Método para la Transformación Facial de Imágenes Sintéticas

– –

Method for Facial Transformation of Synthetic Images

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

CURSO 2022–2023

David del Cerro Domínguez

Directores

Luis Javier García Villalba
Daniel Povedano Álvarez

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

Madrid, Septiembre de 2023

Agradecimientos

En primera instancia agradecer a mis padres por todo el apoyo durante toda mi vida.

En segunda instancia agradecer a todos los tutores por su tiempo y dedicación para el desarrollo de este trabajo.

Índice General

Índice de Figuras

Índice de Tablas

Índice de Algoritmos

Lista de Acrónimos

AP	*Aprendizaje profundo*
FID	*Frechet Inception Distance*
GAN	*Redes Generativas Adversarias*
IA	*Inteligencia Artificial*
KID	*Kernel Inception Distance*
MMD	*Maximum Mean Discrepancy*
RNC	*Redes Neuronales Convolucionales*

Abstract

The huge progress in the development of GANs has produced great advances in areas such as aging, providing a quality in the transformation and in the images generated that has never been seen before. However, this brings several problems, the first is obtaining the same face at different ages and the large number of examples needed to train the model. The other problem is how to maintain the features of the face in the transformation.During the development of this work, these problems will be addressed, for this reason a method was developed for the generation of synthetic faces and the necessary models for their correct classification and a CycleGAN-based model for rejuvenation was developed. In addition, all these models were evaluated with different experiments.

Keywords: Artificial Inteligence, Age Classification, Generative Adversarial Networks, Convolutionals Network, Image Transformacion, Aging, CycleGAN, Faces.

Resumen

El gran desarrollo de las redes GAN ha producido grandes avances en áreas como el rejuvenecimiento facial, produciendo una calidad en la transformación y en las imágenes generadas nunca vista. Sin embargo esto conlleva varios problemas, uno es el problema de obtener el mismo rostro a distintas edades y la gran cantidad de ejemplos necesarios para entrenar el modelo. El otro problema es conseguir mantener las características del rostro en la transformación. Durante el desarrollo de este trabajo se abordarán estos problemas, por ello se desarrolló un método para la generación de rostros sintéticas y los modelos necesarios para su correcta clasificación y también un modelo basado en CycleGAN para el rejuvenecimiento. Además se evaluaron todos estos modelos con distintos experimentos.

Palabras clave: Inteligencia Artificial, Redes GAN, Redes convolucionales, Rejuvenicimiento, Imágenes sintéticas, Rostros, Cálculo edad, CycleGAN.

Capítulo 1

Introducción

1.1. Motivación

El rejuvenecimiento facial es la técnica por la cual dado un rostro, este se transforma a otro de cierta edad. Durante la ultima década ha habido distintos métodos para realizar este rejuvenecimiento, desde aquellos que reducían arrugas o modificaban la textura de la cara. Pero no ha sido hasta la llegada de las redes GAN cuando realmente se ha producido un gran avance en este campo.

Las redes GAN requieren una gran cantidad de datos, en ciertos campos puede no resultar fácil acceder y agrupar estos datos, sobretodo en aquellos casos como el rejuvenecimiento facial donde se necesitan datos de la misma cara pero en distintos edades. Debido a esto distintos artículos han tratado esta problemática desde la exploración del espacio latente de las GAN hasta la construcción de modelos que no necesitan datos emparejados, o también el uso de imágenes sintéticas o aumento de datos.

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

Rejuvenecer un rostro en una imagen conlleva diversos problemas, lo primero es conseguir una *Inteligencia Artificial* (IA), en nuestro caso una red GAN, que sea capaz de inferir que rasgos o características de una persona se mantienen desde que somos niños hasta adultos, el color de ojos o color de la piel,por ejemplo. Y otros como las arrugas o signos de envejecimientos en la piel que claramente cambian años. Conseguir imágenes etiquetadas de la misma persona de cuando era niño y mayor con una buena calidad puede llevar problemas por la falta de datos en estos campos.

Este trabajo abordará un modelo con redes GAN que permita rejuvenecer un rostro dado, manteniendo sus rasgos básicos, que además solucionará el problema de la falta de datos emparejados mediante el uso de IA.

Además también se investigarán distintas métricas que nos permita mejorar y evaluar el modelo construido y así mejorar la calidad de las imágenes construidas y la calidad del rejuvenecimiento en sí.

1.4. Plan de Trabajo

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

1. **Investigación**: Al comienzo del trabajo se realizó una reunión dónde se explicó los conocimientos básicos para comenzar a realizar el trabajo, y que conocimientos sobre redes GAN se deberían de obtener para la realización del trabajo. También se especificó los conocimientos sobre *Keras*, *PyTorch* y *Python* que se necesitarían más adelante, y se nos dió varios recursos para avanzar en esta parte. Además se explicó los objetivos finales del trabajo y el contexto en el que se realizaría. Con todo esto se acordó realizar un seguimiento semanal del trabajo, donde se expondría lo avanzado durante la semana y las dudas o problemas que nos hubiesen surgido.

   Durante los primeros tres meses se adquirieron los conocimientos sobre redes GAN y *Redes Neuronales Convolucionales* (RNC) leyendo numerosos artículos y libros

y también se comenzó probando algunas redes GAN preentrenadas. En esta fase también se empezó a elegir el rumbo del trabajo y que se desarrollaría en las etapas posteriores.

2. **Desarrollo**: Con los conocimientos básicos ya adquiridos, se empezó a preparar los modelos que se usarían mas adelante obteniendo así un nivel más avanzado en el uso de las librerías de *TensorFlow* y *Keras*. También se empezó a recolectar y construir los datos de entrenamiento. No se dejó de seguir con la investigación de más artículos científicos pero si se bajó el ritmo.

3. **Experimentación**: Una vez se tenían los modelos construidos y los datos de entrenamientos preparados. Se empezó con el entrenamiento y ajuste de las redes. Se probaron distintos modelos y se eligió aquellos que mejor resultados obtenía. Se obtuvieron todos los resultados y se analizaron. En esta fase se dejó la investigación.

1.5. Estructura del Trabajo

El resto del trabajo está organizado en 8 capítulos con la siguiente estructura:

El Capítulo 2 introduce los conceptos elementales de las RNC, redes GAN y la transformación de imágenes. Diferenciando entre la transformación de datos emparejados y datos no emparejados. Además explica la red StyleGAN [KLA19] y distintas métricas posibles para la evaluación de redes GAN.

El Capítulo 3 muestra algunos de los modelos más complejos en el campo de las redes GAN como CycleGAN [ZPIE20], además de modelos que profundizan en la falta de datos, como DAGAN [ASE18] o DatasetGAN [FKAL22], todos ellos junto a sus resultados.

El Capítulo 4 presenta el desarrollo del trabajo, mostrando primero como se han obtenido los distintos modelos para la clasificación de rostros faciales, para una posterior creación de un *dataset* sintético. Después se muestra como se construye el modelo basado en CycleGAN y se ajusta para el rejuvenecimiento facial de rostros.

El Capítulo 5 muestra los resultados obtenidos de los modelos obtenidos y de la obtención de los datos sintéticos planteados en el Capítulo 4.

El Capítulo 7 muestra conclusiones de este trabajo y las principales lineas a seguir como trabajo futuro de este trabajo.

Los Capítulos 8 y 6 son las traducciones al inglés de la Introducción y de las Conclusiones.

Capítulo 2

Contexto de la Investigación

El inicio de la IA [TBS+21] se remonta a los años 50, donde se empezaron a sentar las bases de la misma, en 1956, el término IA fue acuñado por John McCarthy. Un año después Rosenblatt desarrolló el perceptrón [Ros58], que permitió la clasificación binaria y sentó las bases para el *Aprendizaje profundo* (AP) y las redes neuronales modernas como las redes GAN.

Durante las siguientes décadas el desarrollo de la inteligencia artificial se estancó debido principalmente a la capacidad computacional de la época.

A partir del año 2000, surgieron nuevos modelos como las redes neuronales artificiales y las redes neuronales profundas que se usaron en diversas aplicaciones.

En los últimos años, la IA ha sufrido los avances más significativos. En 2014 se presentaron las redes GAN [GPAM+14] estas se basan en un generador y discriminador que compiten entre sí. El generador intenta generar un ejemplo lo más parecido a los datos de entrada del modelo, y el discriminador intenta discernir si el ejemplo es real o es creado por el generador y devuelve esa información al generador para mejorar su rendimiento. Las GAN se han utilizado en diversas aplicaciones, pero donde más han destacado es en la generación de imágenes, por ejemplo StyleGAN, que logra generar imágenes con una alta calidad.

2.1. Aprendizaje profundo

El AP es una rama de la IA que se enfoca en el desarrollo de algoritmos que se inspiran en el cerebro humano y el aprendizaje humano. Para ello se basa en redes neuronales artificiales de múltiples capas que aprenden de un conjunto de datos de entrenamiento que contiene ejemplos de entradas con sus respectivos ejemplos de salidas y la red intenta minimizar el error de las predicciones respecto a las salidas reales.

El AP es muy eficaz en una variedad de aplicaciones como por ejemplo en reconocimiento de objetos en imagen o sistemas de recomendación.

2.2. Redes Neuronales Convolucionales

Las RNC [IBM] son un arquitectura para el AP, están diseñadas para procesar datos de tipo matriz, como por ejemplo las imágenes y encontrar patrones. Por ello se utilizan en

clasificación de objetos o segmentación.

La arquitectura se basa en las siguientes capas :

- **1. Capa convolucional.** En esta capa se aplican los filtros a la entrada de la red. Cada filtro realiza distintas operaciones matemáticas para obtener los patrones de la entrada.
- **2. Capa de agrupamiento.** En esta capa se generaliza los patrones obtenidos en la capa convolucional reduciendo a su vez el tamaño de la red.
- **3. Capa totalmente conectada.** Esta capa transforma los patrones en una representación final. Cada neurona de esta capa se conecta con todas las neuronas de la capa anterior.
- **4. Capa de salida.** Es esta capa se transforma la representación anterior en la salida deseada, dependiendo de lo necesitado.

Además de las capas anteriores se pueden incluir alguna capa de regularización como por ejemplo la capa *Dropout* [AAB+15] , esta capa durante el entrenamiento anula ciertas neuronas y deja las otras igual, cambiando en cada iteración, provocando que no haya propagación a través de ellas. Esto evita el sobreajuste de la red.

2.3. Transfer Learning

El *transfer learning* [Bro19b] es una técnica donde se reusa una red ya entrenada en una tarea con un dataset determinado y amplio, en otra tarea relacionada pero del que se dispone de un dataset menor. Esta técnica sobretodo se utiliza en problemas de reconocimiento de objetos y visión artificial, donde la red ha aprendido a extraer las características claves de las imágenes.

2.4. Redes GAN

Las redes GAN o Redes Generativas Antagónicas [Bro19a][GPAM+14], es un modelo generativo basado en el AP, siendo un modelo de aprendizaje no supervisado . En esta red, compiten dos modelos, un generador que genera nuevos ejemplos y un discriminador que intenta clasificar el ejemplo si es real, o esta generado por el generador.

En la Figura 2.1 se puede observar un modelo básico de una red Generativa Antagónica, dónde se puede observar el generador, el discriminador, y con que información se entrena el modelo.

El generador de una GAN recibe una *seed*, o un vector de entrada y genera un ejemplo nuevo, por ejemplo para una imagen, este vector de entrada se convierte al final en la imagen generada, es decir, el vector de entrada contiene una información comprimida de los puntos de la imagen final, que en el proceso de la generación de la imagen, se irá convirtiendo en información de la imagen final, pero comprimida. A esto se le conoce como espacio latente y contiene información relevante, por ejemplo como podemos observar en la Figura 2.2, vemos como según atravesamos las capas del espacio latente, podemos ver

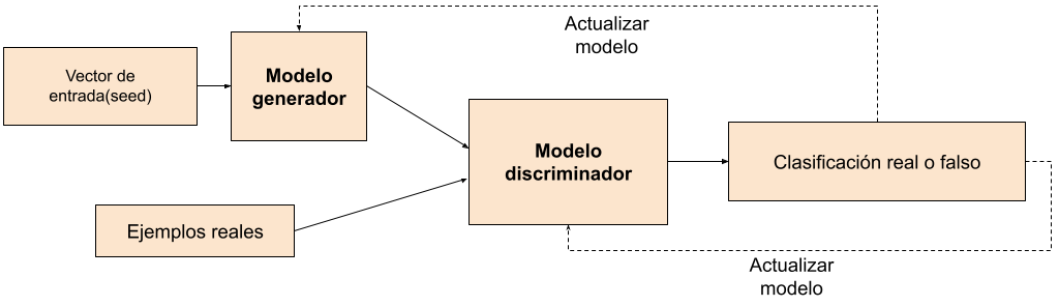

Figura 2.1: Modelo de Red GAN básico [Bro19a]

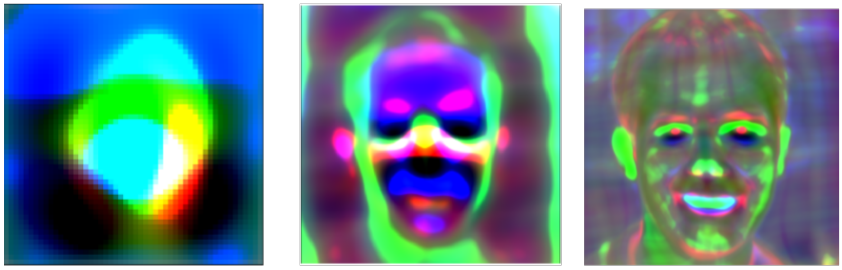

Figura 2.2: Ejemplo de espacio latente en StyleGAN3

la formación de un rostro. Conocerlo bien da la posibilidad de modificarlo a deseo para generar imágenes específicas, con unas características deseadas.

Desde el primer artículo que describía las redes GAN, hasta el día de hoy, han surgido varias implementaciones, que usando estas redes, realizan distintos cometidos, desde la generación de imágenes, la modificación o hasta la transformación de las mismas. Una de ellas que ha sido realmente relevante ha sido StyleGAN.

2.5. Traducción de imágenes

La traducción de imágenes o *image traslation* es una técnica donde se busca convertir una imagen de un dominio a otro dominio, donde hay una relación y se mantienen ciertas características entre las imágenes de distintos dominios. Por ejemplo para transformar un caballo en una cebra, cambiar el estilo artístico de una obra o cambiar el fondo de una imagen. La mayoría de estos modelos usan redes GAN y algunos ejemplos son :

- Pix2Pix [IZZE18]
- CycleGan [ZPIE20]
- Munit [HLBK18]
- ACL-GAN [ZWD21]

Podemos dividir estos modelos en dos grupos, dependiendo del tipo de datos que necesito, pix2pix [IZZE18] necesita datos emparejados, es decir, necesita conocer la imagen de entrada y la imagen transformada correspondiente. La ventaja que aporta por ejemplo

CycleGan y ACL-GAN [ZWD21] es que no necesitan que los datos estén emparejados, para casos donde conseguir los datos emparejados es complicado o casi imposible, esto proporciona una solución.

2.6. Aumento de datos

Las RNC o las redes GAN se nutren de una gran cantidad de datos para obtener un gran rendimiento, sin embargo, no siempre se dispone de estas grandes muestras de datos y se tiene que entrenar a estos modelos con datos limitados, lo que termina produciendo un sobreaprendizaje de las redes y poca generalización. Por ello, durante los últimos años se han desarrollado diferentes técnicas para evitar el sobreaprendizaje, algunas ya tratadas como la capa *dropout* u otras técnicas como el aumento de datos. El aumento de datos [PW17], o en inglés *Data Augmentation*, es una técnica que permite generar más datos a partir de los datos existentes aplicando diversas transformaciones como pueden ser rotaciones, desplazamientos volteos o la introducción de ruido. Esto no solo puede ser usado en *datasets* pequeños, si no también puede mejorar el rendimiento para grandes *datasets*.

2.7. StyleGAN

StyleGAN es una red desarrollada por NVIDIA, que es utilizado para la generación de imágenes realistas, ha destacado sobretodo por su capacidad de generar rostros humanos muy parecidos a los reales. A diferencia de las redes GAN convencionales, StyleGAN [KLA19] divide el generador en dos partes, visibles en la Figura 2.3, esta división crea un nuevo espacio latente intermedio. En vez de, como en la mayoría de modelos GAN, proporcionar a la red un vector Z, mediante este nuevo espacio, denominado espacio W, donde se transforma el vector Z por un vector de 512 dimensiones, permite realizar más modificaciones en el espacio latente, este contiene información única sobre la imagen de una forma comprimida.

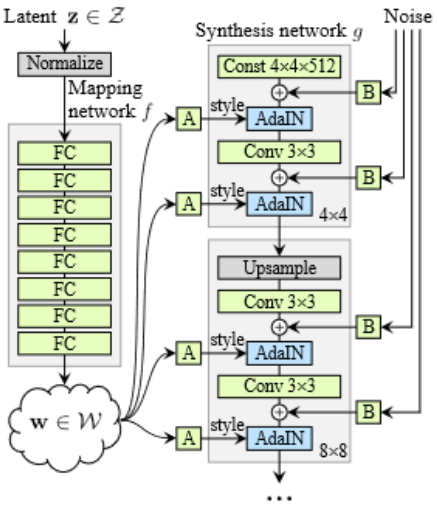

Figura 2.3: Arquitectura del generador de StyleGAN [KLA19]

Durante el proceso de generación a cada una de las capas del generador, se le inyecta un

vector de estilo, que permite un control sobre distintas características de la imagen final, como puede ser el color de piel, la presencia de gafas, la forma o el color del pelo. Cada capa cuenta con RNC que aumentan el tamaño de la imagen, y capas AdaIN[HB17] de normalización que permite realizar la transferencia de estilo mediante la transferencia de la media y covarianza de la imagen de referencia sin perder diversidad y detalles.

2.7.1. Valor de truncación

El valor de truncación [KLA19] ajusta la varianza del espacio latente con el que se genera el rostro. En los *datasets* de entrenamiento hay características que están infrarreperesentadas, por ello la truncación permite generar también estas caras, por ejemplo como se puede observar en la Figura 2.4, a un valor más alto, se generan imágenes más diversas. Rebajando este valor podemos obtener caras de mejor calidad, a costa de obtener un dataset menos variado.

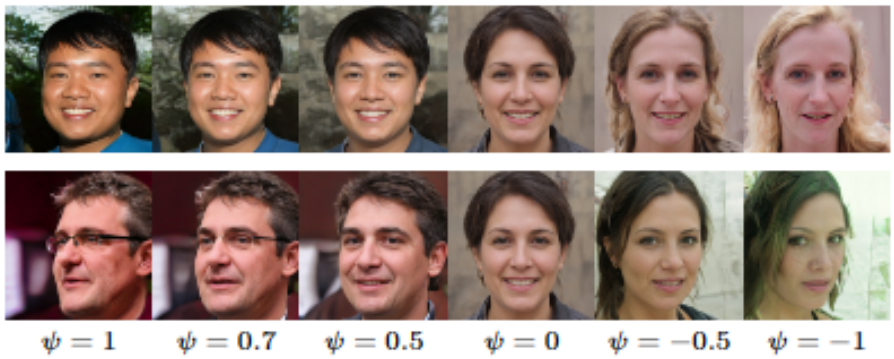

Figura 2.4: Efecto del valor de truncación [KLA19]

2.8. Métricas para evaluar el rendimiento de las redes GANs

La evaluación de las redes GAN ha supuesto un problema en el desarrollo, pues no ha habido una métrica única para la evaluación de las mismas [Bor18]. Algunas de las más comunes son las siguientes :

- **1. FID** - Frechet Inception Distance.
- **2. KID** - Kernel Inception Distance.
- **3. IS** - Inception Score.

2.8.1. FID - Frechet Inception Distance

El FID [HRU+18] es una métrica que evalúa la calidad de las imágenes generadas por las redes GAN. El FID compara las similitudes de las características de dos grupos, normalmente un grupo son las imágenes reales y el otro las imágenes generadas por la red GAN.

Para calcular el FID entre dos grupos de imágenes se tienen que realizar los siguientes pasos :

- Primero se obtienen las características de las imágenes de ambos grupos mediante el uso de una red InceptionV3 preentrenada.
- Se calcula las medias me_1 y me_2 y también las covarianzas cov_1 y cov_2
- Se calcula el FID entre los grupos 1 y 2 siguiendo la fórmula 2.1 Donde Tr es el operador de traza que suma los elementos de la diagonal de la matriz.

$$||me_1 - me_2||^2 + Tr(cov_1 + cov_2 - 2 * sqrt(cov_1 * cov_2)) \tag{2.1}$$

Un valor más cercano a 0 significa que ambos grupos están más próximos entre si, por lo tanto más se parecen ambos grupos.

2.8.2. KID - Kernel Inception Distance

Kernel Inception Distance (KID) [BSAG21] fue propuesto para reemplazar a FID, ambas utilizan una red InceptionV3 para obtener las características de ambos grupos. La diferencia radica en la forma de calcular la distancia entre ambos conjuntos. KID usa *Maximum Mean Discrepancy* (MMD), de tal forma, no importa el número de muestras para calcular la distancia.

Al igual que el FID, un valor más cercano a 0, mejor.

Capítulo 3

Estado del Arte

El rejuvenecimiento facial es la transformación de un rostro dado a otro rango de edad diferente, manteniendo las características básicas y rasgos de la persona durante la transformación. Esto conlleva dos problemas principales, el primero es la necesidad de conseguir imágenes de rostros de la misma persona en distintos rangos de edad. Para ello distintos trabajos han abordado este problema en diferentes áreas, dando lugar a los métodos no emparejados, donde las redes no necesitan conocer los pares de datos, aquí se encuentran algunos modelos como CycleGAN o Dual Conditions GANs. Además estos modelos incluyen algún método para mantener la información básica en las transformaciones, normalmente mediante la reconstrucción de las imágenes generadas. Estas redes a pesar de no necesitar datos emparejados, si que se nutren de una gran cantidad de datos, para ello técnicas como el aumento de datos o la introducción de imágenes sintéticas han sido exploradas como posibles soluciones. Algunas de estas soluciones se tratan en las siguientes secciones.

3.1. Aumento de datos con redes GAN

El aumento de datos explicado en el capítulo 2 intenta generar más ejemplos en un dataset mediante transformaciones simples. Otros modelos como DAGAN [ASE18] exploran el invariante que provoca que una imagen se mantenga en el mismo dominio para generar un mayor número de ejemplos. Para ello aprovechan la capacidad de generalizar de las redes GAN.

3.1.1. Arquitectura de DAGAN

La arquitectura de DAGAN (Figura 3.1) se divide en el generador, que se compone de un *encoder* que codifica una imagen de entrada de la clase correspondiente, un vector aleatorio que se combina con la información codificada en el generador de la red para generar una imagen transformada. La otra parte se compone del discriminador que intenta discernir entre las imágenes reales y las generadas por el generador.

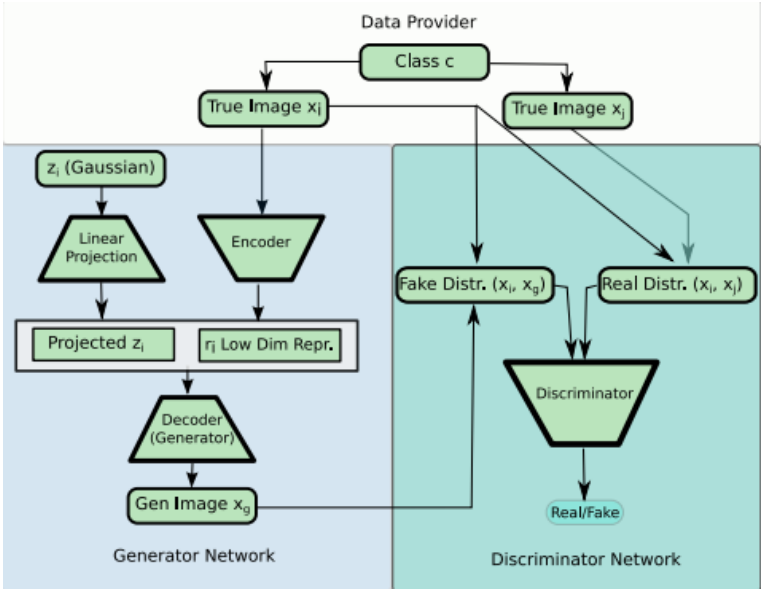

Figura 3.1: Arquitectura de DAGAN [ASE18]

3.1.2. Resultados

DAGAN se evalúa en tres *datasets* de distintos dominios, Omniglot, EMNIST y VGG-Face a los que se les aplica el aumento de datos con DAGAN y después se compara el rendimiento respecto al dataset con el aumento de datos estándar.

Tanto en el dataset de VGG-Face (Tabla 3.1) y en Omniglot(Tabla 3.2) el rendimiento con DAGAN aumenta respecto al rendimiento con el aumento de datos estándar, por lo que usar DAGAN en situaciones de falta de datos puede ser una buena opción para mejorar el rendimiento de distintos modelos.

Tabla 3.1: Resultados VGG-Face [ASE18]

Modelo	**Ejemplos por clase**	**Exactitud**
VGG-Face Standard	5	0.0446948
VGG-Face DAGAN Augmented	5	**0.125969**
VGG-Face Standard	15	0.39329
VGG-Face DAGAN Augmented	15	**0.429385**
VGG-Face Standard	25	0.579942
VGG-Face DAGAN Augmented	25	**0.584666**

3.2. Imágenes sintéticas

Para entrenar los modelos de AP o modelos de redes GAN es necesario una gran cantidad de datos, esto no siempre es posible en todos los campos o se no se disponen de los suficientes. Para ello DatasetGAN [FKAL22] genera imágenes sintéticas apoyándose en una GAN basada en el estilo como StyleGAN2[KLA+20]. Esta red genera imágenes médicas útiles y sus respectivas máscaras para modelos de aprendizaje profundo para detección de enfermedades o lesiones a través de imágenes.

Tabla 3.2: Resultados Omniglot [ASE18]

Modelo	Ejemplos por clase	Exactitud
Omni Standard	5	0.689904
Omni DAGAN Augmented	5	**0.821314**
Omni Standard	10	0.794071
Omni DAGAN Augmented	10	**0.862179**
Omni Standard	15	0.819712
Omni DAGAN Augmented	15	**0.874199**

3.2.1. Arquitectura de DatasetGAN

Su arquitectura se compone de una GAN basada en estilo, en este caso StyleGAN2 con el discriminador ADA [KAH+20]. Este discriminador realiza un aumento de datos adaptativo, realizando transformaciones a las distintas imágenes de entrenamiento pero sin propagarlo al generador, de la forma que mejora el rendimiento general de la red StyleGAN2 sobretodo en *datasets* pequeños, como puede ser este el caso. El resto es un intérprete de las imágenes generadas que genera sus máscaras, para ciertas aplicaciones médicas.

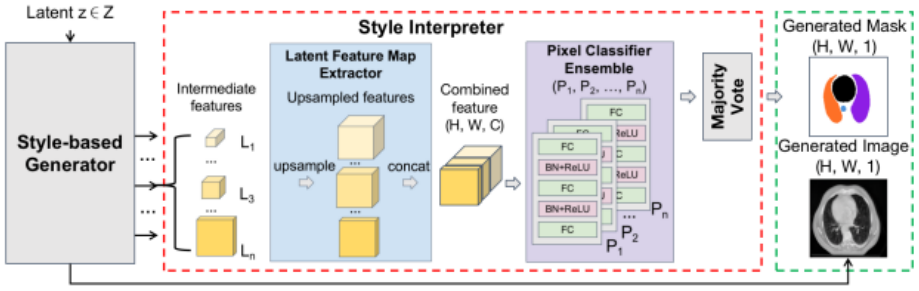

Figura 3.2: Arquitectura de DataSetGAN [FKAL22]

3.2.2. Resultados

El artículo evalúa el rendimiento de su modelo mediante el uso de la métrica mIoU, siguiendo la fórmula 3.1 , donde TP representa los verdaderos positivos, FP los falsos positivos, FN los falsos negativos, i representa las distintas clases y n representa el número total de clases.

$$mIoU = \frac{\sum TP_i / (TP_i + FP_i + FN_i)}{n} \tag{3.1}$$

Para la evaluación de la eficacia de los datos generados, se entrena una red DeepLab-V3-ResNet101 con 500 datos generados (Train-G), 500 datos reales (Train-R), de la misma forma se generan los datos para test Test-G y Test-R. Además también se prueba la red mediante el entrenamiento de la mezcla de datos generados y real, llamado Mix-n, donde n representa el número de datos reales en el dataset de entrenamiento. Como se puede observar en la Tabla 3.3 para el entrenamiento con datos generados se obtiene

un valor mIoU bajo en el test con imágenes reales, al igual que para el entrenamiento con datos reales se obtiene un valor bajo con el test de imágenes generadas. Sin embargo al añadir al menos 10 imágenes reales, ambos valores se igualan, lo que demuestra que se obtiene un buen rendimiento.

Tabla 3.3: Tabla resultados de Dataset-GAN [FKAL22]

Tipo de dato de entrenamiento	mIoU en Test-G	mIoU en Test-R
Train-G	0.770 ± 0.012	0.551 ± 0.011
Train-R	0.459 ± 0.004	0.779 ± 0.010
Mix-1	0.773 ± 0.011	0.585 ± 0.005
Mix-5	0.769 ± 0.012	0.689 ± 0.003
Mix-10	0.768 ± 0.011	0.755 ± 0.004
Mix-20	0.770 ± 0.011	0.780 ± 0.003

3.3. Uso de métodos no emparejados

Uno de los problemas del rejuvenecimiento facial es la dificultad de lograr imágenes del rostro de la misma persona de diferentes edades. Para solucionar la necesidad en general de la falta de datos emparejados, se han propuesto diversos métodos que aprender la transformación entre distintos dominios. Además estos métodos incluyes alguna consistencia entre las imágenes originales y transformadas mediante la introducción de un objetivo que reconstruye la imagen generada y evita que la red transforme una imagen de entrada en cualquiera de los ejemplos de entrenamiento.

3.4. Dual Conditional GANs

Dual Conditional GANs [SZG+18] intenta solucionar dos problemas fundamentales del rejuvenecimiento facial, uno es la necesidad de datos de entrenamiento secuenciales, es decir, tener el rostro de la misma persona en distintos rangos de edad, algo que no siempre es posible y el otro problema es mantener las características faciales de la persona en cuestión.

3.4.1. Arquitectura de DCGAN

Para abordar los problemas anteriores, su modelo primero transforma la imagen de entrada a otra de la edad dada y aprende a realizar la función inversa, por esto una función de perdida tiene en cuenta el error de la imagen reconstruida, así que el discriminador guía al generador para generar las imágenes.

El modelo de Dual Condition GAN se basa en dos cGAN, donde cada una se compone de dos generadores, donde uno obtiene el rostro rejuvenecido y otra la reconstrucción de esta última, y sus correspondientes generadores, como se puede apreciar en la Figura 3.3

El generador (Figura 3.4) por su parte es bastante parecido al generador de CycleGAN [ZPIE20], la parte del *encoder* extrae las características de los rostros. Esta se compone de 3 convoluciones y 9 bloques residuales. La condición de la edad se incluye en el

decoder , donde se representa cada grupo de edad como un vector. El *decoder* contiene 3 deconvoluciones.

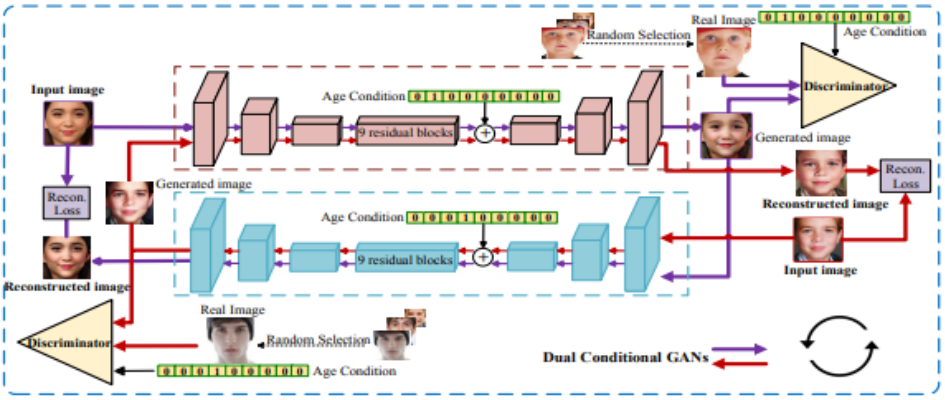

Figura 3.3: Arquitectura de Dual Condition GAN [SZG+18]

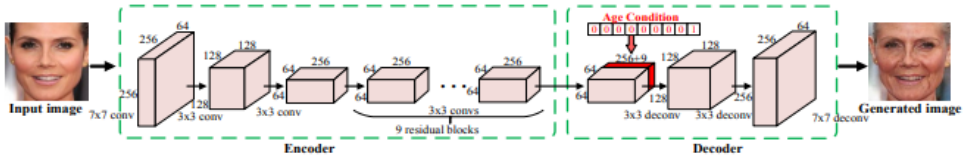

Figura 3.4: Generador de Dual Condition GAN [SZG+18]

3.4.2. Resultados

Los autores evalúan el rendimiento de la red de dos formas, una forma cuantitativa y otra donde comprueban que el rostro preserve la identidad. En ambos casos, comparan los resultados contra otras redes no emparejadas como CAAE y C-GANS y también con la red de datos emparejados FT demo. En la evaluación cuantitativa realizan un estudio con usuarios donde piden a los candidatos seleccionar la mejor imagen. Como se puede observar en la Figura 3.5, Dual CGANs mejora en rendimiento a todos los métodos no emparejados, el único que lo mejora es FT demo que necesita de datos emparejados.

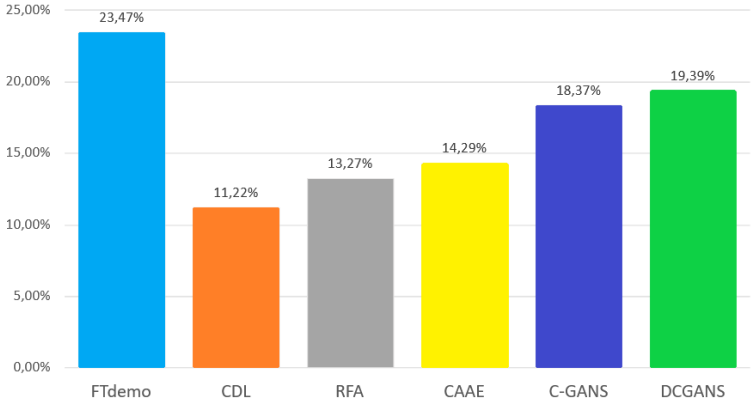

Figura 3.5: Evaluación cuantitativa [SZG+18]

En la preservación de la identidad realizan una comparación de las imágenes generadas con las reales, donde como se puede observar en la Tabla 3.4, la DCGANs mejora el rendimiento del resto, preservando mejor la identidad en las imágenes transformadas.

Tabla 3.4: Resultados de preservación de identidad [SZG+18]

	Input	FT	CDL	RFA	CAAE	C-GANS	DCGANS
4-10	0.98	0.85	0.48	0.59	0.57	0.47	0.86
8-30	0.99	0.86	0.78	0.42	0.51	0.79	0.82
10-20	0.97	0.80	0.68	0.44	0.51	0.70	0.81
10-40	0.95	0.60	0.65	0.33	0.51	0.49	0.80
20-60	0.97	0.77	0.73	0.48	0.65	0.76	0.80
30-60	0.97	0.64	0.70	0.41	0.45	0.65	0.72
40-60	0.98	0.68	0.59	0.52	0.70	0.75	0.84
Average	**0.97**	**0.74**	**0.66**	**0.46**	**0.56**	**0.66**	**0.81**

3.5. TripleGAN

Las redes GAN han demostrado ser efectivas en los problemas de aprendizaje semi supervisado mientras que mantienen sus capacidades generativas. Aunque cuando a esta red contiene un discriminador categórico con más de una clase obtiene dos problemas principal, el generador y el discriminador pueden no ser óptimos al mismo tiempo y el generador no puede controlar las semánticas de los ejemplos generados. Por ejemplo para la generación de imágenes de distintos animales.

Para atajar estos problemas, Triple-GAN [LXZZ17] se presenta como un juego de tres, donde además del generador y discriminador, se incluye un clasificador. El clasificador genera pseudoetiquetas a partir de datos reales, el generador genera pseudodatos a partir de etiquetas reales, y el discriminador distingue si un par dato y etiqueta es real o no. Por esto, hay tres distribuciones en el modelo, la de datos y etiquetas reales, la de datos no reales etiquetas reales y la de datos reales y etiquetas no reales, como se puede observar en la Figura 3.6

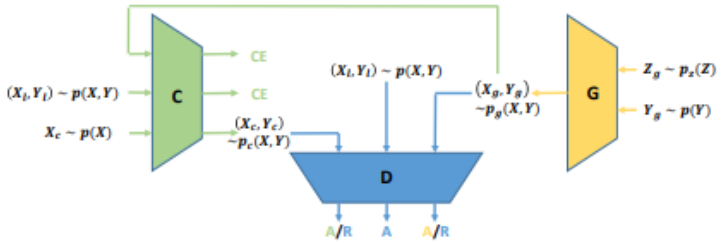

Figura 3.6: Arquitectura de TripleGAN [LXZZ17]

3.5.1. Resultados

TripleGAN se evalúa en *datasets* de imágenes diferenciadas en distintas categorías como son MNIST[Den12], dígitos escritos a mano, SVHN [NWC+11], secuencia de dígitos con

distintos fondos y CIFAR10 [KNH], imágenes de aviones, automóviles, pájaros, etc. En la Tabla 3.5 se puede ver la comparación respecto a otros modelos y sus porcentaje de error.

Tabla 3.5: Resultados de TripleGAN [LXZZ17]

Modelo	MNIST	SVHN	CIFAR
CatGAN	1.39(±0.28)		19.58(±0.58)
Improved-GAN	0.93(±0.07)	8.11(±1.3)	18.63(±2.32)
ALI		7.3	18.3
Triple-GAN	0.91(0. ± 58)	5.77(±0.17)	16.99(±0.3)

TripleGAN logra obtener un mejor rendimiento respecto a los demás modelos del estado del arte.

3.6. CycleGAN

CycleGAN [ZPIE20] es un modelo de aprendizaje no supervisado, ya que no necesita que los datos estén etiquetados entre la entrada y la salida. Para realizar la transformación, CycleGAN utiliza un *dataset* de entrada X y otro de salida Y

Su arquitectura 3.7 consiste en dos generadores G y F y dos discriminadores D_x y D_y, el primer generador intenta aprender la transformación entre el dominio X y el Y, y el segundo entre el dominio Y y el X. El discriminador intenta diferenciar las imágenes generadas por los generadores de las reales.

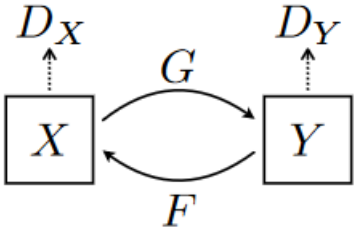

Figura 3.7: Modelo de CycleGAN [ZPIE20]

El generador es una RNC, este consta de tres partes, la primera que reduce la imagen y la codifica, para ello se utiliza primero una *Convolucion-InstanceNorm-ReLU* con 64 filtros y un *stride* de 1. Después dos convoluciones 3 x 3 - *InstanceNorm-Relu* con un *stride* de 2 y filtros de 128 y 256 respectivamente. Posteriormente, en la siguiente fase, se utilizan varios bloques residuales que contienen dos 3x3 convoluciones con el mismo número de filtros en ambas.Y por ultimo en la fase de escalado y decodificación se utiliza 2 convoluciones 3x3 *InstanceNorm-Relu* con un *stride* de 1/2 de 128 y 64 filtros respectivamente y por ultimo una convolución 7x7 con 3 filtros y 1 de *stride*.

El discriminador se compone de 4 convoluciones *InstanceNorm-LeakyReLU* con 64, 128,256 y 512 filtros y todas con un *stride* de 2.

El modelo intenta minimizar dos objetivos, el primero es la minimización de las imágenes generadas y las reales de ambas funciones, tanto G y F. El segundo objetivo a minimizar es la ciclo consistencia, esta trata que después de aplicar la función G a una imagen, esta

se reconstruye aplicando la función F y se minimiza la diferencia entre la imagen original y la reconstruida, esto se puede ver en la Figura 3.8.

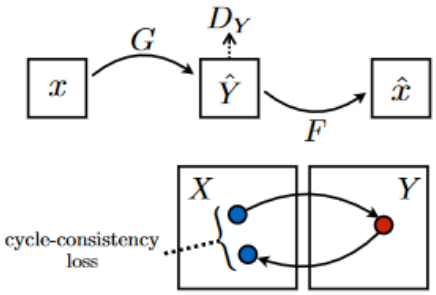

Figura 3.8: Ciclo consistencia de CycleGAN [ZPIE20]

A esto ultimo se le conoce como la ciclo consistencia y es realmente interesante, pues impide a la red que convierta a cada una de las imágenes de un dominio en una imagen de entrenamiento del dominio contrario, ya que al realizar la reconstrucción, tiene que mantener una relación entre las dos imágenes. Específicamente, esto permitirá poder rejuvenecer un rostro manteniendo sus características básicas.

Capítulo 4

Capítulo de Contribución

Una vez visto los mayores problemas que conlleva el rejuvenecimiento facial, se puede realizar un plan donde primero se obtendrán los datos necesarios para ello se utilizará una red StyleGAN, que generará imágenes sintéticas de rostro, se resolverán los problemas que conlleve y se clasificarán las imágenes generadas. Después se creará el modelo para el rejuvenecimiento y se prepararán los datos ya generados para el modelo. Al igual que con los datos generados, se resolverán los problemas y se mejorará el modelo.

4.1. Obtención dataset sintético

Una vez vistas las redes StyleGAN, se pueden usar para generar imágenes y así obtener un dataset de imágenes sintéticas, lo que puede proporcionar es una gran cantidad de imágenes. Para ello utilizaremos StyleGAN3 preentrenada con el dataset de FFHQ con una resolución de 1024x1024. Las imágenes sintéticas solucionan la escasez de datos de entrenamiento, pero a su vez nos introduce errores e imperfecciones a los datos, que se intentarán solucionar en las siguientes secciones, donde también se clasificarán las imágenes para preparar los datos para el modelo construido.

4.1.1. Mejorando la calidad de las imágenes obtenidas

El primer problema que se encuentra con las imágenes es que estas presentas imperfecciones, o incluso no son realistas. Para mejorar esto tenemos dos formas, una primera forma es ajustando los propios parámetros de la red StyleGAN, por ejemplo la truncación usada, explicada anteriormente en el Capítulo 3, otra forma es crear una red neuronal, que determine que imágenes se ajustan mejor a la realidad y cuales no.

4.1.1.1. Red neuronal

Aunque lo anterior elimina un gran número de imágenes con errores, sigue habiendo una cantidad significativa, el objetivo de esta red es eliminar las imágenes con los errores más apreciables. El dataset utilizado contiene 1028 imágenes clasificadas manualmente para entrenamiento y validación y 168 imágenes para test, las imágenes elegidas son aquellas cuyos errores son más apreciables, intentando eliminar dobles rostros o rostros irregulares.

Para ello se utilizará una red convolucional como en la figura 4.1 con 4 capas convolucionales con función de activación *relu*, después la capa totalmente conectada y por ultimo la capa de salida con función de activación *sigmoid* al tratarse de una clasificación binaria.

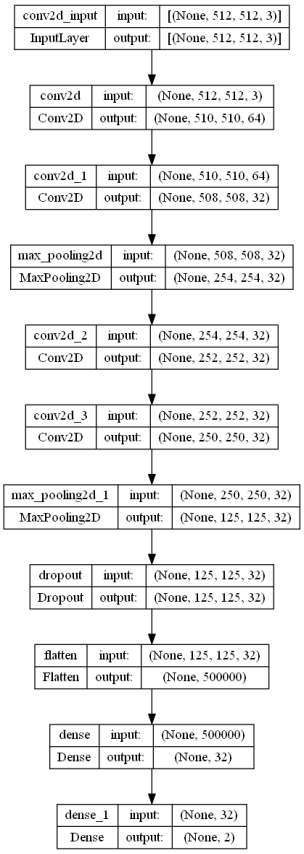

Figura 4.1: Modelo de red convolucional para elegir las imágenes con mayor calidad

Para el entrenamiento se ha usado un tasa de aprendizaje de 0.00001 con optimizador *Adam*, además de 35 épocas. Como se puede observar en la figura 4.2, a partir de la época 25 se produce sobreaprendizaje. Al final se obtiene un 88 % de precisión sobre la validación

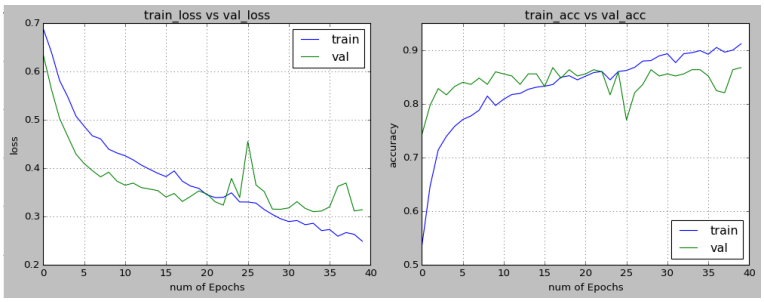

Figura 4.2: Gráfica entrenamiento

Una vez entrenada se puede evaluar la red con las imágenes para test, obteniendo así la matriz de confusión que se puede observar en la figura 4.3. La matriz de confusión arroja una precisión global del 82 por ciento. Produciéndose un 10 % de falsos negativos y solo

un 9 % de falsos positivos, los que más problemas presentarían, ya que los falsos negativos solo nos supondrán perder alguna imagen del dataset, pero los falsos positivos supondrán peor calidad en el dataset.

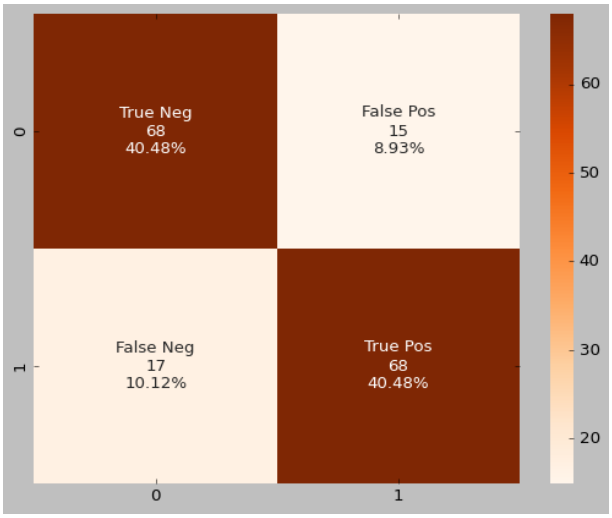

Figura 4.3: Resultados red neuronal

4.1.2. Clasificador edad

Una vez obtenidas las imágenes se necesita clasificarlas por edad, este modelo va a ser muy importante, puesto que el modelo creado posteriormente dependerá de que esta red distinga de la mejor forma posible los distingos rangos de edad. Por esto se probarán distintos modelos y se compararán sus resultados, para elegir correctamente aquel que mejor rendimiento obtenga.

El dataset usado será la combinación de dos *datasets*, estos se tratan de UTKFace y Facial-Age, estos dos *datasets* ofrecen imágenes faciales de personas entre 0 y 105 años, obteniendo entre ambos 33.485 imágenes. Para la clasificación, se va a dividir la edad en distintos grupos, el primero entre 0 y 15 años, el segundo entre 16 y 25, el tercero entre 26 y 40 años, el cuarto entre 41 y 60, y el quinto más de 60 años. El *dataset* está distribuido de la forma en la que se muestra en la figura 4.4. Esto permite sobretodo diferenciar a las personas jóvenes de las más mayores, lo que va a ser importante para la obtención del dataset sintético.

De las 33.485 imágenes se utilizarán 2.644 imágenes para el test de las redes, y de las 30.841 imágenes restantes se usarán el 90 % para el entrenamiento y el 10 % restante para la validación.

Además se utilizará aumento de datos explicado anteriormente en el Capítulo 2. Para este caso se harán las siguientes transformaciones de forma aleatoria:

- Volteo horizontal
- Desplazamiento con un 10 por ciento como máximo tanto horizontal como vertical.
- Rotación con un máximo de 15 grados de la imagen

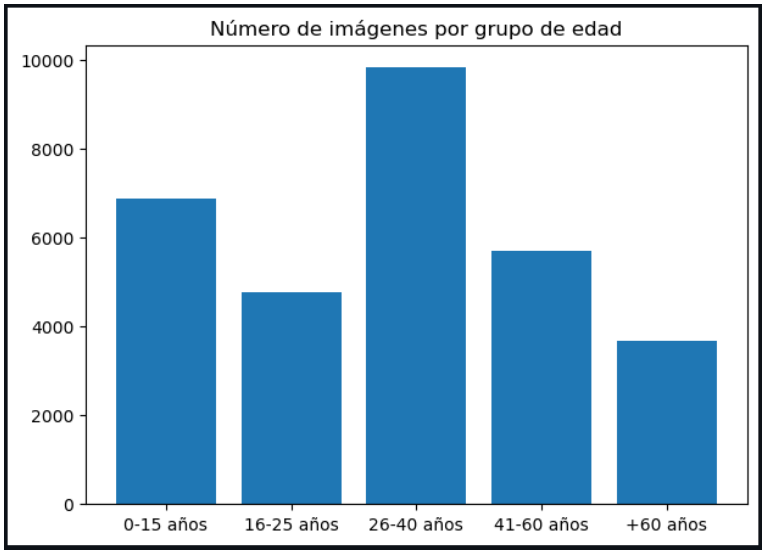

Figura 4.4: Distribución de edades en el dataset

Con estas transformaciones se consigue aumentar el número de datos de entrenamiento del modelo, lo cual permite construir una red más robusta con menos sobreaprendizaje.

4.1.3. Modelos para la clasificación de edad

Para clasificar a los rostros de las imágenes se probarán tres redes, que después se compararán, una de ella se trata de una red MobileNet no inicializada, otra red es una RNC propia, y otra red es una red VGG16 inicializado previamente donde usaremos *transfer learning*.

Para todos los modelos que se verán a continuación se ha usado la mismo tasa de aprendizaje, que comienza en 0.001 y que a partir de la época 20 decae un 10 % por cada época.

4.1.3.1. Red VGG16 y Transfer Learning

En este caso se usará una red VGG16 preentenaada con Imagenet y se transferirá el conocimiento de esta red preentrenada a la red que queremos construir con *transfer learning* explicado en el Capítulo 2. Imagenet [RDS+15] es un gran dataset de imágenes clasificadas en clases, en total el dataset de imagenet tiene 1.281.167 de imágenes clasificadas en 1000 clases distintas.

Una vez obtenido el modelo, se añade una red totalmente conectada y después otra capa para adecuar la salida a las 5 clases que corresponden con los distintos grupos de edad.

La red se entrena por 50 épocas, obteniendo una precisión del 72 % sobre los datos de validación. Como se puede observar en la figura 4.5, el aprendizaje es irregular y con sobreaprendizaje.

Como se puede observar en la matriz de confusión de la figura 4.6 el modelo consigue un 91 % de precisión en el rango de 0 a 15 años, pero confunde el grupo de 16 a 25 con el de 26 a 40, consiguiendo un 45 %. También la red, aunque en menor medida, confunde el

grupo de edad de 41 a 60, con el de 26 a 40 años. Por último, el modelo consigue una precisión del 68 % es el grupo de mayor edad. La precisión total es del 71 %.

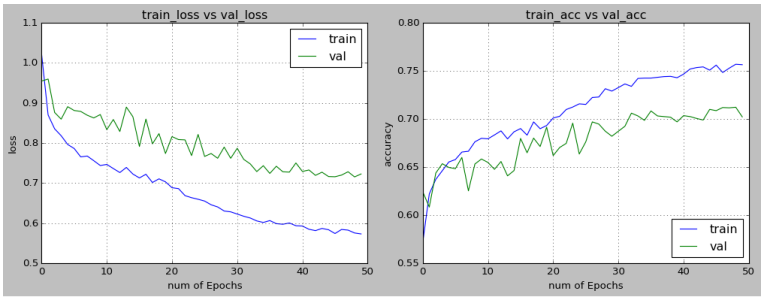

Figura 4.5: Entrenamiento de la red con transfer learning

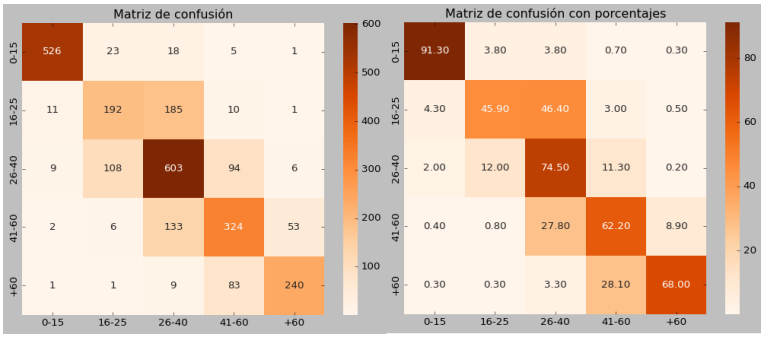

Figura 4.6: Matriz de confusión de la red con transfer learning

4.1.3.2. Red convolucional para la clasificación por edad

Para la red convolucional, que se puede observar en la figura 4.7, se utilizan 5 capas convolucionales en las que se aumenta el número de filtros progresivamente, después se aplica una capa *dropout* que sirve para mejorar la robustez de la red y reducir el sobreaprendizaje. Después se ajusta la salida a los 5 rangos de edad utilizando la activación *softmax*, que es la más indicada para la clasificación en distintas categorías.

La red se entrena en 60 épocas, como se puede observar en la figura 4.8, a partir de la época 40, empieza a producirse un ligero sobreaprendizaje. Al finalizar la red obtiene el 79 % de precisión sobre los datos de validación.

En la figura 4.9 se puede observar que el modelo logra una precisión total del 78 % , para la clase de entre 0 y 15 años, la precisión es del 96 % , como en los demás modelos, el rango de edad más problemático sigue siendo el comprendido entre 16 y 25 años, que obtiene un 62 %, para el grupo de entre 26 y 40 años, la precisión es del 76 %. También como en el resto de modelos el rango entre 41 y 60 se confunde con el rango de 26 y 40, obteniendo un 69 % de precisión. Por ultimo el rango de más de 60 años se obtiene casi un 80 % de precisión.

4.1.3.3. MobileNet

MobileNet [HZC+17] es una RNC que destaca por ser ligera y fácil de entrenar, obteniendo buenos resultados a pesar de esto. Para esto MobileNet usa capas convolucionales *depthwise*

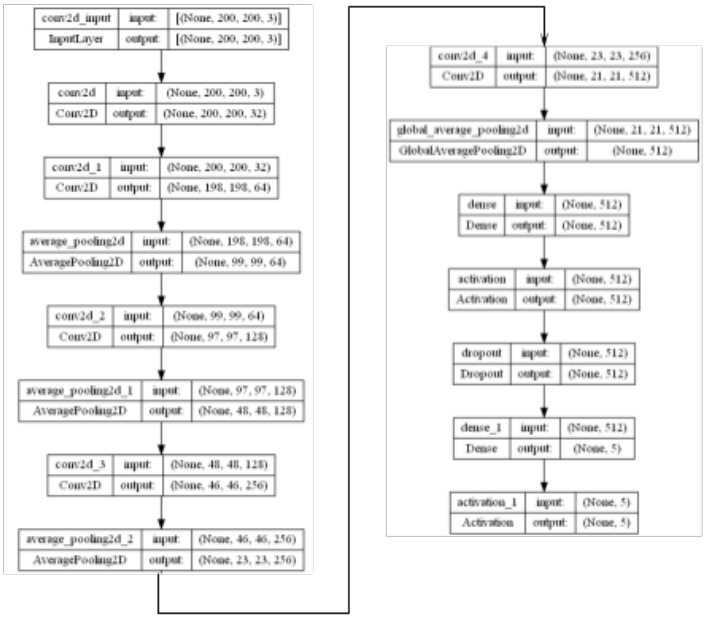

Figura 4.7: Modelo de la red convolucional para la clasificación por edad

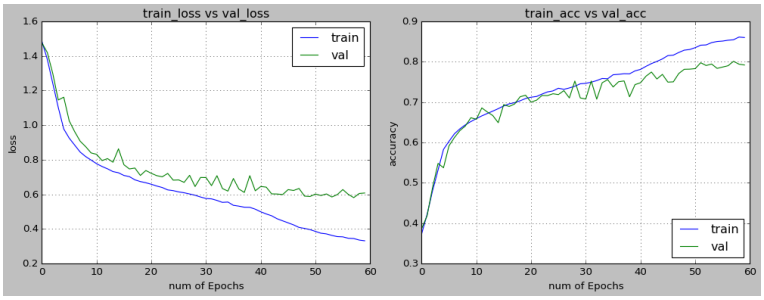

Figura 4.8: Entrenamiento de la red convolucional para la clasificación por edad

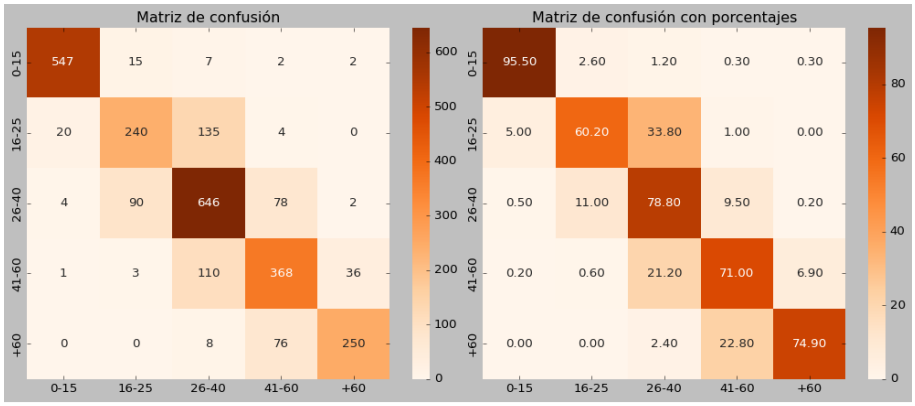

Figura 4.9: Matriz de confusión de la red convolucional para la clasificación por edad

o de profundidad, a diferencia de las capas convolucionales estándar, esta aplica un filtro a cada canal de entrada, posteriormente se aplica una convolución *pointwise* para combinar los resultados de cada canal, lo que reduce el número de parámetros de la red.

Lo único que se modifica de esta red es la salida, para adecuarla a la clasificación de los 5

grupos de edad. La red se entrena por 55 épocas, obteniendo un 0.77 de precisión sobre la validación. Como se puede observar en la figura 4.10, a partir de la época 40 se produce un ligero sobreaprendizaje.

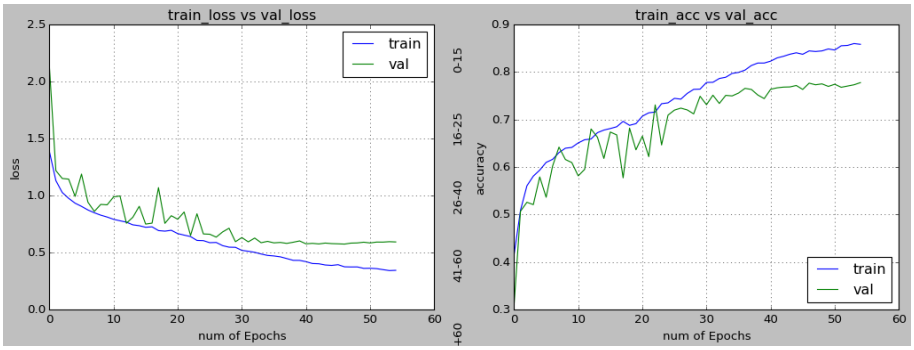

Figura 4.10: Entrenamiento la red MobileNet para la clasificación por edad

La matriz de confusión que se puede apreciar en la figura 4.11, muestra como la red distingue muy bien la clase entre 0 y 15 años, a la red como a las anteriores, tiene más problemas para distinguir entre el grupo de 16 y 25 años y el de 26 a 40 años, y entre este último y el de 41 y 60 años. En todas las clases obtiene una precisión superior al 60 %, logrando tambíen casi un 80 % en la clase de más de 60 años. En total la red obtiene una precisión del 78 %.

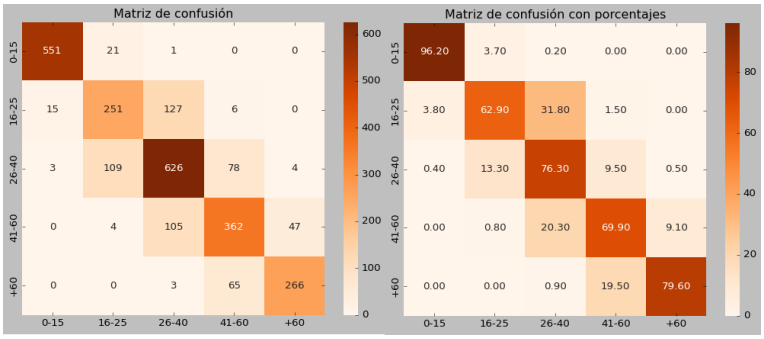

Figura 4.11: Matriz de confusión la red MobileNet para la clasificación por edad

4.2. Obtención del dataset no emparejado

Con lo realizado anteriormente, se puede realizar un *script* que para cada imagen generada por una red StyleGAN3, se determine si es válida o no, y a que grupo de edad pertenece. El primer problema encontrado es la diferencia de entrada de los datos del clasificador de edad, mostrado en la figura 4.12, y las imágenes producidas por StyleGAN3. Para ello se usará un detector de rostros preentrenado y se cortará la zona de la cara detectada.

Aplicando lo visto anteriormente, se generan las imágenes mediante StyleGAN3 con una truncación de 0.7, después el programa procesará cada una de las imágenes, con la RNC usada para mejorar la calidad y la red MobileNet del clasificador de edad. Se elige la red MobileNet ya que como se puede observar en los resultados de cada una de las redes y en la tabla 4.1 donde se presenta la precisión en cada uno de los grupos de edad, la red

Mobilenet es la que mejor distingue entre los dos grupos más conflictivos, el grupo de 16 a 25 años y el de 26 a 40 años.

Figura 4.12: Ejemplo del recorte para adecuar entrada al clasificador de edad

Tabla 4.1: Resultados clasificadores de edad

	Red convolucional	**VGG con Transfer Learning**	**Mobilenet**
0-15	0.955	0.913	0.962
16-25	0.602	0.459	0.629
26-40	0.788	0.745	0.763
41-60	0.710	0.622	0.699
+60	0.749	0.680	0.796
Exactitud	0.78	0.71	0.78

Si la salida del grupo de edad se corresponde al grupo de entre 0 y 15 años,la imagen se copia al *dataset* de personas jóvenes, si la salida es del grupo de edad es entre 16 y 25 años, descartamos la imagen, esto lo hacemos por dos motivos, el primero es para diferenciar algo más los dos datasets, y el segundo motivo para eliminar el grupo de edad más conflictivo.

4.3. Image translation

4.3.1. Trasformación de imágenes con datos no emparejados

A diferencia de los métodos con imágenes emparejadas, estos métodos tienen la ventaja de no necesitarlos, por lo que con el dataset generado anteriormente se puede entrenar a los modelos. En total se han generado 2.510 rostros de personas jóvenes y 5.784 de personas adultas.

4.3.1.1. CycleGAN

Para la implementación del modelo de CycleGAN se ha usado como base la proporcionada por Keras en su documentación, ajustando ciertas características al las del *paper* de CycleGAN, explicado en el Capítulo 3. Además también se ha adaptado la entrada de datos y se han modificado algunos ajustes para reducir el uso de memoria.

Al dataset de entrada se le aplica un aumento de datos, o en inglés *Data Augmentation*, para aumentar de una forma sencilla el número de ejemplos dataset de entrada, en este caso este se compone de dos operaciones, una que realiza volteos horizontales aleatorios y otra que corta la imagen de forma aleatoria. Esta ultima operación corta las imágenes de 140 x 140 a 128x128.

Usando el dataset especificado anteriormente se ha entrenado el modelo durante 130 épocas con un tamaño de *batch* de 1. Este entrenamiento llevó 16 horas de duración en mi ordenador.

Como se puede observar en la figura 4.13 el modelo no es capaz de realizar la transformación correctamente, generando formas raras e imágenes irreales. Esto ocurre en la mayoría de ejemplos de mujeres.

En la figura 4.14, se puede observar como en la transformación de personas con gafas, el modelo falla, generando imperfecciones sobretodo en el área alrededor del ojo. El modelo no es capaz de inferir la presencia de gafas.

Solo en algunos ejemplos, el modelo es capaz de realizar una transformación aceptable, como en los de las figura 4.15. Para solucionar este problema se decide dividir el dataset entre mujeres hombres y personas con y sin gafas, la idea es entrenar una red por cada una de las transformaciones posibles. Para ello habrá que ampliar el *script* anterior y crear una red nueva para el género de las personas en las imágenes y un algoritmo para saber si una persona tiene gafas.

Figura 4.13: Ejemplo de CycleGAN con mujeres

Figura 4.14: Ejemplo de CycleGAN con gafas

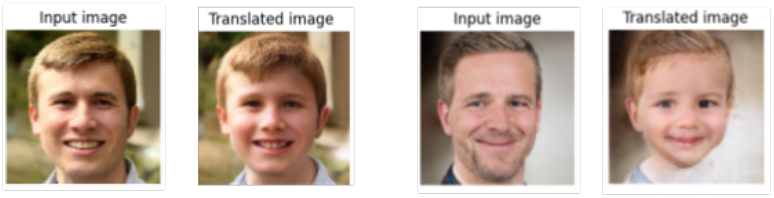

Figura 4.15: Ejemplo de CycleGAN con éxito

4.3.2. Dividiendo el dataset

4.3.2.1. Clasificación por género

Para entrenar este modelo se ha usado el dataset de UTK-Face usado anteriormente para la clasificación por edad, en total se compone de 21.883 imágenes para entrenamiento, de las cuales 11.437 se corresponden a hombres y 10.446 a mujeres. De estas imágenes un 10 por ciento se utiliza para la validación. Y se usan 1825 imágenes para el test del modelo.

Además se utilizará aumento de datos, para ampliar el número de datos de entrenamiento para el modelo, para ello haremos las siguientes operaciones aleatoriamente en las imágenes de entrada :

- Volteos horizontales
- Rotación de la imagen un máximo de 10 grados.
- Desplazamiento vertical y horizontal con un máximo de un 10 por ciento de la imagen.

El modelo consta 5 capas convolucionales de 64,128,256,512 y 512 conectada a una capa *GlobalPooling* para extraer las características y después conectado a una capa densa con un DropOut de 0.25, posteriormente conectada a la salida con función de activación *softmax*.

Para el entrenamiento se utiliza una tasa de aprendizaje de 0.001 y un tamaño de *batch* de 128, es decir 128 imágenes por iteración en el entrenamiento. El modelo se entrena durante 20 épocas y como se puede ver en la Figura 4.16 obtiene una precisión del 73 % en la validación. Sin embargo el modelo alcanza pronto un punto de alto sobreaprendizaje, por lo que se corta antes el entrenamiento y se usa la red entrenada en la época 6.

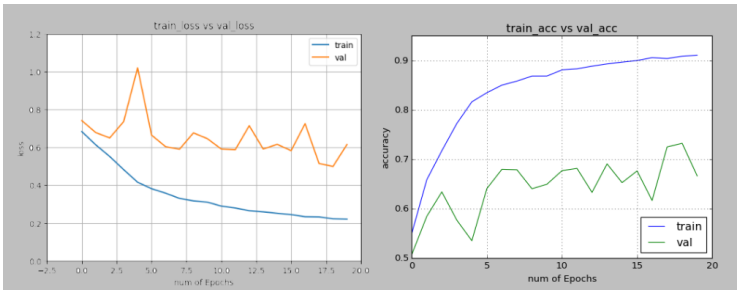

Figura 4.16: Entrenamiento de la red para la clasificación por género

Una vez entrenado se evalúa con las 1.825 imágenes de test, donde el modelo 4.17 obtiene una precisión en el género femenino del 88 % y en el género masculino del 86 %, en total el modelo obtiene un 87 % de precisión .

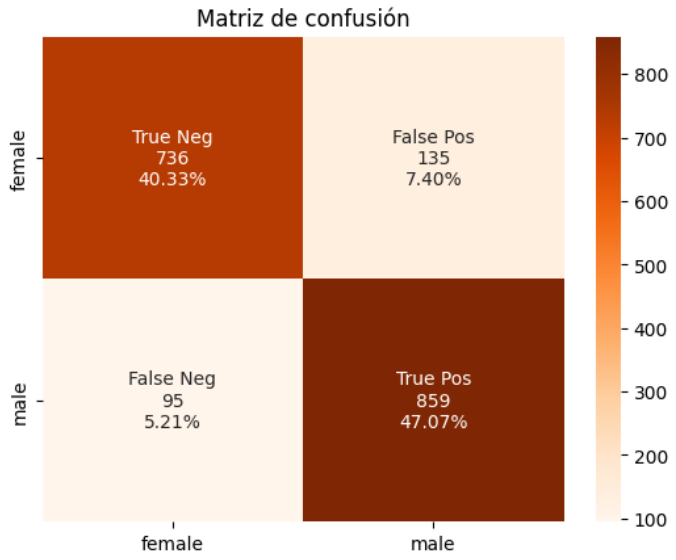

Figura 4.17: Matriz de confusión de la red para la clasificación por género

4.3.2.2. Detectando gafas

Para detectar si una persona tiene o no gafas, se utilizará MTCNN (*Multi-task Cascaded Convoluional Networks*) [ZZLQ16] , esto es un algoritmo de RNC que detecta las caras en una imagen como también los puntos claves de los rostros, posición de los ojos, nariz y boca . Después con la posición de ambos ojos se recortará la parte central, para así obtener la parte del medio de las gafas. Después se aplicará el algoritmo Canny [Ope] que se encuentra en la librería de OpenCV que detecta bordes en la imagen de entrada. Luego simplemente se comprobará si existe algún punto blanco en la imagen, si lo hay es que hay gafas 4.18.

Figura 4.18: Ejemplo de detección de gafas

4.3.3. Mejorando CycleGAN

Con lo realizado anteriormente ahora se tiene el dataset dividido en 8 grupos, divido por edad, género y si tienes gafas, por lo que se tendrá que entrenar 4 redes distintas. Además se realizarán unos cambios en el modelo respecto al anterior. Esto se basa en utilizar *Leaky ReLu* en todas las activaciones de cada bloque.

La principal diferencia entre *ReLu* y *LeakyRelu* [Ral23] es que la función en valores

negativos de entrada a la función *ReLu* devuelve 0 y *LeakyReLu* devuelve un pequeño valor negativo. Las principales ventajas de *LeakyReLu* respecto a *ReLu* es principalmente que se evita el problema que las neuronas mueran porque siempre reciben valores negativos, esto sobretodo ocurre en redes profundas. Además *LeakyReLu* ayuda a reducir el sobreaprendizaje y mejora la generalización.

Además se ha añadido una capa más tanto a la parte de *downsample* como a la de *upsample*. En los bloques residuales de CycleGAN he eliminado la capa de normalización. En la parte del discriminador solo se han modificado las funciones de activación a *LeakyReLu*.

Para el entrenamiento se utilizará un tamaño de *batch* de 4, aumentándolo un poco respecto al entrenamiento anterior, y también se va a entrenar durante más tiempo la red, ya que ahora no se tiene el problema con $ReLU$ del estancamiento del entrenamiento.

Por la dificultad de generar las imágenes sintéticas de personas jóvenes con gafas, la transformación de los adultos con gafas se hace a jóvenes sin ellas.

Capítulo 5

Resultados

A continuación se evaluación tanto la generación de los datos, con las distintas redes construidas para su clasificación, como la transformación mediante la red CycleGAN.

5.1. Generación de datos sintéticos

La generación de los datos parte de StyleGAN3, cuyos resultados son realmente buenos. El uso de 0.7 como valor de truncación provoca que las imágenes generadas tengan mejor calidad aunque una menos diversidad, sin embargo, esto no representa un problema en los datos generados.

Las redes para la clasificación por grupo de edad y por género (véase Tabla 5.1) obtienen una exactitud del 78 % y 87 % respectivamente por lo que introducen pocos errores en el dataset sintético.

Tabla 5.1: Resultados clasificador edad y género

	Precision	**Recall**	**F1-Score**
Clasificador edad	0.78	0.77	0.77
Clasificador género	0.87	0.87	0.87

A pesar de los distintos filtros, algunas imágenes de mala calidad o con una clasificación errónea pueden estar presentes en el dataset pudiéndose filtrarse a la transformación. Sin embargo estas imágenes representan un pequeño porcentaje de las imágenes totales generadas.

5.2. Evaluación de la transformación

Una vez entrenada la red CycleGAN especificada en el capítulo anterior, se evaluará a la red desde dos puntos. Un punto desde la evaluación visual de las imágenes generadas, y otro desde la métrica FID explicada en el capítulo 2.

5.2.1. Evaluación visual

Como podemos observar en las figuras 5.1 y 5.2 , el rejuvenecimiento tiene una buena calidad, la posición de la cara, ojos y boca se mantiene entre las transformaciones. En general la tonalidad del pelo o la piel se mantienen entre ambas imágenes.

Figura 5.1: Ejemplos de imágenes generadas de hombres

Figura 5.2: Ejemplos de imágenes generadas de mújeres

Sin embargo como podemos observar en las figuras 5.3 y 5.4 , en las imágenes con fondos difíciles, o imágenes donde el rostro tiene una posición más extraña, el modelo falla, creando imperfecciones y difuminaciones en la imagen resultante. Esto da a entender que la red puede estar sufriendo de algo de sobreaprendizaje.

5.2.2. Evaluación con el FID

Para evaluar el FID, hice una implementación simple que calcula el FID entre dos grupos de imágenes siguiendo lo especificado en el capítulo 2. Uno de los grupos son las imágenes

Figura 5.3: Ejemplos de imágenes de hombres con errores generadas

Figura 5.4: Ejemplos de imágenes de mujeres con errores generadas

transformadas y el otros son imágenes generadas por StyleGAN sin ningún procesamiento.

Los resultados (Tabla 5.2) arrojan un FID muy elevado, lo que indica que los grupos no son muy similares, si es verdad que en las imágenes generadas no hay mucha diversidad, la mayoría tienden a unos colores parecidos, aun manteniendo las características del rostro. La resolución baja, el uso de imágenes sintéticas y los errores de clasificación lastran el resultado del FID.

Tabla 5.2: Calculo del FID para los distintos modelos

Modelo	**FID**
Modelo de hombre adulto a hombre joven	291.4
Modelo de mujer adulta a mujer joven	237.1
Modelo de hombre con gafas a hombre joven sin gafas	281.1

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

El rejuvenecimiento facial trae dos problemas sustanciales consigo, uno que se hereda de la necesidad de los modelos de aprendizaje profundo y redes GAN que es la necesidad de datos extensos, para solucionar estos, se han visto tanto en los capítulos 2 y 3 diversos métodos que intentan solucionar este problema. Uno mediante el aumento de datos, mediante transformaciones sencillas o métodos mas sostificados como DAGAN, y el uso de imágenes sintéticas generadas por redes GAN. Durante el desarrollo del modelo y los distintos clasificadores, estos métodos han sido utilizados para mejorar el rendimiento de las redes y obtener un gran dataset de imágenes clasificadas en distintos grupos de edad para el entrenamiento del rejuvenecimiento facial.

Para la transformación de las imágenes se ha utilizado CycleGAN, que nos proporciona un método no emparejado para poder entrenarlo con el *dataset* generado. En la gran mayoría de las imágenes generadas se mantienen las características básicas del rostro que se está transformando,a pesar de los malos resultados del FID, debido a la ciclo consistencia de CycleGAN explicada en el Capítulo 3 la transformación tiene una buena calidad.

Sin embargo, no todas las imágenes se asemejan a la realidad. El hecho de usar una baja resolución hace que perdamos detalles significantes en algunas imágenes. A su vez, el uso de un *dataset* sintético provoca que no todos las imágenes de entrenamiento sean fidedignas, ya sea en su generación tenga imperfecciones, o que la clasificación de esa imagen en los distintos grupos no sea correcta.

6.2. Trabajo futuro

Como trabajo futuro se proponen varias líneas a seguir :

- **Mejorar la calidad de las imágenes rejuvenecidas** Para mejorar la calidad se propone el uso de una red que aumente la resolución y la calidad de las imágenes generadas. Al entrenar la red con una resolución baja de 128 píxeles por 128 píxeles, se pierden detalles y calidad. Para solucionar esto se puede usar una red GAN para aumentar la resolución, como lo propuesto en [LTH+17] , que se podría añadir a nuestro modelo. Esto implicaría mayor uso de recursos en el entrenamiento.

- **Eliminar el fondo** Como hemos visto en los resultados, los fondos complejos dificultan la transformación de las imágenes, algunos de los métodos presentados en el Capítulo 3 realizan el entrenamiento sin fondo. Para eliminar los fondos de las imágenes podemos usar la segmentación, por ejemplo un modelo como U2Net [QZH+20] para eliminar el fondo, y entrenar el modelo con imágenes de fondo blanco o gris.

- **Incorporar ejemplos reales al *dataset* sintético** El *dataset* con el que se ha entrenado la red CycleGAN solo se compone de imágenes sintéticas, como hemos visto en DatasetGAN [FKAL22] en el Capítulo 3, el rendimiento del uso de ejemplos no reales mezclado con ejemplos reales mejora a solamente utilizar ejemplos no reales.

Capítulo 7

Introduction

7.1. Motivation

Facial rejuvenation is the technique by which, given a face, it is transformed to another one of a certain age. During the last decade there have been different methods to perform this aging, from those that reduced wrinkles or modified the texture of the face. Not until the arrival of GAN has there been a real breakthrough in this field.

GAN require a large amount of data, in certain fields it may not be easy to gather this data, especially in situations such as facial rejuvenation where data from the same face but at different ages are needed.Therefore, several articles have dealt with this problem, from the exploration of the latent space of GAN to the construction of models that do not require paired data, or even using synthetic images and data augmentation.

7.2. Object of the Investigation

Rejuvenating a face in an image involves several problems, the first is to get an artificial intelligence, in our case a GAN, which is capable of inferring which features of a person are preserved from when we are children to adults, characteristics as eye color or skin color for example. And others as wrinkles or signs of aging in the skin that clearly change with the years. Also, getting labeled images of the same person as a child and as an adult with a decent quality can be problematic due to the lack of data in these areas.

This work will aproach a model with GANs that allows us to rejuvenate a given face, keeping the basic features, that will also solve the problem of the lack of paired data by using artificial intelligence.

In addition, we will also investigate different metrics that will allow us to improve and evaluate the model and improve the quality of the transformed images and the quality of the aging itself.

7.3. Workplan

This work has been developed in three phases:

1. **Research** At the beginning of the work, a meeting was held where the basic knowledge to start the work was explained, and what knowledge about GANs networks should be obtained for the completion of the work. It was also explained the knowledge about Keras, PyTorch and Python that would be needed later on. In addition, the final objectives of the work and the context in which it would be carried out we're explained. It was agreed to make a weekly follow up of the project, where we would explain the progress made during the week and any question or problems that may appear.

   During the first three months the knowledge about GANs and CNN was adquired by reading numerous articles and books, and we also started testing some pre-trained GANs. Also the direction of the work was chosen and what would be developed later on.

2. **Development**: With the basic knowledge adquired,we started to prepare the models that would be used later on, obtaining a higher level in the use of *TensorFlow* and *Keras* libraries. We also started to collect and build the training data. During this time, the investigations of more scientific papers did not stop.

3. **Results**: Once I had the models built and the training data prepared, I started with the training and tuning of the networks. Different models were tested and those with better results were chosen. All the results were obtained and analyzed. The research was stopped at this stage.

7.4. Struture of the Work

The rest of the work is organized in 8 chapters with the following structure:

Chapter 1 is an introduction of this work and the work plan traduced from this chapter.

Chapter 2 introduces the basic concepts of convolutional networks, GAN networks and image translation. Distinguishing between the transformation of paired data with unpaired data. It also explains the StyleGAN network [KLA19] [KAL+21] and differents metrics for the evaluation of GAN networks.

Chapter 3 shows some of the most complex models in the field of GAN, for example CycleGAN [ZPIE20] as well as models that go more in depth into the lack of data, such as DAGAN [ASE18] or DatasetGAN [FKAL22], all of them together with their results.

Chapter 4 presents the development of the work, showing how the different models for face classification have been obtained for the creation of the synthetic dataset.Afterwards it is show how the CycleGAN-based model is built and tuned for the rejuvenation of faces.

Chapter 5 shows the results obtained from the models built in 4.

Chapter 7 shows conclusions of this work and the main lines to be followed as future work of this project.

Chapters 8 and 6 are the English translations of the Introduction and the Conclusions.

Capítulo 8

Conclusions and Future Work

8.1. Conclusions

Aging brings two major problems, one that is inherited of the necessity of extensive data for deep learning and GAN models, to solve these problems, we have seen in chapters 2 and 3 several methods that try to solve this problem. One is using data augmentation, through simple transformations or more sophisticated methods such as DAGAN, and the other is the use of synthetic images generated by GAN networks. During the development of the model and the different classifiers, these methods have been applied to improve the performance of the networks and to obtain a large dataset of images classified into different age groups for facial rejuvenation training.

CycleGAN has been used for the transformation of the images, which provides us with an unpaired method to train it with the generated dataset. In the great majority of images, the features of the face that is being transformed are kept, despite the low FID score, due to the cycle consistency explained in the chapter 3 with a good quality.

However, not all the images correspond to reality. The fact of using a low resolution makes us lose significant details in some images. At the same time, using a synthetic dataset causes that not all the training images be faithful, either in it's generation has imperfections, or that the classification of this image in the different groups is not correct.

8.2. Future Work

As future works, several lines of investigation are proposed:

- **Improving the quality of rejuvenated images** In order to improve the quality, the use of a network is proposed. The network increases the resolution and quality of the generated images. Training the network with a low resolution as 128 pixels per 128 pixels, we lose details and quality. To solve this problem we can use a GAN network to increase the resolution, as proposed in [LTH+17], that can be attached to our model. This would involve an increase use of resources in the training.
- **Background removal** As we have seen in the results, complex backgrounds make it harder to transform images, some of the methods presented in Chapter 3 perform backgroundless training. For this reason, we can use semantic segmentation, for

example a model like U2Net [QZH+20] to remove background, and train the model with white background images.

- **Adding real examples to the synthetic dataset** The dataset on which the CycleGAN network has been trained only consists of synthetic images, as we have seen in DatasetGAN [FKAL22] in Chapter 3, the performance of using non-real examples mixed with real examples improves over just using non-real examples.

Bibliografía

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[ASE18] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks, 2018.

[Bor18] Ali Borji. Pros and cons of gan evaluation measures, 2018.

[Bro19a] J. Brownlee. *Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation.* Machine Learning Mastery, 2019.

[Bro19b] Jason Brownlee. A gentle introduction to transfer learning for deep learning, Sep 2019.

[BSAG21] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021.

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[FKAL22] Zong Fan, Varun Kelkar, Mark A. Anastasio, and Hua Li. Application of datasetgan in medical imaging: preliminary studies, 2022.

[GPAM+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[HB17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017.

[HLBK18] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation, 2018.

[HRU+18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[HZC+17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[IBM]

[IZZE18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[KAH+20] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020.

[KAL+21] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks, 2021.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[KLA+20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020.

[KNH] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[LTH+17] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017.

[LXZZ17] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets, 2017.

[NWC+11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[Ope]

[PW17] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning, 2017.

[QZH+20] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. 106:107404, oct 2020.

[Ral23] Srikari Rallabandi. Activation functions: Relu vs. leaky relu, Mar 2023.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[SZG+18] Jingkuan Song, Jingqiu Zhang, Lianli Gao, Xianglong Liu, and Heng Tao Shen. Dual conditional gans for face aging and rejuvenation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 899–905. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[TBS+21] Amirhosein Toosi, Andrea G. Bottino, Babak Saboury, Eliot Siegel, and Arman Rahmim. A brief history of AI: How to prevent another winter (a critical review). *PET Clinics*, 16(4):449–469, oct 2021.

[ZPIE20] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.

[ZWD21] Yihao Zhao, Ruihai Wu, and Hao Dong. Unpaired image-to-image translation using adversarial consistency loss, 2021.

[ZZLQ16] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, oct 2016.