 2023 [cited 2023 Aug 24]. Available from: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

21. Boulos M, Wheeler S, Tavares C, Jones R. How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX. Biomed Eng OnLine. 2011;10(1):24.

22. Martín CD, Fernández SB, Massip-Salcedo M. Aplicaciones móviles de control de peso y dieta saludable, ¿todas valen? 2018;11.

23. Carter MC, Burley VJ, Nykjaer C, Cade JE. Adherence to a Smartphone Application for Weight Loss Compared to Website and Paper Diary: Pilot Randomized Controlled Trial. J Med Internet Res. 2013 Apr 15;15(4):e32.

24. Carter MC, Burley VJ, Nykjaer C, Cade JE. 'My Meal Mate' (MMM): validation of the diet measures captured on a smartphone application to facilitate weight loss. Br J Nutr. 2013 Feb 14;109(3):539–46.

25. Cafazzo JA, Casselman M, Hamming N, Katzman DK, Palmert MR. Design of an mHealth App for the Self-management of Adolescent Type 1 Diabetes: A Pilot Study. J Med Internet Res. 2012 May 8;14(3):e70.

26. Donker T, Petrie K, Proudfoot J, Clarke J, Birch MR, Christensen H. Smartphones for Smarter Delivery of Mental Health Programs: A Systematic Review. J Med Internet Res. 2013 Nov 15;15(11):e247.

27. Firth J, Torous J, Nicholas J, Carney R, Rosenbaum S, Sarris J. Can smartphone mental health interventions reduce symptoms of anxiety? A meta-analysis of randomized controlled trials. J Affect Disord. 2017 Aug;218:15–22.

28. Fitzpatrick KK, Darcy A, Vierhile M. Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial. JMIR Ment Health. 2017 Jun 6;4(2):e19.

29. Haghi M, Thurow K, Stoll R. Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices. Healthc Inform Res. 2017;23(1):4.

30. Macana Castro TA, María Fernanda MF, Acuña Gómez JS, Jiménez Barbosa WG. Aplicaciones móviles de salud con respaldo en publicaciones científicas, para la mejora en el ejercicio de la optometría y la oftalmología. Cienc Tecnol Para Salud Vis Ocul. 2020 Mar 31;17(2):51–63.

31. Satgunam P, Thakur M, Sachdeva V, Reddy S, Rani P. Validation of visual acuity applications for teleophthalmology during COVID-19. Indian J Ophthalmol. 2021;69(2):385.

32. Ogino M, Salmerón-Campillo RM, Hunter S, Hussey V, Suh D, Gore R, et al. Clinical validation of a novel smartphone application for measuring best corrected visual acuity. J Optom. 2023 Jul;16(3):206–13.

33. Krishna R. A new ophthalmic educational resource for your smartphone.

34. Hossain IT, Malik HH, Franka M. Comment on: 'Effectiveness of a smartphone application for testing near visual acuity.' Eye. 2016 Jun;30(6):899–900.

35. CloudNine [Internet]. [cited 2024 Apr 24]. Available from: https://cloud9development.com/

36. EyeDock [Internet]. [cited 2024 Apr 24]. EyeDock. Available from: https://www.eyedock.com

37. Eye Scholar [Internet]. [cited 2024 Apr 24]. Available from: http://www.eyescholar.com/

38. Eyes On Eyecare [Internet]. 2020 [cited 2024 Apr 24]. EyeScholar: A New Tool for Learning How to Refract. Available from: https://eyesoneyecare.com/resources/eyescholar-new-tool-learn-to-refract/

39. Yeung PH, Lam AK. Fourth nerve palsy with monovision. Clin Exp Optom. 1998 Sep 10;81(5):206–9.

40. Eyes On Eyecare [Internet]. 2017 [cited 2024 Apr 24]. Optranslate – Optometry Translation App : Press Release. Available from: https://eyesoneyecare.com/resources/optranslate/

41. Bhanot S, Sharma A. App Review Series: Epocrates. J Digit Imaging. 2017 Oct;30(5):534–6.

42. Hong M, Shcherbakova N. Comparison of Discounted and Undiscounted Cash Prices for Cardiovascular Medications by Type of US Community Pharmacy. J Gen Intern Med. 2021 Jan;36(1):114–20.

43. GoodRx [Internet]. [cited 2024 Apr 24]. Prescription Prices, Coupons & Pharmacy Information. Available from: https://www.goodrx.com/

44. Eyes On Eyecare [Internet]. 2019 [cited 2024 Apr 24]. OCTaVIA: Your Pocket OCT Visual Reference. Available from: https://eyesoneyecare.com/resources/octavia-pocket-oct-visual-reference/

45. New England College of Optometry [Internet]. 2021 [cited 2024 Apr 24]. Professor's smartphone app, OCTaVIA, upgraded. Available from: https://www.neco.edu/news/professors-app-octavia-upgraded/

46. Goseki T, Kunimi K, Shioya N, Iijima Y, Sebe M, Hosoya K, et al. New device for taking nine-directional ocular photographs: "9Gaze" application. J Eye Mov Res [Internet]. 2022 Mar 6 [cited 2024 Apr 24];15(1). Available from: https://bop.unibe.ch/JEMR/article/view/8152

47. https://fyra.io. Retina Today. Bryn Mawr Communications; [cited 2024 Apr 24]. The New Eyetube. Available from: https://retinatoday.com/articles/2016-may-june/the-new-eyetube

48. Feyenally [Internet]. 2021 [cited 2024 Apr 24]. Available from: https://feyenally.com/

49. Steren BJ, Young B, Chow J. Visual Acuity Testing for Telehealth Using Mobile Applications. JAMA Ophthalmol. 2021 Mar 1;139(3):344.

50. Cheng NM, Chakrabarti R, Kam JK. iPhone Applications for Eye Care Professionals: A Review of Current Capabilities and Concerns. Telemed E-Health. 2014 Apr;20(4):385–7.

51. Gonzalez RC, Woods RE. Digital image processing. Fourth edition. New York, NY: Pearson; 2018. 1168 p.

52. Canny J. A Computational Approach to Edge Detection. IEEE Trans Pattern Anal Mach Intell. 1986 Nov;PAMI-8(6):679–98.

53. Chenyang Xu, Prince JL. Snakes, shapes, and gradient vector flow. IEEE Trans Image Process. 1998 Mar;7(3):359–69.

54. Lefevre T, Dorizzi B, Garcia-Salicetti S, Lemperiere N, Belardi S. Effective elliptic fitting for iris normalization. Comput Vis Image Underst. 2013 Jun;117(6):732–45.

55. Sankur B. Survey over image thresholding techniques and quantitative performance evaluation. J Electron Imaging. 2004 Jan 1;13(1):146.

56. Otsu N. A Threshold Selection Method from Gray-Level Histograms. IEEE Trans Syst Man Cybern. 1979 Jan;9(1):62–6.

57. Bradley D, Roth G. Adaptive Thresholding using the Integral Image. J Graph Tools. 2007 Jan;12(2):13–21.

58. Swain MJ, Ballard DH. Color indexing. Int J Comput Vis. 1991 Nov;7(1):11–32.

59. Jaini K, FARROKHNIA F. UNSUPERVISED TEXTURE SEGMENTATION USING GABOR FILTERS. 1991;

60. Shiode N, Shiode S, Rod-Thatcher E, Rana S, Vinten-Johansen P. The mortality rates and the space-time patterns of John Snow's cholera epidemic map. Int J Health Geogr. 2015 Dec;14(1):21.

61. Yozwiak NL, Schaffner SF, Sabeti PC. Data sharing: Make outbreak research open access. Nature. 2015 Feb 26;518(7540):477–9.

62. Udler MS, Kim J, Von Grotthuss M, Bonàs-Guarch S, Cole JB, Chiou J, et al. Type 2 diabetes genetic loci informed by multi-trait associations point to disease mechanisms and subtypes: A soft clustering analysis. Langenberg C, editor. PLOS Med. 2018 Sep 21;15(9):e1002654.

63. Collett D. Modelling Binary Data [Internet]. 0 ed. Chapman and Hall/CRC; 2002 [cited 2024 Jun 24]. Available from: https://www.taylorfrancis.com/books/9781420057386

64. Malkov YA, Yashunin DA. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. IEEE Trans Pattern Anal Mach Intell. 2020 Apr 1;42(4):824–36.

65. Cristianini N, Shawe-Taylor J. An introduction to support vector machines: and other kernel-based learning methods. Cambridge ; New York: Cambridge University Press; 2000. 189 p.

66. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010.

67. Manning CD, Raghavan P, Schütze H. Introduction to information retrieval. Cambridge: Cambridge university press; 2008.

68. Fisher RA. THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. Ann Eugen. 1936 Sep;7(2):179–88.

69. Fix E, Hodges JL. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. Int Stat Rev Rev Int Stat. 1989 Dec;57(3):238.

70. Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification And Regression Trees [Internet]. 1st ed. Routledge; 2017 [cited 2024 Jun 24]. Available from: https://www.taylorfrancis.com/books/9781351460491

71. Breiman L. Random Forests. Mach Learn. 2001;45(1):5–32.

72. Skurichina M, Duin RPW. Bagging for linear classifiers. Pattern Recognit. 1998 Jul;31(7):909–30.

73. Dumouchel W, O'brien F. Integrating a Robust Option into a Multiple Regression Computing Environment. In: Buja A, Tukey PA, editors. Computing and Graphics in Statistics [Internet]. New York, NY: Springer New York; 1991 [cited 2024 Jun 24]. p. 41–8. (Friedman A, Miller W, editors. The IMA Volumes in Mathematics and its Applications; vol. 36). Available from: http://link.springer.com/10.1007/978-1-4613-9154-8_3

74. Holland PW, Welsch RE. Robust regression using iteratively reweighted least-squares. Commun Stat - Theory Methods. 1977 Jan;6(9):813–27.

75. Seber GAF, Wild CJ. Nonlinear Regression [Internet]. 1st ed. Wiley; 1989 [cited 2024 Jun 24]. (Wiley Series in Probability and Statistics). Available from: https://onlinelibrary.wiley.com/doi/book/10.1002/0471725315

76. Rasmussen CE. Gaussian Processes in Machine Learning. In: Bousquet O, Von Luxburg U, Rätsch G, editors. Advanced Lectures on Machine Learning [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004 [cited 2024 Jun 24]. p. 63–71. (Lecture Notes in Computer Science; vol. 3176). Available from: http://link.springer.com/10.1007/978-3-540-28650-9_4

77. Chen PH, Fan RE, Lin CJ. A Study on SMO-Type Decomposition Methods for Support Vector Machines. IEEE Trans Neural Netw. 2006 Jul;17(4):893–908.

78. Fan RE, Chen PH, Lin CJ. Working Set Selection Using Second Order Information for Training Support Vector Machines. 2005;

79. McCullagh P, Nelder JA. Generalized Linear Models [Internet]. 2nd ed. Routledge; 2019 [cited 2024 Jun 24]. Available from: https://www.taylorfrancis.com/books/9781351445856

80. Bull AD. Convergence rates of efficient global optimization algorithms [Internet]. arXiv; 2011 [cited 2024 Jun 25]. Available from: http://arxiv.org/abs/1101.3501

81. Gelbart MA, Snoek J, Adams RP. Bayesian Optimization with Unknown Constraints [Internet]. arXiv; 2014 [cited 2024 Jun 25]. Available from: http://arxiv.org/abs/1403.5607

82. Nyström M, Andersson R, Niehorster DC, Hessels RS, Hooge ITC. What is a blink? Classifying and characterizing blinks in eye openness signals. Behav Res Methods. 2024 Feb 29;56(4):3280–99.

83. Leigh RJ, Zee DS. The neurology of eye movements. 5th ed. Oxford: Oxford university press; 2015. (Contemporary neurology series).

84. Eckstein MK, Guerra-Carrillo B, Miller Singley AT, Bunge SA. Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? Dev Cogn Neurosci. 2017 Jun;25:69–91.

85. Stern JA, Walrath LC, Goldstein R. The Endogenous Eyeblink. Psychophysiology. 1984 Jan;21(1):22–33.

86. Caffier PP, Erdmann U, Ullsperger P. Experimental evaluation of eye-blink parameters as a drowsiness measure. Eur J Appl Physiol. 2003 May;89(3):319–25.

87. Benedetto S, Pedrotti M, Minin L, Baccino T, Re A, Montanari R. Driver workload and eye blink duration. Transp Res Part F Traffic Psychol Behav. 2011 May;14(3):199–208.

88. Cruz AAV, Garcia DM, Pinto CT, Cechetti SP. Spontaneous Eyeblink Activity. Ocul Surf. 2011 Jan;9(1):29–41.

89. Hoppe D, Helfmann S, Rothkopf CA. Humans quickly learn to blink strategically in response to environmental task demands. Proc Natl Acad Sci. 2018 Feb 27;115(9):2246–51.

90. Lenskiy A, Paprocki R. Blink Rate Variability during resting and reading sessions [Internet]. arXiv; 2016 [cited 2023 Feb 16]. Available from: http://arxiv.org/abs/1605.02037

91. Shin YS, Chang W du, Park J, Im CH, Lee SI, Kim IY, et al. Correlation between Inter-Blink Interval and Episodic Encoding during Movie Watching. Thompson B, editor. PLOS ONE. 2015 Nov 3;10(11):e0141242.

92. Navascues-Cornago M, Sun T, Read ML, Morgan PB. The short-term effect of contact lens wear on blink characteristics. Contact Lens Anterior Eye. 2022 Oct;45(5):101596.

93. López-de La Rosa A, Martín-Montañez V, López-Miguel A, Fernández I, Calonge M, González-Méijome JM, et al. Ocular response to environmental variations in contact lens wearers. Ophthalmic Physiol Opt. 2017 Jan;37(1):60–70.

94. Bron AJ, Tomlinson A, Foulks GN, Pepose JS, Baudouin C, Geerling G, et al. Rethinking Dry Eye Disease: A Perspective on Clinical Implications. Ocul Surf. 2014 Apr;12(2):S1–31.

95. The Epidemiology of Dry Eye Disease: Report of the Epidemiology Subcommittee of the International Dry Eye WorkShop (2007). Ocul Surf. 2007 Apr;5(2):93–107.

96. Nagino K, Okumura Y, Yamaguchi M, Sung J, Nagao M, Fujio K, et al. Diagnostic Ability of a Smartphone App for Dry Eye Disease: Protocol for a Multicenter, Open-Label, Prospective, and Cross-sectional Study. JMIR Res Protoc. 2023 Mar 13;12:e45218.

97. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001 [Internet]. Kauai, HI, USA: IEEE Comput. Soc; 2001 [cited 2023 Nov 24]. p. I-511-I–518. Available from: http://ieeexplore.ieee.org/document/990517/

98. Viola P, Jones MJ. Robust Real-Time Face Detection. 2004;

99. Prof. Assit, Student MSc. Efficient Eye Blink Detection Method for disabled-helping domain. Int J Adv Comput Sci Appl [Internet]. 2014 [cited 2023 Apr 15];5(5). Available from: http://thesai.org/Publications/ViewPaper?Volume=5&Issue=5&Code=IJACSA&SerialNo=30

100. Attivissimo F, D'Alessandro VI, Di Nisio A, Scarcelli G, Schumacher J, Lanzolla AML. Performance evaluation of image processing algorithms for eye blinking detection. Measurement. 2023 Dec;223:113767.

101. Fathi A, Abdali-Mohammadi F. Camera-based eye blinks pattern detection for intelligent mouse. Signal Image Video Process. 2015 Nov;9(8):1907–16.

102. Nie B, Huang X, Chen Y, Li A, Zhang R, Huang J. Experimental study on visual detection for fatigue of fixed-position staff. Appl Ergon. 2017 Nov;65:1–11.

103. Biondi FN, Saberi B, Graf F, Cort J, Pillai P, Balasingam B. Distracted worker: Using pupil size and blink rate to detect cognitive load during manufacturing tasks. Appl Ergon. 2023 Jan;106:103867.

104. Ramalho D, Constantino P, Silva HPD, Constante M, Sanches J. An Augmented Teleconsultation Platform for Depressive Disorders. IEEE Access. 2022;10:130563–71.

105. Dewi C, Chen RC, Jiang X, Yu H. Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. PeerJ Comput Sci. 2022 Apr 18;8:e943.

106. Kamarudin N, Jumadi NA, Mun NL, Keat NC, Ching AHK, Mahmud WMHW, et al. Implementation of Haar Cascade Classifier and Eye Aspect Ratio for Driver Drowsiness Detection Using Raspberry Pi. Univers J Electr Electron Eng. 2019 Dec;6(5B):67–75.

107. Sukumaran A, Manoharan A. Multimodal Engagement Recognition From Image Traits Using Deep Learning Techniques. IEEE Access. 2024;12:25228–44.

108. Yi Y, Zhang H, Zhang W, Yuan Y, Li C. Fatigue Working Detection Based on Facial Multifeature Fusion. IEEE Sens J. 2023 Mar 15;23(6):5956–61.

109. Karmakar S, Chatterjee D, Varghese T, Gavas RD, S MB, Ramakrishnan RK, et al. Quantification of Active Visual Attention using RGB camera. In: 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) [Internet]. Sydney, Australia: IEEE; 2023 [cited 2024 Apr 19]. p. 1–4. Available from: https://ieeexplore.ieee.org/document/10340011/

110. Li K, Gong Y, Ren Z. A Fatigue Driving Detection Algorithm Based on Facial Multi-Feature Fusion. IEEE Access. 2020;8:101244–59.

111. Al-gawwam S, Benaissa M. Robust Eye Blink Detection Based on Eye Landmarks and Savitzky–Golay Filtering. Information. 2018 Apr 15;9(4):93.

112. Ronca V, Giorgi A, Rossi D, Di Florio A, Di Flumeri G, Aricò P, et al. A Video-Based Technique for Heart Rate and Eye Blinks Rate Estimation: A Potential Solution for Telemonitoring and Remote Healthcare. Sensors. 2021 Feb 25;21(5):1607.

113. John SJ, Sharmila ST. Real time blink recognition from various head pose using single eye. Multimed Tools Appl. 2018 Dec;77(23):31331–45.

114. Poursadeghiyan M, Mazloumi A, Saraji GN, Baneshi MM, Khammar A, Ebrahimi MH. Using Image Processing in the Proposed Drowsiness Detection System Design. Iran J Public Health. 2018;47.

115. AlKishri W, Abualkishik A, Al-Bahri M. Enhanced Image Processing and Fuzzy Logic Approach for Optimizing Driver Drowsiness Detection. Ranjan Nayak S, editor. Appl Comput Intell Soft Comput. 2022 Mar 19;2022:1–14.

116. Jasim SS, Abdul Hassan AK, Turner S. Driver Drowsiness Detection Using Gray Wolf Optimizer Based on Face and Eye Tracking. ARO- Sci J KOYA Univ. 2022 May 5;10(1):49–56.

117. Mohammad F, Mahadas K, Hung GK. Drowsy driver mobile application: Development of a novel scleral-area detection method. Comput Biol Med. 2017 Oct;89:76–83.

118. Bamidele AA, Kamardin K, Syazarin N, Mohd S, Shafi I, Azizan A, et al. Non-intrusive Driver Drowsiness Detection based on Face and Eye Tracking. Int J Adv Comput Sci Appl [Internet]. 2019 [cited 2024 Mar 28];10(7). Available from: http://thesai.org/Publications/ViewPaper?Volume=10&Issue=7&Code=IJACSA&SerialNo=75

119. Jang SW, Ahn B. Implementation of Detection System for Drowsy Driving Prevention Using Image Recognition and IoT. Sustainability. 2020 Apr 10;12(7):3037.

120. Morcego B, Argilés M, Cabrerizo M, Cardona G, Pérez R, Pérez-Cabré E, et al. Blinking supervision in a working environment. J Biomed Opt. 2016 Feb 2;21(2):025005.

121. Yauri-Machaca M, Meneses-Claudio B, Vargas-Cuentas N, Roman-Gonzalez A. Design of a Vehicle Driver Drowsiness Detection System Through Image Processing using Matlab. In: 2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII) [Internet]. San Salvador: IEEE; 2018 [cited 2024 Apr 15]. p. 1–6. Available from: https://ieeexplore.ieee.org/document/8596513/

122. Singh H, Singh J. Real-time eye blink and wink detection for object selection in HCI systems. J Multimodal User Interfaces. 2018 Mar;12(1):55–65.

123. Kitazawa M, Yoshimura M, Liang KC, Wada S, Mimura M, Tsubota K, et al. Utilization of Facial Image Analysis Technology for Blink Detection: A Validation Study. Eye Contact Lens Sci Clin Pract. 2018 Nov;44(2):S297–301.

124. Sato H, Abe K, Ohi S, Ohyama M. An Automatic Classification Method for Involuntary and Two Types of Voluntary Blinks. Electron Commun Jpn. 2017 Oct;100(10):48–58.

125. Sato H, Abe K, Matsuno S, Ohyama M. Blink input interface enabling multiple candidate selection through sound feedback. Artif Life Robot. 2021 Aug;26(3):312–7.

126. Nousias G, Panagiotopoulou EK, Delibasis K, Chaliasou AM, Tzounakou AM, Labiris G. Video-Based Eye Blink Identification and Classification. IEEE J Biomed Health Inform. 2022 Jul;26(7):3284–93.

127. Tansakul W, Tangamchit P. Fatigue Driver Detection System Using a Combination of Blinking Rate and Driving Inactivity. J Autom Control Eng. 2015;3(6):33–9.

128. Abiyev RH, Arslan M. Head mouse control system for people with disabilities. Expert Syst. 2020 Feb;37(1):e12398.

129. Vishesh P, S R, Jankatti S, V R. Eye blink detection using CNN to detect drowsiness level in drivers for road safety. Indones J Electr Eng Comput Sci. 2021 Apr 1;22(1):222.

130. Pan G, Wu Z, Su L. Liveness Detection for Face Recognition. In: Delac K, Grgic M, Stewart M, editors. Recent Advances in Face Recognition [Internet]. InTech; 2008 [cited 2024 May 10]. Available from: http://www.intechopen.com/books/recent_advances_in_face_recognition/liveness_detection_for_face_recognition

131. Bobbia S, Macwan R, Benezeth Y, Mansouri A, Dubois J. Unsupervised skin tissue segmentation for remote photoplethysmography. Pattern Recognit Lett. 2019 Jun;124:82–90.

132. Gullapalli AR, Anderson NE, Yerramsetty R, Harenski CL, Kiehl KA. In the blink of an eye: Quantitative blink dynamics predict deceptive personality traits in forensic interviews. Personal Individ Differ. 2021 Jul;176:110764.

133. Fogelton A, Benesova W. Eye blink detection based on motion vectors analysis. Comput Vis Image Underst. 2016 Jul;148:23–33.

134. Bergasa LM, Nuevo J, Sotelo MA, Vhzquez M. Weal-Time System for Monitoring Driver Vigilance. 2004;

135. Matjaz Divjak and Horst Bischof. Eye blink based fatigue detection for prevention of Computer Vision Syndrome. In: Proceedings of IAPR Conference on Machine Vision Applications (MVA). 2009. p. 350--353.

136. A. Alashbi A, Sunar MS, Alqahtani Z. Deep-Learning-CNN for Detecting Covered Faces with Niqab. J Inf Technol Manag [Internet]. 2022 Jan [cited 2024 Apr 26];14(5th International Conference of Reliable Information and Communication Technology (IRICT 2020)). Available from: https://doi.org/10.22059/jitm.2022.84888

137. Song F, Tan X, Liu X, Chen S. Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. Pattern Recognit. 2014 Sep;47(9):2825–38.

138. Lee WO, Lee EC, Park KR. Blink detection robust to various facial poses. J Neurosci Methods. 2010 Nov;193(2):356–72.

139. Danisman T, Bilasco IM, Djeraba C, Ihaddadene N. Drowsy driver detection system using eye blink patterns. In: 2010 International Conference on Machine and Web Intelligence [Internet]. Algiers, Algeria: IEEE; 2010 [cited 2024 May 5]. p. 230–3. Available from: http://ieeexplore.ieee.org/document/5648121/

140. Drutarovsky T, Fogelton A. Eye Blink Detection Using Variance of Motion Vectors. In: Agapito L, Bronstein MM, Rother C, editors. Computer Vision - ECCV 2014 Workshops [Internet]. Cham: Springer International Publishing; 2015 [cited 2023 Feb 14]. p. 436–48. (Lecture Notes in Computer Science; vol. 8927). Available from: http://link.springer.com/10.1007/978-3-319-16199-0_31

141. Biondi FN, Graf F, Pillai P, Balasingam B. On validating a generic camera-based blink detection system for cognitive load assessment. Cogn Comput Syst. 2023 Dec;5(4):255–64.

142. Soukupova T. Real-Time Eye Blink Detection using Facial Landmarks. 2016;

143. Dewi C, Chen RC, Chang CW, Wu SH, Jiang X, Yu H. Eye Aspect Ratio for Real-Time Drowsiness Detection to Improve Driver Safety. Electronics. 2022 Oct 4;11(19):3183.

144. Al-Razgan MS, Alruwaly I, Ali YA. Eye-Blink Event Detection Using a Neural-Network-Trained Frame Segment for Woman Drivers in Saudi Arabia. Electronics. 2023 Jun 16;12(12):2699.

145. Sridevi V, Roshan Srivatsa S, Premnath R, Naresh K. Proactive integrated detection to identify sleepy driver and RFID based auto zone detection for speed control. J Chem Pharm Sci. 2015;2015-March:79–82.

146. Geng Lei 耿 磊, Liang Xiaoyu 梁晓昱, Xiao Zhitao 肖志涛, Li Yuelong 李月龙. Real-time driver fatigue detection based on morphology infrared features and deep learning. Infrared Laser Eng. 2018;47(2):203009.

147. Tychsen-Smith L, Petersson L. Improving Object Localization with Fitness NMS and Bounded IoU Loss [Internet]. arXiv; 2018 [cited 2024 May 5]. Available from: http://arxiv.org/abs/1711.00164

148. Sekar SP, Sathiaseelan J. REAL-TIME ABNORMAL EYE BLINKING DETECTION USING Y-UNET. Int J Sci Eng Res. 2022 Mar 24;4:1357–63.

149. Revelo A, Alvarez R, Grijalva F. Human Drowsiness Detection In Real Time, Using Computer Vision. In: 2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM) [Internet]. Guayaquil, Ecuador: IEEE; 2019 [cited 2024 May 7]. p. 1–6. Available from: https://ieeexplore.ieee.org/document/9014884/

150. Rukhsar Khan, Menon S, Shivraj Patil, Anchan S, Saritha L. R. Human Drowsiness Detection System. 2023 [cited 2024 May 8]; Available from: https://rgdoi.net/10.13140/RG.2.2.24824.01285

151. HTML Standard [Internet]. 2023 [cited 2023 Aug 21]. Available from: https://html.spec.whatwg.org/multipage/

152. Cascading Style Sheets [Internet]. 2023 [cited 2023 Aug 21]. Available from: https://www.w3.org/Style/CSS/

153. Pesudovs K, Garamendi E, Elliott DB. The Contact Lens Impact on Quality of Life (CLIQ) Questionnaire: Development and Validation. Investig Opthalmology Vis Sci. 2006 Jul 1;47(7):2789.

154. Khadka J, McAlinden C, Pesudovs K. Quality Assessment of Ophthalmic Questionnaires: Review and Recommendations. Optom Vis Sci. 2013;90(8).

155. Pesudovs K, Garamendi E, Elliott DB. The Quality of Life Impact of Refractive Correction (QIRC) Questionnaire: Development and Validation: Optom Vis Sci. 2004 Oct;81(10):769–77.

156. Genereaux BW, Dennison DK, Ho K, Horn R, Silver EL, O'Donnell K, et al. DICOMweb™: Background and Application of the Web Standard for Medical Imaging. J Digit Imaging. 2018 Jun;31(3):321–6.

157. González DL. Lambda function [Internet]. 2019 [cited 2023 Aug 28]. Available from: https://github.com/teleyinex/airemadrid

158. Calidad del aire. Datos en tiempo real - Portal de datos abiertos del Ayuntamiento de Madrid [Internet]. [cited 2023 Aug 28]. Available from: https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=41e01e007c9db410VgnVCM2000000c205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD

159. Meteorología AE de. Nota legal - Agencia Estatal de Meteorología - AEMET. Gobierno de España [Internet]. [cited 2023 May 16]. Available from: https://www.aemet.es/es/nota_legal

160. MediaDevices - Referencia de la API Web | MDN [Internet]. 2023 [cited 2023 Aug 22]. Available from: https://developer.mozilla.org/es/docs/Web/API/MediaDevices

161. Python.org [Internet]. [cited 2024 Sep 6]. Python Release Python 3.12.0. Available from: https://www.python.org/downloads/release/python-3120/

162. dlib C++ Library [Internet]. [cited 2024 Aug 31]. Available from: http://dlib.net/

163. Dinh H, Jovanov E, Adhami R. Eye Blink Detection Using Intensity Vertical Projection. 2012. 2012;

164. López-Alonso JM, Alda J, Bernabéu E. Principal-component characterization of noise for infrared images. Appl Opt. 2002 Jan 10;41(2):320.

165. Lo´pez-Alonso JM. Bad pixel identification by means of principal components analysis. Opt Eng. 2002 Sep 1;41(9):2152.

166. Knee Point [Internet]. 2025 [cited 2025 Jan 12]. Available from: https://es.mathworks.com/matlabcentral/fileexchange/35094-knee-point

167. Thomas C, Sheldon B. The Knee of a CurveUseful Clue but Incomplete Support. Mil Oper Res. 1999 Mar 1;4(2):17–24.

168. pca - Principal component analysis of raw data - MATLAB [Internet]. [cited 2024 Oct 15]. Available from: https://www.mathworks.com/help/stats/pca.html#d126e881538

169. Leys C, Ley C, Klein O, Bernard P, Licata L. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. J Exp Soc Psychol. 2013 Jul;49(4):764–6.

170. Andrews LC. Special functions of mathematics for engineers. 2nd ed. Bellingham, Wash. <1000 20th St. Bellingham WA 98225-6705 USA>: SPIE; 1998. 1 p. (SPIE Press monograph).

171. Espinosa J, Domenech B, Vázquez C, Pérez J, Mas D. Blinking characterization from high speed video records. Application to biometric authentication. Sakakibara M, editor. PLOS ONE. 2018 May 7;13(5):e0196125.

172. Espinosa J, Pérez J, Mas D. Comparative analysis of spontaneous blinking and the corneal reflex. R Soc Open Sci. 2020 Dec;7(12):201016.

173. Espinosa J, Martínez M, Perez J, Domenech B, Vazquez C, Mas D. High-speed video analysis of spontaneous and reflex to light blinking. In: Schelkens P, Kozacki T, editors. Optics, Photonics and Digital Technologies for Imaging Applications VI [Internet]. Online Only, France: SPIE; 2020 [cited 2024 Jul 19]. p. 67. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11353/2555403/High-speed-video-analysis-of-spontaneous-and-reflex-to-light/10.1117/12.2555403.full

174. Lacouture Y, Cousineau D. How to use MATLAB to fit the ex-Gaussian and other probability functions to a distribution of response times. Tutor Quant Methods Psychol. 2008 Mar 1;4(1):35–45.

175. Oh J, Jeong SY, Jeong J. The timing and temporal patterns of eye blinking are dynamically modulated by attention. Hum Mov Sci. 2012 Dec;31(6):1353–65.

176. Szendro P, Vincze G, Szasz A. Pink-noise behaviour of biosystems. Eur Biophys J. 2001 Jul 1;30(3):227–31.

177. Kaulakys B. On the intrinsic origin of 1/f noise. Microelectron Reliab. 2000 Nov;40(11):1787–90.

178. Hausdorff JM, Peng CK. Multiscaled randomness: A possible source of 1/ *f* noise in biology. Phys Rev E. 1996 Aug 1;54(2):2154–7.

179. Bak P, Tang C, Wiesenfeld K. Self-organized criticality: An explanation of the 1/ *f* noise. Phys Rev Lett. 1987 Jul 27;59(4):381–4.

180. Hanning NM, Deubel H. A dynamic 1/f noise protocol to assess visual attention without biasing perceptual processing. Behav Res Methods. 2022 Aug 1;55(5):2583–94.

181. Duchowski A, Jörg S, Lawson A, Bolte T, Świrski L, Krejtz K. Eye movement synthesis with 1/ *f* pink noise. In: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games [Internet]. Paris France: ACM; 2015 [cited 2025 Feb 2]. p. 47–56. Available from: https://dl.acm.org/doi/10.1145/2822013.2822014

182. Szendro P, Vincze G, Szasz A. BIO-RESPONSE TO WHITE NOISE EXCITATION. Electro- Magnetobiology. 2001 Jan;20(2):215–29.

183. Chikri YM, Alonso JML, Garzón N, Gonzalez-Perez M. development of e-Health technologies for big data analysis in contact lens. Acta Ophthalmol (Copenh) [Internet]. 2022 [cited 2024 Nov 11];100(S275). Available from: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1755-3768.2022.0371

184. Guirron D, Simard R. Upper Eyelid Movements Measured With a Search Coil During Blinks and Vertical Saccades. Invest Ophthalmol. 1991;32(13).

185. Pult H, Riede-Pult BH, Murphy PJ. A New Perspective on Spontaneous Blinks. Ophthalmology. 2013 May;120(5):1086–91.

186. Sun WS, Baker RS, Chuke JC, Rouholiman BR, Hasan SA, Gaza W, et al. Age-Related Changes in Human Blinks. 1997;

187. findpeaks - Encontrar los máximos locales - MATLAB - MathWorks España [Internet]. [cited 2024 Oct 1]. Available from: https://es.mathworks.com/help/signal/ref/findpeaks.html

188. Moreno SO. IMAGEN COMO HERRAMIENTA DE APOYO EN DIAGNOSIS OCULAR. Proy Fin Máster En Tecnol Ópticas Imagen. 2018 2019;Facultad de Óptica y Optometría, Universidad Complutense de Madrid.

189. Abramowitz M, Stegun IA, editors. Handbook of mathematical functions: with formulas, graphs, and mathematical tables. 9. Dover print.; [Nachdr. der Ausg. von 1972]. New York, NY: Dover Publ; 2013. 1046 p. (Dover books on mathematics).

190. Fairchild MD. Color appearance models. 2nd ed. Chichester, West Sussex, England ; Hoboken, NJ: J. Wiley; 2005. 385 p. (Wiley-IS&T series in imaging science and technology).

191. Gibbons JD, Chakraborti S. Nonparametric Statistical Inference [Internet]. 0 ed. Chapman and Hall/CRC; 2010 [cited 2025 Feb 7]. Available from: https://www.taylorfrancis.com/books/9781439896129

192. Massey FJ. The Kolmogorov-Smirnov Test for Goodness of Fit. J Am Stat Assoc. 1951 Mar;46(253):68–78.

193. Horowitz JL. Bootstrap Methods in Econometrics. Annu Rev Econ. 2019 Aug 2;11(1):193–224.

194. Reprint of: Mahalanobis, P.C. (1936) “On the Generalised Distance in Statistics.” Sankhya A. 2018 Dec;80(S1):1–7.

195. Hu Z, Xiao M, Zhang L, Liu S, Ge Y. Mahalanobis Distance Based Approach for Anomaly Detection of Analog Filters Using Frequency Features and Parzen Window Density Estimation. J Electron Test. 2016 Dec;32(6):681–93.

196. Podolskiy A, Lipin D, Bout A, Artemova E, Piontkovskaya I. Revisiting Mahalanobis Distance for Transformer-Based Out-of-Domain Detection. Proc AAAI Conf Artif Intell. 2021 May 18;35(15):13675–82.

197. Bisgard J. Analysis and linear algebra: the singular value decomposition and applications. Online-Ausg. Providence, Rhode Island: American Mathematical Society; 2021. 1 p. (Student Mathematical Library).

198. Wall ME, Rechtsteiner A, Rocha LM. Singular Value Decomposition and Principal Component Analysis. In: Berrar DP, Dubitzky W, Granzow M, editors. A Practical Approach to Microarray Data Analysis [Internet]. Boston: Kluwer Academic Publishers; 2003 [cited 2025 Feb 18]. p. 91–109. Available from: http://link.springer.com/10.1007/0-306-47815-3_5

199. Golub G, Kahan W. Calculating the Singular Values and Pseudo-Inverse of a Matrix. J Soc Ind Appl Math Ser B Numer Anal. 1965 Jan;2(2):205–24.

200. Liu L, Wang P, Lin J, Liu L. Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning. IEEE Access. 2021;9:7550–63.

201. Ji Z, Zou X, Lin X, Liu X, Huang T, Wu S. An Attention-Driven Two-Stage Clustering Method for Unsupervised Person Re-identification. In: Vedaldi A, Bischof H, Brox T, Frahm JM, editors. Computer Vision – ECCV 2020 [Internet]. Cham: Springer International Publishing; 2020 [cited 2025 Feb 18]. p. 20–36. (Lecture Notes in Computer Science; vol. 12373). Available from: https://link.springer.com/10.1007/978-3-030-58604-1_2

202. JinHuaXu, HongLiu. Web user clustering analysis based on KMeans algorithm. In: 2010 International Conference on Information, Networking and Automation (ICINA) [Internet]. Kunming, China: IEEE; 2010 [cited 2025 Feb 18]. p. V2-6-V2-9. Available from: http://ieeexplore.ieee.org/document/5636772/

203. Ng HP, Ong SH, Foong KWC, Goh PS, Nowinski WL. Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm. In: 2006 IEEE Southwest Symposium on Image Analysis and Interpretation [Internet]. Denver, CO: IEEE; 2006 [cited 2025 Feb 18]. p. 61–5. Available from: http://ieeexplore.ieee.org/document/1633722/

204. Calinski T, Harabasz J. A dendrite method for cluster analysis. Commun Stat - Theory Methods. 1974;3(1):1–27.

205. Jolliffe IT. Principal Component Analysis. 2nd ed. New York, NY: Springer New York; 2002. 1 p. (Springer Series in Statistics Ser).

206. Fernandes PRB, Neves HIF, Lopes-Ferreira DP, Jorge JMM, González-Meijome JM. Adaptation to Multifocal and Monovision Contact Lens Correction. Optom Vis Sci. 2013 Mar;90(3):228–35.

207. Lazon De La Jara P, Sulley A, Pepe P, Walsh K, Guillon M. Multifocal contact lens success predictability. Contact Lens Anterior Eye. 2024 Apr;47(2):102105.

208. Ribeiro M, Vieira MS, Gorgone G, Barbosa LYC, Martini ARAF, David MA, et al. The contact lens dry eyes questionnaire (CLDEQ-8) validation and ocular surface dysfunction among soft contact lens wearers. Arq Bras Oftalmol [Internet]. 2022 [cited 2023 Dec 2];85(1). Available from: http://aboonline.org.br/details/6160/en-US/the-contact-lens-dry-eyes-questionnaire--cldeq-8--validation-and-ocular-surface-dysfunction-among-soft-contact-lens-wearers

209. Statistics and Machine Learning Toolbox [Internet]. [cited 2025 Mar 2]. Available from: https://www.mathworks.com/help/stats/index.html

210. Bergeron F, Labelle G, Leroux P. Combinatorial species and tree-like structures. Cambridge ; New York, NY, USA: Cambridge University Press; 1998. 457 p. (Encyclopedia of mathematics and its applications).

211. Classification Learner App [Internet]. [cited 2025 Mar 2]. Available from: https://www.mathworks.com/help/stats/classification-learner-app.html

212. Stone M. Cross-Validatory Choice and Assessment of Statistical Predictions. J R Stat Soc Ser B Stat Methodol. 1974 Jan 1;36(2):111–33.

213. Hocking RR. A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear Regression. Biometrics. 1976 Mar;32(1):1.

214. Flom PL, Development N, Institutes R, York N. Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use. 2007;

215. Pult H, Riede-Pult BH, Murphy PJ. The Relation Between Blinking and Conjunctival Folds and Dry Eye Symptoms. Optom Vis Sci. 2013 Oct;90(10):1034–9.

8 ANNEX

8. ANNEX

8.1 Protocol for data recollection for patients and app users.

DESARROLLO DE TECNOLOGÍAS E-SALUD PARA ANÁLISIS Y TRATAMIENTO DE DATOS EN PORTADORES DE LENTES DE CONTACTO.

Protocolo APP-EYEH2020

- **Versión 1 Mayo 2023**

FACULTAD DE OPTICA Y OPTOMETRÍA

Programa de doctorado en Óptica, Optometría y Visión
Universidad Complutense de Madrid

- # 1. Resumen

- ***0 Tipo de solicitud.***

Estudio observacional descriptivo para comprobar la precisión de tecnologías basadas en App para móviles (App eyeh2020), utilizadas para la obtención de parámetros físicos del ojo (variables de parpadeo, color esclera, posición y tamaño pupilar), a partir de videos de los mismos, en portadores de lentes de contacto. La calidad de visión se puede evaluar a partir del seguimiento de la velocidad de lectura y de medida de la agudeza visual mediante la visualización de optotipos en el móvil. La comodidad del paciente se evalúa realizando una serie de preguntas mientras usa lentes de contacto.

- ***1 Promotor.***

Facultad de Óptica y Optometría. Universidad Complutense de Madrid.

- ***2 Título del estudio.***

Desarrollo de tecnologías e-Salud para análisis y tratamiento de datos en portadores de lentes de contacto.

- ***3 Código del protocolo.***

APP-EYEH2020 2023.

- ***4 Investigador principal.***

- Jose Manuel López Alonso
- Profesor Dpto. Óptica.
- Facultad de Óptica y Optometría.
- Universidad Complutense de Madrid.
-
- Youssef Marrakchi Chikri.
- Alumno de Doctorado en el programa Óptica, Optomería y Visión.
- Facultad Óptica y Optometría.
- Universidad Complutense de Madrid.
- Fellow Proyecto Europeo EYE (European Young Eye)
-

- ***4.1 Co-investigadores***

- Nuria Garzón Jiménez, PhD.
- Profesor Dpto. Optometría y Visión
- Facultad de Óptica y Optometría.
- Universidad Complutense de Madrid.

- Mariano González Pérez.
- Profesor Dpto. Optometría y Visión.
- Facultad de Óptica y Optometría.
- Universidad Complutense de Madrid.

-

- ***5 Centros en los que se prevé realizar el estudio.***

- Universidad Complutense de Madrid. Facultad de Óptica y Optometría.
- C. de Arcos de Jalón, 118,

- 28037 Madrid, España
- Teléfono: +34 913 94 68 76
-
- Centros Alain Afflelou (España).
-

Se prevé realizar el estudio de forma presencial en el centro de la Universidad Complutense de Madrid, pero el estudio también se realizará de forma remota utilizando igualmente los profesionales de la salud y pacientes de centros ópticos de Alain Afflelou localizados en España. Antes de ser incluido en el estudio cada paciente recibirá toda la información relativa al estudio y para decidir si participa en el estudio firmando el consentimiento. Si la respuesta es afirmativa, se le enviará a cada paciente las credenciales para utilizar la aplicación y realizar la recolección de datos siguiendo las instrucciones necesarias.

- ***6 Comités éticos de investigación clínica que han aprobado el estudio***

Comité Ético de Investigación Clínica del Hospital Clínico San Carlos.

- ***7 Nombre y calificación de la persona responsable de la monitorización***

- No se precisa.

- ***8 Fármacos a utilizar: dosis, formulación, forma farmacéutica y grupo terapéutico.***

- No aplicable

- ***9 Fase del ensayo.***

No aplicable

- ***10 Objetivo principal.***

El objetivo principal es encontrar el tipo de correlación si existe, entre los parámetros ambientales y las variables obtenidas que se relacionan con la adaptación de lentes de contacto. Este resultado se pretende obtener a partir de la recogida objetiva de los datos de pacientes en un determinado intervalo de tiempo.

- ***11 Diseño.***

Estudio observacional descriptivo.

- ***12 Enfermedad o trastorno en estudio.***

Cambios en las características visuales de parpadeo, rojez y calidad visual (velocidad de lectura y agudeza visual) en personas portadoras de lentes de contacto.

- ***13 Variable principal de valoración.***

Intervalo de parpadeo.

- ***13.1. Variables secundarias de valoración***

Con la aplicación de recogida de vídeos de la cara de pacientes y datos ambientales, se han desarrollado algoritmos para la extracción de parámetros de los ojos del paciente con el borrado posterior de cara para no proceder a la identificación visual de personas.

Las variables del algoritmo de detección de parpadeos y rojez de la esclera que se almacenan obtenidas del video del ojo del paciente son:

- Número de parpadeos completos e incompletos en función de la duración del vídeo.
- La duración media de parpadeo.
- Variables dinámicas del parpadeo (amplitud y pendiente lineal del espectrograma en escala logarítmica).
- Índice de la rojez de la esclera.
- El tamaño de la pupila, si visible.
- La posición de la pupila o del iris, si la pupila no es visible.
- La iluminación del ambiente.
- Velocidad de lectura del paciente, a través del tiempo y número de fijaciones.
- Agudeza Visual

También con la app, se obtienen los parámetros ambientales en el momento de la recogida de vídeos, estos son: Presión atmosférica, temperatura, humedad, dióxido de nitrógeno, ozono, partículas de 10 y 2.5 micrones y dióxido de azufre. Estos 8 parámetros atmosféricas se obtienen usando el sistema de geolocalización del móvil del paciente y posteriormente de forma automática conectándose a la estación meteorológica más próxima del grupo "The World Air Quality Index Project Team". Estos parámetros se utilizarán para predecir la comodidad del paciente en condiciones ambientales diferentes y averiguar si afectan en el paciente con el uso de distintos tipos de lentes de contacto.

Otras variables a analizar son las respuestas al cuestionario con preguntas tipo de cada uno los siguientes bloques relacionados con la calidad de vida de los portadores de lentes de contacto; Conveniencia, síntomas generales, síntomas relacionados con la superficie ocular, síntomas visuales, problemas económicos, sociales y cognitivos, limitación de actividades y estado emocional.

Con el uso de las funciones de correlación en los datos obtenidos de parpadeos, rojez de la esclera y tamaño de la pupila si procede y velocidad de lectura del paciente, y teniendo en cuenta los parámetros ambientales, se podrá obtener el tipo de relación (lineal u no lineal) que existe entre estos parámetros.

Finalmente, los sujetos responderán a 3 preguntas demográficas generales relacionadas con el género, edad, etnia y 3 preguntas sobre el uso de lentes de contacto.

- ***14 Población en estudio y número total de sujetos***

El cálculo del tamaño muestral se ha considerado teniendo en cuenta un riesgo alpha del 5% y un riesgo beta del 10%, el tipo de contraste bilateral para obtener una correlación mínima de 0.4 y con una proporción prevista de pérdidas del 20% (estas pérdidas consideran los videos tomados con luminosidad irregular y contraste deficiente de las imágenes del video tomadas por el paciente). Esta correlación se ha calculado a partir de un estudio preliminar interno tomando el número de parpadeos producidos en vídeos de 1 minuto, la duración de los parpadeos, el intervalo en el que se producen y la amplitud y dinámica del parpadeo estudiados a partir del análisis del espectrograma de la función de parpadeo. A partir de estas variables se ha estudiado la correlación existente con los parámetros ambientales y de calidad del aire. El valor de 0.4 se ha elegido punto de partida para que exista una correlación significativa entre estos parámetros ambientales y las variables de parpadeo mencionadas (específicamente se tiene especial consideración del intervalo de parpadeo, variable principal). El mínimo número de personas resultante es de 80 (cálculos realizados a través de la aplicación , Calculadora tamaño muestral GRANMO, versión 7.12 Abril 2012, https://www.imim.es/ofertadeserveis/software-public/granmo/).

La población en estudio engloba a sujetos de distinto género con edad de ser población activa. Los sujetos serán reclutados entre los siguientes colectivos.

- Alumnos de la facultad de Óptica y Optometría de la Universidad Complutense de Madrid, profesores o trabajadores, con edades contempladas en el rango de población activa. Se informará del estudio al personal y alumnado de la UCM y de la facultad mediante el envío de correos informativos a cada colectivo y se pondrá carteles en la facultad para dar visibilidad y facilitar el reclutamiento.
- Pacientes y/o personal de Alain Afflelou España de distintas edades contemplados en el rango de población activa. En primer lugar, se informará a los ópticos-optometristas del desarrollo de las tecnologías descritas y que se ofrecerá por correo electrónico la posibilidad de probarla "como usuarios". En segundo lugar, se les invitará a que ofrezcan estas herramientas a sus pacientes interesados en probar lentes de contacto multifocales o que estén en el proceso de adaptación.

- ***15 Duración del tratamiento.***

No procede.

- ***16 Calendario y fecha prevista de finalización.***

- Fecha de comienzo una vez aprobado el siguiente protocolo por el Comité Ético.
- Fecha estimada de inicio 1 de Junio del 2023.
- Fecha estimada de finalización 31 de Diciembre del 2024.

1. Índice

2.Información general

- ***A. Identificación del estudio***

App-eyeh2020. Versión 1 marzo 2023.

- ***B. Tipo de estudio***

Estudio observacional descriptivo para comprobar la precisión de tecnologías basadas en App para móviles (App eyeh2020), utilizadas para la obtención de parámetros físicos del ojo (variables de parpadeo, color esclera, posición y tamaño pupilar), a partir de videos de los mismos, en portadores de lentes de contacto. La calidad de visión se puede evaluar a partir del seguimiento de la velocidad de lectura y de medida de la agudeza visual mediante la visualización de optotipos en el móvil. La comodidad del paciente se evalúa realizando una serie de preguntas mientras usa lentes de contacto.

- ***C. Descripción de los productos en estudio***

Los productos en estudio son distintas herramientas que funcionan conjuntamente para la recolección de datos de forma continua del paciente estando fuera de la consulta.

Principalmente una aplicación web móvil que funciona en todos los entornos conocidos, web, Android y iOS. Esta App nos permite obtener un vídeo de la cara del paciente junto con los datos ambientales a los que se ve afectado mientras se está grabando.

La App permite la grabación de vídeo, que se usará para guardar un video de los ojos del paciente. Además, mientras se realiza la grabación, el paciente debe activar el sensor GPS del móvil para localizar la estación meteorológica más cercana y extraer y guardar en una base de datos 8 parámetros ambientales, temperatura, presión, humedad, dióxido de nitrógeno, dióxido de azufre, ozono y partículas en el aire de 10 y 2.5 micrones. Tras la toma de datos, la App no guarda ni transmite la localización ni la estación meteorológica. También se le pide al paciente responder a un cuestionario con una pregunta tipo de cada uno los siguientes bloques relacionados con la calidad de vida de los portadores de lentes de contacto; Conveniencia, síntomas generales, síntomas relacionados con la superficie ocular, síntomas visuales, problemas económicos, sociales y cognitivos, limitación de actividades y estado emocional. Finalmente, los sujetos responderán a 3 preguntas demográficas generales relacionadas con el género, edad, etnia y 3 preguntas sobre el uso de lentes de contacto.

Junto con la App se han desarrollado tres algoritmos en Matlab que usan como base el vídeo recogido por la App.

La primera etapa que se realiza de forma automática, utilizando un algoritmo de Matlab cuando el vídeo del paciente se envía al servidor, es el recorte del par de ojos del vídeo de cara. De esta forma, siguiendo el protocolo de ética del proyecto EYE, se anonimizan todos los vídeos enviados por los pacientes para guardarlos posteriormente en el servidor.

El primer algoritmo analiza el vídeo de los ojos para detectar los parpadeos que se producen en este. El algoritmo de detección de parpadeos nos proporciona los siguientes datos; número de parpadeos realizados diferenciando de completos e incompletos, tiempo en que se produce cada uno de los parpadeos y la duración de cada uno de ellos. Finalmente, con todos estos datos nos proporciona el intervalo medio y la desviación estándar entre parpadeos, la duración y desviación estándar de los parpadeos y a partir del estudio del espectrograma de la función se obtiene los parámetros alfa y b índices de la amplitud y dinámica del espectrograma del parpadeo.

El segundo algoritmo tiene como principal objetivo la identificación a partir del vídeo recolectado por la App de indicios de rojez de la esclera y del tamaño de la pupila, si esta es visible. Este algoritmo se basa en sacar una única imagen del ojo del vídeo del paciente y a partir de la detección del iris, analizar parte de la esclera y calcular el tamaño de la pupila.

El tercer algoritmo trabaja con el segundo vídeo obtenido del paciente. La actividad monitorizada en este caso es la lectura de un texto. El número de palabras del texto es conocido para así calcular la rapidez de lectura. Con este algoritmo se obtiene la posición estimada de la pupila (iris) en cada una de las imágenes del vídeo para visualizar la fijación, es decir los momentos en los que la posición de la pupila (iris) es el mismo de forma secuencial. También, como el vídeo es de duración definida por el usuario, se obtiene el tiempo de lectura.

Siguiendo las instrucciones de la App, se puede dar un valor de la agudeza visual del paciente en el momento de visualización.

Las herramientas descritas anteriormente (App y algoritmos), tienen la finalidad de recopilar información ambiental y analizar el vídeo de los ojos del paciente para estudiar el parpadeo, verificar la rojez de la esclera (índice de incomodidad y/o sequedad del ojo) y calcular el tamaño de la pupila teniendo en cuenta la luz del ambiente. Las preguntas respondidas por los pacientes al realizar el cuestionario, que forman parte de los datos subjetivos, se adicionan a las variables objetivas. Toda esta información (medioambiental, datos subjetivos del cuestionario y resultado de análisis) se almacena en una gran base de datos.

Los datos recogidos del paciente se realizarán mediante el uso de la aplicación. Los datos que van a ser recogidos en cada muestra serán: Uno o dos videos de la cara del paciente (actividad del paciente en el video 1 fijación en punto fijo, actividad del segundo vídeo si procede, lectura de un texto a corta distancia), en los que se guarda únicamente la parte de los ojos, y los datos medioambientales del entorno en el que se graba los vídeos el paciente (8 parámetros meteorológicos dependiendo de la localización del paciente), los datos sobre agudeza visual y los datos subjetivos relativos a las respuestas al cuestionario demográfico, de uso y de calidad de vida de portadores de lentes de contacto. El video se guardará en el servidor FTP, mientras que los datos obtenidos del medio ambiente, así como todos los parámetros obtenidos de los vídeos, se enviarán al servidor MySQL para ser almacenados y posteriormente analizados como se ha explicado anteriormente.

Los servidores y todos los datos de los pacientes se encuentran en posesión del proyecto EYEH2020: European Young Eye GA number 956274. Este proyecto ha recibido financiación del programa de investigación e innovación Horizonte 2020 de la Unión Europea en el marco del acuerdo de subvención Marie Skłodowska-Curie No 956274.

- ***BIBLIOGRAFÍA***

- Akoglu, H. (2018). User's guide to correlation coefficients. Turkish Journal of Emergency Medicine, 18(3), 91–93. https://doi.org/10.1016/j.tjem.2018.08.001
- Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation Coefficients: Appropriate Use and Interpretation. Anesthesia & Analgesia, 126(5), 1763–1768. https://doi.org/10.1213/ANE.0000000000002864

- ***D. Datos relativos al promotor.***

Facultad de Óptica y Optometría. Universidad Complutense de Madrid.

- ***E. Identificación del monitor***

No precisa monitor.

- ***F. Datos de los investigadores del estudio***

- Jose Manuel López Alonso
- Profesor Dpto. Óptica.
- Facultad de Óptica y Optometría.
- Universidad Complutense de Madrid.
-
- Youssef Marrakchi Chikri.
- Alumno de Doctorado en el programa Óptica, Optomería y Visión.
- Facultad Óptica y Optometría.

- Universidad Complutense de Madrid.
- Fellow Proyecto Europeo EYE (European Young Eye)
-

- ***Co-investigadores***

- Nuria Garzón Jiménez, PhD.
- Profesor Dpto. Optometría y Visión
- Facultad de Óptica y Optometría.
- Universidad Complutense de Madrid.

- Mariano González Pérez.
- Profesor Dpto. Optometría y Visión.
- Facultad de Óptica y Optometría.
- Universidad Complutense de Madrid.

-

- ***G. Centros en que se realiza el estudio***

- Universidad Complutense de Madrid. Facultad de Óptica y Optometría.
- C. de Arcos de Jalón, 118,
- 28037 Madrid, España
- Teléfono: +34 913 94 68 76
-
- Centros Alain Affelou España

- ***H. Duración prevista del estudio***

- Fecha de inicio sujeta a aprobación por el Comité ético.
- Fecha estimada de inicio 1 de junio del 2023.
- Fecha final 31 de diciembre del 2024.

2. Justificación y objetivos.

- ***Justificación***

El uso de lentes de contacto es muy común y constituye una industria rentable. El tamaño del mercado mundial de lentes de contacto se valoró en USD 9,90. mil millones en 2022. Se prevé que el mercado crezca de USD 10,35 mil millones en 2023 a USD 15,40 mil millones en 2030.

El ajuste, la comodidad y la calidad óptica de las lentes de contacto dependen de un gran número de factores que influyen en sus características, como la temperatura ambiental, la humedad, las condiciones durante el uso de lentes de contacto (Fonn, D. 2007), la subjetividad del paciente (Pesudovs, 2007), los parámetros clínicos del paciente (Richdale, 2007), etc. Por su propia naturaleza, estos parámetros son muy variables. Su medición en una consulta médica no producirá necesariamente los mismos resultados que en las condiciones normales de vida del paciente. Por lo tanto, sería importante recoger estos parámetros de forma continuada a lo largo de las diferentes circunstancias de la vida diaria del usuario. El objetivo principal de este proyecto es desarrollar herramientas capaces de recopilar y monitorear parámetros clínicos (Cheng, C.-Y., 2020), (de la Casa JMM, 2019), resultados informados por el paciente y condiciones ambientales de manera continua a lo largo de las diferentes circunstancias del paciente mientras usa lentes de contacto (Alipour, 2017).

De acuerdo con la bibliografía, la comodidad del paciente al uso de lentes de contacto y la continuidad de uso por parte del paciente de las lentes se deben a síntomas de sequedad del ojo (Fonn, D. 2007), (The Epidemiology of Dry Eye Disease, 2007). Dry eye o sequedad del ojo, implica la aparición de diferentes síntomas que de igual forma se relacionan como antes visto con la comodidad al uso de lentes de contacto. A partir de aquí, se ha empezado a ver aplicaciones orientadas a la oftalmología ya existentes (Macana Castro T. A., 2020) y específicamente sobre sintomatología y/o diagnóstico de dry eye (Nagino, K, 2023).

Posteriormente, debido a la necesidad de recolección de datos a tiempo real, hace inviable el uso de equipamiento técnico específico que se encuentra generalmente en consulta. Por ello, la búsqueda en cuanto a información extraíble con el uso de las técnicas de tratamiento de imagen (Kumar, 2023) se reduce únicamente a imágenes de la córnea a partir del uso de cámaras móviles. De ahí, se crea nuestro primer producto, una app híbrida para la recolección de vídeos y datos atmosféricos. A su vez, las necesidades finales del proyecto del cual nace este estudio inicial con pacientes, es la creación de una amplia base de datos que irá cogiendo amplitud una vez los productos mostrados sean validados. Debido a la gran cantidad de información (vídeos de pacientes a tratar), los productos diseñados son innovadores en ese sentido. Se pretende obtener un resultado preciso requiriendo la mínima memoria computacional.

El uso de técnicas de big data ha crecido mucho en todos los campos. Es una forma de poder encontrar correlaciones entre parámetros cuando el objeto de estudio es complejo, como ocurre en el campo de la oftalmología, especialmente cuando se habla de aspectos objetivos de calidad de imagen y comodidad subjetiva en el uso de lentes de contacto. De ahí que la oftalmología también haya sido un campo en el que ha crecido el uso de este tipo de técnicas [(Alalawi, 2020), (de la Casa JMM, 2019), (Garrica, 2006)].

La mayoría de estas técnicas se basan en asociar etiquetas numéricas a diversas patologías o condiciones del ojo para posteriormente utilizar algoritmos de clasificación. Una parte importante del presente proyecto es la obtención de estas variables numéricas a partir de imágenes, por lo que los mecanismos de procesamiento de imágenes son importantes. Además, este campo está parcialmente desarrollado en otras áreas para otro tipo de aplicaciones distintas al ajuste de lentes de contacto, como la detección del iris del ojo y la pupila en aplicaciones de rastreo del ojo o tamaño de la pupila (Han YJ, 2018), (Bianchetti A, 2009), (Mitra D, 2019), (Kin-Man Lam, 1996). Estas características del ojo han sido estudiadas debido a la relación existente con la comodidad visual portando lentes de contacto (Cardona, G., 2016), (Madrid-Costa D., 2016). Además, el cálculo del tiempo de lectura junto a la duración de la fijación de la pupila y el número de fijaciones es un índice de la calidad de visión de las lentes de contacto (Concepcion-Grande P., 2023).

Centrándonos ahora en las lentes de contacto, un indicador de su comodidad puede ser el cambio en la dinámica del parpadeo y su frecuencia (Lenskiy A, 2016), (Navascues-Cornago M, 2022). Como antes se ha citado, también se aprecia este cambio en la dinámica de parpadeo entre ojo sano y ojo seco (Johnston P., 2013).

Sin embargo, la mayoría de las técnicas actuales de medición y control del parpadeo se basan en imágenes tomadas a través de cámaras (Baccour MH, 2019), (Han YJ, 2018), (Ryan C, 2021) que son tratadas con diferentes algoritmos. El primer paso en este tratamiento suele ser una etapa de segmentación para encontrar el ojo, previa demarcación del rostro en la imagen (Wang YQ, 2014), (Feng GC, 2001), (Ryan C, 2021).

A partir de aquí se han desarrollado diferentes algoritmos basados en el cambio de intensidad de la imagen al cerrar el ojo, cambios en la morfología del párpado, diferencias de color, etc. (Dou D, 2020), (Wang L, 2009), (Sree Sharmila T, 2019), (Drutarovsky T, 2015), (Borza D, 2016), (Fernando BA, 2022), (Al-gawwam S, 2018). D

Una vez extraídas las características (features) más relevantes de las imágenes, toda esta información numérica suele pasarse a extensas bases de datos para su tratamiento con diferentes algoritmos de clasificación y Machine Learning (Garriga GC, 2006).

El uso de los algoritmos diseñados en Matlab para el análisis de videos tomados por smartphone les da un plus a las actuales formas utilizadas para extraer características de un video (Navascues-Cornago M, 2022).

El objetivo final buscado como antes se ha visto, es la obtención por parte del clínico de información sobre el comportamiento de las lentes de contacto en las diferentes condiciones ambientales a las que está expuesto el paciente en los diferentes momentos del día y qué tan cómodos se encuentran. La aplicación también registrará las condiciones ambientales (temperatura, humedad) a los que se exponen los pacientes mientras llevan puesto las lentes de contacto. Los datos de contaminación ambiental donde estuvieron expuestos los pacientes se extraerán de (https://aqicn.org/map/europe/es), ya que permite obtener 8 parámetros importantes en cuanto a la calidad del aire que son: temperatura presión atmosférica, humedad, ozono, dióxido de nitrógeno, dióxido de azufre y partículas de 10 y 2.5 micras. Además, la base de datos del grupo World Air Quality Index Team dispone de accesos a más de 15 000 estaciones meteorológicas en 132 países. El acceso a estas estaciones es muy sencillo a través de la aplicación móvil.

Figure 1: Estaciones meteorológicas disponibles en la base de datos aqicn. Imagen obtenida de World air quality index team y se encuentra disponible en el siguiente enlace: https://aqicn.org/sources/.

Los resultados del paciente englobados en 3 distintos bloques, que son: 1. Datos ambientales durante la toma de vídeo extraído con el uso de la App que conecta con la base de datos aqicn.org, 2. Resultados obtenidos después del análisis del vídeo de paciente, usando algoritmos de Matlab desarrollados específicamente para el estudio (en este caso, el intervalo, la frecuencia y la duración de parpadeo, la rojez de la esclera, etc.) y 3. Respuestas al cuestionario relacionado con la calidad de vida del paciente mientras usa lentes de contacto, se correlacionarán entre sí. Para ello, estos datos recogidos para distintos pacientes en distintos momentos se acumularían en una base de datos y se analizarían estadísticamente utilizando diferentes técnicas de Big Data.

- ***BIBLIOGRAFÍA***

- Agarwal, M., & Sivakumar, R. (2019). Blink: A Fully Automated Unsupervised Algorithm for Eye-Blink Detection in EEG Signals. 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 1113–1121. https://doi.org/10.1109/ALLERTON.2019.8919795
- Alalawi, A. (2020). A Systematic Approach to Big Data Analysis in Cataract Patients In Telangana State, India.
- Al-gawwam, S., & Benaissa, M. (2018). Robust Eye Blink Detection Based on Eye Landmarks and Savitzky–Golay Filtering. Information, 9(4), 93. https://doi.org/10.3390/info9040093
- Alipour, F., Khaheshi, S., Soleimanzadeh, M., & Heidarzadeh, S. (2017). Contact Lens-related Complications: A Review. 12(2).
- Baccour, M. H., Driewer, F., Kasneci, E., & Rosenstiel, W. (2019). Camera-Based Eye Blink Detection Algorithm for Assessing Driver Drowsiness. 2019 IEEE Intelligent Vehicles Symposium (IV), 987–993. https://doi.org/10.1109/IVS.2019.8813871
- Bianchetti, A. (n.d.). Medición de diámetro pupilar ocular.
- Borza, D., Darabant, A., & Danescu, R. (2016). Real-Time Detection and Measurement of Eye Features from Color Images. Sensors, 16(7), 1105. https://doi.org/10.3390/s16071105
- Cardona, G., & López, S. (2016). Pupil diameter, working distance and illumination during habitual tasks. Implications for simultaneous vision contact lenses for presbyopia. Journal of Optometry, 9(2), 78–84. https://doi.org/10.1016/j.optom.2015.06.005
- Cheng, C.-Y., Soh, Z. D., Majithia, S., Thakur, S., Rim, T. H., Tham, Y. C., & Wong, T. Y. (2020). Big Data in Ophthalmology. Asia-Pacific Journal of Ophthalmology, 9(4), 291–298. https://doi.org/10.1097/APO.0000000000000304
- Concepcion-Grande, P., Chamorro, E., Cleva, J. M., Alonso, J., & Gómez-Pedrero, J. A. (2023). Correlation between reading time and characteristics of eye fixations and progressive lens design. PLOS ONE, 18(3), e0281861. https://doi.org/10.1371/journal.pone.0281861
- de la Casa, J. M. M., Urcola, J. A., & Sociedad Española de Oftalmología Congreso. (2019). Big data en oftalmología. Sociedad Española de Oftalmología. https://books.google.pt/books?id=DrbizQEACAAJ
- Dou, D., & Zhang, Z. (2020). Blink Detection Based on Pixel Fluctuation Ratio of Eye Image. Journal of Physics: Conference Series, 1453(1), 012073. https://doi.org/10.1088/1742-6596/1453/1/012073
- Drutarovsky, T., & Fogelton, A. (2015). Eye Blink Detection Using Variance of Motion Vectors. In L. Agapito, M. M. Bronstein, & C. Rother (Eds.), Computer Vision—ECCV 2014 Workshops (Vol. 8927, pp. 436–448). Springer International Publishing. https://doi.org/10.1007/978-3-319-16199-0_31
- Epling, S. L., Middendorf, M., Hoepf, M., Gruenwald, C., Stork, L., & Galster, S. (2015). The Electrooculogram and a New Blink Detection Algorithm.
- Feng, G. C., & Yuen, P. C. (2001). Multi-cues eye detection on gray intensity image. Pattern Recognition, 34(5), 1033–1046. https://doi.org/10.1016/S0031-3203(00)00042-X
- Fernando, B. A., Sridhar, A., Talebi, S., Waczak, J., & Lary, D. J. (2022). Unsupervised Blink Detection Using Eye Aspect Ratio Values [Preprint]. MATHEMATICS & COMPUTER SCIENCE. https://doi.org/10.20944/preprints202203.0200.v1
- Fonn, D. (2007). Targeting Contact Lens Induced Dryness and Discomfort: What Properties Will Make Lenses More Comfortable. Optometry and Vision Science, 84(4), 279–285. https://doi.org/10.1097/OPX.0b013e31804636af
- Garriga, G. C., Kralj, P., & Lavrač, N. (2006). Closed Sets for Labeled Data. In J. Fürnkranz, T. Scheffer, & M. Spiliopoulou (Eds.), Knowledge Discovery in Databases: PKDD 2006 (Vol. 4213, pp. 163–174). Springer Berlin Heidelberg. https://doi.org/10.1007/11871637_19

- Han, Y.-J., Kim, W., & Park, J.-S. (2018). Efficient Eye-Blinking Detection on Smartphones: A Hybrid Approach Based on Deep Learning. Mobile Information Systems, 2018, 1–8. https://doi.org/10.1155/2018/6929762
- Johnston, P., Rodriguez, J., Lane, K., Ousler, & Abelson, M. (2013). The interblink interval in normal and dry eye subjects. Clinical Ophthalmology, 253. https://doi.org/10.2147/OPTH.S39104
- Kin-Man Lam & Hong Yan. (1996). Locating and extracting the eye in human face images. Pattern Recognition, 29(5), 771–779. https://doi.org/10.1016/0031-3203(95)00119-0
- Kumar, S. V. M., & Gunasundari, R. (2023). Computational intelligence in eye disease diagnosis: A comparative study. Medical & Biological Engineering & Computing, 61(3), 593–615. https://doi.org/10.1007/s11517-022-02737-3
- Macana Castro, T. A., María Fernanda, M. F., Acuña Gómez, J. S., & Jiménez Barbosa, W. G. (2020). Aplicaciones móviles de salud con respaldo en publicaciones científicas, para la mejora en el ejercicio de la optometría y la oftalmología. Ciencia y Tecnología para la Salud Visual y Ocular, 17(2), 51–63. https://doi.org/10.19052/sv.vol17.iss2.6
- Madrid-Costa, D., Ruiz-Alcocer, J., García-Lázaro, S., Ferrer-Blasco, T., & Montés-Micó, R. (2015). Optical power distribution of refractive and aspheric multifocal contact lenses: Effect of pupil size. Contact Lens and Anterior Eye, 38(5), 317–321. https://doi.org/10.1016/j.clae.2015.03.008
- Mitra, D., Chakraborty, A., & Thakurta, M. B. (2019). IRIS RECOGNATION USING MATLAB. 6(2).
- Nagino, K., Okumura, Y., Yamaguchi, M., Sung, J., Nagao, M., Fujio, K., Akasaki, Y., Huang, T., Hirosawa, K., Iwagami, M., Midorikawa-Inomata, A., Fujimoto, K., Eguchi, A., Okajima, Y., Kakisu, K., Tei, Y., Yamaguchi, T., Tomida, D., Fukui, M., … Inomata, T. (2023). Diagnostic Ability of a Smartphone App for Dry Eye Disease: Protocol for a Multicenter, Open-Label, Prospective, and Cross-sectional Study. JMIR Research Protocols, 12, e45218. https://doi.org/10.2196/45218
- Navascues-Cornago M, Sun T, Read ML, Morgan PB. The short-term effect of contact lens wear on blink characteristics. Contact Lens and Anterior Eye. 2022 Apr;101596.
- Pesudovs, K., Garamendi, E., & Elliott, D. B. (2006). The Contact Lens Impact on Quality of Life (CLIQ) Questionnaire: Development and Validation. Investigative Opthalmology & Visual Science, 47(7), 2789. https://doi.org/10.1167/iovs.05-0933
- Richdale, K., Sinnott, L. T., Skadahl, E., & Nichols, J. J. (2007). Frequency of and Factors Associated With Contact Lens Dissatisfaction and Discontinuation. Cornea, 26(2), 168–174. https://doi.org/10.1097/01.ico.0000248382.32143.86
- Ryan, C., O'Sullivan, B., Elrasad, A., Cahill, A., Lemley, J., Kielty, P., Posch, C., & Perot, E. (2021). Real-time face & eye tracking and blink detection using event cameras. Neural Networks, 141, 87–97. https://doi.org/10.1016/j.neunet.2021.03.019
- Sree Sharmila, T., Srinivasan, R., Nagarajan, K. K., & Athithya, S. (2019). Eye Blink Detection Using Back Ground Subtraction and Gradient-Based Corner Detection for Preventing CVS. Procedia Computer Science, 165, 781–789. https://doi.org/10.1016/j.procs.2020.01.011
- The Epidemiology of Dry Eye Disease: Report of the Epidemiology Subcommittee of the International Dry Eye WorkShop (2007). (2007). The Ocular Surface, 5(2), 93–107. https://doi.org/10.1016/S1542-0124(12)70082-4
- Wang, L., Ding, X., Fang, C., Liu, C., & Wang, K. (2009). Eye blink detection based on eye contour extraction (J. T. Astola, K. O. Egiazarian, N. M. Nasrabadi, & S. A. Rizvi, Eds.; p. 72450R). https://doi.org/10.1117/12.804916

- ***Objetivos.***

- ***Objetivo principal:***

El objetivo principal es encontrar el tipo de correlación si existe, entre los parámetros ambientales y las variables obtenidas que se relacionan con la adaptación de lentes de contacto. Este resultado se pretende obtener a partir de la recogida objetiva de los datos de pacientes en un determinado intervalo de tiempo. También, dentro de las variables relacionadas con la adaptación, se encuentran las respuestas subjetivas del paciente al cuestionario que se realiza para evaluar la comodidad.

- ***Objetivo secundario:***

Se pretende comprobar la precisión de los algoritmos desarrollados en el entorno Matlab basados en segmentación por colores e intensidad lumínica, Digital Image Processing (DIP) utilizados para la detección de parpadeos a partir de videos de los ojos del paciente, detección de la rojez en la esclera, el tamaño de la pupila (si visible), la velocidad de lectura (tiempo de lectura), seguimiento de la posición de la pupila (iris) para el cálculo de la duración de fijación y número de fijaciones, intensidad lumínica del entorno, calidad visual (agudeza visual).

3. Tipo de estudio y diseño del mismo.

- ***Tipo de estudio.***

Estudio observacional descriptivo para comprobar la precisión de tecnologías basadas en App para móviles (App eyeh2020), utilizadas para la obtención de parámetros físicos del ojo (variables de parpadeo, color esclera, posición y tamaño pupilar), a partir de videos de los mismos, en portadores de lentes de contacto. La calidad de visión se puede evaluar a partir del seguimiento de la velocidad de lectura y de medida de la agudeza visual mediante la visualización de optotipos en el móvil. La comodidad del paciente se evalúa realizando una serie de preguntas mientras usa lentes de contacto.

- ***Diseño del estudio.***

Se reclutará un mínimo de 80 personas a las que se les suministrará la dirección de la aplicación tras la firma del consentimiento. La aplicación tomará videos de sus ojos de los que se extraerán las variables en estudio. También se le realiza al paciente un cuestionario para la recolección de datos demográficos del paciente y relacionados con la comodidad. La aplicación también visualizará optotipos y textos para la valoración de la agudeza visual.

El cálculo del tamaño muestral se ha considerado teniendo en cuenta un riesgo alpha del 5% y un riesgo beta del 10%, el tipo de contraste bilateral para obtener una correlación mínima de 0.4 y con una proporción prevista de pérdidas del 20% (estas pérdidas consideran los videos tomados con luminosidad irregular y contraste deficiente de las imágenes del video tomadas por el paciente). Esta correlación se ha calculado a partir de un estudio preliminar interno tomando el número de parpadeos producidos en vídeos de 1 minuto, la duración de los parpadeos, el intervalo en el que se producen y la amplitud y dinámica del parpadeo estudiados a partir del análisis del espectrograma de la función de parpadeo. A partir de estas variables se ha estudiado la correlación existente con los parámetros ambientales y de calidad del aire. El valor de 0.4 se ha elegido punto de partida para que exista una correlación significativa entre estos parámetros ambientales y las variables de parpadeo mencionadas. El mínimo número de personas resultante es de 80 (cálculos realizados a través de la aplicación, Calculadora tamaño muestral GRANMO, versión 7.12 Abril 2012, https://www.imim.es/ofertadeserveis/software-public/granmo/).

La población en estudio engloba a sujetos de distinto género con edad de ser población activa. Los sujetos serán reclutados entre los siguientes colectivos.

- Alumnos de la facultad de Óptica y Optometría de la Universidad Complutense de Madrid, profesores o trabajadores, con edades contempladas en el rango de población activa.
- Pacientes y/o clientes de Alain Afflelou España de distintas edades contemplados en el rango de población activa.

- ***Duración del estudio.***

- Fecha de inicio sujeta a aprobación por el Comité ético.
- Fecha estimada de inicio 1 de junio del 2023.
- Fecha final 31 de diciembre del 2024.

4. Selección de los sujetos.

- ***Criterios de inclusión y exclusión.***

- ***<u>Criterios de inclusión:</u>***

Todos los pacientes mayores de edad podrán participar al estudio con una buena salud ocular con ausencia de hallazgos oculares que contraindiquen el uso de lentes de contacto.

- ***<u>Criterios de exclusión:</u>***

- Se excluirán los vídeos que no podrán ser analizados por el algoritmo. Estos vídeos serán aquellos con una luminosidad muy baja o alta y en aquellos en los que se dificulta la localización del ojo debido a movimientos de la cámara o por cualquier otra razón. Los pacientes podrán realizar el número de vídeos que consideren.
- Se excluirán los pacientes con antecedentes de sintomatología relacionada con el cierre de párpados, parkinson, ptosis y trastornos psiquiátricos.
- Se excluirán pacientes que no puedan cumplir con el protocolo del estudio debido a razones físicas o mentales.
- Se excluirán pacientes que el investigador principal considere inadecuados para participar en este estudio.

- ***Número de sujetos previstos***

Numero de sujetos previstos 80.

- ***Criterios de retirada y análisis previstos de las retiradas y los abandonos***

Los abandonos por parte de los pacientes son voluntarios y la retirada de los datos proporcionados por el paciente se realiza respetando su petición.

- ***Tratamiento de las pérdidas pre-randomización***

No aplicable

- ***Duración aproximada del periodo de reclutamiento.***

Indiferente, el paciente podrá formar parte en esta primera recogida de datos el número de veces que considere oportuno.

5. Descripción del tratamiento

- ***Duración***

No aplicable.

- ***Posología /régimen posológico***

No aplicable.

- ***Tratamientos concomitantes permitidos y prohibidos.***

No aplicable.

6. Desarrollo del estudio y evaluación de la respuesta

- ***Variable principal y secundaria de evaluación***

- ***Variable principal***

Intervalo de parpadeo.

- ***Variables secundarias***

Con la aplicación de recogida de vídeos de la cara de pacientes y datos ambientales, se han desarrollado algoritmos para la extracción de parámetros de los ojos del paciente con el borrado posterior de cara para no proceder a la identificación visual de personas.

Las variables del algoritmo de detección de parpadeos y rojez de la esclera que se almacenan obtenidas del video del ojo del paciente son:

- Número de parpadeos completos e incompletos en función de la duración del vídeo.
- La duración media de parpadeo.
- Variables dinámicas del parpadeo (amplitud y pendiente lineal del espectrograma en escala logarítmica).
- Índice de la rojez de la esclera.
- El tamaño de la pupila, si visible.
- La posición de la pupila o del iris, si la pupila no es visible.
- La iluminación del ambiente.
- Velocidad de lectura del paciente, a través del tiempo y número de fijaciones.
- Agudeza Visual

También con la app, se obtienen los parámetros ambientales en el momento de la recogida de vídeos, estos son: Presión atmosférica, temperatura, humedad, dióxido de nitrógeno, ozono, partículas de 10 y 2.5 micrones y dióxido de azufre. Estos 8 parámetros atmosféricas se obtienen usando el sistema de geolocalización del móvil del paciente y posteriormente de forma automática conectándose a la estación meteorológica más próxima del grupo "The World Air Quality Index Project Team". Estos parámetros se utilizarán para predecir la comodidad del paciente en condiciones ambientales diferentes y averiguar si afectan en el paciente con el uso de distintos tipos de lentes de contacto.

Otras variables a analizar son las respuestas al cuestionario con preguntas tipo de cada uno los siguientes bloques relacionados con la calidad de vida de los portadores de lentes de contacto; Conveniencia, síntomas generales, síntomas relacionados con la superficie ocular, síntomas visuales, problemas económicos, sociales y cognitivos, limitación de actividades y estado emocional.

Con el uso de las funciones de correlación en los datos obtenidos de parpadeos, rojez de la esclera y tamaño de la pupila si procede y velocidad de lectura del paciente, y teniendo en cuenta los parámetros ambientales, se podrá obtener el tipo de relación (lineal u no lineal) que existe entre estos parámetros.

Finalmente, los sujetos responderán a 3 preguntas demográficas generales relacionadas con el género, edad, etnia y 3 preguntas sobre el uso de lentes de contacto.

7. Acontecimientos adversos.

No se prevé de acontecimientos adversos para el paciente ya que únicamente se pide al paciente que se grabe en vídeo.

8. Aspectos éticos.

- ***Consideraciones generales.***

1. Descripción del método de anonimización/pseudonimización.

A los sujetos se les asignará un código de identificación único del sujeto (USIC) que consta de un código unívoco de 20 caracteres por sujeto (alfanumérico, no redundante, en mayúsculas).

Estos códigos USIC se pedirán directamente al oficial de protección de datos (DOP) del proyecto EYE Juan Casado, PhD para anonimizar a cada uno de los participantes en el estudio.

Además, el rostro de lo pacientes en cada uno de los vídeos enviados se procederá a la ocultación del rostro o recorte del par de ojos de cada una de las imágenes que componen el vídeo.

El proyecto EYE garantizará que se respeten todas las aprobaciones éticas y regulaciones sobre información confidencial. Además, los datos originales de la información se almacenarán y protegerán de forma segura de acuerdo con el GDPR.

2. Fuentes de información, identificación y métodos para la anonimización.

Se recopilarán cuatro fuentes de información en el ensayo EYE: a) datos clínicos, b) datos de cuestionarios de sujetos, c) datos de imágenes y d) análisis estadísticos.

Las cuatro fuentes de información mencionadas anteriormente pueden incluir tanto datos electrónicos como físicamente compatibles (incluidos datos brutos, diseños experimentales, resultados analíticos o análisis estadísticos ...). Los centros de ensayo reclutados inicialmente recopilarán todos los datos necesarios para los objetivos del proyecto. Junto con la información sobre el ensayo, se informará a los sujetos elegibles sobre los procesos de captura, transmisión y análisis de datos. Una vez que un sujeto es elegible para el estudio y firma su consentimiento informado para la participación en el ensayo y la recopilación de datos, se le asignará un código de identificación único del sujeto (USIC).

Los archivos de códigos USIC se generarán previamente y se enviarán a cada centro de reclutamiento. Estos archivos, cada uno de los cuales incluye una lista de códigos USIC y su trazabilidad con los sujetos reclutados, representan una parte crucial del proyecto. Los archivos de códigos USIC permiten tanto la trazabilidad de los sujetos como (si es necesario) su reidentificación. Por lo tanto, los archivos de códigos USIC requieren medidas específicas de almacenamiento y protección de datos. Con este fin, un solo investigador en cada institución (Investigador Principal) será la única persona a cargo de su custodia.

3. Procesamiento posterior de datos personales recopilados previamente

Los datos personales recopilados previamente se tratarán con los mismos procedimientos rigurosos que los datos obtenidos recientemente. Se requerirá una confirmación explícita de los Investigadores Principales que proporcionen los datos confirmando que tienen una base legal para el procesamiento de datos.

4. Medidas de archivo, almacenamiento y protección de datos.

El grueso de la información electrónica y física generada con el proyecto EYE está previsto que se almacene en servidores alojados en la UCM (Universidad Complutense de Madrid). Se prestará especial atención al acceso y almacenamiento de los códigos USIC. La lista de códigos USIC y su trazabilidad con los sujetos reclutados para el proyecto se almacenará como un único archivo digital, cifrado y protegido con un código USIC. Solo los investigadores principales del proyecto tendrán acceso a estos datos y serán instruidos y apoyados en las tareas de custodia correcta de esta información sensible por el oficial de protección de datos (DPO), miembro del Comité Ético y experto en Protección de Datos: Juan Casado, PhD, específicamente designado para este proyecto.

El acceso a los servidores de datos será jerarquizado (se proporcionarán diferentes roles a los participantes), registrado automáticamente y limitado solo a personas autorizadas. El acceso no autorizado a datos de sujetos seudonimizados y/o registros médicos se evita y requeriría una autorización explícita y documentada y una justificación del Coordinador. Esta autorización documentada incluirá no solo a los participantes en el proyecto, sino también a auditores de terceros, así como a miembros de las autoridades locales o federales.

El servidor ftp de almacenamiento es el siguiente: "ftp.eyeh2020.com", perteneciente al proyecto EYE y por ende a la Universidad Complutense de Madrid.

La dirección del servidor en el que se guardan los datos del ambiente y los resultados obtenidos por los algoritmos de Matlab es "eyeh2020.com" y el nombre de la base de datos MySQL es "eyeh20_appeye". El acceso a la base de datos únicamente es posible a través de una dirección IP conocida previamente informada a través del panel de configuración de la dirección web del proyecto "eyeh2020.com". Los datos almacenados tendrán dos números de identificación. El primer número identifica al paciente del cual se han tomado los datos, este número es el código USIC de 20 caracteres alfanuméricos. El segundo número de identificación va asociado tanto al vídeo como al cuestionario y los parámetros ambientales. Este número se genera teniendo en cuenta la fecha en que el usuario accede a la App para tomar las medidas. En cuanto se realiza la última tarea por parte del usuario (cuestionario, toma de vídeo y recogida de datos ambientales) se genera la clave que relaciona los tres tipos de datos tomados.

El archivo con todos los códigos USIC con la información de cada paciente solamente tendrá acceso a él el Investigador Principal y coordinador del proyecto David Madrid Costa. El archivo estará almacenado en uno de los servidores de la Universidad Complutense de Madrid.

Enlaces y Anexos:

- Enlace (acceso a la ley RGPD en todos los idiomas de la UE). **https://eur-lex.europa.eu/eli/reg/2016/679/oj**

9. Consideraciones prácticas.

- ***Suministro de las muestras.***

- No aplicable.

10. Análisis estadístico.

Se realizará una descripción estadística de todos los parámetros tratados y su posible correlación con parámetros ambientales y de calidad del aire. Se usará la correlación de Pearson y la correlación de Spearman para encontrar en el primer caso, las correlaciones lineales, y en segundo caso no lineales, que puedan existir entre los parámetros ambientales y las variables obtenidas relacionadas con el ajuste de las lentes de contacto anteriormente descritas. Además, se pretende estudiar la precisión del algoritmo de detección de parpadeos y se intentará determinar las diferencias entre los resultados y los obtenidos con otros algoritmos, análisis de error usando Los resultados se guardarán en una base de datos MySQL perteneciente al proyecto EYE, posterior a la obtención de resultados con la ayuda del algoritmo de Matlab.

Igualmente se obtienen las desviaciones estándar de la variable principal y variables secundarias y calcula los valores medios. El cálculo se realiza en las siguientes variables:

- Intervalo entre parpadeos.
- Duración del parpadeo.
- La velocidad de lectura.

Cálculo de distribuciones de probabilidad de las variables secundarias (solas o conjuntas). Las distribuciones de probabilidad permiten conocer la tendencia y la probabilidad de obtener un cierto valor de aquella variable discreta analizada.

Realización de modelos de regresión multilineal de la agudeza visual y tiempos de lectura en función de las variables secundarias del parpadeo y espectrograma.

Una vez recopilados todos los datos de la App y de los algoritmos, de 80 pacientes mínimos en distintos momentos se analizará la relación de cada una de las variables obtenidas por los algoritmos de Matlab con los parámetros medioambientales y de la calidad del aire recolectados durante la grabación del vídeo del paciente (Akoglu H, 2018), (Schober P, 2018). Se pretende así, usar con la ayuda de Matlab, la correlación de Pearson y la correlación de Spearman para ver si existen correlaciones significativas y encontrar de forma preliminar, los parámetros que más influyen en el ajuste y comodidad de la visión mientras se porta lentes de contacto.

Se realizarán análisis estadísticos con software comercial con algoritmos y bibliotecas desarrolladas para tal uso (Matlab, MathWorks® 2023a).

• Anexo I. Cuaderno de recogida de datos.

CÁLCULOS TEÓRICOS

- Datos demográficos en el que figuran 3 cuestiones relacionadas con el género, etnia y edad, y 3 cuestiones sobre el uso de lentes de contacto, primer uso, uso general diario e uso específico del día de grabación.
- Datos subjetivos de paciente, recolectados mediante un cuestionario en la propia App relacionados con las 9 siguientes cuestiones; Conveniencia, síntomas generales, síntomas relacionados con la superficie ocular, síntomas visuales, problemas económicos, sociales y cognitivos, limitación de actividades y estado emocional. El cuestionario tiene una pregunta relacionada con cada de las 9 cuestiones.
- Video de duración de un minuto de la cara del paciente y tiempo en el momento que se realiza la toma.
- Datos atmosféricos:
    - Presión atmosférica.
    - Temperatura.
    - Dióxido de nitrógeno en el ambiente.
    - Partículas de 10 micras.
    - Partículas de 2.5 micras.
    - Dióxido de azufre.
    - Ozono.

• Anexo II. Compromiso del investigador

- El Dr. Jose Manuel López Alonso, profesor de la Facultad de Óptica y Optometría de la Universidad Complutense de Madrid

Hace constar:

Que conoce y acepta participar como investigador principal en el estudio con código de protocolo App-eyeh2020 **"Desarrollo de tecnologías e-Salud para análisis y tratamiento de datos en portadores de lentes de contacto."**

Que se compromete a que cada sujeto sea tratado y controlado siguiendo lo establecido en el protocolo autorizado por el Comité Ético de Investigación Clínica y por la Dirección General de Farmacia y Productos Sanitarios.

Que respetará las normas éticas aplicables a este tipo de estudios.

En Madrid, a 22 de mayo de 2023

Firmado:

Dr. Jose Manuel López Alonso
Investigador principal

- # Anexo III. Información al Paciente

Desarrollo de tecnologías e-Salud para análisis y tratamiento de datos en portadores de lentes de contacto.

Se le invita a participar en una tesis de investigación promovida por profesores e investigadores. En este estudio se obtendrán las siguientes variables a partir de videos de los ojos de pacientes con y sin lentes de contacto y de la visualización de optotipos y textos:

- Número de parpadeos completos e incompletos en función de la duración del vídeo.
- El intervalo medio entre parpadeo.
- La duración media de parpadeo.
- Variables dinámicas del parpadeo (amplitud y pendiente lineal del espectrograma en escala logarítmica).
- Índice de la rojez de la esclera.
- El tamaño de la pupila, si visible.
- La posición de la pupila o del iris, si la pupila no es visible.
- La iluminación del ambiente.
- Velocidad de lectura del paciente, a través del tiempo y número de fijaciones.
- Agudeza Visual
- Datos subjetivos de paciente, recolectados mediante un cuestionario en la propia App relacionados con los 9 siguientes tópicos; Conveniencia, síntomas generales, síntomas relacionados con la superficie ocular, síntomas visuales, problemas económicos, sociales y cognitivos, limitación de actividades y estado emocional. El cuestionario tiene una pregunta relacionada con cada una de las 9 cuestiones.
- Datos demográficos y sobre el uso de lentes de contacto.

Posteriormente se obtendrían las correlaciones significativas entre los parámetros ambientales y las variables anteriores, es decir, los parámetros ambientales que afectan cada uno de los parámetros recolectados. Los parámetros ambientales se obtienen a través de la App móvil usada para el estudio en la que el paciente tendrá que dar su consentimiento para la toma de datos. Tras tomar estos, no se guardará ningún dato sobre la localización de la estación ni del paciente. Finalmente, se registrarán también las respuestas a las 3 preguntas demográficas generales relacionadas con el género, edad, etnia y de las 3 preguntas sobre el tiempo de uso de las lentes de contacto, primer uso, tiempo diario medio de uso y uso específica del día de recolección de datos.

¿Cuál es el fundamento y el objetivo de este estudio?

El objetivo final es poder predecir el resultado final de la adaptación de diferentes tipos de lentes de contacto (comodidad, rendimiento visual, etc.), a partir de la recogida objetiva de los datos de un paciente en un determinado intervalo de tiempo. En principio encontrar los parámetros ambientales que influyen en las variables obtenidas relacionadas con la adaptación de lentes de contacto.

¿Es necesario que participe?

Su participación es absolutamente voluntaria. Si decide intervenir, deberá firmar un formulario de CI. Aun así, podrá retirarse del estudio en cualquier momento sin dar ninguna explicación. Si decide abandonar el estudio, le rogaríamos que estableciera contacto con nosotros para poder destruir todos los datos proporcionados. No se esperan beneficios

directos por la participación en el estudio pero los resultados obtenidos en este estudio serán beneficios para futuros pacientes portadores de lentes de contacto.

¿Cómo se efectuará el estudio?

Durante el estudio se le pedirá que se grabe un vídeo de 1 minuto de su rostro a través de la aplicación App a tal fin, manejada por usted mismo, y se enviarán de forma automática los datos de las condiciones meteorológicas, así como de contaminación del aire obtenidas de la siguiente web: https://aqicn.org/. Se usará su correspondiente localización para proporcionar los datos correctos, de la estación meteorológica en cuestión, pero no quedará registrada la localización, ni el nombre de la estación por lo que no podrá ser verificada su localización ni siquiera por parte de los investigadores. También en la misma App, se le pedirá responder a un cuestionario relacionado con los 9 siguientes tópicos; Conveniencia, síntomas generales, síntomas relacionados con la superficie ocular, síntomas visuales, problemas económicos, sociales y cognitivos, limitación de actividades y estado emocional. Por último, se procederá a preguntas demográficas y de uso de lentes de contacto.

La duración de las pruebas será como máximo de 30 minutos siguiendo las instrucciones correspondientes para la grabación de vídeos y recolección de datos

¿Cuáles son los posibles efectos secundarios, los riesgos y las molestias derivadas de mi participación?

No se espera ningún efecto secundario derivado de su participación.

¿Se tratará de manera confidencial la información recogida en el estudio?

Durante el estudio se le tomarán vídeos al paciente y se enviarán datos de las condiciones meteorológicas, así como de contaminación del aire obtenidas de la siguiente web: https://aqicn.org/ Se usará la localización del paciente para proporcionar los datos correctos, de la estación meteorológica en cuestión, pero no quedará registrada la localización, ni el nombre de la estación por lo que no podrá ser verificada la localización del paciente por parte de los investigadores. Esta información se procesará y guardará en un fichero y posteriormente en una base de datos perteneciente al proyecto de investigación EYEH2020: European Young Eye (https://eyeh2020.com/). El resultado del estudio se podrá publicar en una revista médica, pero sin desvelar su identidad. Todos los datos relativos a usted y a su salud que se recojan durante el transcurso del estudio sólo serán utilizados para la realización de este; en caso de otros posibles estudios futuros relacionados con su patología, éstos serán aprobados previamente por un Comité de Ética de la Investigación. Sus datos se gestionarán bajo la confidencialidad más estricta: su nombre y su información médica personal se sustituirán por un código para que no pueda identificarse a ningún participante individual. Solo los investigadores principales del proyecto tendrán acceso a la clave de códigos y han sido instruidos y apoyados en tareas de la correcta custodia de esta sensible información por parte del delegado de protección de datos (RPD), miembro del consejo ético y Experto en Protección de Datos: Dr. Juan Casado, designado específicamente para el proyecto EYE. El responsable del tratamiento de sus datos es el Centro, que guardará todas las medidas de seguridad necesarias para la protección de sus datos. De acuerdo con la Ley Orgánica 3/2018, de Protección de Datos Personales y garantía de los derechos digitales, usted puede ejercer sus derechos de acceso, modificación, oposición y supresión de datos incorrectos, solicitar una copia o que se trasladen a un tercero (portabilidad) los datos que usted ha facilitado para el estudio, en la medida que sean aplicables. Para ejercitar estos derechos, diríjase al Investigador principal. Así mismo tiene derecho a dirigirse a la Agencia de Protección de Datos

si no quedara satisfecho/a. Para obtener información más detallada puede consultar el siguiente enlace: https://www.aepd.es/sites/default/files/2019-10/guia-ciudadano.pdf.

- Por último, es posible que los representantes designados por las autoridades sanitarias o el comité de ética que apruebe el estudio inspeccionen su historial para confirmar que el estudio se ha ejecutado de modo correcto. Toda esta información se tratará de manera rigurosamente confidencial.

• Anexo IV. Consentimiento expreso (CE)

Título del estudio: Desarrollo de tecnologías e-Salud para análisis y tratamiento de datos en portadores de lentes de contacto.

En aras a dar cumplimiento al Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos, y siguiendo las Recomendaciones e Instrucciones emitidas por la Agencia Española de Protección de Datos (A.E.P.D.), SE INFORMA:

- Los datos de carácter personal solicitados y facilitados por usted, son incorporados un fichero de titularidad privada cuyo responsable y único destinatario es EYEH2020 : European Young Eye GA number 956274.
- Solo serán solicitados aquellos datos estrictamente necesarios para prestar adecuadamente los servicios solicitados, pudiendo ser necesario recoger datos de contacto de terceros, tales como representantes legales, tutores, o personas a cargo designadas por los mismos.
- Todos los datos recogidos cuentan con el compromiso de confidencialidad, con las medidas de seguridad establecidas legalmente, y bajo ningún concepto son cedidos o tratados por terceras personas, físicas o jurídicas, sin el previo consentimiento del cliente, tutor o representante legal, salvo en aquellos casos en los que fuere imprescindible para la correcta prestación del servicio.
- Una vez finalizada la relación entre el proyecto y el usuario los datos serán archivados y conservados, durante un periodo tiempo mínimo de 3 años, tras lo cual seguirá archivado o en su defecto serán devueltos íntegramente al cliente o autorizado legal.
- Los datos que facilito serán incluidos en el Tratamiento denominado Pacientes de EYEH2020 : European Young Eye GA number 956274, con la finalidad de gestión del servicio contratado, emisión de facturas, contacto..., todas las gestiones relacionadas con los pacientes y manifiesto mi consentimiento. También se me ha informado de la posibilidad de ejercitar los derechos de acceso, rectificación, cancelación y oposición, indicándolo por escrito a EYEH2020 : European Young Eye GA number 956274 con domicilio en COMPLUTENSE UNIVERSITY OF MADRID Avda. Seneca, 2, Ciudad Universitaria 28040 - MADRID - - Spain .
- Los datos personales sean cedidos por EYEH2020: European Young Eye GA number 956274 a las entidades que prestan servicios a la misma.

Yo:
(Poner nombre y apellidos)

Declaro que he sido amplia y satisfactoriamente informado de forma oral, para participar en presente estudio.
He leído este documento, he comprendido y estoy conforme con las explicaciones del procedimiento, que dicha información ha sido realizada. He podido hacer preguntas sobre el estudio.

He hablado con:
(Nombre del investigador)

Comprendo que mi participación es voluntaria. Comprendo que puedo retirarme del estudio:
l° Cuando quiera
2° Sin tener que dar explicaciones
- *3° Sin que esto repercuta en mis cuidados médicos*

Presto libremente mi conformidad para participar en el estudio

Firma del participante **Fecha**

Firma del investigador **Fecha**

8.2 Script for eye blink detection algorithm

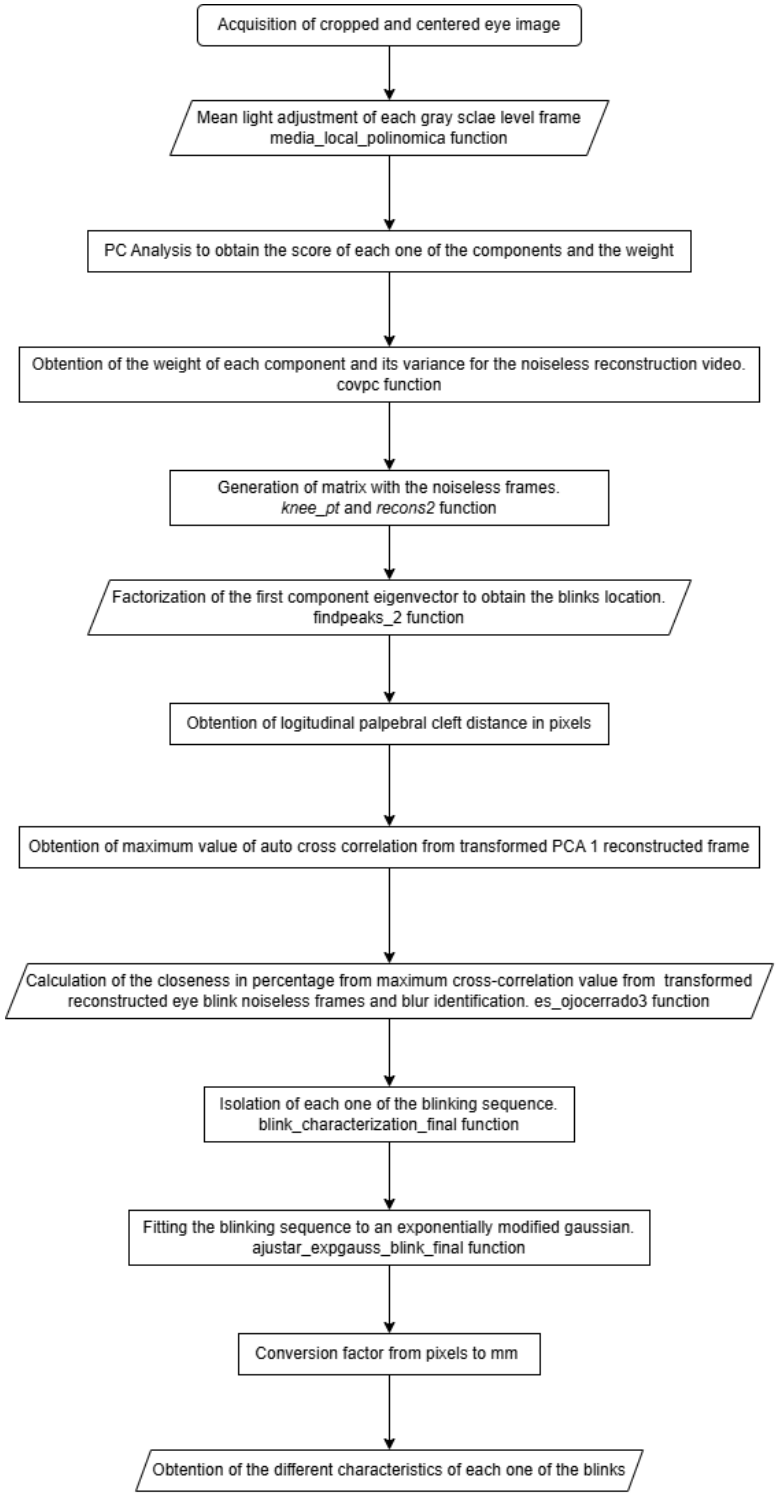

Figure 8-1: Blink detection algorithm flowchart.

Table 8-1: Main Script, blink detection algorithm, Matlab Environment.

```
%% Script para el análisis a partir de imagenes recortadas y centradas.
%% prueba pca1 con interpolación.
```

```
%% Analizar pca video de ojo ya recortado en frames
% Este código es el definitivo.
clear all
close all
% Primero añadimos todas las rutas que se necesitan para el correcto
% funcionamiento, es decir, todas las funciones.
addpath('C:\Users\Marra\OneDrive\Bureau\codigo_ojo_centrado\');
addpath('C:\Users\Marra\OneDrive\Bureau\codigo_ojo_centrado\distrib\');
addpath(['C:\Users\Marra\OneDrive -' ...
' Universidad Complutense de Madrid (UCM)\video_high_fr_face']);
addpath(['C:\Users\Marra\OneDrive - ' ...
'Universidad Complutense de Madrid (UCM)\CODIGOS_MATLAB']);
% añadir el fitting del espectro de la señal completa para la obtención de
% los parámetros alfa y b
addpath('C:\Users\Marra\OneDrive\Bureau')
%% Conexión a la base de datos MySQL.
% Comprobación de la conexión a la base de datos MySQL.
vendor = "MySQL";
opts = databaseConnectionOptions("jdbc",vendor);
opts = setoptions(opts, ...
'DataSourceName',"MySQL", ...
'JDBCDriverLocation',...
"C:\Program Files (x86)\MySQL\Connector J" + ...
" 8.0/mysql-connector-j-8.0.32.jar", ...
'DatabaseName',"eyeh20_appeye",'Server',"eyeh2020.com", ...
'PortNumber',3306);
username = "eyeh20_youssef";
password = "Youssef123456";
try
status = testConnection(opts,username,password);
conn = database('eyeh20_appeye','eyeh20_youssef','Youssef123456');
catch
opts = setoptions(opts, ...
'DataSourceName',"MySQL", ...
'JDBCDriverLocation',"C:\Program Files " + ...
"(x86)\MySQL\Connector J 8.0\mysql-connector-j-8.0.32.jar", ...
'DatabaseName',"eyeh20_appeye",'Server',"eyeh2020.com", ...
'PortNumber',3306);
status = testConnection(opts,username,password);
conn = database('eyeh20_appeye','eyeh20_youssef','Youssef123456');
end
% Verficicación de la conexión a la base de datos utilizando isopen, "1"
% conexión realizada con éxito. "0" conexión no establecida.
status = isopen(conn);
% % Lectura del nombre del vídeo a partir de nombre de fichero
% videoName=['face_video'];
% Carpeta a analizar
%% selección de la carpeta de imagenes a analizar
myFolder='C:\Users\Marra\OneDrive\Bureau\video_patients\Results_dlib';
% cd(myFolder)
filePattern_folder = fullfile(myFolder, 'Cropped_*');
theFolders= dir(filePattern_folder);
fprintf('The videos to analyze are: \n');
% Insertamos una variable para chequear que los videos pueden ser
% analizados. Igualmente definimos una nueva estructura para agregar estos
% videos válidos y poder analizarlos. Para considerar un video analizable,
% la ruta tiene que ser una carpeta y tener imagenes recortadas
cnt00=0;
% crear la misma estructura que en theFolders para ir introduciendo los
```

```
% distintos valores.
videos_to_analyze=struct('name',[],'folder',[],'date',...
[],'bytes',[],'isdir',[],'datetime',[]);
for s=1:max(size(theFolders))
% check if its a folder and also if there is available any cropped
% image.
if theFolders(s).isdir==1 && ~isempty(theFolders(s).name)
cnt00=cnt00+1;
videos_to_analyze(cnt00).name=theFolders(s).name;
videos_to_analyze(cnt00).folder=theFolders(s).folder;
videos_to_analyze(cnt00).date=theFolders(s).date;
videos_to_analyze(cnt00).bytes=theFolders(s).bytes;
videos_to_analyze(cnt00).isdir=theFolders(s).isdir;
videos_to_analyze(cnt00).datetime=theFolders(s).datenum;
fprintf('%i. %s\n', cnt00, videos_to_analyze(cnt00).name);
else
end
end
Ffolder=input('First video to analyse => ');
Lfolder=input('Last video to analyse => ');
inconsistent=[];
inconsistent_cnt=0;
for CNT=Ffolder:Lfolder
close all
folderName=videos_to_analyze(CNT).name;
videoName=[videos_to_analyze(CNT).name(end-12:end)];
current_videoFolder=[myFolder,'/',folderName];
cd(current_videoFolder)
%Ahora se calculan el número de frames que hay en cada una de las
% carpetas.
filePattern = dir(fullfile('*_Image_Centered.png'));
N = size(filePattern);
%% tener presente que hay que introducir el fr del video
% obtener el frame rate de las propiedades del vídeo.
current_video_fullpath=[myFolder,'\',videoName,'.mp4'];
try
cropped_video=VideoReader(current_video_fullpath);
fr_fv=cropped_video.FrameRate;
catch
continue
end
fr_N=N(1)/60;
paso=1;
fr_o=fr_fv/paso;
fr_int=fr_o;
ratio_fr30=30/fr_int;
% Framerate de los vídeos normales se obtiene teniendo en cuenta el número
% de frames recortadps ya que la duración es de 1 minuto.
try
imagen1=imread("0001_Image_Centered.png");
filas=size(imagen1,1);columnas=size(imagen1,2);
rows=size(imagen1,1);columns=size(imagen1,2);
catch
continue
end
r_fr=round(fr_o,0);
fr_str=num2str(r_fr);
% videoName=[videoName,'_',fr_str];
longitud_hendidura_mm=30;
```

```
matriz=[];
matriz_og=[];
medias_locales =[];
blur_idxs=[];
for t=1:paso:N
t;
num=num2str(t,'%04d');
nombre=[num '_Image_Centered.png'];
imagen=imread(nombre);
% figure,imshow(imagen,'InitialMagnification','fit')
imagen=double(rgb2gray(imagen));
imagen=imresize(imagen,[filas,columnas]);
% Obtener el modulo del gradiente de la imagen normalizada.
% min_imagen=min(imagen(:));
% imagen_0=imagen-min_imagen;
% max_imagen=max(imagen_0(:));
% imagen_normal=imagen_0/max_imagen;
% [px_i,py_i]=gradient(imagen_normal);
% mod=sqrt(px_i.^2,py_i.^2);
% [~,modulo_imagen]=normalizar_imagen(imagen);
% [f,xi]=ksdensity(modulo_imagen(:));
% [f_max,idx_max]=max(f);
% blur_idxs=[blur_idxs; [f_max xi(idx_max)]];
medias_locales(:,:,t) = media_local_polinomica( imagen,3 );
% figure,imagesc(media_local_polinomica( imagen,3 ))
% colormap('gray')
end
media_local_global=mean(medias_locales,3);imagen=[];
% figure,imagesc(media_local_global)
% colormap('gray')
for t=1:paso:N
t;
num=num2str(t,'%04d');
nombre=[num '_Image_Centered.png'];
imagen=imread(nombre);
imagen=double(rgb2gray(imagen));
imagen=imresize(imagen,[filas,columnas]);
imagen=imagen-medias_locales(:,:,t)+media_local_global;
matriz=[matriz imagen(:)];
end
%% Análisis de pca.
[pc,score,latent,tsquare]=pca(matriz);
pesos=100*latent/sum(latent);
%% Error de cada componente.
cova=covpc(score);
factor=3;
error_pesos=(100/sum(latent))*sqrt(diag(cova));
factor_por_error_pesos=factor*error_pesos;
figure
errorbar(pesos,factor_por_error_pesos,'.','vertical','LineWidth',3);
xlabel('#pca');
ylabel('% variance');
ax=gca;
ax.FontSize=24;
%% Quitamos la parte de ruido
% quitar=10;
% [v_p1,v_p2,v_p3,num_aislados,num_P3]=...
% clasificador_p1_p2_ruido(pesos,error_pesos,factor,quitar);
% En este caso procedemos a dibujar en y los pesos en escala
```

```
% logaritmica y obtenemos la rodilla de la curva.
[res_x, idx_of_result] = ...
knee_pt(log(pesos(1:round(max(size(pesos))*.92))),...
1:round(max(size(pesos))*.92));
try
matriz_sin_ruido=recons2(pc,matriz,[1:res_x]);
catch
continue
end
% Aquí están los frames originale, ya sin ruido.
%% Análisis del parpardeo a partir del primer componente
a1=abs(pc(:,1)-median(pc(:,1))); % correccion pc1, ajuste de baseline
x1=[1:max(size(a1))]';
tiempo1=x1/fr_o; % tiempo:, tiempo en el que está cada frame.
%% AQUI PROCEDEMOS A INTERPOLAR LA CURVA PC1 PARA PODER TENER
% LA MUESTRA EN DISTINTO FRAMERATE CON EL MISMO NÚMERO DE
% OBSERVACIONES.
% tiempo=[1:1080]/250;
% fr_int=250;
% x=[1:1080];
% a=interp1(tiempo1,a1,tiempo,'pchip');
a=a1;
x=x1;
tiempo=tiempo1;
%% Se obtienena los picos (parpadeos) a partir de la señal pc1 corregida
% restando el valor de la mediana de la señal cada uno de los puntos del pc1.
% [pks,locs,width,proms]=findpeaks(a,x,'MinPeakProminence',...
% 2*std(pc(:,1)),'MaxPeakWidth',round(fr_int),'Annotate','extents');
[pks,locs,width,proms,wxPk]=findpeaks_2(a,x,'MinPeakProminence',...
2*std(pc(:,1)),'Annotate','extents');
% Intervalo de cada uno de los parpadeos a media altura.
wxPk(:,1)=ceil(wxPk(:,1));
wxPk(:,2)=floor(wxPk(:,2));
figure,findpeaks(a,tiempo,'MinPeakProminence',...
2*std(pc(:,1)),'Annotate','extents')
% findpeaks sin interpolar.
xlabel('Time in seconds')
ylabel('Coefficient ratio of 1st PCA')
title('Video analysis with 1st PCA')
ax=gca;
ax.FontSize=24;
% Ahora analizamos, de momento sólo metemos el blink grade, luego habría
% que meter lo de las otras funciones.
NN=max(size(locs));
%% Comparar con la imagen PCA1 que está abierto.
%fotogram_pca1=reshape(score(:,1),rows,columns);
matriz_recons_pca1=recons2(pc,matriz,[1]);
fotogram_pca1=reshape(mean(matriz_recons_pca1,2),rows,columns);
%obtencion de la distancia en pixels entre el parpado superior e
%inferior
[px_pca1,py_pca1]=gradient(fotogram_pca1);
py_pca1=py_pca1(:,round(.15*columns):round(.85*columns));
medians_py_pca1=median(py_pca1,2);
[minimum_pca1, index_minimum_pca1]=...
min(medians_py_pca1(1:round(0.6*rows)));
index_maximum_pca1=rows;
%% obtencion del valor de normalizacion para estudiar la correlacion
% de las imagenes de ojo cerrado con el pca1
%% preparacion imagen pca1
```

```
% [f_pca1_n,m_g1]=normalizar_imagen(fotogram_pca1);
[f_pca1_n,m_g1]=quitar_media(fotogram_pca1);
%% CALCULO AUTOCORRELACION PCA1
% se obtiene el valor de autocorrelación de la imagen normalizada pca1
% que servbirá como medio para la obtencion del threshold que servirá
% como valor para identificar parpadeos de imagenes de movimiento del
% iris horizontal.
C_g1=xcorr2(m_g1,m_g1);
normalizacion=max(max(C_g1));
mascara_pca1=(C_g1>max(C_g1(:)/2));
area_pca1=sum(mascara_pca1(:));
radio_pca1=sqrt(area_pca1/pi);
indices_minimos=[];
indices_maximos=[];
grado=[];
porcentaje=0.8; % Esto es para que escoja algo más de la mitad de
% la promiencia y que tenga sentido el interpolar la curva.
pixels_pca1=index_maximum_pca1-index_minimum_pca1;
figure,imagesc(fotogram_pca1)
axis equal
axis off
colormap('gray')
title('pca1')
ax=gca;
ax.FontSize=24;
contrastes=[];
radios=[];
idx_contrastes=[];
idx_all_openess=[];
allindex_medians_py=[];
allindex_minimum=[];
allindex_maximum=[];
allmedians_py=[];
media_contrastes=[];
std_contrastes=[];
size_contrastes=[];
%% Estudio de grado de cierre a partir de la obtención de valor de grado
% estudiando la correlacion cruzada entre frame del video sin ruido y la
% imagen pca1, este valor se normaliza usando la autocorrelación pca1.
for cnt=1:max(size(locs))
pos=locs(cnt)/paso; % se divide entre el paso debido a la interpolacion
% de la señal
contrastes1=[];
radios1=[];
allindex_minimum1=[];
allindex_maximum1=[];
for frame=wxPk(cnt,1):wxPk(cnt,2)
fotogram=matriz_sin_ruido(:,frame);
% fotogram=matriz(:,pos);
fotogram_sq=reshape(fotogram,rows,columns);
%% Calculo de la correlacion cruzada metodo 1 normalizacion,
% método 2 gradiente
% [contraste,radio]=es_ojocerrado2(...
% frame,m_g1,fotogram_sq,normalizacion);
[contraste,radio]=es_ojocerrado3(...
frame,m_g1,fotogram_sq,normalizacion);
contrastes1=[contrastes1 contraste];
radios1=[radios1 radio];
fotogram=[];
```

```
end
% media_contrastes=[media_contrastes mean(contrastes1)];
% std_contrastes=[std_contrastes std(contrastes1)];
contrastes=[contrastes...
min(contrastes1+(radios1>median(radios1)+...
3*(-1/(sqrt(2)*erfcinv(3/2))*...
median(abs(radios1-median(radios1))))))];
std_contrastes=[std_contrastes std(contrastes1)];
size_contrastes=[size_contrastes max(size(contrastes1))];
[~,idx_contraste1]=min(contrastes1);
idx_contraste=wxPk(cnt,1)-1+idx_contraste1;
idx_contrastes=[idx_contrastes idx_contraste];
radios=[radios radios1];
filename_blink=[videoName,'_',num2str(cnt),'_eyeOpeness.mat'];
save(filename_blink,...
"idx_contraste","contrastes1","idx_contraste1","radios1")
end
% valor aun por ver
idx_parpadeos=find(contrastes<=0.95);
idx_no_parpadeos=find(contrastes>0.95);
idx_no_computan_minimo=(size_contrastes==1);
contrastes_parpadeos=contrastes;
contrastes_parpadeos(idx_no_parpadeos)=NaN;
isblink=zeros(size(contrastes));
isblink(idx_parpadeos)=1;
min_contrastes=min(contrastes_parpadeos+idx_no_computan_minimo);
grado_contrastes=100*(1-contrastes_parpadeos)/(1-min_contrastes);
grado_contrastes(find(grado_contrastes>100))=100;
grado_contrastes_corregido=6.9576e-05*grado_contrastes.^3-...
0.0190*grado_contrastes.^2+2.2129*grado_contrastes;
grado_contrastes_corregido(find(grado_contrastes_corregido>100))=100;
idx_parpadeos_completos=find(grado_contrastes_corregido>=75);
idx_parpadeos_incompletos=find(grado_contrastes_corregido>=25 &...
grado_contrastes_corregido<75);
num_parpadeos_completos=size(idx_parpadeos_completos,2);
num_parpadeos_incompletos=size(idx_parpadeos_incompletos,2);
locs_PC=locs(idx_parpadeos_completos);
locs_PI=locs(idx_parpadeos_incompletos);
if min(size(pixels_pca1))<1
pixels_pca1=size(fotogram,2)*0.5;
end
pixels_cierre=pixels_pca1*grado_contrastes_corregido./100;
filename_all_blinks=[videoName,'_all_eyeOpeness.mat'];
save(filename_all_blinks,...
"idx_contrastes")
aa=a;
%% Teniendo localizada imagen de ojo abierto sin recortar
% Seleccion de imagen del ojo abierta.
%cargamos el vídeo sin recortar de los 50 primeros frames
%y seleccionamos el que tenga nmayor intensidad luminica media.
% for cnt = 1:50
% num=num2str(cnt,'%04d');
% nombre=[num '_Image_Centered.png'];
% imagen=imread(nombre);
% Ig=rgb2gray(imagen);
% mean_luminosity(cnt)=mean(Ig(:));
% end
try
[~,Open_eye]=min(aa(round(max(size(aa))/2:...
```

```
round(max(size(aa))/2+50))));
catch
continue
end
Open_eye=round(max(size(aa))/2)+Open_eye-1;
num=num2str(Open_eye,'%04d');
nombre=[num '_Image_Centered.png'];
I_OE=imread(nombre);
Ig_OE=rgb2gray(I_OE);
media_local = media_local_polinomica(double(Ig_OE),3);
I_eye=double(Ig_OE)-media_local+mean2(media_local);
figure,imagesc(I_eye);
colormap('gray')
figure,plot(mean(I_eye),'LineWidth',3);
title('Mean value of each column of the Open Eye frame')
xlabel('Column number')
ylabel('Mean light intensity')
ax=gca;
ax.FontSize=24;
[pks_sclera,locs_sclera,width_sclera,proms_sclera]=findpeaks...
(mean(I_eye),'MinPeakProminence',...
2*std(mean(I_eye)),'Annotate','extents');
signal_corner=mean(I_eye);
%%detection first corner
mascara_1corner=signal_corner(1:locs_sclera(1))<...
(pks_sclera(1)-.8*proms_sclera(1));
out_x_corner_1=find(mascara_1corner);
columna_corner_1=out_x_corner_1(end);
%%detection 2nd corner
mascara_2corner=signal_corner(locs_sclera(end):end)<...
(pks_sclera(end)-.8*proms_sclera(end));
out_x_corner_2=find(mascara_2corner);
columna_corner_2=out_x_corner_2(1)+locs_sclera(end)-1;
[~,fila_corner_1]=max(I_eye(:,columna_corner_1));
[~,fila_corner_2]=max(I_eye(:,columna_corner_2));
fila_corner_1=0;
fila_corner_2=0;
longitud_hendidura_pixels=sqrt((fila_corner_1-fila_corner_2)^2+...
(columna_corner_1-columna_corner_2)^2);
%% conversión de pixels a mm.
% De pixels a mm "pixels2mm"
pixels2mm=longitud_hendidura_mm/longitud_hendidura_pixels;
alfa=0.85; % este factor sirve para truncar la desviacion estandar
%parpadeos_no analizados, no se detecta inicio y fin en primer
%algoritmo de aislamiento.
%Parpadeos a eliminar o descartados son aquellos en los que no se ha
%podido encontrar el punto inicial o final de la curva interpolada
% con el método de velocidades.
[parpadeos_no_analizados,parpadeos_descartados,...
parpadeos_a_eliminar,results,ini_parpadeo,...
fin_parpadeo,signal_factor]=...
blink_characterization_final(...
aa,alfa,fr_int,locs,width,pixels_cierre);
valid=ones(size(results,1),1);
valid(parpadeos_no_analizados)=0;
valid(parpadeos_a_eliminar)=0;
amplitude_change=[];grado_if_complete=[];
% if size(results,2)<1
All_results=[results locs/fr_int width/fr_int*1000 ...
```

```
amplitude_change'*pixels2mm grado' ...
grado_if_complete' valid isblink' grado_contrastes_corregido'];
%% obtención del valor de alfa y b a partir de la señal pc1.
[P,frec]=pspectrum(aa*pixels2mm,fr_int);
logP=log10(P); %% Para quitar la frecuencia de Nyquist.
logf=log10(frec); %% Para llegar hasta la frecuencia de Nyquist.
try
[fitresult, gof] = createFit(logf, logP);
catch
display(['The following video cannot be analyzed due to' ...
' inconsistency to convert the Fe1 factor ' ...
'to pixel amplitude'])
inconsistent_cnt=inconsistent_cnt+1;
inconsistent(inconsistent_cnt)=videoName;
% continue
end
xlabel('log[f(Hz)]');
ylabel('log PSD');
ax=gca;ax.FontSize=20;
ci=confint(fitresult, 0.65);
[PSD2,frec2,t2]=pspectrum(aa,fr_int,'spectrogram');
figure
pspectrum(aa,fr_int,'spectrogram')
ax=gca;ax.FontSize=20;
alpha=fitresult.p1;
sigmap1=abs(ci(2,1)-alpha);
b=fitresult.p2;
sigmap2=abs(ci(2,2)-b);
R2=gof.rsquare;
disp(['alfa std_alfa b std_b R2'])
resultados=[alpha sigmap1 b sigmap2 R2];
%% calculo de alfa y b a partir de la señal en mm.
%a partir de la señal original pca y posteriormente aisalar y estudiar
%el grado de cada uno de los parpadeos, se procede a modificar la
%señal y transformarla en mm acorde a la amplitud de los parpadeos
%identificados como verdaderos con el estudio de correlacion cruzada e
%identificación de frames con blurr
%% Cambio de señal pca
signal_mm= aa;
try
for i=1:max(size(ini_parpadeo))
signal_mm(ini_parpadeo(i):fin_parpadeo(i))=aa(...
ini_parpadeo(i):fin_parpadeo(i))*signal_factor(i)*...
pixels2mm;
end
catch
continue
end
signal_mm=fillmissing(signal_mm,'constant',0);
[P_mm,frec_mm]=pspectrum(signal_mm,fr_int);
logP_mm=log10(P_mm); %% Para quitar la frecuencia de Nyquist.
logf_mm=log10(frec_mm); %% Para llegar hasta la frecuencia de Nyquist.
try
[fitresult_ab_mm, gof_ab_mm] = createFit(logf_mm, logP_mm);
catch
display(['The following video cannot be analyzed due to' ...
' inconsistency to convert the' ...
' Fe1 factor to pixel amplitude'])
inconsistent_cnt=inconsistent_cnt+1;
```

```
inconsistent(inconsistent_cnt)=videoName;
% continue
end
xlabel('log[f(Hz)]');
ylabel('log PSD');
ax=gca;ax.FontSize=32;
ci=confint(fitresult, 0.65);
[PSD2_mm,frec2_mm,t2_mm]=pspectrum(signal_mm,fr_int,'spectrogram');
figure
pspectrum(signal_mm,fr_int,'spectrogram')
ax=gca;ax.FontSize=32;
alpha_mm=fitresult_ab_mm.p1;
sigmap1_mm=abs(ci(2,1)-alpha);
b_mm=fitresult_ab_mm.p2;
sigmap2_mm=abs(ci(2,2)-b);
R2_mm=gof_ab_mm.rsquare;
disp(['alfa_mm std_alfa_mm b_mm std_b_mm R2_mm'])
resultados_mm=[alpha_mm sigmap1_mm b_mm sigmap2_mm R2_mm];
%% calculo de pico y prominencia grado de parpadeo.
All_results(All_results==0)=NaN;
if size(All_results,1)>1
results_blinks=[fillmissing(mean(All_results,1,'omitnan')',...
'constant',0) fillmissing(std(All_results,1,'omitnan')',...
'constant',0) kurtosis((All_results),1)'...
skewness((All_results),1)'];
if size(idx_parpadeos_completos,2)>1
results_complete=[fillmissing(mean((All_results...
(idx_parpadeos_completos,:)),1,'omitnan')',...
'constant',0) fillmissing(std((All_results(...
idx_parpadeos_completos,:)),1,'omitnan')','constant',...
0) kurtosis((All_results(idx_parpadeos_completos,:))...
,1)' skewness((All_results(idx_parpadeos_completos,...
:)),1)'];
else
results_complete=zeros([53,4]);
end
if size(idx_parpadeos_incompletos,2)>1
results_incomplete=[fillmissing(mean...
((All_results(idx_parpadeos_incompletos,:)),1,...
'omitnan')','constant',0)...
fillmissing(std((All_results...
(idx_parpadeos_incompletos,:)),1,'omitnan')',...
'constant',0) kurtosis((All_results(...
idx_parpadeos_incompletos,:)),1)'...
skewness((All_results(idx_parpadeos_incompletos,:)),1)'];
else
results_incomplete=zeros([53,4]);
end
sumary_results=[results_blinks...
results_complete results_incomplete;
mean(diff(locs'/fr_int)) std(diff(locs'/fr_int))...
kurtosis(diff(locs'/fr_int))...
skewness(diff(locs'/fr_int))... calculo IBI
mean(diff(locs_PC'/fr_int)) std(diff(locs_PC'/fr_int))...
kurtosis(diff(locs_PC'/fr_int))...
skewness(diff(locs_PC'/fr_int))...
mean(diff(locs_PI'/fr_int))...
std(diff(locs_PI'/fr_int)) kurtosis(diff(locs_PI'/fr_int))...
skewness(diff(locs_PI'/fr_int));
```

```
mean(nonzeros(results(:,35))) std(nonzeros(results(:,35)))...
kurtosis(nonzeros(results(:,35)))...
skewness(nonzeros(results(:,35)))... calculo duracion
mean(nonzeros(results(idx_parpadeos_completos,35)))...
std(nonzeros(results(idx_parpadeos_completos,35)))...
kurtosis(nonzeros(results(idx_parpadeos_completos,35)))...
skewness(nonzeros(results(idx_parpadeos_completos,35)))...
mean(nonzeros(results(idx_parpadeos_incompletos,35)))...
std(nonzeros(results(idx_parpadeos_incompletos,35)))...
kurtosis(nonzeros(results(idx_parpadeos_incompletos,35)))...
skewness(nonzeros(results(idx_parpadeos_incompletos,35)));
num_parpadeos_completos+num_parpadeos_incompletos NaN NaN...
NaN ...
num_parpadeos_completos NaN NaN NaN ...
num_parpadeos_incompletos NaN NaN NaN;
alpha sigmap1 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
b sigmap2 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
R2 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
alpha_mm sigmap1_mm NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
b_mm sigmap2_mm NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
R2_mm NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
res_x NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN];
else
sumary_results_0=zeros([55,12]);
summary_results=[sumary_results_0;
num_parpadeos_completos+num_parpadeos_incompletos NaN NaN...
NaN...
num_parpadeos_completos NaN NaN NaN ...
num_parpadeos_incompletos NaN NaN NaN;
alpha sigmap1 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
b sigmap2 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
R2 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
alpha_mm sigmap1_mm NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
b_mm sigmap2_mm NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
R2_mm NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN;
res_x NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN];
end
variables=...
["t0p (s)" "t1p (s)" "t2p (s)" "t3p (s)" "t4p (s)"...
"t5p (s)" "t6p (s)" "t7p (s)" "t8p (s)" "P1"...
"P3" "P5" "P7" "W02" "W24" "W46" "W68"...
"t1a (s)" "t2a (s)" "t3a (s)"...
"a1a (s)" "a2a (s)" "a3a (s)" "J02" "J26" "J68" "v2p"...
"v6p" "S" "R2_ajuste" "norma" "sigma" "mu" "tau" "base"...
"prom" "tiempo_blink (ms)" "half_width_left (ms)"...
"half_width_right (ms)" "half_width (ms)"...
"width_left (ms)" "width_right (ms)"...
"width (ms)" "a1a_sn" "a2a_sn"...
"a3a_sn" "v2p_sn" "v6p_sn" "locs_fp (s)"...
"Hwidth_fp (ms)" "validity" "isblink" "closeness (%)"];
variables_summary=[variables "IBI" "DUR" "num" "alpha" "b" "R2"...
"alpha_mm" "b_mm" "R2_mm" "limit_noise_pc"];
columns_summary=[ "total_mean" "total_std" "total_kurtosis"...
"total_skewness" "complete_blinks_mean" "complete_blinks_std"...
"complete_blinks_kurtosis" "complete_blinks_skewness"...
"incomplete_blinks_mean" "incomplete_blinks_std"...
"incomplete_blinks_kurtosis" "incomplete_blinks_skewness"];
summary_table=array2table(sumary_results,"RowNames",...
variables_summary,"VariableNames",columns_summary);
```

```
% Tabla csv de resultados
table_output= array2table([All_results],"VariableNames",variables);
writetable( table_output,'results_prueba.csv');
summary_name=...
['C:\Users\Marra\OneDrive\Bureau\Figuras_pchip_emg\',...
videoName,'.csv'];
writetable(summary_table,summary_name,'WriteRowNames',true)
% Tabla con los datos de las señales. Pca1 y conversion a mm.
table_signal=array2table([x,x/fr_int,aa,signal_mm],VariableNames=...
["Frame_number" "Time (s)" "Fe1_signal" "signal_mm"]);
signal_name=...
['C:\Users\Marra\OneDrive\Bureau\Figuras_pchip_emg\',...
videoName,'_signal.csv'];
writetable(table_signal,signal_name,'WriteRowNames',true)
table_video_properties=array2table([fr_int,rows,columns,...
longitud_hendidura_mm,pixels2mm],VariableNames=...
["Frame rate" "Video height (pixels)" "Video width (pixels)"...
"Longitude Palpebral Cleft (mm)"...
"Factor Pixels2mm (mm/Pixels)"]);
properties_name=...
['C:\Users\Marra\OneDrive\Bureau\Figuras_pchip_emg\',...
videoName,'_properties.csv'];
writetable(table_video_properties,properties_name,...
'WriteRowNames',true)
%% SAVING WORKSPACE
name_workspace=[videoName,'.mat'];
save(name_workspace)
%% Ahora tendríamos la caracterización de todos los parpadeos.
% procedemos a calcular los valores relativos al vídeo. Duracion e
% intervalo entre parpadeos asi como los distintos valores medias,
% desviacion estandar kurtosis y skewness de los extraidos para cada
% parpadeo. Se tienen en cuenta únicamente aquellos distintos de cero.
%% Exportación de la base de datos a MONGODB
server = "localhost";
port = 27017;
dbname = "mymongodb";
conn = mongoc(server,port,dbname);
isopen(conn)
data = readtable("results_prueba.csv");
data2 = readtable(summary_name);
data3 = readtable(signal_name);
data4 = readtable(properties_name);
tsunamidata = table2struct(data);
syphondata = table2struct(data2);
tornadodata = table2struct(data3);
stormdata = table2struct(data4);
tsunamicoll = videoName;
syphoncoll = [videoName,'_summary'];
tornadocoll = [videoName,'_signal'];
stormcoll = [videoName,'_properties'];
%% Check collection
% Extract collection names from the command result
collectionNames = conn.CollectionNames;
% Check if the collection exists
collectionName = tsunamicoll;
if ismember(collectionName, collectionNames)
fprintf('Collection "%s" exists.\n', collectionName);
% Drop the collection
dropCollection(conn, collectionName);
```

```
fprintf('Collection "%s" has been dropped.\n', collectionName);
else
fprintf('Collection "%s" does not exist.\n', collectionName);
end
collectionName = syphoncoll;
if ismember(collectionName, collectionNames)
fprintf('Collection "%s" exists.\n', collectionName);
% Drop the collection
dropCollection(conn, collectionName);
fprintf('Collection "%s" has been dropped.\n', collectionName);
else
fprintf('Collection "%s" does not exist.\n', collectionName);
end
collectionName = tornadocoll;
if ismember(collectionName, collectionNames)
fprintf('Collection "%s" exists.\n', collectionName);
% Drop the collection
dropCollection(conn, collectionName);
fprintf('Collection "%s" has been dropped.\n', collectionName);
else
fprintf('Collection "%s" does not exist.\n', collectionName);
end
collectionName = stormcoll;
if ismember(collectionName, collectionNames)
fprintf('Collection "%s" exists.\n', collectionName);
% Drop the collection
dropCollection(conn, collectionName);
fprintf('Collection "%s" has been dropped.\n', collectionName);
else
fprintf('Collection "%s" does not exist.\n', collectionName);
end
createCollection(conn,tsunamicoll);
createCollection(conn,syphoncoll);
createCollection(conn,tornadocoll);
createCollection(conn,stormcoll);
n = insert(conn,tsunamicoll,tsunamidata);
n2 = insert(conn,syphoncoll,syphondata);
n3 = insert(conn,tornadocoll,tornadodata);
n3 = insert(conn,stormcoll,stormdata);
end
```

Table 8-2: media_local_polinomica function developed in Matlab Environment. Called on the main script Table 8-1.

```
function [ media_local ] = media_local_polinomica( imagen,n );
[filas columnas]=size(imagen);
ejec=[1:columnas];
ejef=[1:filas];

imagenh=zeros(size(imagen));
imagenhv=zeros(size(imagen));

for f=1:filas
    linea=imagen(f,:);
```

```
    ajuste=polyfit(ejec,linea,n);
    suma=zeros(size(ejec));
    for nn=1:n+1
        suma=suma+ajuste(nn)*(ejec.^(n+1-nn));
    end
    imagenh(f,:)=suma;
end

for c=1:columnas
    linea=imagenh(:,c);
    ajuste=polyfit(ejef',linea,n);
    suma=zeros(size(ejef'));
    for nn=1:n+1
        suma=suma+ajuste(nn)*(ejef'.^(n+1-nn));
    end
    imagenhv(:,c)=suma;
end

media_local=imagenhv;



end
```

Table 8-3: covpc function coded in Matlab Environment.

```
%% funcion para calcular la matriz de covarianzas entre los cp de una matriz.

function [covarianza]=covpc(pc) % Cada columna es un componente principal.
N=min(size(pc));
M=max(size(pc));
covarianz=zeros(N,N);

for t=1:N
   varianzas(t)=(1/M)*2.*(var(pc(:,t))^2)+(1/M)*cum4i(pc(:,t));
end

covarianza=diag(varianzas,0);

for f=1:N
    for c=f+1:N
        covarianza(f,c)=cum22ij(pc(:,f),pc(:,c))*(1/M);
        covarianza(c,f)=covarianza(f,c);
```

```
    end
end
```

Table 8-4: cum4i function called by covpc function on Table 8-3.

```
%% Funcion para calcular el cumulante de orden 4.
function [salida]=cum4i(yi);
M=max(size(yi(:)));
salida=mean2(yi.^4)-3*(mean2(yi.^2)^2);
```

Table 8-5: cum22ij function called by covpc function on Table 8-3.

```
%% funcion para calcular el cumulante de orden 22
function [salida]=cum22ij(yi,yj);
M=max(size(yi(:)));
yi2=yi.^2;yj2=yj.^2;
salida=(1/M)*yi2(:)'*yj2(:)-(mean2(yi2)^2)*(mean2(yj2)^2);
```

Table 8-6: recons2 function on matlab environment. Called on the main script on Table 8-1.

```
%% Programa para la reconstrucción de acuerdo con los componentes
% principales que se escojan.
% numeros es un vector donde se ponen que orden de acuerdo con los
% autovectores son los que hay que seleccionar.
% En la matriz está la reconstrucción correspondiente a los componentes
% seleccionados para cada una de las variables originales, puestas cada
% reconstrucción en una columna.
% La sintaxis es [matriz]=recons2(autovector,x,numeros);
% El autovector es la matriz pc,x son los datos originales.

function [matriz]=recons2(autovector,x,numeros);
[fila columna]=size(numeros);
p=[];
for t=1:columna
       p=[p autovector(:,numeros(t))];
end

p_rec=p*p';
[fila2 col2]=size(x);
N=fila2;

e=ones(N,1);
```

```
x_t=(1/N)*e'*x;
x_m=x-e*x_t;

x_m_tilde=x_m*p_rec;

matriz=e*x_t+x_m_tilde;
```

Table 8-7: Modification on original findpeaks matlab environment function to use findpeaks_2 to extract exact coordinates, based on the location of each peak, values at half width

```
function [Ypk,Xpk,Wpk,Ppk,wxPk] = findpeaks_2(Yin,varargin)
%% (...) after follows the same code used for the findpeaks original function.
% The change on the original function to extract extra information about the
% coordinates at half width is due to observing the original findpeaks matlab
% environment function line 163;
% if needWidth
%   % obtain the indices of each peak (iPk), the prominence base (bPk), and
%   % the x- and y- coordinates of the peak base (bxPk, byPk) and the width
%   % (wxPk)
%   [iPk,bPk,bxPk,byPk,wxPk] = signal.internal.findpeaks.findExtents(...
%       y,x,iPk,iFinite,iInfinite,iInflect,minP,minW,maxW,refW);
% else
%   % combine finite and infinite peaks into one list
%   [iPk,bPk,bxPk,byPk,wxPk] = combinePeaks(iPk,iInfinite);
% end
```

Table 8-8: es_ojocerrado3 Matlab function.

```
%% es_ojocerrado3 funcion para obtencion de correlacion cruzada a partir de la imagen
sin media.
function [contraste,radio]=es_ojocerrado3(cnt,m_g1,fotogram,normalizacion)

str_cnt=string(cnt);

%normalizamos y obtenemos el modulo del fotograma

[fotogram_sm,m_gx]=quitar_media(fotogram);

C_gx=xcorr2(m_g1,m_gx)/normalizacion;

% figure,imagesc(C_gx)
% colorbar
```

```
% title(str_cnt)
% ax=gca;
% ax.FontSize=24;

% figure,mesh(C_gx)
% colorbar
% title(str_cnt)
% ax=gca;
% ax.FontSize=24;
% hold on
% mesh(ones(size(C_gx))*max(C_gx(:)/2))
% hold off

contraste=max(max(C_gx));

% obtencion del blur
mascara=(C_gx>max(C_gx(:)/2));

% figure,imagesc(mascara)
% colorbar
% title(str_cnt)
% ax=gca;
% ax.FontSize=24;

area=sum(mascara(:));
radio=sqrt(area/pi);

end
```

Table 8-9: blink_characterization_final function called by the main script on Table 8-1. Used to isolate and furthermore adjust the signal to an exponentially modified gaussian function.

```
function
[parpadeo_no_analizado,parpadeos_descartados,parpadeo_a_eliminar,results,ini_parpadeo,fi
n_parpadeo,signal_factor]=blink_characterization_final(a,alfa,fr,locs,width,pixels_cierr
e)
parpadeos_descartados=[];
parpadeo_analizado=[];
parpadeo_no_analizado=[];
parpadeo_a_eliminar=[];
nuevo_pixels_cierre=[];
results1=[];
signal_factor=[];
```

```
%% checkear el inicio de los picos a partir del salto producido entre el punto estudiado
y los dos siguientes.
%% test de la funcion script.
valor_maxpico=[];
localizacion_pico=[];

% % Calculo del threshold:
[pdfa,ejea]=ksdensity(a);
[maxpdf,idx_max]=max(pdfa);
media_a=ejea(idx_max);
[CdfY,CdfX] = ecdf(a,'Function','cdf');

%% alfa indica el threshold de probabilidad alfa=0.85
alfa=0.85;
mayor_85=find(CdfY>alfa);
valor_85=CdfX(mayor_85(1));
desv_a=valor_85-media_a;

% desv_a=0.0026;
test=0;test2=0;
incremento=0;
i=0;j=0;

ini_parpadeo=[];
fin_parpadeo=[];
i=[];
for cnt=1:max(size(locs))
    test0=0;
    LOC=locs(cnt);
    WIDTH=width(cnt);
    PIXELS_CIERRE=pixels_cierre(cnt);
    i=locs(cnt);
    while a(i)>mean(a)-std(a) && i>3
        incremento1=(-a(i)+a(i-1));
        incremento2=(-a(i-1)+a(i-2));
        incremento3=(-a(i)+a(i-3));

        % (abs((i-2)-LOC))
        if (incremento1<0 && incremento2<0) || abs(incremento3)>abs(desv_a) || abs(a(i-
1)-a(locs(cnt)))<desv_a && (abs((i-1)-LOC))<ceil(WIDTH*2)
            % || abs(LOC-i)<2*ceil(WIDTH))
            i=i-2;
        else
            test=test+1;
```

```
            i=i-2;
            ini_parpadeo(test)=i;
            test0=1;

            break;
        end
    end
    j=locs(cnt);
    while a(j)>mean(a)-std(a) && j<max(size(a))-3
        incremento1d=(-a(j)+a(j+1));
        incremento2d=(-a(j+1)+a(j+2));
        try
            incremento3d=(-a(j)+a(j+3));
        catch
            incremento3d=0;
        end
        if (incremento1d<0 && incremento2d<0) || abs(incremento3d)>abs(desv_a) ||
abs(a(j+1)-a(locs(cnt)))<desv_a && abs(LOC-j+1)<ceil(WIDTH*2)
            j=j+2;
        else
            test2=test2+1;
            j=j+2;
            fin_parpadeo(test2)=j;
            test0=1;
            break;
        end
    end
    if test0==0
        parpadeo_no_analizado=[parpadeo_no_analizado cnt];

    elseif test<test2
        fin_parpadeo=fin_parpadeo(1:test);
        test2=test;
        parpadeo_no_analizado=[parpadeo_no_analizado cnt];

    elseif test2<test
        test=test2;
        ini_parpadeo=ini_parpadeo(1:test2);
        parpadeo_no_analizado=[parpadeo_no_analizado cnt];
    else
        parpadeo_analizado=[parpadeo_analizado cnt];
        nuevo_pixels_cierre=[nuevo_pixels_cierre PIXELS_CIERRE];
    end
end
```

```
%Interpolamos cada 0.01 frame
factor=0.01;

% factor conversion frame a milisegundos
t_ms=(1/fr)*1000;

results=[];

for i=1:size(ini_parpadeo,2)
    % for i=2:2

    xnew=[];
    %definicion de eje.
    [valor_maxpico(i),localizacion_pico(i)]=max(a(ini_parpadeo(i):fin_parpadeo(i)));

    if fin_parpadeo(i)-ini_parpadeo(i)<4*width(i)

        xnew=[ini_parpadeo(i):factor:fin_parpadeo(i)];
        time_new=xnew*t_ms;
        x_old=[ini_parpadeo(i):fin_parpadeo(i)];
        time_old=x_old*t_ms;
        curva_old=[a(ini_parpadeo(i):fin_parpadeo(i))];

    else
    if round(locs(i)-1.5*width(i))>1

        xnew=[round(locs(i)-1.5*width(i)):factor:round(locs(i)+2*width(i))];
        time_new=xnew*t_ms;
        x_old=[round(locs(i)-1.5*width(i)):round(locs(i)+2*width(i))];
        time_old=x_old*t_ms;
        curva_old=[a(round(locs(i)-1.5*width(i)):round(locs(i)+2*width(i)))];
    else
        xnew=[1:factor:round(locs(i)+2*width(i))];
        time_new=xnew*t_ms;
        x_old=[1:round(locs(i)+2*width(i))];
        time_old=x_old*t_ms;
        curva_old=[a(1:round(locs(i)+2*width(i)))];
    end
    end

    [curva_new]=interp1(x_old,curva_old,xnew,'pchip');

    formatSpec= '%d';
    FigureName1=['blink, pchip: ', num2str(i,formatSpec)];
```

```
    FigureName2=['blink speed from pchip: ', num2str(i,formatSpec)];

    figure,plot(time_new,curva_new,'LineWidth',3)
    set(gcf, 'Name', FigureName1)
    title(FigureName1)
    xlabel('Time (s)')
    ylabel('Factorized pca1 signal (a.u.)')
    ax=gca;
    ax.FontSize=24;
    figure,plot(time_new,gradient(curva_new),'LineWidth',3)
    set(gcf, 'Name', FigureName2)
    title(FigureName2)
    xlabel('Time (s)')
    ylabel('Speed (a.u./s)')
    ax=gca;
    ax.FontSize=24;

    %% búsqueda del punto t0: inicio del parpadeo
    % el inicio del parpadeo se produce cuando la velocidad de parpadeo es
    % positiva, es decir, empieza a moverse el párpado en dirección al
    % párpado inferior.
    gradient_curva_new=gradient(curva_new);
    curva_vn=gradient_curva_new/max(abs(gradient_curva_new));
    % consideramos que la velocidad es cero cuando vale 0.3% de la
    % velocidad máxima.
    valor_0=0.003;


[max_cv1,loc_max_cv1]=max(curva_new(round(.1*size(curva_new,2)):round(.8*size(curva_new,
2))));
    [max_cv2,loc_max_cv2]=max(curva_vn(1:round(.8*size(curva_new,2))));
    if loc_max_cv2<loc_max_cv1
        if loc_max_cv2~=1
            loc_max_cv=loc_max_cv2;
        else
            [v_max,loc_max_cv0,~]=findpeaks(curva_vn(1:round(.8*size(curva_new,2))));
            [~,indx_max_cv]=max(v_max);
            loc_max_cv=loc_max_cv0(indx_max_cv);
        end
    else
        % loc_max_cv=round(.7*loc_max_cv1);
        % [peak_max_cv,loc_max_cv0,~] = findpeaks(curva_vn(1:loc_max_cv1));
        % [~,L]=max(peak_max_cv);
        % loc_max_cv=loc_max_cv0(L);
```

```
        [peak_max_cv,loc_max_cv0] = max(curva_vn(1:loc_max_cv1));
        loc_max_cv=loc_max_cv0;
    end
    curva_vn_inicial=curva_vn(1:loc_max_cv-1);
    pto0=find(curva_vn_inicial<valor_0);
    try
        pto0=pto0(end);
        t0p=xnew(pto0);
    catch
        parpadeos_descartados=[parpadeos_descartados [ ini_parpadeo(i);fin_parpadeo(i)
]];
        parpadeo_a_eliminar=[parpadeo_a_eliminar i];
        output=[];
        results1(i,:)=zeros(1,48);
        signal_factor(i)=NaN;
        continue
    end
    %% nueva curva de analisis después de eliminar todos los puntos anteriores a al
punto t0p

    xnew=xnew(pto0:end);
    curva_new=curva_new(pto0:end);
    curva_vn=curva_vn(pto0:end);

    %% búsqueda del punto t8: final del parpadeo

    [min_cv,loc_min_cv]=min(curva_vn);

    curva_vn_final=curva_vn(loc_min_cv:end);
    pto8=find(curva_vn_final<0);
    try
        pto8=pto8(end)+loc_min_cv-1;
    catch
        pto8=size(xnew,2);
    end
    t8p=xnew(pto8);

    curve_new=curva_new(1:pto8);
    x_new=[t0p:factor:t8p];

    %% Pasar la señal del parpadeo a pixels teniendo en cuenta el cierre
    % del parpado máximo obtenido a partir de los contrastes

    % Normalizamos la curva
```

```
    curve_new_n=(curve_new-min(curve_new))/max(curve_new-min(curve_new));
    curve_new_pixels=curve_new_n.*nuevo_pixels_cierre(i);

    signal_factor(i)=1/max(curve_new)*nuevo_pixels_cierre(i);



    %Este es el nivel base (para calculod e prom)
    % base(i)=(a(ini_parpadeo(i))+a(fin_parpadeo(i)))/2;
    % prom(i)=valor_maxpico(i)-base(i);
    % [valor_newmax(i),localizacion_new(i)]=max(curva_new);
    % tiempo_blink(i)=xnew(localizacion_new(i))*t_ms;
    % half_width_axis=xnew(curva_new>(valor_maxpico(i)-(prom(i)/2)));
    % half_width_left(i)=(xnew(localizacion_new(i))-half_width_axis(1))*t_ms;
    % half_width_right(i)=(half_width_axis(end)-xnew(localizacion_new(i)))*t_ms;
    % half_width(i)=half_width_left(i)+half_width_right(i);
    % width_axis=xnew(curva_new>(valor_maxpico(i)-(prom(i))));
    % width_left(i)=(xnew(localizacion_new(i))-width_axis(1))*t_ms;
    % width_right(i)=(width_axis(end)-xnew(localizacion_new(i)))*t_ms;
    % width(i)=(width_left(i)+width_right(i));

    %% aquí calculo todas las features de los parpadeos para la caraterizació.
    %% dentro de la curva se encuentra todo lor relativo a la caracterizacióm,
    %% calculo de velocidades y accecleraciones.

    [output,R,norma,x,curva_ajustada]=ajustar_expgauss_blink_final...
        (curve_new_pixels,x_new,t_ms,i);

    results1(i,:)=output;

end

if size(parpadeo_a_eliminar,2)>0
    results1(parpadeo_a_eliminar,:)=[];
    parpadeo_analizado(parpadeo_a_eliminar)=[];
end
results=zeros(max(size(locs)),size(results1,2));
results(parpadeo_analizado,:)=results1;
if size(results,2)==0;
    results=zeros([size(locs,1),48]);
end
```

Table 8-10: ajustar_expgauss_blink_final function called on Table 8-9. It purpose is to adjust the interpolated isolated blink sequence signal to an exponentially modified gaussian function.

```
%% Funcion ajustar a exp_ mod_ gaussian blink final
%% Script del día 25/12/2024
%% Código para hacer una ajuste a una
% exponencial gaussian.tau=std(data).*0.8;
% reasonable starting value for tau
%% La distribución la calculamos a mano a patir del pulso.
function [output,R,norma,x,curva_ajustada]=...
    ajustar_expgauss_blink_final(curve_new,xnew,t_ms,cnt)
% function [output,R,norma]=ajustar_expgauss_blink2(curve_new)
x=[1:max(size(curve_new))];

fr=1000/t_ms;
% dx=x(2)-x(1);
norma=sum(curve_new);
curve_new_pdf=curve_new/norma;
media=sum(x.*curve_new_pdf); % no lleva dx ya es "1".
sigma=sqrt(sum(curve_new_pdf.*((x-media).^2)));
tau=sigma.*0.8;
mu=media-tau;   % reasonable starting value for mu
sig=sqrt((sigma^2)-(tau^2));    % reasonable starting value for sig

pinit  = [mu sig tau]; % put starting parameter values in an array

R=fminsearch(@(params) MSE_eglike(params,x,curve_new_pdf),pinit);
    % given the data, and starting parameters in
    % pinit, find the parameter values that minimize eglike
    % the function returns R=[mu, sig, tau]

curva_ajustada=exgausspdf(R(1),R(2),R(3), x)*norma;
% la función del ajuste necesita que sea una pdf por eso se normaliza
% "norma", y luego multiplicaomos por la norma.

%% Calculo del R2

sigma_r2=mean((curve_new-curva_ajustada).^2);
sigma_y2=mean((curve_new-mean(curve_new)).^2);

R2_ajuste=1-(sigma_r2/sigma_y2);



%% a partir de la curva ajustada calculo todos los puntos de interés de la
% nueva curva.
```

```
%% Visualización curva pchip vs curva ajustada
xnew_fr=xnew/fr;
figure,
plot(xnew_fr,curve_new,'DisplayName','Only interpoled','LineWidth',3)
xlabel('Video time ocurrence of the blink')
ylabel('Amplitude in pixels of the eye closiness')
title(...
    'Comparison interpolated curve blink sequence vs EMG characterization')
hold on
plot(xnew_fr,curva_ajustada,'DisplayName','Characterized','LineWidth',3)
hold off
ax=gca;
ax.FontSize=24;

%% Guardado de datos
r_fr=round(fr,0);
fr_str=num2str(r_fr);
cnt_str=num2str(cnt);
nombre_pto_mat=['pchip_emg_',fr_str,'_',cnt_str];
fullname=['C:\Users\Marra\OneDrive\Bureau\Figuras_pchip_emg\',nombre_pto_mat];
save(fullname,"xnew_fr","curva_ajustada","curve_new" )

%% visualizacion del input
formatSpec= '%d';
FigureName=['blink characteristics: ', num2str(cnt,formatSpec)];
figure
set(gcf, 'Name', FigureName)
title(FigureName)
ax=gca;
ax.FontSize=24;
yyaxis right
plot(xnew/fr,curva_ajustada,...
    'DisplayName','Vertical amplitude in pixels','LineWidth',2)

% calculo de la velocidad como gradiente

curva_v=gradient(curva_ajustada);
curva_vn=curva_v/max(abs(curva_v));

curva_a=gradient(gradient(curva_ajustada));
curva_an=curva_a/max(abs(curva_a));

curva_P=curva_v.*curva_a;
curva_Pn=curva_P/max(abs(curva_P));
```

```
yyaxis left
plot(xnew/fr,curva_vn,...
    'DisplayName','Normalized speed','LineWidth',2)

hold on
plot(xnew/fr,curva_an,...
    'DisplayName','Normalized acceleration','LineWidth',2)

plot(xnew/fr,curva_Pn,...
    'DisplayName','Normalized power','LineWidth',2)

%
% hold off

%% extraccion de los cuatro puntos calculables: t0p, t2p, t4p, t6p, t8p.
% con relación a la velocidad
% t0p; punto en que se considera el parpado está abierto, velocidad igual a
% 0
pto0=1;
t0p=xnew(1);

% t2p:punto en el que la fuerza desarrollada por el párpado es nula y la
% velocidad es máxima. En este punto el mmúsculo del parpado pasa de
% intentar cerrar el ojo a empezar a frenar el cierre.

[v2p,pto2] = max(curva_vn);
t2p = xnew(pto2);
v2p_sn=curva_v(pto2);

% t4p: punto en el que parpado cnsigue el mayor estado cierre (parpado
% completamente cerrado o punto de máximo cierre.

[v4p,pto4] = max(curva_ajustada);
t4p = xnew(pto4);
v4p_sn=curva_v(pto4);

% t6p: PUNTO DE MÁXIMA VELOCIDAD, en este momento los músculos del
% párpado pasan de intentar abrir el ojo a frenar la apertura.

[v6p,pto6] = min(curva_vn);
t6p = xnew(pto6);
v6p_sn=curva_v(pto6);
```

```
% t8p: punto en el que se considera el ojo totalmente abierto velocidad
% igual a0.

pto8=max(size(curva_ajustada));
t8p=xnew(end);
v8p_sn=curva_v(pto8);

%% ptos calculables  a la relacion de aceleración. t1p, t3p, t5p, t7p.
try
[~,pto1] = findpeaks(curva_P); %
t1p=xnew(pto1(1));

[~,pto3] = min(curva_P);
t3p=xnew(pto3);

curva_P_apertura=curva_P(pto4:end);

[~,pto5i] = max(curva_P_apertura);
pto5=pto5i+pto4;
t5p=xnew(pto5);

[~,pto7i] = min(curva_P_apertura);
pto7=pto7i+pto4;
t7p=xnew(pto7);

puntos_importantes= ...
    [t0p/fr t1p/fr t2p/fr t3p/fr t4p/fr t5p/fr t6p/fr t7p/fr t8p/fr];
xline(puntos_importantes)

% leyenda
legend

hold off

%% S es la relación de velocidades:

S=((curva_ajustada(pto4)-curva_ajustada(pto0))*(t8p-t4p))/...
    ((curva_ajustada(pto4)-curva_ajustada(pto8))*(t4p-t0p));

%% features related to normalized absolute power:

P1=curva_Pn(pto1(1));
P3=curva_Pn(pto3);
P5=curva_Pn(pto5);
```

```
P7=curva_Pn(pto7);

%% Work performed in each step calculated from tghe normalized power:

% W02=sum(curva_Pn(1:pto2));
% W24=sum(curva_Pn(pto2:pto4));
% W46=sum(curva_Pn(pto4:pto6));
% W68=sum(curva_Pn(pto6:end));

W02=trapz(xnew(1:pto2)/fr,curva_Pn(1:pto2));
W24=trapz(xnew(pto2:pto4)/fr,curva_Pn(pto2:pto4));
W46=trapz(xnew(pto4:pto6)/fr,curva_Pn(pto4:pto6));

try
    W68=trapz(xnew(pto6:end)/fr,curva_Pn(pto6:end));
catch
    W68=0;
end

%% Acceleration time;
curva_an_apertura=curva_an(pto6:end);
[a1a,pto1a]=max(curva_an);
t1a=xnew(pto1a);
a1a_sn=curva_an(pto1a);

[a2a,pto2a]=min(curva_an);
t2a=xnew(pto2a);
a2a_sn=curva_an(pto2a);

[a3a,pto3ai]=max(curva_an_apertura);
pto3a=pto3ai+pto6-1;
t3a=xnew(pto3a);
a3a_sn=curva_an(pto3a);

% a1a, a2a, a3a : acceleraciones correspondientes al tiempo 1, 2 y 3.

% J02=sum(curva_an(1:pto2));
% J26=sum(curva_an(pto2:pto6));
% J68=sum(curva_an(pto6:end));

J02=trapz(xnew(1:pto2)/fr,curva_an(1:pto2));
J26=trapz(xnew(pto2:pto6)/fr,curva_an(pto2:pto6));
try
    J68=trapz(xnew(pto6:end)/fr,curva_an(pto6:end));
```

```
catch
    J68=0;
end

%% normalized absolute velocity, v2p, v6p.

%% half width w= already calculated from peaks location

%% ratio between velocities S;

%% Cálculo de la base, prominencia, tiempo de parpadeo, half width lft y
% rgt width left y right

base=(curva_ajustada(1)+curva_ajustada(end))/2;
prom=max(curva_ajustada)-base;
% pks=max(curva_ajustada);
tiempo_blink=(t8p-t0p)/fr;
half_width_axis=xnew(curva_ajustada>(max(curva_ajustada)-prom/2));
half_width_left=(xnew(pto4)-half_width_axis(1))*t_ms;
half_width_right=(half_width_axis(end)-xnew(pto4))*t_ms;
half_width=half_width_left+half_width_right;
% width_axis=xnew(curva_ajustada>(max(curva_ajustada)-(prom)));
width_left=(xnew(pto4)-xnew(1))*t_ms;
width_right=(xnew(end)-xnew(pto4))*t_ms;
width=(width_left+width_right);

catch
    t0p=0;t1p=0;t2p=0;t3p=0;t4p=0;t5p=0;t6p=0;t7p=0;t8p=0;
    P1=0; P3=0; P5=0; P7=0; W02=0; W24=0; W46=0; W68=0;
    t1a=0;t2a=0;t3a=0;a1a=0; a2a=0; a3a=0; J02=0; J26=0; J68=0; v2p=0; v6p=0; S=0;
R2_ajuste=0; sigma=0;tau=0;mu=0;
    base=0; prom=0; tiempo_blink=0; half_width_left=0; half_width_right=0; half_width=0;
    width_left=0; width_right=0; width=0; a1a_sn=0; a2a_sn=0; a3a_sn=0; v2p_sn=0;
v6p_sn=0;
end
%% Total results

output=[(t0p-t0p)/fr (t1p-t0p)/fr (t2p-t0p)/fr (t3p-t0p)/fr (t4p-t0p)/fr...
    (t5p-t0p)/fr (t6p-t0p)/fr (t7p-t0p)/fr (t8p-t0p)/fr...
    P1 P3 P5 P7 W02 W24 W46 W68...
    (t1a-t0p)/fr (t2a-t0p)/fr (t3a-t0p)/fr a1a a2a a3a J02 J26 J68 v2p v6p S R2_ajuste
norma sig...
    mu tau...
    base prom tiempo_blink half_width_left half_width_right half_width...
```

```
    width_left width_right width a1a_sn a2a_sn a3a_sn v2p_sn v6p_sn];

end

function [error2m]=MSE_eglike(params,x,pdf_ajustar)
mu=params(1);
sig=params(2);
tau=params(3);
exponential_gauss=exgausspdf(mu,sig,tau, x);
error2m=mean((exponential_gauss-pdf_ajustar).^2);
end

function f=exgausspdf(mu,sig,tau, x)
% given parameter mu, sig and tau, returns density at x for the ex-Gaussian
% mu, sig, tau are scalars
% x is either a scalar, vector or matrix
% f has the same shape as x
% version 2.0 2/10/99
% (c) Yves Lacouture, UniversiteŽ Laval
arg1=(mu./tau)+((sig.*sig)./(2.*tau.*tau))-(x/tau);
arg2=((x-mu)-((sig.*sig)./tau))./sig;
f=(1./tau)*(exp(arg1).*pnf(arg2));
end
```

8.3 Script for Sclera Segmentation Algorithm.

Table 8-11: Main algorithm for sclera detection.

```
clear all
clc
% close all

%% Algoritmo de recorte de esclera para paper.
% A partir del recorte del ojo usando el algoritmo de viola Jones en cada
% uno de los vídeos se procede a analizar la rojez de la esclera.

% ruta completa donde se encuentran las imágenes de ojo recortadas.
cropped_images_address='C:\Users\Marra\OneDrive\Bureau\video_patients\Results_dlib\Anali
sis_color\';

% ruta completa donde se encuentra el código
code_folder='C:\Users\Marra\OneDrive\Bureau\video_patients\cropped_videos\';
```

```
%Inclusión de la ruta donde se encuentra la carpeta de códgio
addpath(code_folder)

% Inclusión de la ruta donde se encuentra la función de la
% media_local_polinomica para quitar el ruido de la imagen.
addpath('C:/Users/Marra/OneDrive/Bureau/codigo_ojo_centrado/')

% Ruta completa de la funcion de hough transform
addpath('C:\Users\Marra\OneDrive\Bureau\JM-SanValentin\output');

% Ruta completa donde se encuentran los archivos output de color
% color_output=...
%     'C:\Users\Marra\OneDrive\Bureau\JM-SanValentin\output\Analysis\';
color_output=...
    'C:\Users\Marra\OneDrive\Bureau\video_patients\Results_dlib\Analisis_color\results';


%creacion carpeta
mkdir(color_output)

% Cambiar carpeta actual a carpeta de vídeos
cd(cropped_images_address)

% Carga de archivos de imagen con extensión jpg.
jpg_path=[cropped_images_address,'*.jpg'];
theFiles=dir(jpg_path);

for image_number=1:size(theFiles,1)
    cd(cropped_images_address)
    close all
    clear i i_o i_g Xsize Ysize

% image_number=21;

% Elección de número de imagen a analizar
% image_number=1;

try
% carga de imagen a analizar
i_o=imread(theFiles(image_number).name);
catch
    continue
end
```

```
% visualizacion imagen original
figure,imshow(i_o,'InitialMagnification', 'fit');

% Extracción de valores de largo y ancho de la imagen recortada por VJ.
Xsize=size(i_o,1);
Ysize=size(i_o,2);

% Imagen de ojo a analizar.
% try
% i=i_o(floor(0.15*Xsize/2)*2:floor(0.85*Xsize/2)*2,1:Ysize,:);
% catch
%     continue
% end
% i=i(1:floor(0.60*Xsize/2)*2,floor(.2*Ysize):Ysize,:);

% coger imagen sin recortar
i=i_o;



% imagen en escala de grises.
i_g=im2gray(i);

% Visualización de la imagen del ojo en escala de grises
figure, imshow(i_g,'InitialMagnification', 'fit');

% Transformar a unidades escalares dobles para poder realizar operaciones
% de gradiente.
i_g=double(i_g);

% eliminación de ruido a partir del ajuste a grado 3 del valor de cada fila
% de pixeles y calculando el error cometido entre el valor de intesidad
% real y el valor de la curva de ajuste. Este mismo proceso se realiza para
% cada columna antes de quedarnos con la diferencia teniendo en cuenta los
% valores del ajuste previos. Este error entre la curva de ajuste y el
% valor real de intesidad será nuestra imagen sin ruido.

media_local=media_local_polinomica(i_g,3);

% Cálculo de la intensidad media de la imagen
total_mean_i_g=mean(i_g(:));

% Nueva imagen en escala de grises sin ruido.
i_g=i_g+total_mean_i_g*ones(size(i_g))-media_local;
```

```
% Visualización de la imagen escala de grises sin ruido.
figure, imshow(uint8(i_g),'InitialMagnification', 'fit');

%Aplicación filtro de la mediana
i_g=medfilt2((i_g));
figure,imshow(uint8(i_g),'InitialMagnification', 'fit');

% Cálculo de la mediana de cada columna
column_median=median(i_g);

%Obtención de la columna de localización de zona central pupila/iris, para
%poder detectar mejor cada una de las escleras, zona esclera exterior y
%zona de esclera nasal.

[pks_ip,locs_ip,width_ip,proms_ip]=...
    findpeaks(-column_median(floor...
    (.25*(size(column_median,2))):ceil(.75*(size(column_median,2)))),...
    floor(.25*(size(column_median,2))):ceil(.75*(size(column_median,2))));

figure,findpeaks(-column_median(floor...
    (.25*(size(column_median,2))):ceil(.75*(size(column_median,2)))),...
    floor(.25*(size(column_median,2))):ceil(.75*(size(column_median,2))),...
    Annotate="extents");
title('Obtention of iris column position reference')
xlabel('Column number')
ylabel('Median intensity value of each column')

% Se selecciona el pico con la mayor prominencia teniendo en cuenta
% únicamente el centro de la imagen, es decir desde la columna 25% del
% ancho total hasta la columna 75%.
[~,idx_ip]=max(proms_ip);
pos_ip=locs_ip(idx_ip);


% Obtención de la curva de mediana del valor de intensidades de pixels de
% cada columna.

% figure,plot(column_median)

% Obtención de las esquinas de la hendidura palpebral. Se hace a partir de
% la curva de mediana de cada columna. Con la función findpeaks, obtenemos
% los picos que corresponden a las zonas más blancas, con mayor intensidad.
% detectamos cada una de las zonas blancas por separado. Para evitar la
% detección de zonas de reflejo o de piel con intensidad luminica excesiva
```

```
% se procede a restringir los picos a localizar con ancho a media altura
% superior al 5% del ancho total de la imagen del ojo aproximadamente. De
% igual forma únicamente se tiene  en cuenta los picos con una prominnencia
% superior a la desviacion estándar de la señalp en estudio, en este caso
% la mediana del valor de intensidad teniendo en cuenta cada uno de los
% píxeles de cada columna de la imagen.

[pks_os,locs_os,width_os,proms_os,wxPk_os]=...
    findpeaks_2(column_median(1:pos_ip),'MinPeakProminence',...
    std(column_median(1:pos_ip)),'MinPeakWidth',...
    0.1*max(size(column_median(1:pos_ip))),'Annotate','extents');

figure,findpeaks(column_median(1:pos_ip),1:pos_ip,'MinPeakProminence',...
    std(column_median(1:pos_ip)),'MinPeakWidth',...
    0.1*max(size(column_median(1:pos_ip))),'Annotate','extents');

title('Obtention of outer sclera column position reference')
xlabel('Column number')
ylabel('Median intensity value of each column')

try
pos_os=locs_os(end)-1;
wxPk_os=wxPk_os(end,:);
b_prom_os=pks_os(end)-proms_os(end);
catch
    continue
end
% Teniendo en cuenta este valor límite y los puntos de los picos que
% pertenecen a la zona de esclera, podemos obtener los puntos extremos de
% la hendidura. Esto se consigue obteniendo la base de los picos de región
% de esclera teniendo en cuenta el valor de intesidad de pixel límite.

column_border_1=find(column_median(1:pos_os)<=b_prom_os);
try
column_border_1=column_border_1(end);
catch
    continue
end
if column_border_1<ceil(wxPk_os(1))-(pos_os-floor(wxPk_os(1)))
    % column_border_1=ceil(wxPk_os(1))-(pos_os-floor(wxPk_os(1)));
    [~,column_border_1]=...
        min(column_median...
        (ceil(wxPk_os(1))-(pos_os-floor(wxPk_os(1))):pos_os));
    column_border_1=column_border_1...
```

```
        +ceil(wxPk_os(1))-(pos_os-floor(wxPk_os(1)));
end

% La obtención del límite de la esclera se realiza con la nueva funcion
% de findpeak s modificada que permite obtener la base del pico a media
% altura
%
% column_border_1=floor(wxPk_os(1));
% column_border_2=ceil(wxPk_ns(2))+pos_ip;



%% Análisis de la parte exterior de la esclera
p_LOW=[];
p_HIGH=[];

edgeimage=zeros(size(i_g));

ini_value=floor((pos_os-column_border_1)/2+column_border_1-1);

c1=i_g(:,pos_os);

[Pk1,LOCS1,W1,~,wxPk1]=findpeaks_2(c1,1:size(c1,1));

figure,findpeaks_2(c1,1:size(c1,1),annotate='extents');
num_os=num2str(pos_os);
title_c1=['Analysis of column number ',num_os];
title(title_c1)
xlabel('Row number')
ylabel('Pixel intesity')
% [~,posw1]=max(W1);
[~,posw1]=max(Pk1);
loc1=LOCS1(posw1);
wxpk1=wxPk1(posw1,:);
try
ini_os=floor(wxpk1(1)-(loc1-wxpk1(1)));
catch
    continue
end
if ini_os<1
    ini_os=1;
end

fin_os=ceil(wxpk1(2)+abs((wxpk1(1)-wxpk1(2))/2));
```

```
if fin_os>size(i_g,1)
    fin_os=size(i_g,1);
end

upper_limit_os=floor(wxpk1(1));
bottom_limit_os=ceil(wxpk1(2));
middle_limit_os=round((wxpk1(1)+wxpk1(2))/2);
try
for p_col=pos_os:-1:ini_value+1
    c=i_g(ini_os:fin_os,p_col);
    % figure,plot(c)
    % figure,findpeaks(c,ini:fin,'MinPeakProminence',30,...
    %     'Annotate','extents','WidthReference','halfprom')
    % figure,findpeaks(c,ini:fin,'Annotate','extents','WidthReference','halfprom')
    % column_num=num2str(p_col);
    % title_col=['Column number being analyzed ', column_num];
    % title(title_col)
    % xlabel('Row number')
    % ylabel('Pixel intensity')
    [~,LOCS,W,~,wxPk]=findpeaks_2(c,1:size(c,1));
    [~,posw]=max(W);
    loc=LOCS(posw);
    wxPk=wxPk(posw,:);

    c_row_low_eyelid=ceil(wxPk(2))+ini_os-1;
    p_low=[c_row_low_eyelid,p_col];
    p_LOW=[p_LOW;p_low];

    c_row_high_eyelid=floor(wxPk(1))+ini_os-1;
    p_high=[c_row_high_eyelid,p_col];
    p_HIGH=[p_HIGH;p_high];
end
catch
    continue
end
% Chequeo tamaño de la curva de limite de parpado en zona exterior de
% esclera

sz_sclera_outer_limit=size(p_LOW);

% ajustar curva de borde esclera exterior con parpado inferior
% a un arco de elipse
try
limit_eyelid_outer_bot_x=p_LOW(:,2);
```

```
limit_eyelid_outer_bot_y=p_LOW(:,1);
figure,plot(p_LOW(:,1),p_LOW(:,2))
catch
    continue
end
x_low_outer_nr=...
    [limit_eyelid_outer_bot_x(end):1/5:limit_eyelid_outer_bot_x(1)]';

% make the fitting curve taking out the outsiders considering only the 95
% % of the sample.

mean_value=mean(limit_eyelid_outer_bot_y);
std_value=std(limit_eyelid_outer_bot_y);

idx1=find(limit_eyelid_outer_bot_y>mean_value+2*std_value);
idx2=find(limit_eyelid_outer_bot_y<mean_value-2*std_value);
idx=[idx1;idx2];

limit_eyelid_outer_bot_y(idx)=[];
limit_eyelid_outer_bot_x(idx)=[];
try
[fitresult_outer_bot, gof] = fit2degree...
    (limit_eyelid_outer_bot_x, limit_eyelid_outer_bot_y,...
    0,Inf,Inf,-Inf,0,-Inf,upper_limit_os,bottom_limit_os);
catch
    continue
end
if 2*x_low_outer_nr(end)*fitresult_outer_bot.p1+fitresult_outer_bot.p2<0
    display('new fitting needed')
    ft = fittype( 'poly1' );
    opts = fitoptions( 'Method', 'LinearLeastSquares' );
    [fitresult_outer_bot, gof] =...
        fit( limit_eyelid_outer_bot_x, limit_eyelid_outer_bot_y, ft, opts );
end

y_test=fitresult_outer_bot(x_low_outer_nr);
x_low_outer_test=round(x_low_outer_nr);
y_low_outer_test=round(y_test);

% ajustar curva de borde esclera exterior con parpado inferior a un arco
% de elipse

limit_eyelid_outer_top_x=p_HIGH(:,2);
limit_eyelid_outer_top_y=p_HIGH(:,1);
```

```
figure,plot(p_HIGH(:,1),p_HIGH(:,2))
try
[fitresult_outer_top, gof] = fit2degree...
    (limit_eyelid_outer_top_x, limit_eyelid_outer_top_y,...
    Inf,0,Inf,0,-Inf,-Inf,upper_limit_os,bottom_limit_os);
catch
    continue
end
x_high_outer_test=...
    [limit_eyelid_outer_top_x(end):1/5:limit_eyelid_outer_top_x(1)]';
y_test=fitresult_outer_top(x_high_outer_test);
x_high_outer_test=round(x_high_outer_test);
y_high_outer_test=round(y_test);

% cálculo de la curva límite de la esclera exterior considerando el borde
% de la hendirua palpebral y la zona limite de la curva de parpado inferior
% y superior.

% punto parpado superior
y1=y_high_outer_test(1);
x1=x_high_outer_test(1);
p1=[x1 y1];

%punto esquina hendidura.
x2=floor((pos_os-column_border_1)*.2+column_border_1);
y2=round((y_high_outer_test(1)+y_low_outer_test(1))/2);
p2=[x2 y2];

%punto parpado inferior
y3=y_low_outer_test(1);
x3=x_low_outer_test(1);
p3=[x3 y3];

[center,radius]=circleThroughThreePoints(p1, p2, p3);

if center(1)<x1
    x2=floor((pos_os-column_border_1)*.4+column_border_1);
    p2=[x2 y2];
    [center,radius]=circleThroughThreePoints(p1, p2, p3);
end

% Generate points along the circle
theta = linspace(0, 2*pi, 1000);
x_circle = center(1) + radius * cos(theta);
```

```
y_circle = center(2) + radius * sin(theta);

x_arc_idx=find(x_circle<=x1);
x_arc=x_circle(x_arc_idx);
y_arc=y_circle(x_arc_idx);

figure,plot(x_arc,y_arc)

x_arc=round(x_arc);
y_arc=round(y_arc);
try
if center(1)>x1
    for cnt00=1:max(size(x_arc))
        edgeimage(y_arc(cnt00),x_arc(cnt00))=1;
    end
end
catch
    continue
end
% edgefor_outer(y1,x1)=1;
% edgefor_outer(y2,x2)=1;
% edgefor_outer(y3,x3)=1;
%
% figure,imshow(edgefor_outer)

%% Análisis de la parte nasal de la esclera
% encontrar la zona de esclera nasal.

if ini_os>floor(size(i_g,1)*.15)
    ini_os=ini_os-floor(size(i_g,1)*.15);
else
    ini_os=1;
end

if fin_os+floor(size(i_g,1)*.15)<size(i_g,1)
    fin_os=fin_os+floor(size(i_g,1)*.15);
else
    fin_os=size(i_g,1);
end

column_median_2=median(i_g(ini_os:fin_os,:));

[pks_ns,locs_ns,width_ns,proms_ns,wxPk_ns]=...
    findpeaks_2(column_median_2(pos_ip:end),'MinPeakProminence',...
```

```
    std(column_median_2(pos_ip:end)),'MinPeakWidth',...
    0.05*max(size(column_median_2(pos_ip:end))),'Annotate','extents');

figure,findpeaks_2(column_median_2(pos_ip:end),pos_ip:size(i_g,2),...
    'MinPeakProminence',std(column_median_2(pos_ip:end)),'MinPeakWidth',...
    0.05*max(size(column_median_2(pos_ip:end))),'Annotate','extents');
title('Obtention of nasal sclera column position reference')
xlabel('Column number')
ylabel('Median intensity value of each column')


[~,idx_pos_ns]=max(pks_ns);
pos_ns=locs_ns(idx_pos_ns);
pos_ns=pos_ns+pos_ip-1;
wxPk_ns=wxPk_ns(idx_pos_ns,:);
b_prom_ns=pks_ns(idx_pos_ns)-proms_ns(idx_pos_ns);

column_border_2=find(column_median(pos_ns:end)<=b_prom_ns);

try
if min(size(column_border_2))>0
    column_border_2=pos_ns+column_border_2(1)-1;

    if column_border_2>floor(pos_ip+wxPk_ns(2))+abs((pos_ns)...
            -ceil(pos_ip+wxPk_ns(2)))
        column_border_2=floor(pos_ip+wxPk_ns(2))+abs((pos_ns)...
        -ceil(pos_ip+wxPk_ns(2)));
    end

else
    column_border_2=floor(pos_ip+wxPk_ns(2))+abs((pos_ns)...
        -ceil(pos_ip+wxPk_ns(2)));
end
catch
    continue
end
    if column_border_2>size(i_g,2)
        column_border_2=size(i_g,2);
    end

final_value=ceil(+column_border_2-(column_border_2-pos_ns)/2);

c1=i_g(:,pos_ns);
```

```
figure,findpeaks(c1,'Annotate','extents')
num_ns=num2str(pos_ns);
title_c1=['Analysis of column number ',num_ns];
title(title_c1)
xlabel('Row number')
ylabel('Pixel intesity')


[~,LOCS1,W1,~,wxPk1]=findpeaks_2(c1,1:size(c1,1));
[~,posw1]=max(W1);
loc1=LOCS1(posw1);
wxpk1=wxPk1(posw1,:);
try
ini_ns=floor(wxpk1(1)-(loc1-wxpk1(1)));
catch
    continue
end
if ini_ns<1
    ini_ns=1;
end

fin_ns=ceil(wxpk1(2)+abs((wxpk1(1)-wxpk1(2))/2));
if fin_ns>size(i_g,1)
    fin_ns=size(i_g,1);
end

upper_limit_ns=floor(wxpk1(1));
mid_limit_ns=round((wxpk1(1)+wxpk1(2))/2);
bottom_limit_ns=ceil(wxpk1(2));
try
for p_col=pos_ns:final_value
    c=i_g(ini_ns:fin_ns,p_col);
    [~,LOCS,W,~,wxPk]=findpeaks_2(c,1:size(c,1));
    [~,posw]=max(W);
    loc=LOCS(posw);
    wxPk=wxPk(posw,:);
    c_row_low_eyelid=ceil(wxPk(2))+ini_ns-1;
    p_low=[c_row_low_eyelid,p_col];
    p_LOW=[p_LOW;p_low];
    c_row_high_eyelid=floor(wxPk(1))+ini_ns-1;
    p_high=[c_row_high_eyelid,p_col];
    p_HIGH=[p_HIGH;p_high];
end
catch
```

```
    continue
end

%% ajustar curva de borde parpado a un arco de elipse
% curva de ajuste para zona nasal de esclera, párpado inferior
limit_eyelid_nasal_bot_x=p_LOW(sz_sclera_outer_limit+1:end,2);
limit_eyelid_nasal_bot_y=p_LOW(sz_sclera_outer_limit+1:end,1);
figure,plot(limit_eyelid_nasal_bot_x,limit_eyelid_nasal_bot_y)

try
[fitresult_nasal_bot, gof] = fit2degree...
    (limit_eyelid_nasal_bot_x, limit_eyelid_nasal_bot_y,...
    0,0,Inf,-Inf,-Inf,-Inf,mid_limit_ns,bottom_limit_ns);
catch
    continue
end


x_low_nasal_test=...
    [limit_eyelid_nasal_bot_x(1):1/5:limit_eyelid_nasal_bot_x(end)]';
y_test=fitresult_nasal_bot(x_low_nasal_test);
y_low_nasal_test=round(y_test);
x_low_nasal_test=round(x_low_nasal_test);

% x_low_nasal=limit_eyelid_nasal_bot_x;
% y=fitresult_nasal_bot(x_low_nasal);
% y_low_nasal=round(y);
% x_low_nasal=round(x_low_nasal);

% curva de ajuste para zona nasal de esclera, párpado superior.
limit_eyelid_nasal_top_x=p_HIGH(sz_sclera_outer_limit+1:end,2);
limit_eyelid_nasal_top_y=p_HIGH(sz_sclera_outer_limit+1:end,1);
figure,plot(limit_eyelid_nasal_top_x,limit_eyelid_nasal_top_y)

try
    [fitresult_nasal_top, gof_nasal_top] = fit2degree...
        (limit_eyelid_nasal_top_x, limit_eyelid_nasal_top_y,...
        Inf,Inf,Inf,0,0,-Inf,upper_limit_ns,mid_limit_ns);
catch
    [fitresult_nasal_top, gof_nasal_top] = fit2degree...
        (limit_eyelid_nasal_top_x, limit_eyelid_nasal_top_y,...
        Inf,Inf,Inf,0,0,-Inf,[],[]);
end
```

```
x_high_nasal_test=...
    [limit_eyelid_nasal_top_x(1):1/5:limit_eyelid_nasal_top_x(end)]';
y_test=fitresult_nasal_top(x_high_nasal_test);
y_high_nasal_test=round(y_test);
x_high_nasal_test=round(x_high_nasal_test);

% x_high_nasal=limit_eyelid_nasal_top_x;
% y=fitresult_nasal_top(x_high_nasal);
% y_high_nasal=round(y);
% x_high_nasal=round(x_high_nasal);

% cálculo de la curva límite de la esclera exterior considerando el borde
% de la hendirua palpebral y la zona limite de la curva de parpado inferior
% y superior.

% punto parpado superior
y1=y_high_nasal_test(end);
x1=x_high_nasal_test(end);
p1=[x1 y1];

%punto esquina hendidura.
x2=floor(column_border_2-abs(pos_ns-column_border_2)*.2);
y2=round((y_high_nasal_test(end)+y_low_nasal_test(end))/2);
p2=[x2 y2];

%punto parpado inferior
y3=y_low_nasal_test(end);
x3=x_low_nasal_test(end);
p3=[x3 y3];

[center,radius]=circleThroughThreePoints(p1, p2, p3);

if center(1)>x1
    x2=floor(column_border_2-abs(pos_ns-column_border_2)*.4);
    p2=[x2 y2];
    [center,radius]=circleThroughThreePoints(p1, p2, p3);
end

% %radio
% radius=(abs(y3-y1))/2;
% %centro
% center=[y2,x2+radius];

% Generate points along the circle
```

```
theta = linspace(0, 2*pi, 1000);
x_circle = center(1) + radius * cos(theta);
y_circle = center(2) + radius * sin(theta);

x_arc_idx=find(x_circle>=x1);
x_arc=x_circle(x_arc_idx);
y_arc=y_circle(x_arc_idx);

figure,plot(x_arc,y_arc)

x_arc=round(x_arc);
y_arc=round(y_arc);
try
if center(1)<x1
    for cnt00=1:max(size(x_arc))
        edgeimage(y_arc(cnt00),x_arc(cnt00))=1;
    end
end
catch
    continue
end
% variables x e y de las curvas ajustadas de los limites superior e
% inferior

x_low=[x_low_outer_test;x_low_nasal_test];
y_low=[y_low_outer_test;y_low_nasal_test];

x_high=[x_high_outer_test;x_high_nasal_test];
y_high=[y_high_outer_test;y_high_nasal_test];

% representación de límites en imagen blanco y negro
try
for cnt0=1:max(size(x_high))
    edgeimage(y_high(cnt0),x_high(cnt0))=1;
end

for cnt0=1:max(size(x_low))
    edgeimage(y_low(cnt0),x_low(cnt0))=1;
end
catch
    continue
end
figure,imshow(edgeimage)
```

```
edgeimage2=zeros(size(i_g));
for cnt0=1:max(size(p_HIGH))
    edgeimage2(p_HIGH(cnt0,1),p_HIGH(cnt0,2))=1;
end
for cnt0=1:max(size(p_LOW))
    edgeimage2(p_LOW(cnt0,1),p_LOW(cnt0,2))=1;
end

figure,imshow(edgeimage2)

%limite esclera parpado inferior zona exterior.
x_low_outer=x_low_outer_test;
y_low_outer=y_low_outer_test;

%limite esclera parpado inferior zona nasal.
x_low_nasal=x_low_nasal_test;
y_low_nasal=y_low_nasal_test;

%limite esclera parpado superior zona exterior.
x_high_outer=x_high_outer_test;
y_high_outer=y_high_outer_test;

%limite esclera parpado superior zona nasal.
x_high_nasal=x_high_nasal_test;
y_high_nasal=y_high_nasal_test;

%% Hasta aquí limite parpadeo superior e inferior.
% Obtención de la circunferencia de borde del iris.
% Teniendo en cuenta que el diámetro ronda el valor de 2 quintos el valor
% de la longitud de hendidura palpebral, se estima que el diámetro mínimo
% será igual a 1/3 de la longitud hp y el diámetro máximo .5 del valor de
% Lhp.
Lhp=column_border_2-column_border_1;

% Obtención de la distancia entre las dos regiones de la esclera.
column_nasal_sclera=pos_ns;
column_outer_sclera=pos_os;

% Calculo de la columna media entre la columna referencia de la region
% nasal y exterior de la esclera.
column_mean_iris_region=round((column_nasal_sclera+column_outer_sclera)/2);

%% Análisis de filas del iris para encontrar la columna límite de la parte
% exterior de la esclera con el iris.
```

```
column_limit_outer=[];
row_limit_outer=[];
STOP_VALUES=[];

for cnt=upper_limit_os:bottom_limit_os
    current_line_outer=...
        i_g(cnt,column_outer_sclera:column_mean_iris_region);

    % figure,plot(...
    %     column_outer_sclera:column_mean_iris_region,current_line_outer)
    % row_number=num2str(cnt);
    % hold on
    % figure_name=[...
    %     ['Comparing the pixel variation to the median value' ...
    %     ' of row number '],row_number];
    % title(figure_name);
    % xlabel('Column number')
    % ylabel('Pixel intesity')
    find_stop_value=median(current_line_outer);
    min_value=min(current_line_outer);
    %
plot(column_outer_sclera:column_mean_iris_region,find_stop_value*ones(size(current_line_
outer)))
    % hold off
    stop_value=min(find(current_line_outer<find_stop_value));
    [~,min_grad_line_outer]=min(gradient(current_line_outer(1:stop_value)));
    % figure,plot(column_outer_sclera:column_outer_sclera+stop_value-
1,gradient(current_line_outer(1:stop_value)))
    % figure_title=['Obtention of column limit of sclera region' ...
    %     ' of row number ',row_number];
    % title(figure_title)
    % xlabel('Column number')
    % ylabel('Pixel intesity')

    column_limit_outer=[column_limit_outer;...
        column_outer_sclera+min_grad_line_outer-1];
    row_limit_outer=[row_limit_outer;cnt];
    STOP_VALUES=[STOP_VALUES min_value];
end

[~,last_row_iris_outer,width_last,~,wxPk_o]=...
    findpeaks_2(gradient(STOP_VALUES),...
    'MinPeakProminence',3*std(gradient(STOP_VALUES)),'Annotate','extents');
figure,findpeaks(gradient(STOP_VALUES),upper_limit_os:bottom_limit_os,...
```

```
    'MinPeakProminence',3*std(gradient(STOP_VALUES)),'Annotate','extents');
if size(last_row_iris_outer,2)>0
    last_row_iris_outer=round(wxPk_o(1));

    if last_row_iris_outer<max(size(row_limit_outer))
        row_limit_outer=row_limit_outer(1:last_row_iris_outer);
        column_limit_outer=column_limit_outer(1:last_row_iris_outer);
    end
end

%% Análisis de filas del iris para encontrar la columna límite de la parte
% nasal de la esclera con el iris.

column_limit_nasal=[];
row_limit_nasal=[];
STOP_VALUES=[];
for cnt=upper_limit_ns:bottom_limit_ns
    current_line_nasal=...
        i_g(cnt,column_mean_iris_region:column_nasal_sclera);
    find_stop_value=median(current_line_nasal);
    stop_value=max(find(current_line_nasal<find_stop_value));
    min_value=min(current_line_nasal);
    [~,max_grad_line_nasal]=...
        max(gradient(current_line_nasal(stop_value:end)));
    column_limit_nasal=[column_limit_nasal;...
        column_mean_iris_region+stop_value+max_grad_line_nasal-1];
    row_limit_nasal=[row_limit_nasal;cnt];
    %este valor sirve para detectar cuando no hay iris
    STOP_VALUES=[STOP_VALUES find_stop_value];
end

[~,last_row_iris_nasal,width_last,~,wxPk_n]=...
    findpeaks_2(gradient(STOP_VALUES),...
    'MinPeakProminence',3*std(gradient(STOP_VALUES)),'Annotate','extents');

figure,plot(upper_limit_ns:bottom_limit_ns,STOP_VALUES)

figure,findpeaks(gradient(STOP_VALUES),upper_limit_ns:bottom_limit_ns,...
    'MinPeakProminence',3*std(gradient(STOP_VALUES)),'Annotate','extents');

if size(last_row_iris_nasal,2)>0
    last_row_iris_nasal=wxPk_n(1);

    if last_row_iris_nasal<max(size(row_limit_nasal))
```

```
        row_limit_nasal=row_limit_nasal(1:last_row_iris_nasal);
        column_limit_nasal=column_limit_nasal(1:last_row_iris_nasal);
    end
end

%% ajustar curva de borde de iris a un arco de circunferencia.
% En esta parte se estudia el ajuste de la curva limite del iris con la
% parte exterior de la esclera.

limit_outer_sclera_iris=[column_limit_outer row_limit_outer];
figure,plot(row_limit_outer,column_limit_outer)
try
[fitresult_outer, gof] = fit2degree(row_limit_outer, column_limit_outer,...
    Inf,Inf,Inf,-Inf,-Inf,-Inf,[],[]);
catch
    continue
end

x_outer=[row_limit_outer(1):1/5:row_limit_outer(end)]';
y=fitresult_outer(x_outer);
y_outer=round(y);
x_outer=round(x_outer);

%% En esta parte se estudia el ajuste de la curva limite del iris con la
% parte nasal de la esclera.
try
[fitresult_inner, gof2] = ...
    fit2degree(row_limit_nasal, column_limit_nasal,...
    Inf,Inf,Inf,-Inf,-Inf,-Inf,[],[]);
catch
    continue
end
x_nasal=[row_limit_nasal(1):1/5:row_limit_nasal(end)]';
y2=fitresult_inner(x_nasal);
y_nasal=round(y2);
x_nasal=round(x_nasal);

% Hacer que los limites de parpado inferior sean la restriccion para
% encontrar los pixels de la esclera y no los puntos obtenidos por el
% limite del iris con la esclera. Tanto para la zona exterior conmo la
% esclera nasal

if x_outer(end)>y_low_outer(end)
    idx=find(x_outer<=y_low_outer(end));
```

```
    x_outer=x_outer(idx);
    y_outer=y_outer(idx);
end

if x_nasal(end)>y_low_nasal(1)
    idx=find(x_nasal<=y_low_nasal(1));
    x_nasal=x_nasal(idx);
    y_nasal=y_nasal(idx);
end

for cnt2=1:size(y_nasal,1)
    edgeimage(x_nasal(cnt2),y_nasal(cnt2))=1;
end

for cnt3=1:size(y_outer,1)
    edgeimage(x_outer(cnt3),y_outer(cnt3))=1;
end

%% asignación de variable únicamnete para prueba
%distancia entre borde de parpado y borde de iris

% coordenadas zona parpado superior e inferior de la esclera exterior.
columna_limite_exterior=x_high_outer(1);
edgeimage(y_high_outer(1):y_low_outer(1),columna_limite_exterior)=1;

% coordenadas zona parpado superior e inferior de la esclera nasal.
columna_limite_nasal=x_high_nasal(end);
edgeimage(y_high_nasal(end):y_low_nasal(end),columna_limite_nasal)=1;

%zona parpado superior con borde de iris exterior
if (y_high_outer(end)>x_outer(1))
    edgeimage(y_high_outer(end),x_high_outer(end):y_outer(1))=1;
    edgeimage(x_outer(1):y_high_outer(end),y_outer(1))=1;
else
    edgeimage(x_outer(1),x_high_outer(end):y_outer(1))=1;
    edgeimage(y_high_outer(end):x_outer(1),x_high_outer(end))=1;
end

%zona parpado inferior con borde de iris exterior
if (y_low_outer(end)<x_outer(end))
    edgeimage(y_low_outer(end),x_low_outer(end):y_outer(end))=1;
    edgeimage(y_low_outer(end):x_outer(end),y_outer(end))=1;
else
    edgeimage(y_low_outer(end),x_low_outer(end):y_outer(end))=1;
```

```
    edgeimage(x_outer(end):y_low_outer(end),y_outer(end))=1;
end

%zona parpado superior con borde de iris nasal
try
if (y_high_nasal(1)>x_nasal(1))
    edgeimage(y_high_nasal(1),y_nasal(1):x_high_nasal(1))=1;
    edgeimage(y_high_nasal(1):x_nasal(1),x_high_nasal(1) )=1;
else
    edgeimage(x_nasal(1),y_nasal(1):x_high_nasal(1))=1;
    edgeimage(x_nasal(1):y_high_nasal(1),x_high_nasal(1))=1;
end
catch
    continue
end


%zona parpado inferior con borde de iris nasal
try
if (y_low_nasal(1)<x_nasal(end))
    edgeimage(x_nasal(end),y_nasal(end):x_low_nasal(1))=1;
    edgeimage(y_low_nasal(1):x_nasal(end),x_low_nasal(1))=1;
else
    edgeimage(y_low_nasal(1),y_nasal(end):x_low_nasal(1))=1;
    edgeimage(x_nasal(end):y_low_nasal(1),y_nasal(end))=1;
end
catch
    continue
end

figure,imshow(edgeimage,'InitialMagnification', 'fit');

mask=imfill(edgeimage);
try
figure,imshow(mask,'InitialMagnification', 'fit');
figure,imshow(uint8(i_g.*mask),'InitialMagnification', 'fit');
figure,imshow(uint8(i_g),'InitialMagnification', 'fit');
catch
    continue
end
cd(color_output)
esclera=uint8(i_g.*mask);
% imwrite(i,theFiles(image_number).name)
imwrite(mask, ['mask_',theFiles(image_number).name])
```

```
imwrite(esclera, ['mask_es',theFiles(image_number).name])
% figure,imagesc(edgeimage)
% colormap('gray');

% Eliminación de bordes en la máscara.

%% Estudio del sistema de color
% mascara temporal
mask_temporal=zeros(size(mask));
mask_temporal(:,1:pos_ip)=mask(:,1:pos_ip);

%mascara nasal
mask_nasal=zeros(size(mask));
mask_nasal(:,pos_ip+1:end)=mask(:,pos_ip+1:end);

%Obtención de todos los puntos de la imagen original correspondientes a la
%esclera.
% imagen original del ojo es i, la máscara es mask
% pixeles de la esclera
i_rgb=i;
i_lab=rgb2lab(i_rgb);
i_hsv=rgb2hsv(i_rgb);
mask_1d=find(mask);
mask_nasal_1d=find(mask_nasal);
mask_temporal_1d=find(mask_temporal);
S_name_rgb=['S_',theFiles(image_number).name(1:end-4),'_rgb'];
traza_name_rgb=['traza_',theFiles(image_number).name(1:end-4),'_rgb'];
S_name_lab=['S_',theFiles(image_number).name(1:end-4),'_lab'];
traza_name_lab=['traza_',theFiles(image_number).name(1:end-4),'_lab'];
S_name_hsv=['S_',theFiles(image_number).name(1:end-4),'_hsv'];
traza_name_hsv=['traza_',theFiles(image_number).name(1:end-4),'_hsv'];
pdf_name_nasal_rgb=['pdf_nasal',theFiles(image_number).name(1:end-4),'_rgb'];
pdf_name_nasal_lab=['pdf_nasal',theFiles(image_number).name(1:end-4),'_lab'];
pdf_name_nasal_hsv=['pdf_nasal',theFiles(image_number).name(1:end-4),'_hsv'];
pdf_name_temporal_rgb=['pdf_temporal',theFiles(image_number).name(1:end-4),'_rgb'];
pdf_name_temporal_lab=['pdf_temporal',theFiles(image_number).name(1:end-4),'_lab'];
pdf_name_temporal_hsv=['pdf_temporal',theFiles(image_number).name(1:end-4),'_hsv'];
channels_name_nasal_rgb=['channels_nasal',theFiles(image_number).name(1:end-4),'_rgb'];
channels_name_nasal_lab=['channels_nasal',theFiles(image_number).name(1:end-4),'_lab'];
channels_name_nasal_hsv=['channels_nasal',theFiles(image_number).name(1:end-4),'_hsv'];
channels_name_temporal_rgb=['channels_temporal',theFiles(image_number).name(1:end-
4),'_rgb'];
channels_name_temporal_lab=['channels_temporal',theFiles(image_number).name(1:end-
4),'_lab'];
```

```
channels_name_temporal_hsv=['channels_temporal',theFiles(image_number).name(1:end-
4),'_hsv'];
eval(['[',S_name_rgb ',' traza_name_rgb,',' pdf_name_nasal_rgb,','
pdf_name_temporal_rgb,',' channels_name_nasal_rgb,','
channels_name_temporal_rgb,']=var_color(i_rgb,mask_1d,mask_nasal_1d,mask_temporal_1d);']
);
eval(['[',S_name_lab ',' traza_name_lab,',' pdf_name_nasal_lab,','
pdf_name_temporal_lab,',' channels_name_nasal_lab,','
channels_name_temporal_lab,']=var_color(i_lab,mask_1d,mask_nasal_1d,mask_temporal_1d);']
);
eval(['[',S_name_hsv ',' traza_name_hsv,',' pdf_name_nasal_hsv,','
pdf_name_temporal_hsv,',' channels_name_nasal_hsv,','
channels_name_temporal_hsv,']=var_color(i_hsv,mask_1d,mask_nasal_1d,mask_temporal_1d);']
);
dotmat_name=[theFiles(image_number).name(1:end-4),'.mat'];
save(dotmat_name,eval(['"',S_name_rgb,'"']),eval(['"',traza_name_rgb,'"']),...
    eval(['"',S_name_lab,'"']),eval(['"',traza_name_lab,'"']),...
    eval(['"',S_name_hsv,'"']),eval(['"',traza_name_hsv,'"']),...
    eval(['"',pdf_name_nasal_rgb,'"']),eval(['"',pdf_name_nasal_lab,'"']),...
    eval(['"',pdf_name_nasal_hsv,'"']),...
    eval(['"',pdf_name_temporal_rgb,'"']),eval(['"',pdf_name_temporal_lab,'"']),...
    eval(['"',pdf_name_temporal_hsv,'"']),...
    eval(['"',channels_name_nasal_rgb,'"']),...
    eval(['"',channels_name_nasal_lab,'"']),...
    eval(['"',channels_name_nasal_hsv,'"']),...
    eval(['"',channels_name_temporal_rgb,'"']),...
    eval(['"',channels_name_temporal_lab,'"']),...
    eval(['"',channels_name_temporal_hsv,'"'])...
    )



end
```

Table 8-12: function for the calculation of the trace of the color matrix.

```
%% funcion cálculo de la traza de la matriz de color
function
[S_mean,traza,pdf_nasal,pdf_temporal,channels_nasal,channels_temporal]=var_color(fotogra
m,mask,mask_nasal,mask_temporal)
%% Estudio del sistema de color
```

```
%Obtención de todos los puntos de la imagen original correspondientes a la
%esclera.
% imagen original del ojo es i con la mascara ya proporcionada
% pixeles de la esclera
i_sclera_rgb=fotogram;
rows=size(i,1);
columns=size(i,2);

% figure,imshow(i_sclera_rgb)
%separate channel
canal_sclera_r=i_sclera_rgb(:,:,1);
canal_sclera_g=i_sclera_rgb(:,:,2);
canal_sclera_b=i_sclera_rgb(:,:,3);
% Transform matrix channel to 1-D
canal_sclera_r_v=canal_sclera_r(:);
canal_sclera_g_v=canal_sclera_g(:);
canal_sclera_b_v=canal_sclera_b(:);
% erase all data that is not sclera
% idx_only_sclera=find(canal_sclera_r_v>0);
canal_r_only_sclera=canal_sclera_r_v(mask);
canal_g_only_sclera=canal_sclera_g_v(mask);
canal_b_only_sclera=canal_sclera_b_v(mask);

%only nasal
canal_r_only_sclera_nasal=canal_sclera_r_v(mask_nasal);
canal_g_only_sclera_nasal=canal_sclera_g_v(mask_nasal);
canal_b_only_sclera_nasal=canal_sclera_b_v(mask_nasal);

%only temporal
canal_r_only_sclera_temporal=canal_sclera_r_v(mask_temporal);
canal_g_only_sclera_temporal=canal_sclera_g_v(mask_temporal);
canal_b_only_sclera_temporal=canal_sclera_b_v(mask_temporal);

%plot data in a 3d coordinate system
figure,scatter3(canal_r_only_sclera,canal_g_only_sclera,canal_b_only_sclera,...
    'filled','o','ColorVariable',canal_r_only_sclera)
xlabel('Channel 1')
ylabel('Channel 2')
zlabel('Channel 3')
ax=gca;
ax.FontSize=24;

% Preparación de la matriz con los datos de la distribución
```

```
data_rgb=[canal_r_only_sclera,canal_g_only_sclera,canal_b_only_sclera];
data_rgb=double(data_rgb);

% Calculo de la matriz de covarianzas
S=zeros(3,3,size(canal_b_only_sclera,1));
for x=1:3
    for y=1:3
        for p=1:size(canal_r_only_sclera,1)
            S(x,y,p)=(data_rgb(p,x)-mean(data_rgb(:,x)))*(data_rgb(p,y)-
mean(data_rgb(:,y)));
        end
    end
end

S_sum=sum(S,3);
S_mean=mean(S,3);
S_std=std(S,0,3);
traza=sum(diag(S_mean));

%transform to double the data
% display(canal_r_only_sclera)
canal_r_only_sclera_nasal_d=double(canal_r_only_sclera_nasal);
canal_g_only_sclera_nasal_d=double(canal_g_only_sclera_nasal);
canal_b_only_sclera_nasal_d=double(canal_b_only_sclera_nasal);

canal_r_only_sclera_temporal_d=double(canal_r_only_sclera_temporal);
canal_g_only_sclera_temporal_d=double(canal_g_only_sclera_temporal);
canal_b_only_sclera_temporal_d=double(canal_b_only_sclera_temporal);

% cambiar variables de nombre
[pdf_y1_n,pdf_x1_n]=ksdensity(canal_r_only_sclera_nasal_d);
[pdf_y2_n,pdf_x2_n]=ksdensity(canal_g_only_sclera_nasal_d);
[pdf_y3_n,pdf_x3_n]=ksdensity(canal_b_only_sclera_nasal_d);

[pdf_y1_t,pdf_x1_t]=ksdensity(canal_r_only_sclera_temporal_d);
[pdf_y2_t,pdf_x2_t]=ksdensity(canal_g_only_sclera_temporal_d);
[pdf_y3_t,pdf_x3_t]=ksdensity(canal_b_only_sclera_temporal_d);

pdf_nasal=[pdf_x1_n',pdf_y1_n',pdf_x2_n',pdf_y2_n',pdf_x3_n',pdf_y3_n'];
pdf_temporal=[pdf_x1_t',pdf_y1_t',pdf_x2_t',pdf_y2_t',pdf_x3_t',pdf_y3_t'];

channels_nasal=[canal_r_only_sclera_nasal_d canal_g_only_sclera_nasal_d
canal_b_only_sclera_nasal_d];
```

```
channels_temporal=[canal_r_only_sclera_temporal_d canal_g_only_sclera_temporal_d
canal_b_only_sclera_temporal_d];
end
```

Table 8-13: media_local_polinomica function to perform luminance equalization.

```
function [ media_local ] = media_local_polinomica( imagen,n );
[filas columnas]=size(imagen);
ejec=[1:columnas];
ejef=[1:filas];

imagenh=zeros(size(imagen));
imagenhv=zeros(size(imagen));

for f=1:filas
    linea=imagen(f,:);
    ajuste=polyfit(ejec,linea,n);
    suma=zeros(size(ejec));
    for nn=1:n+1
        suma=suma+ajuste(nn)*(ejec.^(n+1-nn));
    end
    imagenh(f,:)=suma;
end

for c=1:columnas
    linea=imagenh(:,c);
    ajuste=polyfit(ejef',linea,n);
    suma=zeros(size(ejef'));
    for nn=1:n+1
        suma=suma+ajuste(nn)*(ejef'.^(n+1-nn));
    end
    imagenhv(:,c)=suma;
end

media_local=imagenhv;


end
```

Table 8-14: Modified fitting function to polynomial 2nd degree to consider bottom and upper limits.

```
function [fitresult, gof] = fit2degree(row_limit_outer,...
    column_limit_outer,pu1,pu2,pu3,pl1,pl2,pl3,upper_limit,bottom_limit)
%CREATEFIT(ROW_LIMIT_OUTER,COLUMN_LIMIT_OUTER)
%  Create a fit.
%
%  Data for 'untitled fit 1' fit:
%      X Input: row_limit
%      Y Output: column_limit
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.

%  Auto-generated by MATLAB on 12-Sep-2024 15:29:35


%% Fit: 'untitled fit 1'.
[xData, yData] = prepareCurveData( row_limit_outer, column_limit_outer );

% Set up fittype and options.
ft = fittype( 'poly2' );
if min(size(upper_limit))>0
    excludedPoints = (yData < upper_limit | yData > bottom_limit);
else
    excludedPoints =[];
end
opts = fitoptions( 'Method', 'LinearLeastSquares' );
opts.Upper = [pu1,pu2,pu3];
opts.Lower = [pl1,pl2,pl3];
opts.Normalize = 'off';
opts.Robust = 'LAR';
opts.Exclude = excludedPoints;

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'untitled fit 1' );
h = plot( fitresult, xData, yData );
legend( h, 'column_limit vs. row_limit', 'untitled fit 1', 'Location', 'NorthEast',
'Interpreter', 'none' );
```

```
% Label axes
xlabel( 'row_limit', 'Interpreter', 'none' );
ylabel( 'column_limit', 'Interpreter', 'none' );
grid on
```

Table 8-15: Self modified findpeaks function for the detection of half width initial and ending points.

```
function [Ypk,Xpk,Wpk,Ppk,wxPk] = findpeaks_2(Yin,varargin)
%FINDPEAKS Find local peaks in data
%   PKS = FINDPEAKS(Y) finds local peaks in the data vector Y. A local peak
%   is defined as a data sample which is either larger than the two
%   neighboring samples or is equal to Inf.
%
%   [PKS,LOCS] = FINDPEAKS(Y) also returns the indices LOCS at which the
%   peaks occur.
%
%   [PKS,LOCS] = FINDPEAKS(Y,X) specifies X as the location vector of data
%   vector Y. X must be a strictly increasing vector of the same length as
%   Y. LOCS returns the corresponding value of X for each peak detected.
%   If X is omitted, then X will correspond to the indices of Y.
%
%   [PKS,LOCS] = FINDPEAKS(Y,Fs) specifies the sample rate, Fs, as a
%   positive scalar, where the first sample instant of Y corresponds to a
%   time of zero.
%
%   [...] = FINDPEAKS(...,'MinPeakHeight',MPH) finds only those peaks that
%   are greater than the minimum peak height, MPH. MPH is a real-valued
%   scalar. The default value of MPH is -Inf.
%
%   [...] = FINDPEAKS(...,'MinPeakProminence',MPP) finds peaks guaranteed
%   to have a vertical drop of more than MPP from the peak on both sides
%   without encountering either the end of the signal or a larger
%   intervening peak. The default value of MPP is zero.
%
%   [...] = FINDPEAKS(...,'Threshold',TH) finds peaks that are at least
%   greater than both adjacent samples by the threshold, TH. TH is a
%   real-valued scalar greater than or equal to zero. The default value of
%   TH is zero.
%
```

```
%   FINDPEAKS(...,'WidthReference',WR) estimates the width of the peak as
%   the distance between the points where the signal intercepts a
%   horizontal reference line. The points are found by linear
%   interpolation. The height of the line is selected using the criterion
%   specified in WR:
%
%    'halfprom' - the reference line is positioned beneath the peak at a
%       vertical distance equal to half the peak prominence.
%
%    'halfheight' - the reference line is positioned at one-half the peak
%       height. The line is truncated if any of its intercept points lie
%       beyond the borders of the peaks selected by the 'MinPeakHeight',
%       'MinPeakProminence' and 'Threshold' parameters. The border between
%       peaks is defined by the horizontal position of the lowest valley
%       between them. Peaks with heights less than zero are discarded.
%
%    The default value of WR is 'halfprom'.
%
%   [...] = FINDPEAKS(...,'MinPeakWidth',MINW) finds peaks whose width is
%   at least MINW. The default value of MINW is zero.
%
%   [...] = FINDPEAKS(...,'MaxPeakWidth',MAXW) finds peaks whose width is
%   at most MAXW. The default value of MAXW is Inf.
%
%   [...] = FINDPEAKS(...,'MinPeakDistance',MPD) finds peaks separated by
%   more than the minimum peak distance, MPD. This parameter may be
%   specified to ignore smaller peaks that may occur in close proximity to
%   a large local peak. For example, if a large local peak occurs at LOC,
%   then all smaller peaks in the range [N-MPD, N+MPD] are ignored. If not
%   specified, MPD is assigned a value of zero.
%
%   [...] = FINDPEAKS(...,'SortStr',DIR) specifies the direction of sorting
%   of peaks. DIR can take values of 'ascend', 'descend' or 'none'. If not
%   specified, DIR takes the value of 'none' and the peaks are returned in
%   the order of their occurrence.
%
%   [...] = FINDPEAKS(...,'NPeaks',NP) specifies the maximum number of peaks
%   to be found. NP is an integer greater than zero. If not specified, all
%   peaks are returned. Use this parameter in conjunction with setting the
%   sort direction to 'descend' to return the NP largest peaks. (see
%   'SortStr')
%
%   [PKS,LOCS,W] = FINDPEAKS(...) returns the width, W, of each peak by
%   linear interpolation of the left- and right- intercept points to the
```

```
%   reference defined by 'WidthReference'.
%
%   [PKS,LOCS,W,P] = FINDPEAKS(...) returns the prominence, P, of each
%   peak.
%
%   FINDPEAKS(...) without output arguments plots the signal and the peak
%   values it finds
%
%   FINDPEAKS(...,'Annotate',PLOTSTYLE) will annotate a plot of the
%   signal with PLOTSTYLE. If PLOTSTYLE is 'peaks' the peaks will be
%   plotted. If PLOTSTYLE is 'extents' the signal, peak values, widths,
%   prominences of each peak will be annotated. 'Annotate' will be ignored
%   if called with output arguments. The default value of PLOTSTYLE is
%   'peaks'.
%
%   % Example 1:
%   %   Plot the Zurich numbers of sunspot activity from years 1700-1987
%   %   and identify all local maxima at least six years apart
%   load sunspot.dat
%   findpeaks(sunspot(:,2),sunspot(:,1),'MinPeakDistance',6)
%   xlabel('Year');
%   ylabel('Zurich number');
%
%   % Example 2:
%   %   Plot peak values of an audio signal that drop at least 1V on either
%   %   side without encountering values larger than the peak.
%   load mtlb
%   findpeaks(mtlb,Fs,'MinPeakProminence',1)
%
%   % Example 3:
%   %   Plot all peaks of a chirp signal whose widths are between .5 and 1
%   %   milliseconds.
%   Fs = 44.1e3; N = 1000;
%   x = sin(2*pi*(1:N)/N + (10*(1:N)/N).^2);
%   findpeaks(x,Fs,'MinPeakWidth',.5e-3,'MaxPeakWidth',1e-3, ...
%             'Annotate','extents')
%
%   See also MAX, FINDSIGNAL, FINDCHANGEPTS.

%   Copyright 2007-2023 The MathWorks, Inc.
%#ok<*EMCLS>
%#ok<*EMCA>
%#codegen
```

```
narginchk(1,22);
isInMATLAB = coder.target('MATLAB');

if nargout == 0 && ~isInMATLAB
    % Plotting is not supported for code generation. If this is running in
    % MATLAB, just call MATLAB's FINDPEAKS, else error.
    coder.internal.assert(coder.target('MEX') || coder.target('Sfun'), ...
        'signal:codegeneration:PlottingNotSupported');
    feval('findpeaks',Yin,varargin{:});
    return
end

% extract the parameters from the input argument list
[y,yIsRow,x,xIsRow,minH,minP,minW,maxW,minD,minT,maxN,sortDir,annotate,refW] ...
  = parse_inputs(isInMATLAB,Yin,varargin{:});

% indicate if we need to compute the extent of a peak
needWidth = minW>0 || maxW<inf || minP>0 || nargout>2 || strcmp(annotate,'extents');

if isInMATLAB
    % find indices of all finite and infinite peaks and the inflection points
    [iFinite,iInfinite,iInflect] = getAllPeaks(y);

    % keep only the indices of finite peaks that meet the required
    % minimum height and threshold
    iPk = removePeaksBelowMinPeakHeight(y,iFinite,minH,refW);
    iPk = removePeaksBelowThreshold(y,iPk,minT);
else
    % Use equivalent one-pass code generation algorithms
    [iFinite,iInfinite,iInflect] = getAllPeaksCodegen(y);
    iPk = removeSmallPeaks(y,iFinite,minH,minT);
end

if needWidth
  % obtain the indices of each peak (iPk), the prominence base (bPk), and
  % the x- and y- coordinates of the peak base (bxPk, byPk) and the width
  % (wxPk)
  [iPk,bPk,bxPk,byPk,wxPk] = signal.internal.findpeaks.findExtents(...
      y,x,iPk,iFinite,iInfinite,iInflect,minP,minW,maxW,refW);
else
  % combine finite and infinite peaks into one list
  [iPk,bPk,bxPk,byPk,wxPk] = combinePeaks(iPk,iInfinite);
end
```

```
% find the indices of the largest peaks within the specified distance
idx = findPeaksSeparatedByMoreThanMinPeakDistance(y,x,iPk,minD,sortDir);

% use the index vector to fetch the correct peaks.
% explicit creation of peaks to prevent resizing of iPk
if(length(idx)>maxN)
    fPk = coder.nullcopy(zeros(maxN, 1, class(iPk)));
    % Keep at most maxN peaks
    idx = idx(1:maxN);
else
    fPk = coder.nullcopy(zeros(size(idx), class(iPk)));
    maxN = cast(length(idx), class(maxN));
end

fPk = iPk(idx(1:length(fPk)));



if nargout > 0
  % assign output variables
  if needWidth
    [Ypk,Xpk,Wpk,Ppk] = assignFullOutputs(y,x,fPk,wxPk,bPk,yIsRow,xIsRow,idx,maxN);
  else
    [Ypk,Xpk] = assignOutputs(y,x,fPk,yIsRow,xIsRow);
  end
else
  % no output arguments specified. plot and optionally annotate
  if needWidth
    [bPk, bxPk, byPk, wxPk] = fetchPeakExtents(idx,bPk,bxPk,byPk,wxPk);
  end
  signal.internal.findpeaks.plot(x,y,fPk,bPk,bxPk,byPk,wxPk,refW,annotate)
end

%-------------------------------------------------------------------------
function [y,yIsRow,x,xIsRow,Ph,Pp,Wmin,Wmax,Pd,Th,NpOut,Str,Ann,Ref] =
parse_inputs(isInMATLAB,Yin,varargin)

% Validate input signal
validateattributes(Yin,{'double','single'},{'nonempty','real','vector'},...
    'findpeaks','Y');
y = Yin(:);
M = length(y);

if isInMATLAB
    if M < 3
```

```
        error(message('signal:findpeaks:emptyDataSet'));
    end
    yIsRow = isrow(Yin);
else
    coder.internal.assert(M >= 3,'signal:findpeaks:emptyDataSet');
    % To return row vectors we require Yin to be a row vector *type*, i.e.
    % length(size(y)) == 2, size(y,1) is constant 1, and size(y,2) ==
    % length(y). Otherwise, the output allocation might be O(n^2) instead
    % of O(n).
    yIsRow = coder.internal.isConst(isrow(Yin)) && isrow(Yin);
end

% indicate if the user specified an Fs or X
hasTimeInfo = ~(isempty(varargin) || coder.internal.isCharOrScalarString(varargin{1}));
if hasTimeInfo
  startArg = 2;
  if isInMATLAB
      FsSupplied = isscalar(varargin{1});
  else
      FsSupplied = coder.internal.isConst(isscalar(varargin{1})) &&
isscalar(varargin{1});
  end
  if FsSupplied
    % Fs
    Fs1 = varargin{1};
    validateattributes(Fs1,{'numeric'},{'real','finite','positive'},'findpeaks','Fs');
    Fs = double(Fs1(1));
    x = (0:M-1).'/Fs;
    xIsRow = yIsRow;
  else
    % X
    Xin1 = varargin{1};
    validateattributes(Xin1,{'numeric','datetime'},{'vector'},'findpeaks','X');
    if ~isInMATLAB

coder.internal.errorIf(isdatetime(Xin1),'signal:findpeaks:DatetimeInputsNotSupported');
    end
    if isnumeric(Xin1)

validateattributes(Xin1,{'numeric'},{'real','finite','increasing'},'findpeaks','X');
      Xin = double(Xin1);
    else % isdatetime(Xin)
      validateattributes(seconds(Xin1-
Xin1(1)),{'double'},{'real','finite','increasing'},'findpeaks','X');
```

```
      Xin = Xin1;
    end

    if isInMATLAB
        if length(Xin) ~= M
            throwAsCaller(MException(message('signal:findpeaks:mismatchYX')));
        end
        xIsRow = isrow(Xin);
    else
        coder.internal.assert(length(Xin) == M, ...
            'signal:findpeaks:mismatchYX');
        xIsRow = coder.internal.isConst(isrow(Xin)) && isrow(Xin);
    end
    x = Xin(:);
  end
else
  startArg = 1;
  % unspecified, use index vector
  x = (1:M).';
  xIsRow = yIsRow;
end

%#function dspopts.findpeaks
if isInMATLAB
    p = signal.internal.findpeaks.getParser();
    parse(p,varargin{startArg:end});
    Ph = p.Results.MinPeakHeight;
    Pp = p.Results.MinPeakProminence;
    Wmin = p.Results.MinPeakWidth;
    Wmax = p.Results.MaxPeakWidth;
    Pd = p.Results.MinPeakDistance;
    Th = p.Results.Threshold;
    Np = p.Results.NPeaks;
    Str = p.Results.SortStr;
    Ann = p.Results.Annotate;
    Ref = p.Results.WidthReference;
else
    defaultMinPeakHeight = -inf;
    defaultMinPeakProminence = 0;
    defaultMinPeakWidth = 0;
    defaultMaxPeakWidth = Inf;
    defaultMinPeakDistance = 0;
    defaultThreshold = 0;
    defaultNPeaks = [];
```

```
    defaultSortStr = 'none';
    defaultAnnotate = 'peaks';
    defaultWidthReference = 'halfprom';

    parms = struct('MinPeakHeight',uint32(0), ...
                'MinPeakProminence',uint32(0), ...
                'MinPeakWidth',uint32(0), ...
                'MaxPeakWidth',uint32(0), ...
                'MinPeakDistance',uint32(0), ...
                'Threshold',uint32(0), ...
                'NPeaks',uint32(0), ...
                'SortStr',uint32(0), ...
                'Annotate',uint32(0), ...
                'WidthReference',uint32(0));
    pstruct = eml_parse_parameter_inputs(parms,[],varargin{startArg:end});
    Ph =
eml_get_parameter_value(pstruct.MinPeakHeight,defaultMinPeakHeight,varargin{startArg:end
});
    Pp =
eml_get_parameter_value(pstruct.MinPeakProminence,defaultMinPeakProminence,varargin{star
tArg:end});
    Wmin =
eml_get_parameter_value(pstruct.MinPeakWidth,defaultMinPeakWidth,varargin{startArg:end})
;
    Wmax =
eml_get_parameter_value(pstruct.MaxPeakWidth,defaultMaxPeakWidth,varargin{startArg:end})
;
    Pd =
eml_get_parameter_value(pstruct.MinPeakDistance,defaultMinPeakDistance,varargin{startArg
:end});
    Th =
eml_get_parameter_value(pstruct.Threshold,defaultThreshold,varargin{startArg:end});
    Np = eml_get_parameter_value(pstruct.NPeaks,defaultNPeaks,varargin{startArg:end});
    Str =
eml_get_parameter_value(pstruct.SortStr,defaultSortStr,varargin{startArg:end});
    Ann =
eml_get_parameter_value(pstruct.Annotate,defaultAnnotate,varargin{startArg:end});
    Ref =
eml_get_parameter_value(pstruct.WidthReference,defaultWidthReference,varargin{startArg:e
nd});
end

% limit the number of peaks to the number of input samples
if isempty(Np)
```

```
    NpOut = M;
else
    NpOut = Np;
end

% ignore peaks below zero when using halfheight width reference
if strcmp(Ref,'halfheight')
  Ph = max(Ph,0);
end

validateattributes(Ph,{'numeric'},{'real','scalar','nonempty'},'findpeaks','MinPeakHeigh
t');
if isnumeric(x)
  validateattributes(Pd,{'numeric'},{'real','scalar','nonempty','nonnegative','<',x(M)-
x(1)},'findpeaks','MinPeakDistance');
else
  if isInMATLAB && isduration(Pd)

validateattributes(seconds(Pd),{'numeric'},{'real','scalar','nonempty','nonnegative'},'f
indpeaks','MinPeakDistance');
  else

validateattributes(Pd,{'numeric'},{'real','scalar','nonempty','nonnegative'},'findpeaks'
,'MinPeakDistance');
  end
end
validateattributes(Pp,{'numeric'},{'real','scalar','nonempty','nonnegative'},'findpeaks'
,'MinPeakProminence');
if isInMATLAB && isduration(Wmin)

validateattributes(seconds(Wmin),{'numeric'},{'real','scalar','finite','nonempty','nonne
gative'},'findpeaks','MinPeakWidth');
else

validateattributes(Wmin,{'numeric'},{'real','scalar','finite','nonempty','nonnegative'},
'findpeaks','MinPeakWidth');
end
if isInMATLAB && isduration(Wmax)

validateattributes(seconds(Wmax),{'numeric'},{'real','scalar','nonnan','nonempty','nonne
gative'},'findpeaks','MaxPeakWidth');
else
```

```
validateattributes(Wmax,{'numeric'},{'real','scalar','nonnan','nonempty','nonnegative'},
'findpeaks','MaxPeakWidth');
end
if isInMATLAB && isduration(Pd)

validateattributes(seconds(Pd),{'numeric'},{'real','scalar','nonempty','nonnegative'},'f
indpeaks','MinPeakDistance');
else

validateattributes(Pd,{'numeric'},{'real','scalar','nonempty','nonnegative'},'findpeaks'
,'MinPeakDistance');
end
validateattributes(Th,{'numeric'},{'real','scalar','nonempty','nonnegative'},'findpeaks'
,'Threshold');
validateattributes(NpOut,{'numeric'},{'real','scalar','nonempty','integer','positive'},'
findpeaks','NPeaks');
Str = validatestring(Str,{'ascend','none','descend'},'findpeaks','SortStr');
Ann = validatestring(Ann,{'peaks','extents'},'findpeaks','SortStr');
Ref = validatestring(Ref,{'halfprom','halfheight'},'findpeaks','WidthReference');

%-------------------------------------------------------------------------
function [iPk,iInf,iInflect] = getAllPeaks(y)
% fetch indices all infinite peaks
iInf = find(isinf(y) & y>0);

% temporarily remove all +Inf values
yTemp = y;
yTemp(iInf) = NaN;

% determine the peaks and inflection points of the signal
[iPk,iInflect] = findLocalMaxima(yTemp);



%-------------------------------------------------------------------------
function [iPk, iInflect] = findLocalMaxima(yTemp)
% bookend Y by NaN and make index vector
yTemp = [NaN; yTemp; NaN];
iTemp = (1:length(yTemp)).';

% keep only the first of any adjacent pairs of equal values (including NaN).
yFinite = ~isnan(yTemp);
iNeq = [1; 1 + find((yTemp(1:end-1) ~= yTemp(2:end)) & ...
                    (yFinite(1:end-1) | yFinite(2:end)))];
```

```
iTemp = iTemp(iNeq);

% take the sign of the first sample derivative
s = sign(diff(yTemp(iTemp)));

% find local maxima
iMax = 1 + find(diff(s)<0);

% find all transitions from rising to falling or to NaN
iAny = 1 + find(s(1:end-1)~=s(2:end));

% index into the original index vector without the NaN bookend.
iInflect = iTemp(iAny)-1;
iPk = iTemp(iMax)-1;



%--------------------------------------------------------------------------
function [iPk,iInf,iInflect] = getAllPeaksCodegen(y)
% One-pass code generation version of getAllPeaks
coder.varsize('iPk');
coder.varsize('iInf');
coder.varsize('iInflect');
% Define constants.
ZERO = coder.internal.indexInt(0);
ONE = coder.internal.indexInt(1);
DECREASING = 'd';
INCREASING = 'i';
NEITHER = 'n';
NonFiniteSupport = coder.internal.get_eml_option('NonFinitesSupport');
% Allocate output arrays.
iPk = coder.nullcopy(zeros(size(y),'like',ONE));
iInf = coder.nullcopy(zeros(size(y),'like',ONE));
iInflect = coder.nullcopy(zeros(size(y),'like',ONE));
ny = coder.internal.indexInt(length(y));
% Counter variables to store the number of elements in each array that are
% in use.
nPk = ZERO;
nInf = ZERO;
nInflect = ZERO;
% Initial direction.
dir = NEITHER;
if NonFiniteSupport || ny == 0
    % This is the typical start. With kfirst = 0 and ykfirst = +Inf, the
    % first value is an artificial +Inf. Unless the signal begins with
```

```
    % +Infs and/or NaNs, we'll pick up the first non-NaN, non-Inf value in
    % the first iteration, replace ykfirst with it, and proceed from there.
    kfirst = ZERO; % index of first element of a series of equal values
    ykfirst = coder.internal.inf('like',y); % first element of a series of equal values
    isinfykfirst = true;
else
    % With no non-finite support, we know the first element of the signal
    % is finite, so we can start with it.
    kfirst = ONE; % index of first element of a series of equal values
    ykfirst = y(1); % first element of a series of equal values
    isinfykfirst = false;
end
for k = kfirst + 1:ny
    yk = y(k);
    if isnan(yk)
        % yk is NaN. Convert it to +Inf.
        yk = coder.internal.inf('like',yk);
        isinfyk = true;
    elseif isinf(yk) && yk > 0
        % yk is +Inf. Record its position in the iInf array.
        isinfyk = true;
        nInf = nInf + 1;
        iInf(nInf) = k;
    else
        isinfyk = false;
    end
    if yk ~= ykfirst
        previousdir = dir;
        if NonFiniteSupport && (isinfyk || isinfykfirst)
            dir = NEITHER;
            % kfirst == 0 implies that ykfirst was just the artificial
            % starting value. We don't want to add the artificial value to
            % the array of inflection points, so we only append if kfirst
            % is at least 1.
            if kfirst >= 1
                nInflect = nInflect + 1;
                iInflect(nInflect) = kfirst;
            end
        elseif yk < ykfirst
            dir = DECREASING;
            if dir ~= previousdir
                % Previously the direction was not decreasing and now it
                % is. At least record an inflection point.
                nInflect = nInflect + 1;
```

```
                iInflect(nInflect) = kfirst;
                if previousdir == INCREASING
                    % Since the direction was previously increasing and now
                    % is decreasing, y(kfirst) is a peak.
                    nPk = nPk + 1;
                    iPk(nPk) = kfirst;
                end
            end
        else % if yk > ykfirst
            dir = INCREASING;
            if dir ~= previousdir
                % Direction was previously not increasing. Record the
                % inflection point y(kfirst).
                nInflect = nInflect + 1;
                iInflect(nInflect) = kfirst;
            end
        end
        % yk becomes the new ykfirst.
        ykfirst = yk;
        kfirst = k;
        isinfykfirst = isinfyk;
    end
end
% Add last point as inflection point if it is finite and not already there.
if ny > 0 && ~isinfykfirst && (nInflect == 0 || iInflect(nInflect) < ny)
    nInflect = nInflect + 1;
    iInflect(nInflect) = ny;
end
% Shorten the variable-size arrays down to the number of elements in use.
iPk = iPk(1:nPk,1);
iInf = iInf(1:nInf,1);
iInflect = iInflect(1:nInflect,1);

%-------------------------------------------------------------------------
function iPk = removePeaksBelowMinPeakHeight(Y,iPk,Ph,widthRef)
if ~isempty(iPk)
  iPk = iPk(Y(iPk) > Ph);
  if isempty(iPk) && ~strcmp(widthRef,'halfheight')
    warning(message('signal:findpeaks:largeMinPeakHeight', 'MinPeakHeight',
'MinPeakHeight'));
  end
end

%-------------------------------------------------------------------------
```

```
function iPk = removePeaksBelowThreshold(Y,iPk,Th)
base = max(Y(iPk-1),Y(iPk+1));
iPk = iPk(Y(iPk)-base >= Th);

%--------------------------------------------------------------------------
function iPk = removeSmallPeaks(y,iFinite,minH,thresh)
% Combination of removePeaksBelowMinPeakHeight and
% removePeaksBelowThreshold for code generation
iPk = coder.nullcopy(iFinite);
nPk = coder.internal.indexInt(0);
n = coder.internal.indexInt(length(iFinite));
for k = 1:n
    j = iFinite(k);
    pk = y(j);
    if pk > minH
        base = max(y(j - 1),y(j + 1));
        if pk - base >= thresh
            nPk = nPk + 1;
            iPk(nPk) = j;
        end
    end
end
iPk = iPk(1:nPk,1);

%--------------------------------------------------------------------------
function [iPkOut,bPk,bxPk,byPk,wxPk] = combinePeaks(iPk,iInf)
iPkOut = union(iPk,iInf);
bPk = zeros(0,1);
bxPk = zeros(0,2);
byPk = zeros(0,2);
wxPk = zeros(0,2);

%--------------------------------------------------------------------------
function idx = findPeaksSeparatedByMoreThanMinPeakDistance(y,x,iPk,Pd,sortDir)
% Start with the larger peaks to make sure we don't accidentally keep a
% small peak and remove a large peak in its neighborhood.

if isempty(iPk) || Pd==0
  IONE = ones('like',getIZERO);
  idx = orderPeaks(y,iPk,(IONE:length(iPk)).',sortDir);
  return
end

% copy peak values and locations to a temporary place
```

```
pks = y(iPk);
locs_temp = x(iPk);

% Order peaks from large to small
if coder.target('MATLAB')
    [~, sortIdx] = sort(pks,'d');
else

    ZERO = coder.internal.indexInt(0);
    sortIdx = coder.nullcopy(zeros(numel(pks), 1, 'like', ZERO));
    sortIdx = coder.internal.mergesort(sortIdx,pks,'d', ...
        ZERO,coder.internal.indexInt(numel(pks)));
end

locs_temp = locs_temp(sortIdx);

idelete = zeros(size(locs_temp), 'logical');

for i = 1:length(idelete)
  if ~idelete(i)
    % If the peak is not in the neighborhood of a larger peak, find
    % secondary peaks to eliminate.

    if coder.target('MATLAB')
        idelete = idelete | (locs_temp>=locs_temp(i)-Pd)&(locs_temp<=locs_temp(i)+Pd);
    else
               % explicit loop for better memory profile
        for jj = length(idelete):-1:1
            idelete(jj) = idelete(jj) | (locs_temp(jj)>=locs_temp(i)-
Pd)&(locs_temp(jj)<=locs_temp(i)+Pd);
        end
    end

    idelete(i) = 0; % Keep current peak
  end
end

% report back indices in consecutive order
idx = sortIdx(~idelete);

if isempty(idx)
    return
end
```

```
% re-order and bound the number of peaks based upon the index vector and
% sortDir.

if strcmp(sortDir,'none')
    if coder.target('MATLAB')
        idx = sort(idx);
    else
        idx = coder.internal.introsort(idx,coder.internal.indexInt(1), ...
            coder.internal.indexInt(length(idx)));
    end
elseif sortDir(1) == 'a'
    idx = flipud(idx);
end

%--------------------------------------------------------------------------
function idx = orderPeaks(Y,iPk,idx,Str)

if isempty(idx) || strcmp(Str,'none')
  return
end

if coder.target('MATLAB')
  [~,s]  = sort(Y(iPk(idx)),Str);
else
  ZERO = coder.internal.indexInt(0);
  s = zeros(numel(idx), 1, 'like', ZERO);
  s = coder.internal.mergesort(s, Y(iPk(idx)), Str(1), ...
      ZERO,coder.internal.indexInt(numel(idx)));
end
idx = idx(s);

%--------------------------------------------------------------------------
function [bPk,bxPk,byPk,wxPk] = fetchPeakExtents(idx,bPk,bxPk,byPk,wxPk)


bPk = bPk(idx);
bxPk = bxPk(idx,:);
byPk = byPk(idx,:);
wxPk = wxPk(idx,:);

%--------------------------------------------------------------------------
function [YpkOut,XpkOut] = assignOutputs(y,x,iPk,yIsRow,xIsRow)
coder.internal.prefer_const(yIsRow,xIsRow);
```

```
% fetch the coordinates of the peak
Ypk = y(iPk);
Xpk = x(iPk);

% preserve orientation of Y
if yIsRow
  YpkOut = Ypk.';
else
  YpkOut = Ypk;
end

% preserve orientation of X
if xIsRow
  XpkOut = Xpk.';
else
  XpkOut = Xpk;
end

%--------------------------------------------------------------------------
function [YpkOut,XpkOut,WpkOut,PpkOut] =
assignFullOutputs(y,x,iPk,wxPk,bPk,yIsRow,xIsRow,idx,maxN)
coder.internal.prefer_const(yIsRow,xIsRow);

% fetch the coordinates of the peak
Ypk = y(iPk);
Xpk = x(iPk);

% compute the width and prominence

Wpk = diff(wxPk(idx(1:maxN),:),1,2);

Ppk = Ypk - bPk(idx(1:maxN));

% preserve orientation of Y (and P)
if yIsRow
  YpkOut = Ypk.';
  PpkOut = Ppk.';
else
  YpkOut = Ypk;
  PpkOut = Ppk;
end

% preserve orientation of X (and W)
if xIsRow
```

```
  XpkOut = Xpk.';
  WpkOut = Wpk.';
else
  XpkOut = Xpk;
  WpkOut = Wpk;
end

%--------------------------------------------------------------------------
function y = getIZERO
% Return zero of the indexing type: double 0 in MATLAB,
% coder.internal.indexInt(0) for code generation targets.
if coder.target('MATLAB')
    y = 0;
else
    y = coder.internal.indexInt(0);
end

% [EOF]
```

Table 8-16: circleThroughThreePoints function

```
function [center, radius] = circleThroughThreePoints(p1, p2, p3)
A = [p1(1), p1(2), 1; p2(1), p2(2), 1; p3(1), p3(2), 1];
B = -[p1(1)^2 + p1(2)^2; p2(1)^2 + p2(2)^2; p3(1)^2 + p3(2)^2];
X = A \ B;

center = [-X(1)/2, -X(2)/2];
radius = 0.5*sqrt(X(1)^2 + X(2)^2 - 4*X(3));
end
```

8.4 Script for database creation

Table 8-17: Main script for database creation

```
%% Creacion de la base de datos.
%Cargar la tabla de respuestas de la BBDD de MySQL
vendor = "MySQL";
opts = databaseConnectionOptions("jdbc",vendor);
```

```
opts = setoptions(opts, ...
    'DataSourceName',"MySQL", ...
    'JDBCDriverLocation',...
    "C:\Program Files (x86)\MySQL\Connector J 8.0/mysql-connector-j-8.0.32.jar", ...
    'DatabaseName',"eyeh20_appeye",'Server',"eyeh2020.com", ...
    'PortNumber',3306);

username = "eyeh20_youssef";
password = "Youssef123456";

try
    status = testConnection(opts,username,password);
    conn = database(…);
catch
    opts = setoptions(opts, ...
        'DataSourceName',"MySQL", ...
        'JDBCDriverLocation',"C:\Program Files (x86)\MySQL\Connector J 8.0\mysql-
connector-j-8.0.32.jar", ...
        'DatabaseName',"…",'Server',"eyeh2020.com", ...
        'PortNumber',3306);
    status = testConnection(opts,username,password);
    conn = database(…);
end
% Verficicación de la conexión a la base de datos utilizando isopen, "1"
% conexión realizada con éxito. "0" conexión no establecida.
status = isopen(conn);

answers=sqlread(conn,"answers");
answers.id_iaqi=uint64(str2num(char(string(answers.id_iaqi))));
answers.answer=string(answers.answer);

iaqi=sqlread(conn,"iaqi");
iaqi.id_iaqi=uint64(str2num(char(string(iaqi.id_iaqi))));
weather=sqlread(conn,"weather");
weather.id_iaqi=uint64(str2num(char(string(weather.id_iaqi))));


%% CARAGAR ARCHIVO: Databse_29122024.mat
% Obtener el nombre excel de cada vídeo.
% Cargar la lista de videos
cd('C:\Users\Marra\OneDrive - Universidad Complutense de Madrid (UCM)\')
codigos=readtable('codigo_email.csv');
codigos.email=lower(string(codigos.email));
codigos.iaqi=uint64(codigos.iaqi);
```

```
% leer tabla de codigo paciente
patient_number=readtable('patient_code.csv');
patient_number.email=lower(string(patient_number.email));
% check CL
CL_use=readtable('CL_1.csv');
CL_use.email=lower(string(CL_use.email));
CL_use.id_iaqi=uint64(CL_use.id_iaqi);

% Create table with all the possible registers name.
code_patientn=innerjoin(codigos,patient_number,'Keys','email');
code_patientn_CL=outerjoin(code_patientn,CL_use(:,[2,7]),"LeftKey","iaqi","RightKey","id
_iaqi","MergeKeys",true);
[~,idx_missing]=rmmissing(code_patientn_CL.email);
code_patientn_CL=code_patientn_CL(find(~idx_missing),:);
code_patientn_CL.CL=fillmissing(code_patientn_CL.CL,'constant',0);
code_patientn_CL.iaqi=code_patientn_CL.iaqi_id_iaqi;
code_patientn_CL.iaqi_id_iaqi=[];
%creacion table codigo paciente email.
emails_codes=code_patientn_CL(:,1:2);

code_patientn_CL=code_patientn_CL(:,2:end);
rf=rowfilter(code_patientn_CL);

all_patient_codes=unique(code_patientn.email_code);

%% arreglo tabla answers
%asignando las keys de las respuestas.
answers_c=answers;
rf_answers_c=rowfilter(answers_c);
all_bad_answers=answers_c(rf_answers_c.id_iaqi <= 1692192118290,"answer");
bad_answers=unique(all_bad_answers);
bad_answers_corrected=array2table(["2. 18-29";
"3. Latino or Hispanic"            ;
"1. 0-17"                           ;
"2. 18-29"                          ;
"3. 30-44"                          ;
"4. 45-59"                          ;
"2. A veces"                        ;
"2. Afroamericano"                  ;
"5. Always"                         ;
"4. Asian"                          ;
"4. Assez souvent"                  ;
"3. Bastante a menudo"              ;
"3. Between 1 and 2 hours."         ;
```

```
"4. Between 2 and 4 hours."         ;
"4. Between 6 months and 1 year."   ;
"3. Between one month and 6 months.";
"1. Caucasian"                      ;
"1. Caucásico"                      ;
"3. Con bastante frecuencia"        ;
"3. Entre 1 y 2 horas"              ;
"4. Entre 2 y 4 horas"              ;
"3. Entre un mes y 6 meses"         ;
"3. Fairly often"                   ;
"2. Femenino"                       ;
"2. Femme"                          ;
"3. Half the time"                  ;
"1. I didn’t wear contact lenses."  ;
"1. I didn’t wear contact lenses."   ;
"3. La mitad del tiempo"            ;
"4. La plupart du temps"            ;
"3. Latino o hispano"               ;
"2. Less than 1 hour."              ;
"1. Male"                           ;
"1. Masculino"                      ;
"2. Menos de 1 hora"                ;
"3. Menos de 2 horas"               ;
"2. Menos de un mes"                ;
"5. More than 4 hours."             ;
"5. More than 4 hours."             ;
"4. Most of the time"               ;
"4. Muy a menudo"                   ;
"5. Más de 1 año."                  ;
"1. Never"                          ;
"1. Never "                         ;
"1. No he usado lentes de contacto" ;
"1. No usaba lentes de contacto"    ;
"1. No uso lentes de contacto"      ;
"1. No uso lentillas"               ;
"1. Nunca"                          ;
"Ocasionalmente"                ;
"2. Occasional"                     ;
"2. Occasionnellement"              ;
"2. Parfois"                        ;
"3. Prefer not to say"              ;
"3. Prefiero no decir"              ;
"5. Siempre"                        ;
"5. Toujours"                       ;
```

```
"4. Très souvent"                    ;
"4. Very often"  ],"VariableNames","corrected_answers");

correction_table=[bad_answers,bad_answers_corrected];
all_corrected_answers=join(all_bad_answers,correction_table);

answers_c(rf_answers_c.id_iaqi <= 1692192118290,"answer")=all_corrected_answers(:,2);

%% Make excels only for videos analyzed successfully by the blink algorithm
server = "localhost";
port = 27017;
dbname = "newdatabase";
conn = mongoc(server,port,dbname);

isopen(conn)

collectionlist=conn.CollectionNames;
success=endsWith(collectionlist,"properties");

videos_succesfully_analyzed=collectionlist(success);
videos_succesfully_analyzed=char(videos_succesfully_analyzed);
videos_succesfully_analyzed=videos_succesfully_analyzed(:,1:13);
videos_succesfully_analyzed=string(videos_succesfully_analyzed);
v_success=array2table(videos_succesfully_analyzed,VariableNames="iaqi");

conn=[];

code_patientn_CL_1=code_patientn_CL;
code_patientn_CL=innerjoin(code_patientn_CL_1,v_success);




all_data_patients=[];

for cnt=1:size(all_patient_codes,1)
    % cnt=4;
    patient=all_patient_codes(cnt);
    data_patient= code_patientn_CL(rf.email_code==patient,:);
    data_patient.total_CL=ones(size(data_patient,1),1)*size(data_patient(rf.CL ==
1,:),1);
    data_patient.total_wCL=ones(size(data_patient,1),1)*size(data_patient(rf.CL ==
0,:),1);
    data_patient_CL=data_patient(rf.CL == 1,:);
```

```
    data_patient_CL.video_numer=[1:size(data_patient(rf.CL == 1,:),1)]';
    data_patient_wCL=data_patient(rf.CL == 0,:);
    data_patient_wCL.video_numer=[1:size(data_patient(rf.CL == 0,:),1)]';
    data_patient=[data_patient_wCL;data_patient_CL];
    data_patient=sortrows(data_patient,'iaqi');
    all_data_patients=[all_data_patients;data_patient];
    data_patient=[];
end
all_data_patients.iaqi=uint64(all_data_patients.iaqi);
iaqis=unique(all_data_patients.iaqi);
q_type=zeros(size(iaqis));
q_type(find(iaqis>1706006352413))=2;
q_type(find(iaqis<=1706006352413))=1;

q_type=array2table([iaqis,q_type],VariableNames=["iaqi" "q_type"]);

all_data_patients=join(all_data_patients,q_type);

% for video_number=1:size(all_data_patients,1)
% for video_number=1:max(size(all_data_patients))
% for video_number=169:max(size(all_data_patients))
for video_number=1:169
patient_code=all_data_patients(video_number,"email_code");
%% extract data from blink analysis.
videoName=num2str(table2array(all_data_patients(video_number,"iaqi")));
excelName=strcat(['P' num2str(table2array(all_data_patients(video_number,"email_code")))
'_' ...
    num2str(table2array(all_data_patients(video_number,"CL"))) '_' ...
    num2str(table2array(all_data_patients(video_number,"total_CL"))) '_' ...
    num2str(table2array(all_data_patients(video_number,"total_wCL"))) '_' ...
    num2str(table2array(all_data_patients(video_number,"video_numer"))) '_Q'...
    num2str(table2array(all_data_patients(video_number,"q_type"))) '.xls']);

%% Access mongoDB database to extract data from collections where all the info is
stored.

server = "localhost";
port = 27017;
dbname = "newdatabase";
conn_mg = mongoc(server,port,dbname);

isopen(conn_mg)

%% Access sql database to extract answers from the patient
```

```
%% Conexión a la base de datos MySQL.
% Comprobación de la conexión a la base de datos MySQL.
try
    status = testConnection(opts,username,password);
    conn = database('eyeh20_appeye','eyeh20_youssef','Youssef123456');
catch
    opts = setoptions(opts, ...
        'DataSourceName',"MySQL", ...
        'JDBCDriverLocation',"C:\Program Files (x86)\MySQL\Connector J 8.0\mysql-
connector-j-8.0.32.jar", ...
        'DatabaseName',"eyeh20_appeye",'Server',"eyeh2020.com", ...
        'PortNumber',3306);
    status = testConnection(opts,username,password);
    conn = database('eyeh20_appeye','eyeh20_youssef','Youssef123456');
end
% Verficicación de la conexión a la base de datos utilizando isopen, "1"
% conexión realizada con éxito. "0" conexión no establecida.
status = isopen(conn);

answers_eye=sqlread(conn,"answers_eye");
answers_eye.id_iaqi=uint64(str2num(char(string(answers_eye.id_iaqi))));


%% Excel summary
summary_collectionName=string(char([videoName , '_summary']));
% antes de cambio: quitar luego (solo prueba)
summary_collectionName=string(char([videoName , '_summary']));


%% INCLUIR TRY CATCH POR SI EL VIDEO NO SE HA ANALIZADO
summary_data_blink=find(conn_mg,summary_collectionName);
summary_data_blink=rmfield(summary_data_blink,"_id");
fieldNames=fieldnames(summary_data_blink);



Rows=num2cell(string({summary_data_blink.Row}));
[summary_data_blink.Parameter]=Rows{:};
summary_data_blink=rmfield(summary_data_blink,"Row");

total_mean=     num2cell(string({summary_data_blink.total_mean}));
total_std=      num2cell(string({summary_data_blink.total_std}));
% new database
total_kurtosis= num2cell(string({summary_data_blink.total_moment_3}));
```

```
total_skewness= num2cell(string({summary_data_blink.total_moment_4}));
% total_kurtosis= num2cell(string({summary_data_blink.total_kurtosis}));
% total_skewness= num2cell(string({summary_data_blink.total_skewness}));

complete_blinks_mean=     num2cell(string({summary_data_blink.complete_blinks_mean}));
complete_blinks_std=      num2cell(string({summary_data_blink.complete_blinks_std}));

% new database
complete_blinks_kurtosis=
num2cell(string({summary_data_blink.complete_blinks_moment_3}));
complete_blinks_skewness=
num2cell(string({summary_data_blink.complete_blinks_moment_4}));

% complete_blinks_kurtosis=
num2cell(string({summary_data_blink.complete_blinks_kurtosis}));
% complete_blinks_skewness=
num2cell(string({summary_data_blink.complete_blinks_skewness}));

incomplete_blinks_mean=
num2cell(string({summary_data_blink.incomplete_blinks_mean}));
incomplete_blinks_std=
num2cell(string({summary_data_blink.incomplete_blinks_std}));
%new database
incomplete_blinks_kurtosis=
num2cell(string({summary_data_blink.incomplete_blinks_moment_3}));
incomplete_blinks_skewness=
num2cell(string({summary_data_blink.incomplete_blinks_moment_4}));

% incomplete_blinks_kurtosis=
num2cell(string({summary_data_blink.incomplete_blinks_kurtosis}));
% incomplete_blinks_skewness=
num2cell(string({summary_data_blink.incomplete_blinks_skewness}));

[summary_data_blink.total_mean]=      total_mean{:};
[summary_data_blink.total_std]=       total_std{:};

%new database
[summary_data_blink.total_moment_3]=  total_kurtosis{:};
[summary_data_blink.total_moment_4]=  total_skewness{:};

% [summary_data_blink.total_kurtosis]=  total_kurtosis{:};
% [summary_data_blink.total_skewness]=  total_skewness{:};

[summary_data_blink.complete_blinks_mean]=      complete_blinks_mean{:};
```

```
[summary_data_blink.complete_blinks_std]=       complete_blinks_std{:};
%new database
[summary_data_blink.complete_blinks_moment_3]=  complete_blinks_kurtosis{:};
[summary_data_blink.complete_blinks_moment_4]=  complete_blinks_skewness{:};

% [summary_data_blink.complete_blinks_kurtosis]=  complete_blinks_kurtosis{:};
% [summary_data_blink.complete_blinks_skewness]=  complete_blinks_skewness{:};

[summary_data_blink.incomplete_blinks_mean]=      incomplete_blinks_mean{:};
[summary_data_blink.incomplete_blinks_std]=       incomplete_blinks_std{:};

[summary_data_blink.incomplete_blinks_moment_3]=  incomplete_blinks_kurtosis{:};
[summary_data_blink.incomplete_blinks_moment_4]=  incomplete_blinks_skewness{:};

% [summary_data_blink.incomplete_blinks_kurtosis]=  incomplete_blinks_kurtosis{:};
% [summary_data_blink.incomplete_blinks_skewness]=  incomplete_blinks_skewness{:};

%new database
summary_data_blink=orderfields(summary_data_blink,["Parameter", "total_mean",
"total_std", "total_moment_3","total_moment_4","complete_blinks_mean",
"complete_blinks_std",
"complete_blinks_moment_3","complete_blinks_moment_4","incomplete_blinks_mean",
"incomplete_blinks_std", "incomplete_blinks_moment_3","incomplete_blinks_moment_4"]);

% summary_data_blink=orderfields(summary_data_blink,["Parameter", "total_mean",
"total_std", "total_kurtosis","total_skewness","complete_blinks_mean",
"complete_blinks_std",
"complete_blinks_kurtosis","complete_blinks_skewness","incomplete_blinks_mean",
"incomplete_blinks_std", "incomplete_blinks_kurtosis","incomplete_blinks_skewness"]);


summary_data_blink_cell=struct2cell(summary_data_blink);
% summary_data_blink_table=cell2table(summary_data_blink_cell(1:end-
1,:)','RowNames',cellstr(summary_data_blink_cell(end,:)),"VariableNames",[ "total_mean",
"total_std", "total_kurtosis","total_skewness","complete_blinks_mean",
"complete_blinks_std",
"complete_blinks_kurtosis","complete_blinks_skewness","incomplete_blinks_mean",
"incomplete_blinks_std", "incomplete_blinks_kurtosis","incomplete_blinks_skewness"]);

%new database
summary_data_blink_table=cell2table(summary_data_blink_cell(1:end,:),"RowNames",["Parame
ters", "total_mean", "total_std",
"total_moment_3","total_moment_4","complete_blinks_mean", "complete_blinks_std",
```

```
"complete_blinks_moment_3","complete_blinks_moment_4","incomplete_blinks_mean",
"incomplete_blinks_std", "incomplete_blinks_moment_3","incomplete_blinks_moment_4"]);

%
summary_data_blink_table=cell2table(summary_data_blink_cell(1:end,:),"RowNames",["Parame
ters", "total_mean", "total_std",
"total_kurtosis","total_skewness","complete_blinks_mean", "complete_blinks_std",
"complete_blinks_kurtosis","complete_blinks_skewness","incomplete_blinks_mean",
"incomplete_blinks_std", "incomplete_blinks_kurtosis","incomplete_blinks_skewness"]);

writetable(summary_data_blink_table,excelName,"Sheet","1","WriteRowNames",true,"WriteVar
iableNames",false)

%% ALL blinks
blinks_collectionName=string(char([videoName]));
% antes de cambio: quitar luego (solo prueba)

blinks_data_blink=find(conn_mg,blinks_collectionName);
blinks_data_blink=rmfield(blinks_data_blink,"_id");
blinks_fieldNames=fieldnames(blinks_data_blink);

blinks_data_blink_table=struct2table(blinks_data_blink,"AsArray",true);
blink_id=[1:size(blinks_data_blink_table,1)]';
blinks_data_blink_table=addvars(blinks_data_blink_table,blink_id,'Before','t0p_s_');

writetable(blinks_data_blink_table,excelName,"Sheet","2","WriteRowNames",false)

%% Video signal
signal_collectionName=string(char([videoName,'_signal']));
signal_data_blink=find(conn_mg,signal_collectionName);
signal_data_blink=rmfield(signal_data_blink,"_id");
signal_fields=fieldnames(signal_data_blink);
signal_data_blink_cell=struct2cell(signal_data_blink);
signal_data_blink_cell=[signal_fields';signal_data_blink_cell'];
writecell(signal_data_blink_cell,excelName,"Sheet","3")

%% Video Properties
properties_collectionName=string(char([videoName,'_properties']));
properties_data_blink=find(conn_mg,properties_collectionName);
properties_data_blink=rmfield(properties_data_blink,"_id");
properties_data_blink_table=struct2table(properties_data_blink);
properties_data_blink_table.CL_brand=NaN;

if uint64(str2num(videoName)) > 1707313460337
```

```
    rf_ec=rowfilter(emails_codes);
    current_email=emails_codes(rf_ec.email_code==patient_code.email_code,1);
    rf_ae=rowfilter(answers_eye);

    CL_brand=answers_eye(rf_ae.email == current_email.email(end) &
rf_ae.id_question==2,6);
    if size(CL_brand,1)>0
        properties_data_blink_table.CL_brand=CL_brand;
    else
        properties_data_blink_table.CL_brand=NaN;
    end
end



writetable(properties_data_blink_table,excelName,"Sheet","4","WriteRowNames",false)

%% Color data
%% creacion tabla nasal

%% CAMBIAR Y ELIMINAR TODOS LOS SYSTE,AS QUE NO ESTANN BIEN (NO TIENEN TODOS LOS
CANALES)

eval(['try  channel_nasal_hsv_1=channels_nasal',videoName,'_1_hsv; catch
channel_nasal_hsv_1=[]; end']);
eval(['try  channel_nasal_hsv_2=channels_nasal',videoName,'_2_hsv; catch
channel_nasal_hsv_2=[]; end']);
eval(['try  channel_nasal_hsv_3=channels_nasal',videoName,'_3_hsv; catch
channel_nasal_hsv_3=[]; end']);
eval(['try  channel_nasal_hsv_4=channels_nasal',videoName,'_4_hsv; catch
channel_nasal_hsv_4=[]; end']);
eval(['try  channel_nasal_hsv_5=channels_nasal',videoName,'_5_hsv; catch
channel_nasal_hsv_5=[]; end']);


if min(size(channel_nasal_hsv_1))<2
    channel_nasal_hsv_1=[];
end
if min(size(channel_nasal_hsv_2))<2
    channel_nasal_hsv_2=[];
end
if min(size(channel_nasal_hsv_3))<2
    channel_nasal_hsv_3=[];
end
```

```
if min(size(channel_nasal_hsv_4))<2
    channel_nasal_hsv_4=[];
end
if min(size(channel_nasal_hsv_5))<2
    channel_nasal_hsv_5=[];
end

channels_nasal_hsv=[channel_nasal_hsv_1;
channel_nasal_hsv_2;
channel_nasal_hsv_3;
channel_nasal_hsv_4;
channel_nasal_hsv_5];

eval(['try  channel_nasal_lab_1=channels_nasal',videoName,'_1_lab; catch
channel_nasal_lab_1=[]; end']);
eval(['try  channel_nasal_lab_2=channels_nasal',videoName,'_2_lab; catch
channel_nasal_lab_2=[]; end']);
eval(['try  channel_nasal_lab_3=channels_nasal',videoName,'_3_lab; catch
channel_nasal_lab_3=[]; end']);
eval(['try  channel_nasal_lab_4=channels_nasal',videoName,'_4_lab; catch
channel_nasal_lab_4=[]; end']);
eval(['try  channel_nasal_lab_5=channels_nasal',videoName,'_5_lab; catch
channel_nasal_lab_5=[]; end']);

if min(size(channel_nasal_lab_1))<2
    channel_nasal_lab_1=[];
end
if min(size(channel_nasal_lab_2))<2
    channel_nasal_lab_2=[];
end
if min(size(channel_nasal_lab_3))<2
    channel_nasal_lab_3=[];
end
if min(size(channel_nasal_lab_4))<2
    channel_nasal_lab_4=[];
end
if min(size(channel_nasal_lab_5))<2
    channel_nasal_lab_5=[];
end

channels_nasal_lab=[channel_nasal_lab_1;
              channel_nasal_lab_2;
              channel_nasal_lab_3;
              channel_nasal_lab_4;
```

```
                channel_nasal_lab_5];

eval(['try  channel_nasal_rgb_1=channels_nasal',videoName,'_1_rgb; catch
channel_nasal_rgb_1=[]; end']);
eval(['try  channel_nasal_rgb_2=channels_nasal',videoName,'_2_rgb; catch
channel_nasal_rgb_2=[]; end']);
eval(['try  channel_nasal_rgb_3=channels_nasal',videoName,'_3_rgb; catch
channel_nasal_rgb_3=[]; end']);
eval(['try  channel_nasal_rgb_4=channels_nasal',videoName,'_4_rgb; catch
channel_nasal_rgb_4=[]; end']);
eval(['try  channel_nasal_rgb_5=channels_nasal',videoName,'_5_rgb; catch
channel_nasal_rgb_5=[]; end']);

if min(size(channel_nasal_rgb_1))<2
    channel_nasal_rgb_1=[];
end
if min(size(channel_nasal_rgb_2))<2
    channel_nasal_rgb_2=[];
end
if min(size(channel_nasal_rgb_3))<2
    channel_nasal_rgb_3=[];
end
if min(size(channel_nasal_rgb_4))<2
    channel_nasal_rgb_4=[];
end
if min(size(channel_nasal_rgb_5))<2
    channel_nasal_rgb_5=[];
end

channels_nasal_rgb=[channel_nasal_rgb_1;
                channel_nasal_rgb_2;
                channel_nasal_rgb_3;
                channel_nasal_rgb_4;
                channel_nasal_rgb_5];

all_channels_nasal=[];
try
    all_channels_nasal=array2table([channels_nasal_hsv channels_nasal_lab
channels_nasal_rgb],VariableNames=["channel_h" "channel_s" "channel_v" "channel_l"
"channel_a" "channel_b_lab" "channel_r" "channel_g" "channel_b_rgb"]);
catch
    all_channels_nasal=array2table(zeros([1 9]),VariableNames=["channel_h" "channel_s"
"channel_v" "channel_l" "channel_a" "channel_b_lab" "channel_r" "channel_g"
"channel_b_rgb"]);
```

```
end

%% creacion tabla temporal

eval(['try  channel_temporal_hsv_1=channels_temporal',videoName,'_1_hsv; catch
channel_temporal_hsv_1=[]; end']);
eval(['try  channel_temporal_hsv_2=channels_temporal',videoName,'_2_hsv; catch
channel_temporal_hsv_2=[]; end']);
eval(['try  channel_temporal_hsv_3=channels_temporal',videoName,'_3_hsv; catch
channel_temporal_hsv_3=[]; end']);
eval(['try  channel_temporal_hsv_4=channels_temporal',videoName,'_4_hsv; catch
channel_temporal_hsv_4=[]; end']);
eval(['try  channel_temporal_hsv_5=channels_temporal',videoName,'_5_hsv; catch
channel_temporal_hsv_5=[]; end']);

if min(size(channel_temporal_hsv_1))<2
    channel_temporal_hsv_1=[];
end
if min(size(channel_temporal_hsv_2))<2
    channel_temporal_hsv_2=[];
end
if min(size(channel_temporal_hsv_3))<2
    channel_temporal_hsv_3=[];
end
if min(size(channel_temporal_hsv_4))<2
    channel_temporal_hsv_4=[];
end
if min(size(channel_temporal_hsv_5))<2
    channel_temporal_hsv_5=[];
end

channels_hsv=[channel_temporal_hsv_1;
              channel_temporal_hsv_2;
              channel_temporal_hsv_3;
              channel_temporal_hsv_4;
              channel_temporal_hsv_5];

eval(['try  channel_temporal_lab_1=channels_temporal',videoName,'_1_lab; catch
channel_temporal_lab_1=[]; end']);
eval(['try  channel_temporal_lab_2=channels_temporal',videoName,'_2_lab; catch
channel_temporal_lab_2=[]; end']);
eval(['try  channel_temporal_lab_3=channels_temporal',videoName,'_3_lab; catch
channel_temporal_lab_3=[]; end']);
```

```
eval(['try  channel_temporal_lab_4=channels_temporal',videoName,'_4_lab; catch
channel_temporal_lab_4=[]; end']);
eval(['try  channel_temporal_lab_5=channels_temporal',videoName,'_5_lab; catch
channel_temporal_lab_5=[]; end']);

if min(size(channel_temporal_lab_1))<2
    channel_temporal_lab_1=[];
end
if min(size(channel_temporal_lab_2))<2
    channel_temporal_lab_2=[];
end
if min(size(channel_temporal_lab_3))<2
    channel_temporal_lab_3=[];
end
if min(size(channel_temporal_lab_4))<2
    channel_temporal_lab_4=[];
end
if min(size(channel_temporal_lab_5))<2
    channel_temporal_lab_5=[];
end

channels_lab=[channel_temporal_lab_1;
              channel_temporal_lab_2;
              channel_temporal_lab_3;
              channel_temporal_lab_4;
              channel_temporal_lab_5];

eval(['try  channel_temporal_rgb_1=channels_temporal',videoName,'_1_rgb; catch
channel_temporal_rgb_1=[]; end']);
eval(['try  channel_temporal_rgb_2=channels_temporal',videoName,'_2_rgb; catch
channel_temporal_rgb_2=[]; end']);
eval(['try  channel_temporal_rgb_3=channels_temporal',videoName,'_3_rgb; catch
channel_temporal_rgb_3=[]; end']);
eval(['try  channel_temporal_rgb_4=channels_temporal',videoName,'_4_rgb; catch
channel_temporal_rgb_4=[]; end']);
eval(['try  channel_temporal_rgb_5=channels_temporal',videoName,'_5_rgb; catch
channel_temporal_rgb_5=[]; end']);

if min(size(channel_temporal_rgb_1))<2
    channel_temporal_rgb_1=[];
end
if min(size(channel_temporal_rgb_2))<2
    channel_temporal_rgb_2=[];
end
```

```
if min(size(channel_temporal_rgb_3))<2
    channel_temporal_rgb_3=[];
end
if min(size(channel_temporal_rgb_4))<2
    channel_temporal_rgb_4=[];
end
if min(size(channel_temporal_rgb_5))<2
    channel_temporal_rgb_5=[];
end

channels_rgb=[channel_temporal_rgb_1;
              channel_temporal_rgb_2;
              channel_temporal_rgb_3;
              channel_temporal_rgb_4;
              channel_temporal_rgb_5];

all_channels_temporal=[];

try
    all_channels_temporal=array2table([channels_hsv channels_lab
channels_rgb],VariableNames=["channel_h" "channel_s" "channel_v" "channel_l" "channel_a"
"channel_b_lab" "channel_r" "channel_g" "channel_b_rgb"]);
catch
    all_channels_temporal=array2table(zeros([1 9]),VariableNames=["channel_h"
"channel_s" "channel_v" "channel_l" "channel_a" "channel_b_lab" "channel_r" "channel_g"
"channel_b_rgb"]);
end

all_mean_nasal=mean(all_channels_nasal,1);
all_std_nasal=std(all_channels_nasal,0,1);
try
    all_kurtosis_nasal=kurtosis(all_channels_nasal,1);
    all_skewness_nasal=skewness(all_channels_nasal,1);
catch
    all_kurtosis_nasal=array2table(zeros([1 9]),"VariableNames",["channel_h" "channel_s"
"channel_v" "channel_l" "channel_a" "channel_b_lab" "channel_r" "channel_g"
"channel_b_rgb"]);
    all_skewness_nasal=array2table(zeros([1 9]),"VariableNames",["channel_h" "channel_s"
"channel_v" "channel_l" "channel_a" "channel_b_lab" "channel_r" "channel_g"
"channel_b_rgb"]);
end

all_mean_temporal=mean(all_channels_temporal,1);
all_std_temporal=std(all_channels_temporal,0,1);
```

```
try
    all_kurtosis_temporal=kurtosis(all_channels_temporal,1);
    all_skewness_temporal=skewness(all_channels_temporal,1);
catch
    all_kurtosis_temporal=array2table(zeros([1 9]),"VariableNames",["channel_h"
"channel_s" "channel_v" "channel_l" "channel_a" "channel_b_lab" "channel_r" "channel_g"
"channel_b_rgb"]);
    all_skewness_temporal=array2table(zeros([1 9]),"VariableNames",["channel_h"
"channel_s" "channel_v" "channel_l" "channel_a" "channel_b_lab" "channel_r" "channel_g"
"channel_b_rgb"]);
end

color_database=[all_mean_nasal;all_std_nasal;all_kurtosis_nasal;all_skewness_nasal;...
    all_mean_temporal;all_std_temporal;all_kurtosis_temporal;all_skewness_temporal];

color_database.Properties.RowNames=["mean_nasal" "std_nasal" "kurtosis_nasal"
"skewness_nasal" "mean_temporal" "std_temporal" "kurtosis_temporal"
"skewness_temporal"];

%% Get color database from script analisis_resultados_color.
% table name T
% rf=rowfilter(T);
% color_database=T(rf.iaqi == uint64(str2num(videoName)),:);
% color_database(:,["OriginalVariableNames", "iaqi"])=[];

writetable(color_database,excelName,"Sheet","5","WriteRowNames",true)
writetable(all_channels_nasal,excelName,"Sheet","nasal_color_data","WriteRowNames",false
)
writetable(all_channels_temporal,excelName,"Sheet","temporal_color_data","WriteRowNames"
,false)

%% COntamination data
rf_iaqi=rowfilter(iaqi);
env_database=iaqi(rf_iaqi.id_iaqi ==uint64(str2num(videoName)),:);
env_database(:,["id", "id_iaqi","dew","h","p","t","w","wg"])=[];
% writetable(env_database,excelName,"Sheet","6","WriteRowNames",false)

%% Weather data
rf_weather=rowfilter(weather);
weather_database=weather(rf_weather.id_iaqi ==uint64(str2num(videoName)),:);
weather_database(:,["id", "id_iaqi", "email", "ce"])=[];
% writetable(weather_database,excelName,"Sheet","7","WriteRowNames",false)
```

```
% Connect weather + contamination data
if size(env_database,1)==0
    env_database=array2table(NaN([1 6]),VariableNames=["dew","h","p","t","w","wg"]);
end

if size(weather_database,1)==0
    weather_database=array2table(NaN([1 7]),VariableNames=["ta"     "vmax"    "vv"
"dv"     "prec"    "pres"     "hr"]);
end

wc_data=[env_database(1,:), weather_database(1,:)];


%% Get Temperature and humidity in the inside
rf_ae=rowfilter(answers_eye);
if uint64(str2num(videoName)) <= 1707313460337
    temp_int=answers_eye(rf_ae.id_iaqi == uint64(str2num(videoName)) &
rf_ae.id_question==2 ,6);
    hum_int=answers_eye(rf_ae.id_iaqi == uint64(str2num(videoName)) &
rf_ae.id_question==3 ,6);
else
    temp_int=answers_eye(rf_ae.id_iaqi == uint64(str2num(videoName)) &
rf_ae.id_question==3 ,6);
    hum_int=answers_eye(rf_ae.id_iaqi == uint64(str2num(videoName)) &
rf_ae.id_question==4 ,6);
end

if size(temp_int,1)==0
   temp_int=array2table([NaN],VariableNames=["T_int"]);
end
if size(hum_int,1)==0
    hum_int=array2table([NaN],VariableNames=["hr_int"]);
end

temp_int.Properties.VariableNames="T_int";
hum_int.Properties.VariableNames="hr_int";
wc_data=[wc_data,temp_int,hum_int];

writetable(wc_data,excelName,"Sheet","Environment_data","WriteRowNames",false)

%% creacion de la tabla answers
rf_answers=rowfilter(answers_c);
% videoName='1691058258596';
```

```
answers_database=answers_c(rf_answers.id_iaqi==uint64(str2num(videoName)),:);
answers_database=answers_database(:,["id_question" "question" "answer"]);
check_questionnaire=max(answers_database.id_question);
if check_questionnaire>20 | check_questionnaire==11;
    %CLDEQ8
    answers_database=answers_database(rf_answers.id_question>7,:);

writetable(answers_database,excelName,"Sheet","Questionnaire_type_2","WriteRowNames",fal
se)
else
    %Cuestionario inicial
    answers_database=answers_database(rf_answers.id_question>5,:);

writetable(answers_database,excelName,"Sheet","Questionnaire_type_1","WriteRowNames",fal
se)
end
end
```