Detección de Contenido Sexual mediante Aprendizaje Profundo y Aprendizaje por Transferencia

Sexual Content Detection through Deep Learning and Transfer Learning

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA DEL SOFTWARE
CURSO 2022–2023

Íñigo Sanz Torres

Directores
Luis Javier García Villalba
Daniel Povedano Álvarez

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, Junio de 2023

Agradecimientos

A mi director del TFG, que confió en mí para trabajar en este proyecto. También a Ana, Luis y Daniel por ayudarme durante estos meses con todas las dudas que me surgían sobre la documentación e implementación.

A mi familia por apoyarme durante todos mis años de estudio.

Índice General

Índice de Figuras

Índice de Tablas

Lista de Acrónimos

AI	*Artificial Intelligence*
CNN	*Convolutional Neural Networks*
DL	*Deep Learning*
GPU	*Graphics Processing Unit*
IA	Inteligencia Artificial
KNN	*KNeighbors*
LSTM	*Long Short Term Memory*
MASI	Material de Abuso Sexual Infantil
ML	*Machine Learning*
MLP	*Multi-Layer Perceptrons*
NSFW	*Not Safe For Work*

RGB *Red-Green-Blue*

RNN *Recurrent Neural Networks*

SIFT *Scale-invariant Feature Transform*

STIP *Spatio-Temporal Interest Point*

SVM Máquina de vectores de soporte

ViT *Vision Transformer*

XML *Extensible Markup Language*

Abstract

Due to the amount of sexual content that exists on the internet, a method capable of differentiating it between safe content is necessary. Therefore, this Bachelor's thesis will be focused on developing a tool that will analyse images automatically classify sexual content. Research was carried out on other sexual content detection tools to understand the methods that will be used on this project. To create this tool, we will use deep learning and transfer learning techniques, using as pretrained model Open nsfw model by Yahoo for sexual content classification. Finally, a data set consisting of enough images to train similar models will be developed.

Keywords: Artificial Intelligence, Computer Vision, Deep Learning, Image Classifiction, Neural Networks, Sexual content detection, Transfer Learning.

Resumen

Debido a la cantidad de contenido sexual que existe en internet, es necesario un método capaz de discernirlo entre el contenido seguro. Por ello, se trata de enfocar este Trabajo Fin de Grado en desarrollar una herramienta que realizará un análisis de las imágenes para la identificación automática de contenido sexual. Para comprender los métodos que se van a usar en este proyecto, se realizó una investigación sobre otras herramientas diseñadas para la detección de contenido sexual. Para la creación de esta herramienta se utilizarán técnicas de aprendizaje profundo y aprendizaje por transferencia, utilizando como base para ello el modelo de detección de contenido sexual en imágenes *Open nsfw model* desarrollado por Yahoo. Además, se desarrollará y limpiará un conjunto de datos formado por suficientes imágenes para el entrenamiento de modelos con este mismo objetivo.

Palabras clave: Aprendizaje por transferencia, aprendizaje profundo, clasificación de imágenes, detección de contenido sexual, inteligencia artificial, redes neuronales, visión artificial

Capítulo 1

Introducción

1.1. Motivación

Durante los últimos 65 años, internet ha ido creciendo cada vez más rápidamente. En un primer momento creado con intereses militares [oG23], poco a poco ha ido evolucionando hasta convertirse en lo que es hoy en día. La popularidad del internet ha llevado a aumentar su velocidad, permitiendo el rápido intercambio de imágenes y vídeos.

La mayoría de este contenido es seguro y puede ser visto por todo el mundo, pero no todo. Parte de este contenido puede no ser seguro o incluso ilegal, como la pornografía infantil. Debido a la cantidad de contenido que se sube a internet constantemente, es imposible asegurarse manualmente de que todo lo que se suba sea legal. Por ello, la mayoría de las plataformas que permiten subir contenido a sus aplicaciones utilizan algoritmos que comprueban si el contenido subido es seguro para que se suba.

Estos algoritmos suelen utilizar inteligencias artificiales y visión artificial para detectar si el contenido se puede permitir en dichas plataformas. Actualmente, estas tecnologías están evolucionando constantemente ya que se usan en todo tipo de aplicaciones, por lo que encontrar el mejor modelo es muy importante para estos algoritmos.

Los algoritmos de detección de contenido sexual son muy comunes en redes sociales, pero no se usan tanto en los cuerpos de seguridad. Estos algoritmos podrían ayudar a la policía en la búsqueda de contenido ilegal, ya que muchas veces necesitan encontrar este contenido en grandes cantidades de archivos con un tiempo limitado. Debido al límite de tiempo que se da en estos casos, revisar todo este contenido manualmente es inviable, así que la posibilidad de usar un algoritmo que sea capaz de localizar aquellos archivos rápidamente sería muy beneficioso.

Un modelo de visión artificial bien entrenado y lo suficientemente rápido sería muy útil para los cuerpos de seguridad, y por ello realizamos este trabajo.

1.2. Contexto

El presente Trabajo Fin de Grado se enmarca dentro de un proyecto de investigación titulado Novel Strategies to Fight Child Sexual Exploitation and Human Trafficking Crimes and Protect their Victims – HEROES, aprobado por la Comisión Europea dentro del Programa Marco Horizonte 2020 (convocatoria H2020-SU-SEC-2020) en virtud del

acuerdo de subvención número 101021801 y en el que participa como coordinador del proyecto el Grupo GASS de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, https://gass.ucm.es, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

Además de la Universidad Complutense de Madrid participan en HEROES 21 entidades ubicadas en 17 países: 11 de países de la UE (Austria, Bélgica, Bulgaria, Francia, Grecia, Irlanda, Letonia, Lituania, Portugal, España, Reino Unido), 1 país asociado (Suiza) y 5 terceros países (Bangladesh, Brasil, Colombia, Perú, Uruguay). Dichas entidades son: University of Kent (Reino Unido), The Free University of Brussels (Bélgica), The French National Research Institute for Digital Science and Technology – INRIA (Francia), Center for Security Studies – KEMEA (Grecia), International Centre for Migration Policy Development – ICMPD (Austria), International Center for Missing and Exploited Children – ICMEC (Suiza), IDENER Research & Development Agrupación de Interés Económico (España), Athena Research Center – ARC (Grecia), Trilateral Research and Consulting (Reino Unido), Centre for Women and Children Studies – CWCS (Bangladesh), Center Against Human Trafficking and Exploitation – KOPZI (Lituania), Portuguese Association for Victim Support – APAV (Portugal), Fundación Renacer (Colombia), The Greek Council for Refugees – GCR (Grecia), Brazilian Association for the Defense of Children of Children and Youth – ASBRAD (Brasil), Hellenic Police (Grecia), Latvia National Police (Letonia), General Directorate for the Fight against Organized Crime (Bulgaria), Dirección General de la Policía – DGP (España), Federal Police (Brasil), Federal Highway Police (Brasil), Secretaría de Inteligencia Estratégica de Estado – Presidencia de la República Oriental del Uruguay (Uruguay)

Tienen más información en:

https://cordis.europa.eu/project/id/101021801

https://heroes-fct.eu

1.3. Objeto de la Investigación

El objetivo de este Trabajo Fin de Grado es diseñar e implementar un modelo de visión artificial capaz de clasificar imágenes con contenido sexual. Este modelo permitirá la revisión de grandes cantidades de imágenes en poco tiempo buscando contenido no seguro, ayudando a cuerpos de seguridad a encontrar este contenido. Por tanto, se debe utilizar una arquitectura con buen rendimiento además de rápida y preparar un conjunto de datos que permita su correcto entrenamiento. Se realizarán extensas pruebas para encontrar los mejores parámetros y métodos de entrenamiento del modelo usando un sistema de pruebas estándar, con el objetivo de poder comparar con otros modelos similares. De igual forma, este proyecto pretende comprender el funcionamiento de los modelos de visión artificial, su estructura y entrenamiento, además de las dificultades que propone crear uno.

1.4. Plan de Trabajo

El desarrollo de este trabajo se ha realizado en tres fases:

1. **Investigación**: Durante los primeros meses del proyecto se llevó a cabo el período de aprendizaje del marco teórico y del estado del arte. Para empezar, se hizo una reunión

en la que se explicaron los objetivos y conocimientos necesarios para llevar a cabo el proyecto. También se acordaron reuniones semanales para realizar el seguimiento del proyecto, resolver dudas y, en caso de que fuese necesario, realizar explicaciones sobre las herramientas que se fuesen a usar. Algunas de estas herramientas estaban centradas en la preparación de la documentación del proyecto, como *Mendeley Reference Manager*. Una vez se tenía el conocimiento necesario y se habían estudiados otros modelos del estado del arte, se comenzó con el desarrollo.

2. **Desarrollo**: Tras obtener el conocimiento necesario, se comenzó el desarrollo del proyecto. La fase de investigación no llegó a detenerse, debido a que se tuvo que seguir investigando sobre las librerías que se iban a utilizar. Entre estas librerías se encontraban *TensorFlow* y *keras* de *Python*. Durante esta fase también se preparó el conjunto de datos que se utilizaría para el entrenamiento, para ello se revisó y limpió manualmente para asegurar los mejores resultados. El modelo y el conjunto de datos original se obtuvieron de uno de los artículos que se leyeron durante la fase de investigación.

3. **Experimentación**: Tras avanzar lo suficiente en el desarrollo del proyecto, el proceso de experimentación comenzó. En este realizamos predicciones y entrenamientos con el modelo para comparar los resultados entre versiones del conjunto de datos, métodos y parámetros de entrenamiento. Durante esta fase tampoco se detuvieron las fases de investigación y desarrollo, necesarias para aprender y desarrollar herramientas y métodos usados durante la realización de pruebas, como fue la herramienta *Grad-Cam* que ofrece explicabilidad a los modelos de visión artificial.

4. **Documentación**: A lo largo de todas las fases del proyecto se fueron documentando los resultados. Durante la fase de investigación se realizaron resúmenes sobre los artículos que se leyeron, durante la fase de desarrollo se fue comentando el código pertinente para su fácil entendimiento y durante la fase de experimentación se documentaron los resultados de cada prueba.

En la Figura 1.1 muestra el flujo de trabajo en un diagrama de Gantt.

Figura 1.1: Flujo del trabajo

1.5. Estructura del Trabajo

El resto del trabajo está organizado en 8 capítulos estructurados de la siguiente forma: El Capítulo 2 se explica la historia de la Inteligencia Artificial (IA), además de una explicación teórica del funcionamiento de sus arquitecturas más comunes. Entre estas arquitecturas se encuentran el aprendizaje automático, el aprendizaje profundo y la visión artificial.

El Capítulo 3 se describen cronológicamente los algoritmos y modelos que se han usado con el objetivo de detectar y clasificar imágenes con contenido sexual. También se muestran los resultados que obtuvieron en sus pruebas e información sobre el tamaño y contenido de los conjuntos de datos que usaron en cada proyecto.

El Capítulo 4 se muestran las contribuciones de este proyecto. Se presentan el modelo propuesto que será entrenado, el conjunto de datos elegido para entrenarlo y cómo se entrenará. También se explicarán el preproceso y el paso de aumento de datos que se realizan antes usar las imágenes para el entrenamiento del modelo. Por último, se explica el uso que se le ha dado a la herramienta de *Grad-Cam*.

El Capítulo 5 se describen el procedimiento que se siguió para la realización de los entrenamientos y presentan los resultados obtenidos para cada entrenamiento.

El Capítulo 6 muestran las conclusiones obtenidas tras la realización de este proyecto y las acciones que se pueden realizar para ampliar.

En los Capítulos 7 y 8 se encuentran las traducciones al inglés de la Introducción y de las Conclusiones.

Por último, en el Capítulo del Anexo A se muestran todos los resultados de todos los entrenamientos.

Capítulo 2

Marco teórico

En esta ección se va a explicar el marco teórico necesario para comprender este Trabajo Fin de Grado. En la Sección 2.1 se realizará una breve introducción a la historia de la IA. Se explicará el origen y funcionamiento del aprendizaje automático (del inglés *Machine Learning* (ML)) en la Sección 2.2. En la Sección 2.3 se explicará el aprendizaje profundo (del inglés *Deep Learning* (DL)), junto con una explicación sobre las redes neuronales, las redes neuronales convolucionales y el aprendizaje por transferencia. Por último, en la Sección 2.4 se mostrarán los usos y evolución de la visión artificial.

2.1. Historia de la Inteligencia Artificial

La IA es una rama de las ciencias de la computación que se refiere al desarrollo de tareas que requieren inteligencia humana para ser resueltas, tareas como el reconocimiento del habla, traducción de lenguajes, toma de decisiones o percepción visual. Para realizar estas tareas, las IAs utilizan algoritmos y modelos estadísticos creados tras analizar grandes cantidades de datos.

El término de IA se acuñó durante la Conferencia de Dartmouth [Any23] en el año 1956, aunque anteriormente ya se habían realizado trabajos relacionados con ella usando otros nombres como el programa de ajedrez que Alan Turing diseñó [MMN23] entre 1948 y 1950 pero no pudo implementar debido a la falta de potencia en los ordenadores del momento. Tras esta conferencia, el Departamento de Defensa de los Estados Unidos empezó a financiar investigaciones sobre IA y se fundaron laboratorios por todo el mundo.

En ese momento, aparecieron dos enfoques en el diseño de la IA [RNCRJA04]. El primero, conocido como IA simbólica, era diseñado para crear un representación simbólica de los sistemas y medio en el que trabajaba, y debía ser diseñado basándose en el proceso de decisión de un experto humano. El segundo enfoque, llamado IA subsimbólica, buscaba que la IA aprendiese por sí misma. Para ello, se basaron en el concepto de las conexiones que crean las neuronas biológicas y diseñaron el perceptrón. El perceptrón es un discriminador lineal que es capaz de aprender a clasificar valores entre dos clases.

A pesar del optimismo inicial, el progreso se fue ralentizando hasta que en 1974 los gobiernos de Estados Unidos y Gran Bretaña retiraron la financiación [Equ23] dando comienzo a lo que se conoció como el primer Invierno IA. Durante el primer invierno IA el interés disminuyó hasta principios de la década de los 80, cuando debido al éxito comercial de sistemas expertos como el Mycin, especializado en diagnosticar enfermedades infecciosas

y escrito para máquinas Lisp [SA23]. Esto llevó a los gobiernos de Estados Unidos y Gran Bretaña a restaurar la financiación hasta que volvieron a retirarla en 1987 con el colapso del mercado de máquinas Lisp [Hen08].

Durante la década de los años 90 y principios del siglo XXI, la reputación de las IAs fue mejorando poco a poco tras producir buenos resultados y colaborar en otros campos, como las matemáticas, estadística y economía. Debido a la mejora de velocidad de los ordenadores del momento y la posibilidad de acceder a grandes cantidades de datos, en el año 2012, las técnicas DL empezaron a dominar por su precisión en las pruebas de rendimiento [Equ23].

Actualmente, la IA está siendo rápidamente desarrollada debido a sus múltiples aplicaciones en campos como la salud, finanzas, sistemas de recomendación o transporte.

2.2. Aprendizaje automático

"El aprendizaje automático o ML es el campo de estudio que permite a los ordenadores aprender a partir de datos sin ser explícitamente programados." (Arthur Samuel, 1959). Existen varios tipos de algoritmos como el aprendizaje supervisado, aprendizaje no supervisado, sistemas de recomendación y el aprendizaje por refuerzo [Ng23a].

En el aprendizaje supervisado, el aprendizaje se lleva a cabo usando casos con la solución correcta ya propuesta. Dentro del aprendizaje supervisado existen dos tipos principales de tareas: regresión y clasificación. La regresión sirve para predecir un número entre infinitos valores posibles, por ejemplo, se podría usar para calcular el precio de una casa sabiendo los metros cuadrados de la misma. La clasificación se utiliza para predecir una etiqueta o clase, distinguiendo entre un número finito de clases, como por ejemplo clasificar si un tumor es maligno o benigno conociendo su tamaño y la edad del paciente.

El aprendizaje no supervisado se usa cuando los datos no tienen la respuesta correcta y se necesita que el algoritmo encuentre una estructura en los datos. Entre los usos más habituales que se da a este método están el análisis de grupos que junta datos similares, la reducción de dimensionalidad que se usa para comprimir los datos en un espacio latente con menos dimensiones, y la detección de anomalías que se usa para encontrar datos extraños dentro del conjunto.

2.2.1. Regresión

El modelo más sencillo que se puede crear de aprendizaje supervisado es el de regresión lineal usando una variable [Ng23a]. El algoritmo de aprendizaje deberá crear una función que, recibiendo unos valores de entrada, devuelva una predicción estimada, esta función está representada como se muestra en la Ecuación 2.1.

$$f_{w,b}(x) = wx + b \tag{2.1}$$

Para encontrar los valores de w y b, el algoritmo de aprendizaje deberá minimizar la función de coste que haga que todas las distancias de los valores estimados de x estén lo más cerca posible del valor real, para ello se calcula el error cuadrático medio usando la Ecuación 2.2.

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2 \tag{2.2}$$

Donde m es la cantidad de ejemplos en el entrenamiento, $\hat{y}^{(i)}$ es el valor estimado del caso $x^{(i)}$ y el valor real es $y^{(i)}$, donde (i) es el índice del caso. Para encontrar el valor mínimo de esta función se usa el descenso de gradiente que encuentra mínimos locales, pero dado que la función de coste está elevada al cuadrado, sólo habrá un mínimo. Este algoritmo actualiza los valores de w y b al mismo tiempo hasta que llegue al mínimo o a un valor lo suficientemente cercano, para lo que se usan las Ecuaciones 2.3 y 2.4.

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)} \tag{2.3}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)}) \tag{2.4}$$

Donde α es la tasa de aprendizaje, que determina el tamaño de los pasos que se deben dar para llegar al mínimo. El valor no puede ser demasiado pequeño porque puede hacer que el descenso gradiente tarde demasiado ni muy grande ya que podría no llegar a no encontrar el valor mínimo que se busca.

Para los modelos en los que no hay sólo una variable si no varias, se utiliza la Ecuación 2.5 llamada producto escalar.

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b = w_1 x_1 + w_2 x_2 + w_3 x_3 + ... + w_n x_n + b \tag{2.5}$$

El descenso gradiente para varias variables se calcula usando las Ecuaciones 2.6 y 2.7.

$$\begin{aligned} w_1 &= w_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_1^{(i)} \\ &\vdots \\ w_n &= w_n - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_n^{(i)} \end{aligned} \tag{2.6}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) \tag{2.7}$$

2.2.2. Clasificación

La forma más básica de clasificación es la clasificación binaria, donde la salida sólo puede adoptar dos valores, que se suelen representar como 0 y 1 [Ng23a]. Para el cálculo de las probabilidades se utiliza la función sigmoide como se muestra en la Ecuación 2.8.

$$g(z) = \frac{1}{1 + e^{-z}} \tag{2.8}$$

La Ecuación 2.9 representa la probabilidad de que y sea 1 dada la entrada $\vec{x}$ y los parámetros $\vec{w}$ y b y se la conoce como regresión logística.

$$f_{\vec{w},b}(\vec{x}) = g(\vec{w} \cdot \vec{x} + b) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}} \tag{2.9}$$

Para encontrar los valores de $\vec{w}$ que den las predicciones más precisas se buscará minimizar el valor de la función de coste, que se muestra en la Ecuación 2.10.

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(f_{\vec{w},b}(\vec{x}^{(i)})) + (1 - y^{(i)})log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))] \tag{2.10}$$

Para ello se utiliza la misma fórmula de descenso gradiente que se usaba en la regresión lineal de múltiples variables.

2.3. Aprendizaje Profundo

El DL se puede definir como un tipo de red neuronal con tres o más capas ocultas con el objetivo de funcionar de manera similar al cerebro humano. A diferencia de otros algoritmos, los modelos DL no requieren que los datos con los que se entrenan estén estructurados, automatizando la extracción de características. Esto les permite llegar a comprender imágenes, texto o voces [Ng23b].

Aunque para ello, los modelos de DL requieren de grandes cantidades de datos en sus entrenamientos. Esto hace que los modelos de DL sean computacionalmente más costosos de entrenar y obliga al uso de *Graphics Processing Unit* (GPU) para su entrenamiento. Dado que el objetivo de este trabajo es la clasificación de imágenes, a continuación, se describirán las principales técnicas para el procesamiento de imágenes: las redes neuronales y las redes neuronales convolucionales.

2.3.1. Redes Neuronales

Las redes neuronales son modelos de IA que se originaron basándose en el funcionamiento de las neuronas del cerebro humano. En las décadas de los 80 y 90 tuvieron un avance significativo, pero a finales de los años 90 se perdió interés debido a la falta de potencia en los ordenadores del momento. En el año 2005, volvió el interés en las redes neuronales debido a la mejora de potencia de los ordenadores y el auge del *Big Data* o la cantidad de información disponible en todos los ámbitos (imágenes, texto, audio, etc.), y siguió aumentando al empezar a usar unidades de procesamiento gráfico (en inglés GPU) para mejorar el rendimiento. Algunos de los usos que se les dio a estas redes neuronales fueron la detección de habla, clasificación de imágenes y el procesamiento de lenguaje natural.

El componente básico de una red neuronal es la neurona [Ng23b]. Cada neurona recibe varios valores y devuelve un valor. Una red neuronal es un conjunto de neuronas separadas en capas y conectadas entre ellas. Las redes neuronales están formadas por una capa de entrada, donde recibe los valores a procesar, una capa de salida, donde devuelve el resultado; y una o varias capas ocultas formadas por neuronas que realizan los cálculos, como se puede ver en la Figura 2.1.

Cada neurona devuelve un valor $\vec{a}_j^{[i]}$ donde i es la posición de la capa oculta y j la posición de la neurona en esa capa. Para calcular el valor de $\vec{a}_j^{[i]}$ se calcula usando la función de activación. Algunas de las funciones de activación más comunes son: la función sigmoide, la función lineal y la función rectificadora, que se pueden ver en las Ecuaciones 2.8, 2.11 y 2.12 respectivamente.

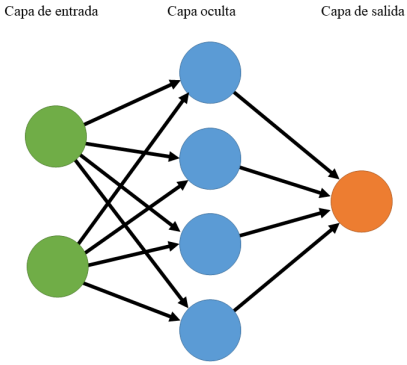

Figura 2.1: Diseño de una red neuronal

$$g(z) = z \tag{2.11}$$

$$g(z) = max(0, z) \tag{2.12}$$

El valor de z se calcula como $\vec{w}_j^{[i]} \cdot \vec{a}^{[i-1]} + b_j$. En el caso de la primera capa, en vez de usarse $\vec{a}^{[i-1]}$ se usan los valores de entrada $\vec{x}$. El valor devuelto por la capa de salida será un valor entre 0 y 1, por lo que en clasificadores binarios el resultado será 1 si la salida es mayor o igual que 0,5 y 0 en caso contrario, utilizando la función sigmoide, representada en la Ecuación 2.8, para obtener estas probabilidades. Para clasificación entre múltiples clases, la última capa estará formada por tantas neuronas como clases se quieran clasificar. Cada neurona se activará usando la función *Softmax* representada en la Ecuación 2.13.

$$\begin{aligned} z_j &= \vec{w}_j \cdot \vec{x} + b_j \\ a_j &= \frac{e^{z_j}}{\sum_{k=1}^{N} e^{z_k}} \end{aligned} \tag{2.13}$$

Donde j va de 1 a N, siendo N el número de clases posibles.

El entrenamiento de las redes neuronales comienza inicializando aleatoriamente los pesos $\vec{w}_j^{[i]}$ y b_j. Una vez hecho esto, la red neuronal recibirá los datos de entrenamiento y realizará los cálculos necesarios para clasificar estos datos, lo que se conoce como propagación hacia delante. Al terminar de clasificar los datos de entrenamiento se calcula el coste, para ello se realiza la propagación hacia atrás, que consiste en calcular el gradiente de la función de coste con respecto a los valores de $\vec{w}_j^{[i]}$ y b_j de cada capa usando la regla de la cadena. Por último, se actualizan los valores de $\vec{w}_j^{[i]}$ y b_j con el objetivo de reducir el valor de la función de coste, y para ello se suele utilizar el descenso gradiente. Estos pasos se repetirán a partir de la propagación hacia delante hasta que termine de entrenarse.

2.3.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales son un tipo de red neuronal utilizado para reconocimiento de imágenes y vídeos mediante el análisis de estructuras espaciales [Ng23b]. Estas redes están formadas por capas de filtros convolucionales, que se encargan de la extracción de características; seguidas de una fase de reducción por muestreo, y por último, la capa *fully-connected*, formada por neuronas sencillas que se encargan de clasificar las características extraídas.

Los valores de entrada consisten en los píxeles de una imagen. Para las imágenes con color (*Red-Green-Blue* (RGB)) cada píxel debe entrar tres veces, una por cada canal. Cada color de cada píxel estará representado como un valor entre 0 y 255, por lo que hay que normalizarlas entre 0 y 1.

La operación de convolución consiste en realizar el producto escalar de un conjunto de píxeles cercanos y una pequeña matriz conocida como kernel. Este cálculo se debe realizar en todos los píxeles al menos una vez, por lo que se recorre la imagen de izquierda a derecha y de arriba a abajo, lo que genera una matriz de salida. Si la imagen tiene color, el resultado del cálculo para cada color se sumaría dando lugar a una sóla salida. Al conjunto de kernels se le llama filtro convolucional. Cada capa realizará cálculos para todos los kernels del filtro dando resultado a varias matrices de salida.

A continuación, se encuentra la fase de muestreo, donde se toman las neuronas más representativas para hacer una convolución. Es necesario reducir la cantidad de neuronas que se toman debido al tamaño que puede llegar a tener una red si se usasen todos los píxeles de la imagen en cada capa. El método de muestreo más común es el de *Max-Pooling*, que crea una matriz de menor tamaño y toma los valores más altos de subregiones del tamaño de la matriz del *Max-Pooling* de la matriz de entrada. Esto devuelve una matriz más pequeña que debería conservar los valores más importantes de la matriz original. Esta matriz resultante pasará a la siguiente capa de filtros convolucionales que repetirá el proceso hasta que llegue a la capa *fully-connected*.

En la capa *fully-connected*, la matriz que recibe es transformada en un vector lineal. Esta capa funciona como la capa anterior a la capa de salida de una red neuronal, por lo que las neuronas de esta capa se activan con la función de activación *Softmax*. Se puede ver un ejemplo de red neuronal en la Figura 2.2. Con ellas se pueden clasificar las imágenes entres varias clases, devolviendo la probabilidad de que la imagen de entrada pertenezca a cada clase. Para entrenar las redes convolucionales también se usa la propagación hacia delante y hacia atrás, pero en vez de ajustarse el peso de las neuronas, se ajustan los pesos de los kernels.

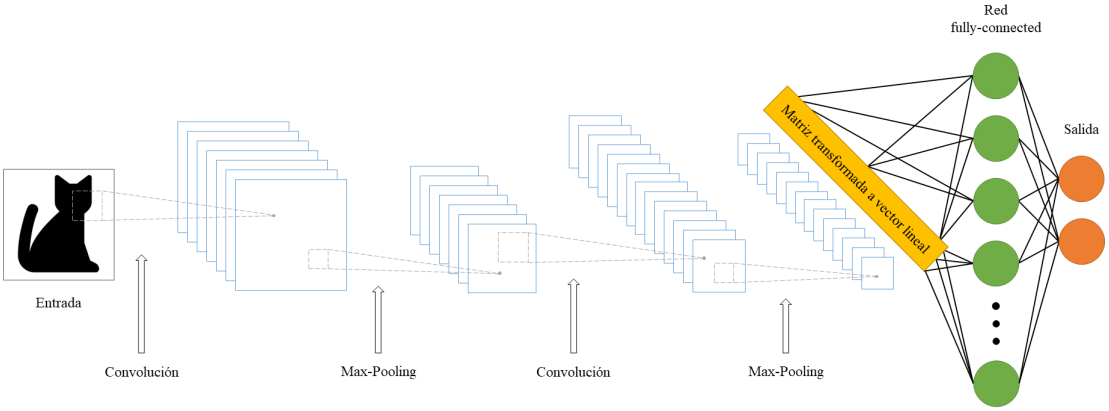

Figura 2.2: Red neuronal convolucional

2.3.3. Aprendizaje por Transferencia

El aprendizaje por transferencia o *Transfer Learning* es un método de entrenamiento de DL en el que un modelo desarrollado para una tarea genérica, como los modelos entrenados con el conjunto de datos ImageNet [DDS+09], formado por 1000 clases diferentes, se usa

como punto de partida para el entrenamiento de otro modelo con una tarea similar al primero. Esto se hace para poder ahorrar tiempo y recursos en entrenar un modelo de cero.

Este método se basa en la idea de que los modelos de DL aprenden características similares en sus primeras capas sin importar la tarea para la que se entrene. El método más común para aplicar el *Transfer Learning* es congelar las primeras capas de una modelo de DL ya entrenado y entrenar las capas finales en la nueva tarea. Para que esto funcione las dos tareas deben ser similares, como se puede ver en el ejemplo de la Figura 2.3, se podrían usar las primeras capas de un modelo que clasifica animales para entrenar un modelo que clasifica perros y gatos, ya que ambos utilizarán técnicas de visión artificial para detectar características en las imágenes.

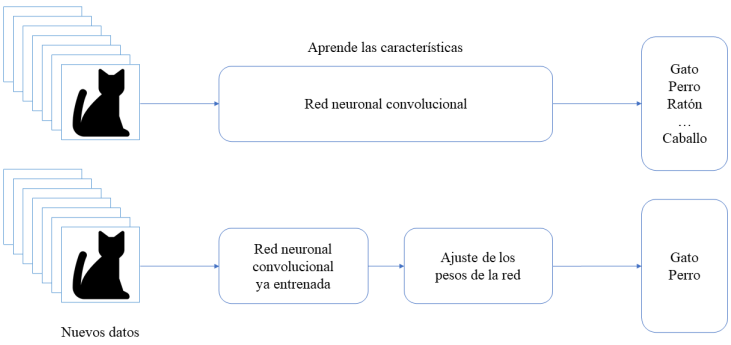

Figura 2.3: Aprendizaje por transferencia

2.3.4. Localización basada en gradiente

Unos de los mayores problemas de las redes neuronales es su falta de explicabilidad. Para solucionar esto, [SCD+16], tratan de proporcionar explicaciones visuales de redes profundas mediante activaciones de clase usando los pesos gradientes. Al ejecutar esta herramienta sobre una imagen dada una clase, devuelve un mapa de calor que indica qué píxeles de la imagen han activado las neuronas del modelo. Esto permite comprender cuáles son las características en las que más se fijan el modelo y comprobar si está clasificando las características importantes.

2.4. Visión Artificial

La visión artificial es el campo de la IA que permite a los ordenadores identificar y reconocer objetos y personas en imágenes. De la misma forma que las redes neuronales están basadas en las neuronas humanas, la visión artificial está basada en la visión humana. Aunque su principal diferencia es que los humanos pasamos toda una vida viendo y aprendiendo mientras que estos modelos deben aprender en poco tiempo utilizando grandes cantidades de datos [IBM23].

Para poder realizar tareas como reconocimiento y clasificación de imágenes, el algoritmo de visión artificial deberá analizar las imágenes repetidas veces hasta encontrar los patrones subyacentes. Para ello, en los últimos años, la arquitectura de DL más empleada en este campo han sido las redes neuronales convolucionales (del inglés *Convolutional Neural*

Networks (CNN)). Las CNN son un tipo de red neuronal capaz de comprender el contexto de los datos visuales utilizando algoritmos que le permiten aprender a diferenciar imágenes. Estos datos suelen estar representados como la posición del píxel en el plano y varios valores numéricos que permitirán diferenciar el color, como podrían ser los valores para el rojo, verde y azul. Las CNN aplican filtros de convolución a la imagen de entrada para extraer características de bajo nivel, como bordes y formas. Para ello realiza iteraciones en las que ejecuta la operación matemática de las convoluciones hasta que empieza a detectar y reconocer estos patrones en imágenes. Este sistema es similar a lo que hace la vista humana, en un primer momento se disciernen los bordes y las formas, y a continuación va rellenando hasta que comienza a ver la imagen completa. Para los vídeos, dado que son secuencias de imágenes, a veces se utilizan redes neuronales recurrentes (del inglés *Recurrent Neural Networks* (RNN)), que son redes capaces de procesar secuencias de imágenes y tienen la capacidad de recordar, en este contexto, las características de la imagen anterior.

El último avance realizado en el ámbito de la visión artificial son los *Vision Transformer* (ViT). Los *Transformers* se usaban originalmente para el procesamiento de lenguaje natural, midiendo las relaciones entre pares o grupos de palabras mediante mecanismos de atención (del inglés, *attention mechanisms*). En el caso de las imágenes, en vez de palabras se usarían píxeles, pero eso es inviable a nivel computacional por la cantidad de píxeles que hay en una imagen. Para solucionar esto, las relaciones se realizaban entre conjuntos de píxeles conocidos como parches que se ordenan secuencialmente en un vector, estas relaciones luego las recibía una red neuronal que se encarga de la clasificación final, como se puede ver en la Figura 2.4.

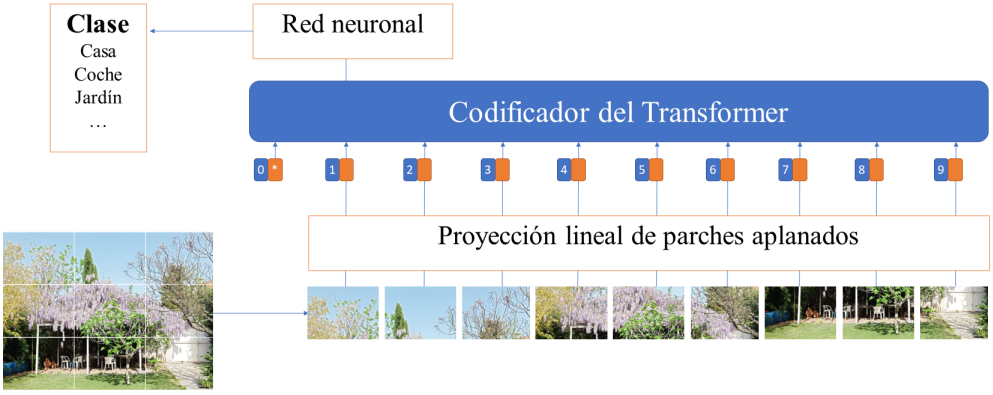

Figura 2.4: Diseño de un *Transformer*

En la actualidad se realizan constantemente investigaciones relacionadas con la visión artificial, pero también se usan en el día a día. Por ejemplo, se usan para automatizar inspección de calidad en productos, clasificación de imágenes, control de procesos, contadores de personas, detección facial, modelado de objeto y terrenos, navegación de vehículos autónomos o la organización de bases de datos de imágenes. Además, gracias a la cantidad de datos visuales que se generan hoy en día por los teléfonos móviles, cámaras de seguridad y cámaras de tráficos entre otros, es fácil acceder a las grandes cantidades de imágenes y vídeos necesarias para entrenar estos modelos.

Capítulo 3

Estado del Arte

El uso de visión artificial para la detección de contenido sexual en vídeos e imágenes es uno de los campos de investigación más maduros y avanzados. Por ello, se ha utilizado ampliamente en la detección de contenido sexual en imágenes para automatizar la revisión manual de estas mismas.

En un primer momento para detectar si en una imagen había contenido sexual se utilizó el número de píxeles de piel que había en ella [FF99]. Para este modelo utilizaron un conjunto de datos de 565 imágenes sexuales y 4302 no sexuales con el objetivo de filtrar las imágenes que había en internet, evitando aquellas que consideran ofensivas. El método que utilizaron fue primero detectar zonas de piel según el color y la textura y, a continuación, utilizaron un algoritmo que detectaba partes del cuerpo humano como los miembros, la posición de la cadera y la cara. Aquellas imágenes en las que detectaba piel donde se encontraba la espina o la cadera las consideraba como desnudos, este método obtuvo resultados bastante malos con una *precision* del 57 % y un *recall* del 43 %. Más tarde, los autores de [AA05] presentaron el algoritmo de detección de desnudez basándose en el método de detección de piel y agruparlas en regiones de piel. Tras detectar las regiones de piel, este algoritmo clasificará las imágenes como desnudos si cumple las siguientes condiciones:

1. Si el número de píxeles es menor que el 15 % del tamaño de la imagen no se clasifica como desnudo, en caso contrario se sigue en el paso 2.
2. Si la región de piel más grande es menor que el 35 % o la segunda o tercera son menor que el 30 % no se clasifica como desnudo, en caso contrario se sigue con el paso 3.
3. Si el número de píxeles de piel de la mayor región de píxeles de piel es menor que el 45 % del total de píxeles de piel no se clasifica como desnudo, en caso contrario se sigue con el paso 4.
4. Si el número de regiones de piel es mayor que el 60 % y la intensidad media dentro del polígono es menor que el 0.25 entonces no se clasifica, en caso contrario sí se clasifica como desnudo.

Como conjunto de datos utilizó 685 imágenes con desnudos y 935 imágenes sin desnudos, además obtuvo un buen resultado, con un *recall* del 94,32 %. Debido a la aparición de la plataforma YouTube, la necesidad de detectar contenido sexual en vídeos aumentó. Para ello, [AdLdS+11] utilizó los fotogramas de los vídeos y los pasaba por un algoritmo de extracción de características, usaban clasificadores Máquina de vectores de soporte

(SVM) que clasificaba como pornográfica o no pornográfica la característica. Finalmente, el vídeo se clasificaba por voto mayoritario. Como algoritmos de extracción de características probaron *Scale-invariant Feature Transform* (SIFT), *HueSIFT* y *Spatio-Temporal Interest Point* (STIP) siendo este el que mejores resultados dio, acertando nueve de cada diez vídeos de su conjunto de datos. Fue en este artículo donde se presentó el conjunto de datos *Pornography-800* [ATC+13], predecesor del *Pornography2k* [MAP+16] que es el conjunto que se usará como base para entrenamiento y prueba del modelo propuesto. Además de las características extraídas de los fotogramas utilizaron los descriptores de vídeo. Finalmente, utilizaron un clasificador SVM para la detección final del vídeo, como en el caso de [CAJ+14] que utilizaron el descriptor de vídeo *BinBoost* junto con la representación *BossaNova*. Siguieron utilizando el *Pornography-800* como conjunto de datos y con él obtuvieron un 90,9 % de *accuracy*.

A pesar de los avances en otros métodos, [GRH+18] siguió tratando de mejorar el método de detección de píxeles de piel, esta vez usando el detector de piel de [Mah17] y manteniendo el algoritmo de detección de desnudez. Obtuvieron una *precision* del 90,33 % y una *accuracy* del 80,23 % usando un conjunto de datos formados por 986 imágenes y 253 vídeos, estos resultados no lograron superar aquellos que usaban otros métodos más modernos, como el aprendizaje automático. Otro enfoque a la detección de contenido sexual fue el de [HSWA18], que consideraban que el problema provenía de que los teléfonos móviles permitían tomar fotografías con contenido sexual en ellas y que se debían censurar aquellas fotografías que contenían contenido sexual. Para ello utilizaron el clasificador en cascada *Haar* de la biblioteca *OpenCV* que crea un archivo *Extensible Markup Language* (XML) donde se buscarán los patrones de la imagen. Utilizando este método obtuvieron una *accuracy* del 85 %. Otro de los métodos que no se había probado para los vídeos era incluir un clasificador de audio, [DFDSB+19] procesaba los 5 primeros minutos de cada vídeo, usando un fotograma por segundo para el vídeo en la red *InceptionV3* y el audio pasaba por la red *AudioVGG*. Una vez obtenían las características de vídeo y audio probaron a clasificarlas usando *SVN*, *KNeighbors* (KNN) y *Multi-Layer Perceptrons* (MLP) y con este último modelo obtuvieron un 98,03 % de *precision* en el conjunto de datos *Pornography2k*. Un año más tarde, [FBGC20] mejoraron este sistema sustituyendo MLP por un clasificador *Long Short Term Memory* (LSTM), alcanzando una *F1-score* del 99,00 % superando el 97,99 % que obtuvieron anteriormente.

Con el aumento de uso de aplicaciones de *streaming* en tiempo real por la llegada de la pandemia del coronavirus, se necesitó aumentar la velocidad de detección de contenido sexual para poder detener la reproducción si tuviese algún tipo de contenido sexual. Para ello, [SK20] diseñó un modelo por capas que primero concatenaba las características extraídas del audio y vídeo y las clasificaba, si la clasificación era pornográfica la ejecución terminaba. En caso contrario, volvía a clasificar esta vez únicamente las características del vídeo y, de nuevo, si se clasificaba como pornográfico terminaba la ejecución, pero en caso contrario vuelve a clasificar únicamente las características del audio, y esa sería la clasificación final. Para extraer las características del vídeo utilizaron la CNN *VGG-16* junto con una red neuronal recurrente LSTM y para extraer las de audio usaron el *Static Feature Extraction de Mel-Scaled Spectogram* y una CNN. Usando el conjunto de datos *Pornography2k* obtuvieron un 92,33 % de *accuracy* pero con un ratio de un 4,6 % de falsos negativos que creen que se debió al mal rendimiento del clasificador de audio.

Para mejorar la velocidad de procesamiento sin perder precisión y con el objetivo de crear un modelo capaz de encontrar genitales en imágenes, [TBC+20] utilizó *transfer learning* empleando la red *MobileNet*, sustituyendo la capa de clasificación por una capa

Dense ReLU seguida por una capa de salida *softmax*. Entrenaron el modelo utilizando el *Adult Pornography Dataset 2M* (*Pornography2M*), que está formado por dos millones de imágenes, y obtuvieron un *accuracy* del 95 % en clasificación pornográfica y una velocidad de menos de 5 milisegundos en comparación a *VGG-16* o *Inceptaion-V3*. Continuando con el aumento de velocidad de procesamiento, [ANFR+20] hicieron pruebas de velocidad en distintos ordenadores con distintos sistemas operativos usando el modelo *Not Safe For Work* (NSFW) de Yahoo [MP23]. Estas pruebas demostraron que el sistema operativo Ubuntu es más rápido que Windows, además hicieron pruebas usando como unidad de procesamiento una tarjeta gráfica en vez de un procesador y obtuvieron velocidades mucho mayores y, por último, comprobaron el impacto que tiene el reescalado de imágenes en la precisión del modelo y obtuvieron mejores resultados reescalando la imagen en un 25 %.

Uno de los más recientes avances en el campo de la visión artificial son los mecanismos de atención, [CLH+20] diseñó una arquitectura formada por una CNN conectada a un módulo de atención que se aseguraba de seleccionar únicamente las características que realmente les interesaba y para poder utilizar imágenes de distintos tamaños propone el uso de *Scale Constraint Pooling*, que convierte las entradas de diferentes tamaños a salidas de un mismo tamaño. Además, para evitar posibles ataques adversarios añadió al preproceso un paso de comprensión y descomprensión de la imagen. Para entrenar este modelo utilizaron un conjunto de datos propio formado por 62500 imágenes que aumentaron a 500000 usando técnicas de aumentación de datos y obtuvieron una *accuracy* de 98,41 %.

[GGCAF21] presenta un método cuyo objetivo era detectar Material de Abuso Sexual Infantil (MASI). Para ello, dividió el problema en dos partes. Por un lado, primero se detectaba si en la imagen hay o no contenido sexual y seguidamente la detección de la edad, mediante un modelo que sea capaz de estimar la edad de las personas que aparecen en la imagen utilizando mecanismo de atención combinados con CNN. Para ello crearon la arquitectura *AttM-CNN* que fue construida usando como unidades básicas las redes *Inception* y *ResNet* además de aplicar el mecanismo de atención a varias capas. Utilizaron el conjunto de datos *Pornography2M* para entrenar el modelo y obtuvieron un *accuracy* del 97,10 % en vídeos haciendo pruebas usando *Pornography2k*.

Uno de los problemas de todos estos modelos es que no consideran el contexto de la escena del vídeo, en el caso de vídeos subidos a páginas webs este contexto podría ser el texto de la página web o las etiquetas del vídeo. [Kum21] ofrece un sistema capaz de detectar contenido sexual en páginas web, para ello primero usan el clasificador de lenguaje natural *distilBERT* que comprueba si en la página web hay texto tóxico y si hay vídeos con etiquetas. En caso de que la página tenga vídeos con etiquetas usan su modelo LSTM para detectar si las etiquetas se pueden considerar tóxicas. En caso de que se consideren tóxicas, el vídeo pasa por un módulo de detección de edad de *OpenCV* y en caso de que alguien del vídeo tiene menos de 18 años bloquea el vídeo. Para el entrenamiento y pruebas usaron el conjunto de datos *Jigsaw Multilingual Toxic* en la detección de texto tóxico obteniendo una *accuracy* del 98 % y para el modelo de clasificación de etiquetas obtuvieron una accuracy de un 99 % usando como base de datos etiquetas obtenidas de páginas web adultas y Twitter. Otro trabajo que buscaba el contexto de las imágenes fue el de [TCB+21], en este trataban de detectar órganos sexuales en cada imagen. Para ello usaron el modelo detector de objetos *YoloV3* y un conjunto de datos propio formado por más de 20000 imágenes debidamente etiquetadas. Obtuvieron una p*recision* del 97,81 % en detección de órganos sexuales y un 63,63 % de *mean average precision* en detección de material pornográfico.

Con el avance de la visión artificial cada vez hay más técnicas para crear aplicaciones,

por ello encontrar el mejor modelo o clasificador puede ser un reto. [BK21] comparó los resultados al mezclar una red LSTM con tres posibles CNNs. Las redes que eligieron para probar fueron *ResNet50*, *VGG16* y una CNN simple, que entrenaron con el conjunto de datos *PornDbSetTiUnram* del que sacaron 1000 imágenes seguras y 1000 imágenes pornográficas. La red que mejores resultados dio fue *ResNet50* con la que obtuvieron un *accuracy* de un 98 %. Este aumento de opciones también afecta a los conjuntos de datos que pueden ser mucho mayores, como lo es el conjunto de datos que crearon [BRFdA22]. Este conjunto de datos está formado por 127000 vídeos etiquetados de los cuales más de 67000 son vídeos no seguros en los que hay contenido sexual o violento. Su modelo extraía las características visuales usando la red *InceptionV3* y extraían las características de audio usando la red *VGGish*. Una vez extraídas usaban un clasificador para decidir si el contenido no era seguro. Los clasificadores utilizados fueron MLP, LSTM, SVM y KNN. Los mejores resultados los obtuvieron con MLP, así que lo usaron para probar su modelo con el conjunto de datos *Pornograhpy2k* con lo que obtuvieron una *precision* del 90,08 %. Otro enfoque que se probó fue el de clasificar los valores del autoespacio de las imágenes [KR22]. Para calcular dichos valores se utilizaron dos módulos, uno que obtenía los valores mediante conversiones de la imagen y después cálculos, y otro módulo que utiliza una red neuronal que además devuelve una etiqueta. Una vez calculados estos valores y la etiqueta, el módulo de decisión clasifica la imagen o vídeo entre varias categorías. En cuanto a los experimentos, los autores compararon la velocidad entre un equipo humano y el modelo construido, por lo que no pudieron ser muy extensas. Realizaron pruebas con 60 imágenes y 50 vídeos y su modelo obtuvo una precisión del 100 % superando la velocidad humana.

Por último, el modelo que mejores resultados ha dado haciendo las pruebas usando el conjunto de datos *Pornography2k* fue el de [GV22]. Este modelo recibe un vídeo del cual extrae cuatro fotogramas por segundo, que preprocesa y se le aplican técnicas de aumento de datos; a continuación, utilizan la red *Faster RCNN-Inception ResNet V2* con el objetivo de detectar humanos para no utilizar aquellas imágenes en las que no aparezcan. Estas imágenes pasan a una red *ResNet-18* junto con una red neuronal que se encargarán de clasificar la secuencia de fotogramas. Para las pruebas del conjunto de datos *Pornography-800* usaron el modelo completo, mientras que para el conjunto de datos *Pornography2k* no incluyeron el detector de humanos y obtuvieron una *accuracy* del 98,25 % y del 97,15 % respectivamente, aunque este resultado lo obtuvieron utilizando una metodología diferente a la de los autores del conjunto de datos, por lo que la comparación no es justa. Así que sin contar este último artículo el mejor resultado fue [GGCAF21] con un *accuracy* del 97,10 %.

Finalmente en la Tabla 3.1 se resumen los trabajos más importantes del estado del arte.

Tabla 3.1: Artículos más relevantes en detección de contenido sexual

Referencia	**Tamaño del conjunto de datos**	**Arquitectura**	**Resultados**
[FF99]	4867 imágenes	Detección de piel	Pre: 57 %
[AA05]	1620 imágenes	Detección de piel	Rec: 94,32 %
[AdLdS+11]	800 vídeos	STIP + SVM	*Acc: 91,9 %*
[CAJ+14]	800 vídeos	BinBoost + BossaNova	*Acc: 90,9 %*
[GRH+18]	986 imágenes +253 vídeos	Detección de piel	Acc: 80,23 %
[HSWA18]	19756 imágenes	Haar + OpenCV	*Acc: 85 %*
[DFDSB+19]	2000 vídeos	InceptionV3 + AudioVGG + MLP	*Pre: 98,03 %*
[FBGC20]	2000 vídeos	InceptionV3 + AudioVGG + LSTM	*F1-score: 99,00 %*
[SK20]	2000 vídeos	VGG-16 + Mel-Scaled Spectogram + LSTM	*Acc: 92,33 %*
[TBC+20]	2000000 imágenes	MobileNet	Acc: 95 %
[CLH+20]	500000 imágenes	CNN + Mecanismo de atención	Acc: 98,41 %
[GGCAF21]	2000000 imágenes	Inception + ResNet + Mecanismo de atención	Acc: 97,10 %
[Kum21]	-	distilBERT + LSTM + OpenCV	Acc: 98/99 %
[TCB+21]	20000 imágenes	YoloV3	Pre: 97,81 + MAP:63,63 %
[BK21]	2000 imágenes	ResNet50	Acc: 98 %
[BRFdA22]	127000 vídeos	InceptionV3 + VGGish + MLP	*Pre: 90,08 %*
[GV22]	2000 vídeos	Faster RCNN-Inception + ResNet V2 + ResNet-18	*Acc: 97,15 %*

Capítulo 4

Modelo propuesto

En este capítulo se mostrará la metodología que se ha seguido para desarrollar este proyecto, mostrando los pasos que han sido necesarios para crear el modelo. En la primera Sección 4.1 se explicará el flujo de trabajo que se siguió a lo largo del proyecto. Seguido por la Sección 4.2 donde se explica cómo funciona la arquitectura elegida. En la Sección 4.3 se mostrarán los pasos que sigue el preproceso de las imágenes antes de pasar por el modelo. A continuación, en la Sección 4.4 se explicará cómo se realizó la limpieza y preparación del conjunto de datos. En la Sección 4.5 se explicará el algoritmo Grad-Cam que proporciona explicaciones visuales a las decisiones del modelo, seguido por la Sección 4.6 donde se explicarán los métodos de aumento de datos que se probaron durante el entrenamiento del modelo. Por último, en la Sección 4.7 se tratará la manera en la que se entrenó el modelo y en la Sección 4.8 se explican las métricas que se usaron para evaluar el rendimiento del modelo.

4.1. Flujo del trabajo

El primer paso del desarrollo del proyecto fue la limpieza del conjunto de datos y para comprobar la calidad de la limpieza se revisaba la matriz de confusión, especialmente comprobando si había cambios en los falsos positivos. En caso de que no hubiese una mejora significativa, volveríamos a realizar la limpieza del conjunto. A continuación, se revisó la explicabilidad del modelo usando el algoritmo Grad-Cam para comprobar qué características de las imágenes daba más importancia. Una vez revisada la explicabilidad, se comenzó con el entrenamiento del modelo. Se entrenó repetidas veces y se evaluaron los resultados en cada entrenamiento, hasta que obtuvimos resultados satisfactorios. Este flujo de trabajo se puede ver gráficamente en la Figura 4.1.

4.2. Arquitectura del Modelo

El modelo *Open nsfw model* fue creado por Yahoo con el objetivo de ofrecer software libre para detectar imágenes NSFW. Este modelo recibe una imagen y devuelve la probabilidad entre 0 y 1 de que una imagen sea NSFW. Este modelo se creó usando la librería *Caffe* y *CaffeOnSpark*, un *framework* de software libre que permite entrenar modelos de DL en clústers de *Apache Hadoop* y *Apache Spark*.

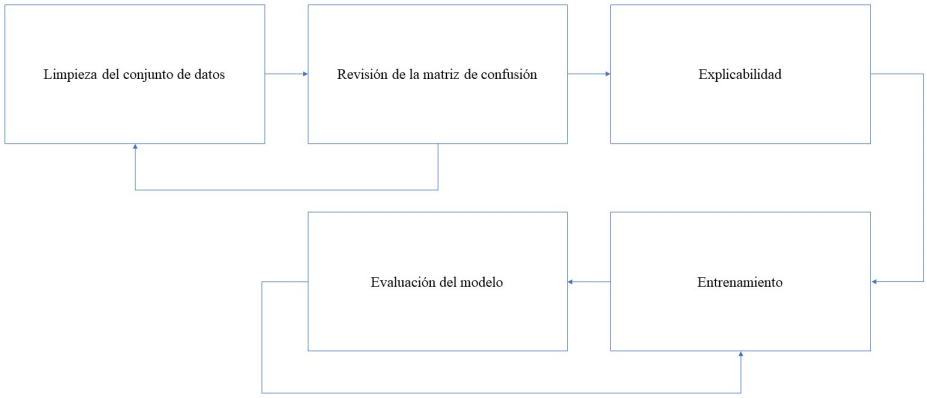

Figura 4.1: Flujo de trabajo

Probaron varias arquitecturas para ver la precisión y velocidad de ejecución de cada una de ellas, y para las redes residuales, además del aumento de datos anterior, se usó el método de aumento de escala. Las arquitecturas probadas fueron MS_CTC, Squeezenet, VGG, GoogLeNet, ResNet-50 y ResNet-50-thin. Los modelos primero fueron entrenados usando el conjunto de datos de ImageNet y a continuación, se sustituyó la última capa por una capa *fully-connected* con dos neuronas. Usando su propio conjunto de datos afinaron los pesos de los modelos. Se dieron cuenta de que el rendimiento de la detección de imágenes NSFW estaba ligado con el rendimiento de los modelos al clasificar imágenes de ImageNet. Tras realizar las mediciones de rendimiento obtuvieron una *accuracy* del 71,75 % con ResNet-50 pero se decantaron por ResNet-50-thin con una *accuracy* del 66,79 % un debido a que requiere menor capacidad de computación.

Se decidió utilizar este modelo por encima de otros por varias razones. Una de las primeras y de mayor peso es el hecho de que sea software libre y por lo tanto podamos utilizarlo abiertamente, además de que es de los pocos modelos que está especializado en la detección de contenido sexual en imágenes. Otra de las razones es que ya existía una versión para Tensorflow 2 [Yun23] que podía adaptarse fácilmente a nuestras necesidades.

Dado que entrenar un modelo de visión artificial de cero conlleva mucho tiempo y recursos computacionales, decidimos aplicar aprendizaje por transferencia en este modelo ya entrenado. Este modelo recibirá los fotogramas clave extraídos del conjunto de datos y los clasificaría individualmente. Los fotogramas que recibe este modelo tendrán que pasar por un preprocesamiento antes de clasificarlos. Este modelo servirá de base y, posteriormente, se volverán a entrenar los pesos de sus últimas capas con nuestro conjunto de datos.

4.2.1. ResNet

Las redes neuronales se crearon debido a la dificultad que supone entrenar una red neuronal profunda. Por ello, [HZRS15] propone reformular las capas como funciones de aprendizaje residual que hace referencia a los valores de entrada. Usan bloques formados por una capa convolucional y un atajo. Estos atajos permiten pasar información de entrada a la salida de la capa convolucional. Estos bloques son conectados secuencialmente y se dividen en bloques básicos y bloques de cuello de botella, que se diferencian en que deben modificar el tamaño de los valores que pasan por el atajo sin perder la precisión. La arquitectura ResNet50 está formada por 50 capas que sigue la estructura mostrada en la Figura 4.2.

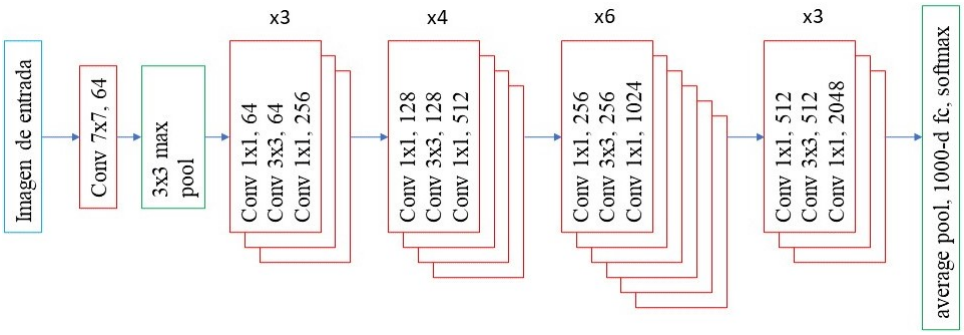

Figura 4.2: Diseño de ResNet

Estas arquitecturas están diseñadas de manera que es fácil entrenarlas con un conjunto de datos mayor y luego afinarlas para una tarea más específica mediante el aprendizaje por transferencia.

4.3. Preproceso

Para el correcto funcionamiento del modelo *Open nsfw model*, las imágenes deben pasar por una etapa de preproceso. La implementación de Tensorflow 2 ofrece dos posibles preprocesamientos: la versión por defecto utilizada en la implementación original de Yahoo y una versión más simple con menos pasos. Tras realizar distintas pruebas de velocidad y precisión decidimos utilizar la versión original realizando un cambio al último paso. Este cambio reduce ligeramente la precisión del modelo, pero aumenta la velocidad de entrenamiento considerablemente. El primer paso del preproceso consiste en redimensionar la imagen a 256 píxeles por 256 píxeles y, posteriormente, es recortada a un tamaño de 224 píxeles por 224 píxeles. Por último, nuestro preproceso aplica una función de preproceso de la librería de ImageNet de Keras a las imágenes. Esta función convierte las imágenes de RGB a BGR y resta el valor medio de los píxeles de VGG a los píxeles de la imagen. En el preproceso de Yahoo restaban el valor medio de su conjunto de datos a cada píxel de las imágenes. El método implementado era más lento y por ello lo sustituimos. Este preproceso dificulta la explicabilidad del modelo debido al cambio de colores que provoca en la imagen.

4.4. Preparación del Conjunto de Datos

Se suele considerar el conjunto de datos de entrenamiento como la parte más importante de una IA, por lo que preparar el conjunto de datos sería crucial para este proyecto. Como base para el conjunto de datos usamos el conjunto de datos *Pornography2k* [MAP+16]. Este conjunto de datos está formado por 2000 vídeos, 1000 de ellos seguros y 1000 pornográficos. Dado que el modelo se entrena con imágenes, se extrajeron los fotogramas clave de cada vídeo usando la herramienta ffmpeg.

Una vez hecho esto pudimos realizar las primeras pruebas con el modelo *Open nsfw model*. Estas pruebas clasificaron todas las imágenes y, usando un pequeño script, separamos los falsos negativos, es decir, las imágenes que se clasificaron como seguras cuando no lo eran. Entre estas imágenes había falsos negativos reales, pero también imágenes que no

mostraban ningún tipo de contenido sexual. Esto se debe a que algunos vídeos tenían introducciones o títulos que no tienen contenido sexual, pero como forman parte de vídeos sexuales esos fotogramas se clasifican como tal. Para solucionar esto, había que eliminar todos estos fotogramas del conjunto de datos pornográfico.

Los primeros en ser eliminados fueron los fotogramas de un color plano, ya sea todo negro o todo blanco; estos suelen ser el primer o último fotograma de cada vídeo. Una vez eliminados estos fotogramas se continuó con los fotogramas en los que sólo aparecía texto, incluso aunque el texto haga referencia a algo relacionado con el contenido sexual. A continuación, se revisaron todas las imágenes pornográficas individualmente para asegurarse de que todas ellas mostraban contenido sexual. Algunas de estas imágenes eran primeros planos de mobiliario o en paisajes abiertos, lo que hacía que el modelo tomara esas características como pornográficas. Para solucionar esto, eliminamos aquellas imágenes que podían dar información incorrecta.

Finalmente se empleó la librería *Grad-Cam*, que tiene como objetivo principal conocer las características de la imagen que han llevado al modelo a su decisión final. El algoritmo se aplicó en un subconjunto de datos, donde se pudo observar que se centraba en las características sexuales de la imagen en vez de en otros aspectos. Pero se observó que muchas imágenes clasificadas como contenido sexual podían dar información incorrecta al futuro entrenamiento del modelo.

4.5. Localización basada en gradiente

Para comprobar que el modelo clasifica las imágenes utilizando las características sexuales de la imagen utilizamos el algoritmo *Grad-Cam*. Pasamos todas las imágenes que clasificaba como positivas por este algoritmo. Primero revisamos los verdaderos positivos, donde pudimos ver que las zonas más marcadas del mapa de calor solían corresponder con las características más sexuales de la imagen. A continuación, revisamos los falsos positivos con el objetivo de comprobar que no existieran sesgos. Entre los falsos positivos, no encontramos sesgos más allá de que el tipo de imagen que más fallaba eran las que se suelen considerar de difícil clasificación (escenas de playa, lucha libre, sumo, bebés, etc.).

Esto demuestra que el modelo se centra en las características sexuales y sólo se ve afectado por las mismas a la hora de clasificar una imagen. A continuación, en la Figura 4.3 podemos ver cuatro imágenes de ejemplo que han pasado por el algoritmo Grad-Cam, dos imágenes fáciles y dos imágenes difíciles de clasificar.

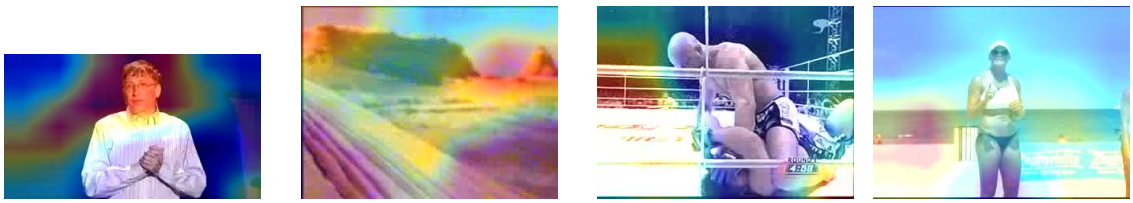

Figura 4.3: Cuatro ejemplos de uso de la algoritmo Grad-Cam

4.6. Aumento de Datos

Como ya se ha comentado con anterioridad, se suele considerar el conjunto de datos de entrenamiento como la parte más importante de una IA. Pero muchas veces no se tienen suficientes datos para realizar un correcto entrenamiento del modelo. Para solucionar este problema y con el objetivo de aumentar la capacidad de generalización del modelo, se utilizan técnicas de aumento de datos. En el caso de conjuntos de datos de imágenes, existen muchas opciones para aumentar la cantidad de datos y en este proyecto se probaron varias. Este aumento de datos se realiza como un paso adicional en la etapa de preproceso, pero sólo para las imágenes que se usan al realizar el entrenamiento.

Los métodos de aumento de datos probados a lo largo de este trabajo fueron: *horizontal flip* (en español, volteo horizontal), contraste aleatorio, brillo aleatorio, rotación aleatoria y zoom aleatorio. Las pruebas que se realizaron fueron con los métodos individualmente y también combinándolos entre ellos, permitiendo varias formas de usarlos. Estos métodos modifican la imagen ligeramente, generando nuevas imágenes con las que el modelo puede entrenar, lo cual debería aumentar la precisión del modelo al haber sido entrenado con mayor cantidad de imágenes.

4.7. Entrenamiento del modelo con el conjunto de datos final

El entrenamiento del modelo se llevó a cabo usando como base el *Open nsfw model*. Utilizando técnicas de aprendizaje por transferencia congelamos todas las capas menos las últimas capas, siendo estas las únicas que entrenaríamos. En un primer momento, el entrenamiento se realizó entrenando las cuatro últimas capas aplicando varias combinaciones de aumentos de datos.

Después, escogimos los métodos que mejores resultados habían dado y el modelo fue entrenado utilizando el método de división entre datos de entrenamiento y datos de prueba propuesto por [MAP+16]. Este método consistía en dividir los 2000 vídeos en dos grupos de 1000 vídeos cada uno, uno para entrenamiento y otro para pruebas. Era importante que se dividiesen los vídeos y no las imágenes debido a que, si usaba parte de los fotogramas de un vídeo para entrenar y otra parte para realizar las pruebas, estas pruebas no serían realistas debido a que ya habría visto fotogramas de ese mismo vídeo. Además, [MAP+16] incluyó cinco subconjuntos de datos con los que realizar el entrenamiento y pruebas. Cada uno era un archivo de texto con los 1000 vídeos que correspondían a entrenamiento y pruebas.

Más tarde, utilizando el mejor método de aumento de datos, se realizaron entrenamientos descongelando las últimas 30 capas y entrenamientos añadiendo una capa *fully-connected* de 512 neuronas antes de la capa de salida.

4.8. Evaluación del modelo

Uno de los pasos más importantes de la creación de un modelo es su evaluación. Para ello se utilizan distintas métricas que indican el rendimiento del modelo. Al realizar una predicción del conjunto de datos propuesto se obtendrá la matriz de confusión con

cuatro tipos de resultados: los verdaderos positivos (TP), los falsos positivos (FP), los falsos negativos (FN) y los verdaderos negativos (TN). Estos indican respectivamente el número de predicciones sexuales clasificadas correctamente, el número de predicciones sexuales clasificadas incorrectamente, el número de predicciones seguras clasificadas incorrectamente y el número de predicciones seguras clasificadas correctamente. La métrica con la que se van a mostrar los resultados es la *accuracy* que se calcula como se muestra en la Ecuación 4.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Otras de las métricas más comunes son la *precision*, el *recall* y el *F1 score*, que se calculan como se puede ver en las Ecuaciones 4.2, 4.3 y 4.4.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$F1score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4.4}$$

Las métricas permiten determinar si se está cumpliendo el objetivo del entrenamiento, para poder iterar sobre ello. Finalmente se obtuvieron los resultados del mejor modelo.

Capítulo 5

Experimentos y Resultados

Para la realización de los experimentos de este proyecto, el modelo se entrenaba con el conjunto de datos y se tomaba el valor del *accuracy test* para las comparaciones. Este valor indica el porcentaje de imágenes de prueba que ha clasificado correctamente. La primera Sección 5.1 explica las pruebas que se realizaron en primer lugar, dividiendo aleatoriamente el conjunto de datos. Seguido por la Sección 5.2 donde se muestran los resultados de los experimentos realizados con los subconjuntos de datos [MAP+16]. A continuación, en la Sección 5.3 se explican los últimos experimentos realizados utilizando el conjunto de datos limpiado completamente. Y por último, en las Secciones 5.4 y 5.5 se mostrarán los resultados y la predicción usando el mejor modelo obtenido.

5.1. Primeros Experimentos

Tras realizar las primeras limpiezas de datos comenzamos a entrenar el modelo usando distintas combinaciones de métodos de aumento de datos, como se muestra en la Tabla 5.1, donde usaremos el acrónimo Configuración para referirnos a las configuraciones. Al preproceso de las imágenes de entrenamiento se le añadía un paso de aumento de datos. Los métodos de aumento de datos pueden recibir un parámetro indicando el valor máximo en el que puede llegar a modificar la imagen, en caso de que no se indique es porque estaba usando el valor por defecto. En el caso del método *flip* siempre se utilizó la opción *horizontal*. Los datos se dividieron aleatoriamente entre datos de entrenamiento y de validación, manteniendo una relación del 80/20 % respectivamente. Se realizaron 10 épocas de entrenamiento, utilizando un tamaño de *batch* de 32, el optimizador Adam con una tasa de aprendizaje del 0,001 y se apuntaron la duración del entrenamiento, el *accuracy* y el *accuracy test*, siendo este último el que tomamos en consideración para las comparaciones. Se debe notar que el tiempo no siempre es representativo, ya que la primera vez que se cargan las imágenes en memoria tarda más que las siguientes. Los resultados se pueden observar en la Tabla 5.2 y para ver cada prueba con más detalle se pueden ver en el Capítulo del Anexo A.

5.2. Experimentos Usando los Cinco Subconjuntos de Datos

Una vez realizadas estas primeras pruebas, decidimos quedarnos con los diez mejores resultados. Pero debido a que varios de los mejores resultados eran del mismo método

variando el parámetro, decidimos elegir únicamente los que daban el mejor resultado para cada método. Finalmente, también decidimos no usar el método de *contrast* debido a los malos resultados que había mostrado individualmente, dejando así sólo los diez mejores métodos. Para las siguientes pruebas, utilizamos la división de imágenes recomendada por [MAP+16]. En estas se dividen los vídeos en dos grupos de 1000, un grupo para el entrenamiento y otro para la validación, cada uno formado por 500 vídeos seguros y 500 vídeos pornográficos. [MAP+16] ofrece cinco de estas divisiones en subconjuntos de datos, cada uno de ellos compuesto por dos conjuntos de 1000 vídeos diferentes.

Una vez elegidos los métodos y preparadas las imágenes de cada subconjunto, realizamos los entrenamientos. Además, realizamos entrenamientos de cada subconjunto sin métodos de aumentos de datos para poder comprobar si mejoran los resultados. Por último, en la Tabla 5.3 se puede ver la media del *accuracy test* de todos los subconjuntos para cada método probado. De esta manera sabemos que el método *flip* es el que mejores resultados da.

5.3. Experimentos con el Conjunto de Datos Final

Una vez encontrado el mejor método de aumento de datos, se volvió a realizar una limpieza de datos más estricta, dado que el modelo había mejorado lo suficiente para que hubiese una reducción significativa de falsos positivos en las predicciones. Después de esto, el conjunto de datos no se volvió a modificar. Con esta nueva versión del conjunto de datos, entrenamos el modelo usando el método de aumento de datos *flip* y sin usar ningún método de aumento de datos. Cada uno de estos entrenamientos se realizaron cuatro veces, entrenando las cuatro últimas capas y las treinta últimas capas, y se probó a entrenar añadiendo una capa *fully connected* de 512 neuronas. Además, para estas pruebas en vez de las 10 épocas anteriores, se hacen 30 épocas que se detienen si el cambio de precisión es muy bajo. Los resultados individuales se pueden encontrar en el Capítulo del Anexo A. Además en la Tabla 5.4 podemos observar los resultados medios de los subconjuntos de cada uno de los métodos probados.

5.4. Resultados

Los experimentos nos han permitido observar una mejoría en el rendimiento de los modelos según los métodos usados. Para comparar, podemos observar en la Tabla 5.4 la *accuracy* del modelo *Open nsfw model* antes del ajuste de pesos. En las pruebas realizadas con los subconjuntos de datos se puede ver que el mejor resultado se consigue utilizando el método de aumento de datos de *flip*, así que se siguió usando para las pruebas con el conjunto de datos final. En estas, los resultados muestran que la opción que mejor rendimiento da es el entrenamiento de las 30 últimas capas del modelo sin aumento de datos, seguido de cerca el entrenamiento de las cuatro últimas capas del modelo con la capa *fully connected* y el método de aumento de datos de *flip*.

5.5. Predicción usando el mejor modelo

Una vez se encontró el mejor modelo se realizó una predicción sobre el conjunto de datos completo. Las métricas obtenidas para esta predicción fueron una *accuracy* del 97,76 %,

una *precision* del 97,82 %, un *recall* del 97,91 % y una *F1 score* del 97,86 %. La matriz de confusión que se obtuvo se muestra en la Figura 5.1.

Tabla 5.1: Configuraciones de los métodos

Configuración	**Método de aumento de datos**	**Parámetros**
Conf 1	Zoom	0,15
Conf 2	Zoom	Por defecto
Conf 3	Flip	Por defecto
Conf 4	Zoom	0,1
Conf 5	Zoom + Flip	0,15 + Por defecto
Conf 6	Rotation	0,1
Conf 7	Zoom + Rotation	0,15 + 0,1
Conf 8	Zoom	0,3
Conf 9	Rotation	Por defecto
Conf 10	Zoom + Flip	Por defecto + Por defecto
Conf 11	Zoom	0,3
Conf 12	Flip + Rotation	Por defecto + 0,1
Conf 13	Zoom + Flip + Rotation	0,15 + Por defecto + 0,1
Conf 14	Brightness	0,1
Conf 15	Zoom + Contrast	Por defecto + Por defecto
Conf 16	Brightness	0,3
Conf 17	Brightness	Por defecto
Conf 18	Contrast + Brightness	Por defecto + Por defecto
Conf 19	Flip + Brightness	Por defecto + 0,1
Conf 20	Contrast	Por defecto
Conf 21	Zoom + Brightness	0,15 + 0,1

Tabla 5.2: Resultados de entrenamiento de las primeras pruebas

Método de aumento de datos	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Conf 1	50	95,01	95,33
Conf 2	50	94,88	95,06
Conf 3	25	95,51	94,73
Conf 4	61	95,31	94,65
Conf 5	64	93,93	93,89
Conf 6	49	94,27	93,52
Conf 7	83	93,08	93,44
Conf 8	53	84,55	92,96
Conf 9	66	93,54	92,77
Conf 10	53	95,01	92,52
Conf 11	60	93,48	91,83
Conf 12	68	92,95	91,71
Conf 13	72	92,66	91,59
Conf 14	36	89,87	90,28
Conf 15	52	89,14	90,28
Conf 16	35	88,31	89,65
Conf 17	35	89,24	88,95
Conf 18	36	88,31	84,70
Conf 19	36	89,14	83,28
Conf 20	34	89,97	80,72
Conf 21	53	88,33	80,46

Tabla 5.3: Media de los resultados

Método de aumento de datos	**Media Accuracy Test (%)**
Sin aumento de datos	92,01
Conf 1	92,80
Conf 3	93,27
Conf 5	93,20
Conf 6	92,86
Conf 7	92,25
Conf 12	92,82
Conf 13	91,72
Conf 14	91,34
Conf 19	91,14
Conf 21	91,85

Tabla 5.4: Media de los entrenamientos realizados con el conjunto de datos final

Tabla	Número de capas entrenadas	Aumento de datos	Fully Connected	Media Accuracy Test (%)
A.6	4	No	No	95,16
A.7	30	No	No	95,84
A.8	4	Flip	No	94,72
A.9	30	Flip	No	95,22
A.10	4	No	Sí	94,92
A.11	30	No	Sí	92,95
A.12	4	Flip	Sí	95,35
A.13	30	Flip	Sí	94,42

Tabla 5.5: Resultados comparativos

Modelo	Media Accuracy Test (%)
Open nsfw model	89,22
Conjunto de datos no final con Flip	93,27
Conjunto de datos final entrenando 4 capas con Fully Connected y Flip	95,35
Conjunto de datos final entrenando 30 capas	95,84

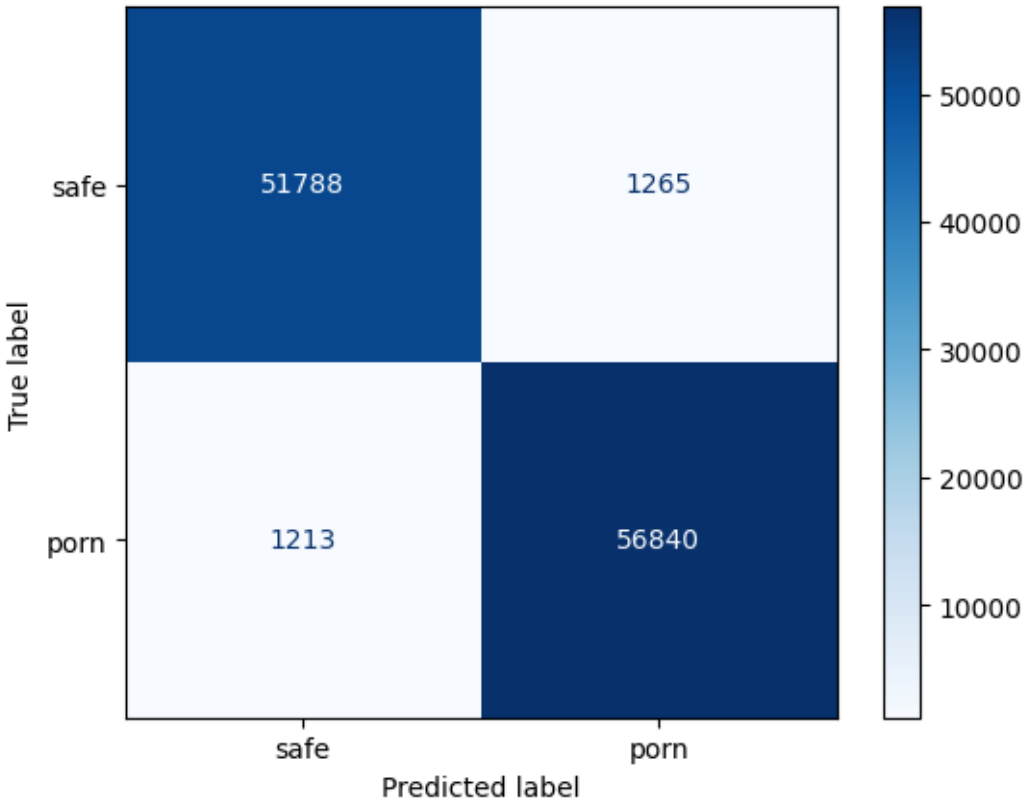

Figura 5.1: Matriz de confusión

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

Este proyecto buscaba crear un modelo de visión artificial capaz de clasificar imágenes con contenido sexual de manera rápida y consistente. Los resultados fueron satisfactorios, pero dado que los artículos nombrados en el capítulo 3 que usan este mismo conjunto de datos dan resultados sobre clasificación de vídeos en vez de clasificación de imágenes, no se pueden realizar comparaciones justas. Utilizando como base el modelo *Open nsfw model* de Yahoo y ajustando sus pesos mediante técnicas de aprendizaje por transferencia, nuestro modelo obtuvo un *accuracy* medio del 95,84 %. Para la realización de este entrenamiento, se limpió el conjunto de datos de [MAP+16] formado por 2000 vídeos, al que se le extrajeron los fotogramas clave de cada vídeo para formar un conjunto de datos de más de 120000 imágenes. Se ve cómo por una buena limpieza del conjunto de datos se pueden obtener buenos resultados incluso utilizando una arquitectura antigua.

Se realizaron pruebas con varias combinaciones de métodos de aumento de datos, hasta llegar a la conclusión que el mejor método es *flip*, pero los mejores resultados se obtienen sin utilizar ningún método de aumento de datos. Esto creemos que se trata por la limpieza realizada al conjunto de datos, ya que al ser tan estricta los cambios que provocan los métodos de aumento de datos sólo dificultan el proceso de aprendizaje del modelo.

Personalmente, ha sido todo un desafío realizar un proyecto de estas magnitudes. Realizar todos los pasos necesarios para un trabajo así me ha permitido aprender el proceso por el que pasan estos proyectos para poder conseguir los resultados que se buscan. También me ha permitido adentrarme en el mundo de la visión artificial y la investigación en el ámbito de la informática.

Este tipo de herramientas están tomando cada vez más protagonismo, por el rápido avance que está teniendo la IA durante estos últimos años. Especialmente, cuando se trata de una herramienta que tiene como objetivo ayudar a enfrentarse a delitos de esta índole. Por ello, me alegro de haber podido trabajar en este proyecto y sobre todo de poder ayudar con un propósito tan bien intencionado.

6.2. Trabajo Futuro

Algunos posibles trabajos futuros que pueden señalarse son los siguientes:

- Probar otras arquitecturas más actuales y potentes, como los ViT.
- Aumentar el conjunto de datos o continuar su limpieza.
- Probar más métodos de aumento de datos, combinaciones y cambiar sus parámetros.
- Cambiar el número de capas entrenables a números mayores.

Capítulo 7

Introduction

7.1. Motivation

Over the last 65 years, the internet has been growing faster and faster. Initially created with military interests, it has evolved to become what it is today. The popularity of the internet has led to increase it's speed, allowing the rapid exchange of images and videos.

Most of this content is safe and can be seen by everyone, but not all of it. Some of this content may not be safe or may even be illegal, such as child pornography. Due to the amount of content that is constantly being uploaded to the internet, it is impossible to check manually that everything is legal. Because of that, most platforms use algorithms that whether the content uploaded is safe to upload.

These algorithms usually use artificial intelligence and computer vision to detect if content can be allowed on those platforms. Currently, these technologies are constantly evolving as they are being used in all kinds of applications, so finding the best model is very important for these algorithms.

Sexual content detection algorithms are very common in social networks, but they are not used as much in law enforcement agencies. These algorithms could help the police in the search of illegal content, due to the fact that they need to find this content in large numbers of files within limited time. Due to the time limit in these cases, searching through all this content manually is not possible, so using an algorithm that is capable of locating those files quickly would be very beneficial.

A well-trained and fast enough computer vision model would be useful for law enforcement agencies, and that is why we are doing this project.

7.2. Context

This Final Degree Project is part of a research project called Novel Strategies to Fight Child Sexual Exploitation and Human Trafficking Crimes and Protect their Victims - HEROES, approved by the European Commission within the Horizon 2020 Framework Programme (call H2020-SU-SEC-2020) under grant agreement number 101021801 and in which the GASS Group of the Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, https://gass.ucm.es, group 910623 of the catalogue of research groups recognised by the UCM).

In addition to the Universidad complutense de Madrid, 21 organisations from 17 countries are participating in HEROES: 11 from EU countries (Austria, Belgium, Bulgaria, France, Greece, Ireland, Latvia, Lithuania, Portugal, Spain, United Kingdom), 1 associated country (Switzerland) and 5 third countries (Bangladesh, Brazil, Colombia, Peru, Uruguay). These entities are: University of Kent (UK), The Free University of Brussels (Belgium), The French National Research Institute for Digital Science and Technology - INRIA (France), Center for Security Studies - KEMEA (Greece), International Centre for Migration Policy Development - ICMPD (Austria), International Center for Missing and Exploited Children - ICMEC (Switzerland), IDENER Research & Development Agrupación de Interés Económico (Spain), Athena Research Center - ARC (Greece), Trilateral Research and Consulting (United Kingdom), Centre for Women and Children Studies - CWCS (Bangladesh), Center Against Human Trafficking and Exploitation - KOPZI (Lithuania), Portuguese Association for Victim Support - APAV (Portugal), Fundación Renacer (Colombia), The Greek Council for Refugees - GCR (Greece), Brazilian Association for the Defense of Children of Children and Youth - ASBRAD (Brazil), Hellenic Police (Greece), Latvia National Police (Latvia), General Directorate for the Fight against Organized Crime (Bulgaria), Dirección General de la Policía - DGP (Spain), Federal Police (Brazil), Federal Highway Police (Brazil), Secretaria de Inteligencia Estratégica de Estado - Presidencia de la Republica Oriental del Uruguay (Uruguay).

More information is available at:

https://cordis.europa.eu/project/id/101021801
https://heroes-fct.eu

7.3. Object of the Investigation

The objective of this Bachelor's thesis is to design and implement a computer vision model capable of classifying images with sexual content. This model will allow to review large amounts of images in short time looking for not safe content, helping law enforcement agencies to find this content. Therefore, a fast architecture with great performance must be used and we must prepare a data set that allows to train it. To find the best parameters and training methods, extensive tests will be carried out using the standard test system, allowing us to compare it to other similar models. Similarly, this projects aims to understand how computer vision models works, their structure and training, as well as the difficulties of creating one.

7.4. Workplan

The development of this project has been carried out in three phases:

1. **Research**: During the first months of the project, the learning phase of the theoretical framework and state of the art was carried out. Firstly, a meeting was held in which the objectives and knowledge needed to carry out this project were explained. We agreed to perform weekly meetings to monitor the project, resolve doubts and, if necessary, explain the tools that were being used. Some of these tools were focused on the preparation of project documentation, such as Mendeley

Reference Manager. Once we had the needed knowledge and other state of the art models had been studied, we began developing.

2. **Development**: After we obtained the needed knowledge, project development started. The research phase never stopped, because we had to continue investigating the libraries that were going to be used. Among this libraries we could find Python's Tensorflow and Keras. Also during this phase, the data set that was going to be used for training was being prepared, for this, it was reviewed and cleaned manually to ensure the best results. Both model and original data set were obtained from articles that we read during the reasearch phase.

3. **Experimentation**: The experimentation process began once we made enough progress in the development of this project. During this phase, we perform predictions and trained the model to compare the results between versions of the data set, methods and training parameters. During this phase, the research and development phases, necessary to learn and develop tools and methods used during this phase, such as the Grad-Cam tool that offers explainability to artificial vision models, did not stop either.

7.5. Struture of the Work

The rest of the work is organised in 6 chapters with the structure explained below:

Chapter 2 explains the history of artificial intelligence, as well as a theoretical explanation of the operation of its most common architectures. Among this architectures are machine learning, deep learning and computer vision.

Chapter 3 chronologically describes the algorithms and models that have been used in order to detect and classify with sexual content. The results they obtained in their tests and information about the size and content of the data sets they used in each project are also shown.

Chapter 4 shows the contributions of this project. The model that will be trained, the data set chosen to train it and how it will be trained are presented. It will also explain the preprocessing and data augmentation step that is performed in the images before they are fed to the model for training. Finally, it will be explained how Grad Cam has been used.

Chapter 5 describes the procedure that was followed to carry out the training and presents the results obtained for each training.

Chapter 6 shows the conclusions obtained after carrying out this project and the possible future work.

Chapters 7 and 8 are the English translations of the Introduction and the Conclusions.

Finally, we show all the results of all the trainings in the Chapter of the Annex A.

Capítulo 8

Conclusions and Future Work

8.1. Conclusions

This project sought to create a computer vision model capable of detecting images with explicit content quickly and consistently. The results were satisfactory, but since the articles named in chapter 3 using this data set give results on video classification instead of image classification, we cannot make a fair comparison.Using Yahoo's Open nsfw model as the pretrained model and fine-tuning its weights using transfer learning techniques, our model obtained an average accuracy of 95.84 %. To carry out this training, we cleaned the [MAP+16] data set consisting of 2000 videos, from which the key frames of each video were extracted forming a data set of 120,000 images. It can be seen how by a good cleaning of the data set, we can obtain good results even using an old architecture.

Tests were carried out with various combinations of data augmentation methods, until we reached the conclusion that the best method is flip, but the best results were obtained without using any data augmentation method. We believe that this is due to the strict cleaning carried out on the data set, the changes caused by the data augmentation methods only hinder the learning process of the model.

Personally, it has been a real challenge to carry out a project of these magnitudes. Carrying out all the necessary steps for a job like this has allowed me to learn the process that these projects go through in order to achieve the results that are sought. It has also allowed me to delve into the field of computer vision and investigation in computing.

This types of tools are standing more and more, due to the rapid progress that *Artificial Intelligence* (AI) has had during recent years. Especially, when it comes to a tool that tries to help to deal with crimes of this nature. For this reasons, I am glad to have been able to work on this project and above all to be able to help with such noble purpose.

8.2. Future Work

As possible future works, there are proposed the following ones:

- Try other more prevalent and powerful, such as ViT.
- Increment the size of the data set or keep cleaning it.

- Test more data augmentation methods, combinations and change its parameters.
- Change the number of trainable layers to larger numbers.

Apéndice A

Resultados de entrenamientos

A.1. Resultados de entrenamientos

A continuación, se mostrarán en tablas los resultados de todos los entrenamientos realizados a lo largo del proyecto. Los resultados para cada subconjunto se pueden ver en las Tablas A.1, A.2, A.3, A.4 y A.5, donde usaremos Conf para referirnos a las configuraciones.

Se pueden observar los resultados de cada método para cada subconjunto en las Tablas A.6, A.7, A.8, A.9, A.10, A.11, A.12 y A.13.

Tabla A.1: Resultados de entrenamiento del subconjunto 1

Método de aumento de datos	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Sin aumento de datos	71	83,93	89,82
Conf 1	31	94,60	93,27
Conf 3	29	95,19	94,01
Conf 5	43	94,20	93,58
Conf 6	31	92,66	93,36
Conf 7	38	92,53	92,87
Conf 12	34	92,75	93,34
Conf 13	47	92,40	93,06
Conf 14	23	89,33	92,33
Conf 19	31	89,39	92,34
Conf 21	32	88,62	92,30

Tabla A.2: Resultados de entrenamiento del subconjunto 2

Método de aumento de datos	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Sin aumento de datos	29	95,06	91,86
Conf 1	70	95,03	91,81
Conf 3	21	95,32	93,35
Conf 5	58	94,87	93,02
Conf 6	70	93,16	92,62
Conf 7	41	92,68	89,99
Conf 12	55	93,15	92,69
Conf 13	47	92,75	91,66
Conf 14	51	89,91	90,58
Conf 19	26	89,93	91,00
Conf 21	42	89,09	91,15

Tabla A.3: Resultados de entrenamiento del subconjunto 3

Método de aumento de datos	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Sin aumento de datos	41	95,41	92,24
Conf 1	30	94,55	92,52
Conf 3	30	95,15	92,33
Conf 5	37	94,52	92,60
Conf 6	30	93,96	91,92
Conf 7	58	93,62	92,22
Conf 12	30	93,84	92,34
Conf 13	36	93,60	90,97
Conf 14	33	90,02	90,58
Conf 19	23	89,94	90,27
Conf 21	36	93,57	91,92

Tabla A.4: Resultados de entrenamiento del subconjunto 4

Método de aumento de datos	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Sin aumento de datos	31	94,72	93,07
Conf 1	31	94,16	93,01
Conf 3	30	94,68	93,28
Conf 5	34	94,07	93,31
Conf 6	37	93,46	93,22
Conf 7	37	93,10	93,12
Conf 12	40	93,33	92,74
Conf 13	37	93,04	89,89
Conf 14	26	89,14	91,51
Conf 19	33	89,17	90,05
Conf 21	29	88,28	91,73

Tabla A.5: Resultados de entrenamiento del subconjunto 5

Método de aumento de datos	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Sin aumento de datos	30	94,67	93,09
Conf 1	29	93,92	93,39
Conf 3	19	94,54	93,42
Conf 5	29	93,96	93,52
Conf 6	29	93,19	93,18
Conf 7	48	92,88	93,05
Conf 12	31	93,17	93,00
Conf 13	39	92,80	93,05
Conf 14	33	88,98	91,73
Conf 19	23	88,91	92,08
Conf 21	31	88,24	92,17

Tabla A.6: Resultados de entrenamiento de 4 capas entrenables sin aumento de datos

Subconjunto	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Subds 1	34	96,31	95,61
Subds 2	43	96,70	94,89
Subds 3	39	96,55	94,68
Subds 4	37	96,31	95,27
Subds 5	30	96,10	95,35
Media Accuracy Test			95,16

Tabla A.7: Resultados de entrenamiento de 30 capas entrenables sin aumento de datos

Subconjunto	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Subds 1	36	99,25	96,06
Subds 2	45	99,53	95,77
Subds 3	47	99,50	95,72
Subds 4	31	99,24	95,95
Subds 5	42	99,43	95,70
Media Accuracy Test			95,84

Tabla A.8: Resultados de entrenamiento de 4 entrenables capas con *flip*

Subconjunto	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Subds 1	27	96,13	95,34
Subds 2	29	96,60	94,03
Subds 3	21	96,44	93,35
Subds 4	29	96,34	95,45
Subds 5	43	96,43	95,43
Media Accuracy Test			94,72

Tabla A.9: Resultados de entrenamiento de 30 capas entrenables con *flip*

Subconjunto	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Subds 1	36	99,16	94,10
Subds 2	44	99,24	95,23
Subds 3	33	99,31	95,07
Subds 4	44	99,47	96,09
Subds 5	44	99,38	95,65
Media Accuracy Test			95,22

Tabla A.10: Resultados de entrenamiento de 4 capas entrenables con la capa *fully connected*

Subconjunto	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Subds 1	23	95,97	95,07
Subds 2	25	96,44	94,92
Subds 3	26	96,23	94,97
Subds 4	32	96,29	95,17
Subds 5	25	96,02	94,48
Media Accuracy Test			94,92

Tabla A.11: Resultados de entrenamiento de 30 capas entrenables con la capa *fully connected*

Subconjunto	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Subds 1	41	99,05	94,71
Subds 2	35	99,06	95,05
Subds 3	23	98,31	91,46
Subds 4	27	98,70	94,97
Subds 5	42	99,10	88,60
Media Accuracy Test			92,95

Tabla A.12: Resultados de entrenamiento de 4 capas entrenables con la capa *fully connected* y *flip*

Subconjunto	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Subds 1	36	95,85	95,67
Subds 2	43	96,53	94,99
Subds 3	39	96,34	95,20
Subds 4	28	95,67	95,30
Subds 5	43	95,79	95,62
Media Accuracy Test			95,35

Tabla A.13: Resultados de entrenamiento de 30 capas entrenables con la capa *fully connected* y *flip*

Subconjunto	**Tiempo en minutos**	**Accuracy (%)**	**Accuracy Test (%)**
Subds 1	45	98,99	93,48
Subds 2	38	98,81	92,80
Subds 3	27	98,49	94,04
Subds 4	30	98,68	95,78
Subds 5	47	98,90	96,01
Media Accuracy Test			94,42

Bibliografía

[AA05] R. Ap-Apid. An algorithm for nudity detection, 2005.

[AdLdS+11] S. Avila, A. da Luz, F. de Souza, M. Coelho, E. Valle, and A. Araújo. Content-based filtering for video sharing social networks segmentation of the coronary artery view project sensitive content detection in cartoons view project content-based filtering for video sharing social networks, 2011.

[ANFR+20] M. Al-Nabki, E. Fidalgo, Vasco-Carofilis R., Jañez-Martino F., and Velasco-Mata J. Evaluating performance of an adult pornography classifier for child sexual abuse detection. 5 2020.

[Any23] R. Anyoha. The history of artificial intelligence. https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/, 2023.

[ATC+13] S. Avila, N. Thome, M. Cord, E. Valle, and A. Araújo. Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117:453–465, 2013.

[BK21] I. Bintang and G. Kusuma. Porn detection in a video streaming using hybrid network of cnn and lstm. *International Journal of Engineering Trends and Technology*, 69:248–255, 11 2021.

[BRFdA22] A. Barbosa Raposo and S. Fontes de Avila. Sensitive content detection in video with deep learning, 2022.

[CAJ+14] C. Caetano, S. Avila, S. Jamil, F. Guimarães, S. Guimarães, and A. Araújo. Pornography detection using bossanova video descriptor exploring feature distribution to create mid-level representations: A case study in human action recognition view project sensitive content detection in cartoons view project pornography detection using bossanova video descriptor, 2014.

[CLH+20] J. Chen, G. Liang, W. He, C. Xu, J. Yang, and R. Liu. A pornographic images recognition model based on deep one-class classification with visual attention mechanism. *IEEE Access*, 8:122709–122721, 2020.

[DDS+09] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. IEEE, 2009.

[DFDSB+19] P. De Freitas, G. Dos Santos, A. Busson, A. Guedes, and S. Colcher. A baseline for nsfw video detection in e-learning environments. pages 357–360. Association for Computing Machinery, Inc, 10 2019.

[Equ23] Equipo redactor interno de DataScientest. Inteligencia artificial : definición, historia, usos, peligros. https://datascientest.com/es/inteligencia-artificial-definicion, 2023.

[FBGC20] P. Freitas, A. Busson, A. Guedes, and S. Colcher. A deep learning approach to detect pornography videos in educational repositories. pages 1253–1262. Sociedade Brasileira de Computacao - SB, 11 2020.

[FF99] D. Forsyth and M. Fleck. Automatic detection of human nudes, 1999.

[GGCAF21] A. Gangwar, V. González-Castro, E. Alegre, and E. Fidalgo. Attm-cnn: Attention and metric learning based cnn for pornography, age and child sexual abuse (csa) detection in images. *Neurocomputing*, 445:81–104, 7 2021.

[GRH+18] M. Garcia, T. Revano, B. Habal, J. Contreras, and J. Enriquez. A pornographic image and video filtering application using optimized nudity recognition and detection algorithm. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018*. Institute of Electrical and Electronics Engineers Inc., 3 2018.

[GV22] N. Gautam and D. Vishwakarma. Obscenity detection in videos through a sequential convnet pipeline classifier. *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[Hen08] J. Hendler. Avoiding another ai winter, 2008.

[HSWA18] A. Husodo, G. Suta Wijaya, and W. Arimbawa. *Realtime Porn Image Censor Method for PreventingSmartphone Users to Take a Pornographic Photo*. IEEE, 2018.

[HZRS15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 12 2015.

[IBM23] IBM. ¿qué es la visión artificial? https://www.ibm.com/es-es/topics/computer-vision, 2023.

[KR22] R. Kang and P. Rau. Iivrs: An intelligent image and video rating system to provide scenario-based content for different users, 2022.

[Kum21] S. Kumar. Smart system to detect adult content and child pornography on web. *International Journal for Research in Applied Science and Engineering Technology*, 9:1704–1706, 9 2021.

[Mah17] M. Mahmoodi. Fast and efficient skin detection for facial detection, 2017.

[MAP+16] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha. Pornography classification: The hidden clues in video space–time. *Forensic Science International*, 268:46–61, 11 2016.

[MMN23] P. Meseguer and J. Moreno Navarro. Turing y el ajedrez. https://blogs.elpais.com/turing/2012/07/turing-y-el-ajedrez.html, 2023.

[MP23] J. Mahadeokar and G. Pesavento. Open sourcing a deep learning solution for detecting nsfw images. https://yahooeng.tumblr.com/post/151148689421/open-sourcing-a-deep-learning-solution-for, 2023.

[Ng23a] A. Ng. Programa especializado: Aprendizaje automático. https://www.coursera.org/specializations/machine-learning-introduction, 2023.

[Ng23b] A. Ng. Programa especializado: Aprendizaje profundo. https://www.coursera.org/specializations/deep-learning, 2023.

[oG23] University System of Georgia. A brief history of the internet. https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml, 2023.

[RNCRJA04] S. Russell, P. Norvig, J. Corchado Rodríguez, and L. Joyanes Aguilar. *Inteligencia artificial : un enfoque moderno*. Pearson Prentice Hall, 2004.

[SA23] S. Sancho Azcoitia. Mycin, el comienzo de la inteligencia artificial en el mundo de la medicina. https://empresas.blogthinkbig.com/mycin-el-comienzo-de-la-inteligencia/, 2023.

[SCD+16] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. 10 2016.

[SK20] K. Song and Y. Kim. An enhanced multimodal stacking scheme for online pornographic content detection. *Applied Sciences (Switzerland)*, 10, 4 2020.

[TBC+20] A. Tabone, A. Bonnici, S. Cristina, R. Farrugia, and K. Camilleri. Private body part detection using deep learning. pages 205–211. SciTePress, 2020.

[TCB+21] A. Tabone, K. Camilleri, A. Bonnici, S. Cristina, R. Farrugia, and M. Borg. Pornographic content classification using deep-learning. Association for Computing Machinery, Inc, 8 2021.

[Yun23] Bosco Yung. opennsfw2. https://github.com/bhky/opennsfw2, 2023.