ESCUELA DE INGENIERÍA DE FUENLABRADA

GRADO EN INGENIERÍA EN SISTEMAS AUDIOVISUALES Y MULTIMEDIA

TRABAJO FIN DE GRADO

RECONOCIMIENTO FACIAL A TRAVÉS DE DEEP LEARNING

Autor: Iván Pérez Lorenzo

Tutor: Jorge Calero Sanz

Curso académico 2022 / 2023

Agradecimientos

En primer lugar, quiero agradecer a mi tutor Jorge Calero Sanz por ayudarme en este proyecto. Ha sido un trabajo que ha requerido mucho tiempo y quebraderos de cabeza. A su vez también quería agradecer a mis padres, Nicolas Pérez Antón y M.ª Del Carmen Lorenzo Ballesteros por haberme proporcionado las facilidades para que pueda estar escribiendo esto hoy en día y por haberme apoyado desde el día en que nací. Además de agradecerle a mi hermano David Pérez Lorenzo por pincharme a que funcionará mejor, estoy satisfecho de haber luchado por una base de datos más amplia y no haberme conformado con algo sencillo. Y por ultimo y no menos importante, agradecer a mis amigos, en especial a los que se han ofrecido a prestarme sus caras para mi base de datos pese a la vergüenza que pudiese acarrear, sus nombres son Alexia García, Lucia Mezquita, Gabriela Maxwell, Rafael Sleam , Oscar Esteban, Raúl Izquierdo, Samuel Maroto, Sulaiman El Azouan, Roberto Ramos, Álvaro Fernández, Bea López, Ramon Ye, Paula Carranza, Lylia Godino, Lucia San Miguel, Víctor Buenadicha, Lourdes Lozano, Sara Fernández, Jorge Alcácer, Álvaro Sánchez y Lorena Martínez.

Resumen

El reconocimiento facial ha ganado un gran interés en los últimos años debido a su amplia gama de aplicaciones en diversas industrias, incluyendo seguridad, vigilancia, autenticación y entretenimiento. El aprendizaje profundo ha revolucionado el campo del reconocimiento facial al proporcionar métodos altamente eficientes y precisos para la detección y clasificación de objetos en imágenes. La detección de caras es una tarea crítica en el reconocimiento facial, ya que sirve como paso inicial para identificar y localizar rostros en una imagen.

El enfoque principal de este trabajo de fin de grado (TFG) se centra en el uso del aprendizaje profundo (deep learning) para la detección de caras en imágenes. Se explora y analiza el estado del arte en técnicas de detección de caras basadas en deep learning, centrándose en arquitecturas de redes neuronales convolucionales (CNNs) especialmente diseñadas para esta tarea. Se revisan diferentes enfoques, evaluando su rendimiento en términos de precisión, velocidad de procesamiento y capacidad de detección en diferentes condiciones, como variaciones de iluminación, poses y oclusiones parciales.

El TFG también incluye la implementación y evaluación de un sistema de detección de caras basado en deep learning utilizando una arquitectura VGG-16. Se describen las etapas del proceso, desde la recopilación y preparación de los datos hasta la evaluación de los resultados obtenidos. Se analizan las métricas de rendimiento, como precisión, recall y matriz de confusión, para evaluar la efectividad del modelo propuesto.

Índice general

1. Introducción

1.1 Motivación, hipótesis y objetivo

El objetivo principal de este trabajo es desarrollar un software capaz de llevar a cabo el tratamiento de un video que tomará el video de una cámara en la que aparece una persona con la cara descubierta, el sistema tendrá que ser capaz de detectar a través de una red neuronal convolucional la cara que aparezca en el video de la cámara dibujando alrededor suyo un rectángulo.

Como estudiante de Ingeniería de Sistemas Audiovisuales, tengo la motivación de aprender a crear proyectos autómatas que sean capaces de procesar imágenes, videos y sonidos para el desarrollo del marco tecnológico en ámbitos como la seguridad, la restauración, el propio procesamiento y la extracción de características ya que ello permite realizar la elaboración de sistemas autómatas que faciliten la carga de trabajo de las personas.

1.2 Inteligencia Artificial

La inteligencia artificial [1] es un campo de la informática y la ingeniería que se centra en la creación de sistemas y tecnologías que pueden realizar tareas que normalmente requieren inteligencia humana, como el aprendizaje, el razonamiento y la resolución de problemas. Una de las idead de la inteligencia artificial se basa en que las máquinas pueden aprender de la experiencia y mejorar en el tiempo, de manera similar a cómo lo hacen los seres humanos. Esta tecnología ha avanzado enormemente en los últimos años, lo que ha permitido el desarrollo de aplicaciones y sistemas que antes se consideraban imposibles. Desde el reconocimiento facial hasta la identificación de patrones en grandes conjuntos de datos, la IA está transformando la manera en que trabajamos y vivimos.

Pese a que la idea de la inteligencia artificial aparece ya en documentos del siglo III a.C [2], donde se describe como "*un artífice automático mecánico"*. Las bases de la inteligencia artificial no se asientan como disciplina científica hasta que Warren McCulloch y Walter Pitts publican un artículo en 1943 titulado "*A logical calculus of the ideas immanent in nervous activity. Bulletin of*

Mathematical Biophysics" [3]**,** en el cual se propone un modelo matemático que simula la interacción entre neuronas del cerebro. Otro trabajo importante que cabe recalcar fue escrito por Alan Turing en 1950, titulado "*Computing Machinery and Intelligence"* [4] en el cual se plantea la idea de que las máquinas puedan a llegar a pensar por sí mismas algún día. También se menciona el famoso Test de Turing [5], que todavía es una de las pruebas más populares para inteligencia artificial. Posteriormente, en la década de los 60, su enfoque se desvió al razonamiento simbólico, llegando a desarrollar programas capaces de manipular símbolos y realizar interferencias lógicas. En los 70 y 80 la presentación del conocimiento devoró el área de investigación de la inteligencia artificial aplicándose en diversas áreas, como por ejemplo la ingeniería. En parte de la década de los 80 también se observó un interés por el área del razonamiento probabilístico destacando los modelos y redes bayesianos desempeñaron un papel importante.

A medida que avanzaba el campo, se hicieron intentos de desarrollar sistemas que pudieran aprender de manera autónoma. Los algoritmos genéticos, inspirados en la evolución biológica, permitían la optimización y la búsqueda de soluciones en problemas complejos. A lo largo de las décadas siguientes, la IA continuó evolucionando con avances en diversas áreas como la visión por computadora, el procesamiento del lenguaje natural y los sistemas basados en cookies, logrado, gracias a algoritmos que presentaban técnicas capaces de reconocer patrones.

1.3 Machine Learning

El machine learning [6] (aprendizaje automático) es un subcampo de la inteligencia artificial que se enfoca en desarrollar algoritmos y modelos que permiten a los sistemas informáticos aprender y mejorar de manera autónoma a partir de los datos. su inicio se remonta a 1952, por el profesor e informático Arthur Samuel [7] que desarrolló un software capaz de jugar a las damas. A diferencia de los algoritmos tradicionales, que están diseñados para seguir reglas explícitas, los algoritmos de machine learning son capaces analizar grandes cantidades de datos para detectar patrones y relaciones complejas, lo que les permite tomar decisiones y hacer predicciones con una precisión cada vez mayor.

Entre las aplicaciones del machine learning podemos encontrar desde el análisis de riesgo financiero y la detección de fraudes, hasta la clasificación de imágenes y el procesamiento de lenguaje natural. Gracias al aprendizaje automático, los sistemas informáticos pueden realizar

tareas de manera más eficiente y efectiva, lo que está transformando la forma en que las empresas y los individuos interactúan con la tecnología.

En el año 1958 Frank Rosenblatt, psicólogo y científico de la computación, desarrolló el modelo de una neurona artificial como se muestra en la figura 1, denominado perceptrón [8], una de las primeras y más simples redes neuronales artificiales diseñada originalmente para reconocer patrones en imágenes. Considerado un hito importante pese a estar muy limitado, este modelo fue capaz de distinguir imágenes entre hombres y mujeres utilizando un sistema de clasificación binaria.

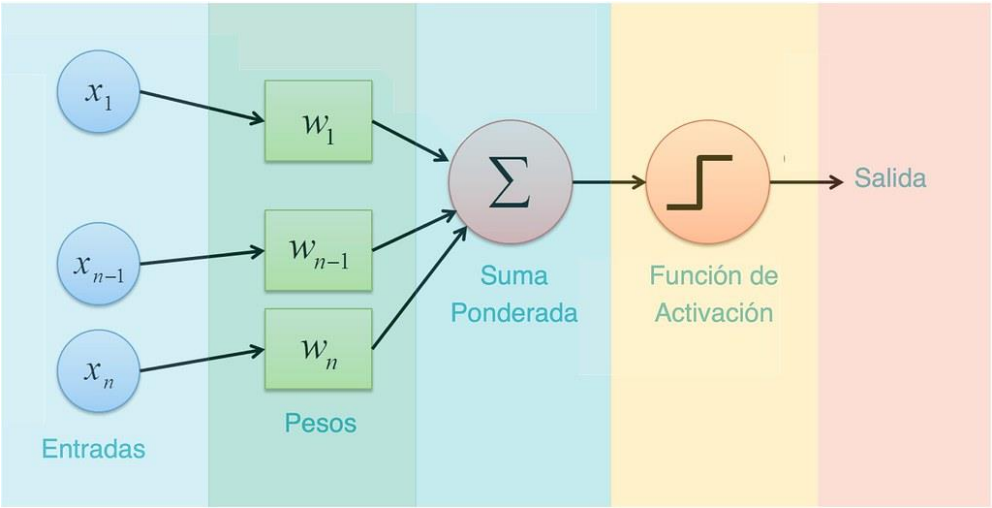

Figura 1. Organización de un perceptrón el cual consta de una entrada, que sufre una operación para obtener una salida y que dependiendo de la función de activación tendrá un umbral distinto para determinar la salida final [9].

Esta idea se quedaría estancada debido a la falta de financiación además de la falta de bases de datos y capacidad computacional. No sería hasta 1979 cuando un grupo de estudiantes de la universidad de Stanford crearían un robot denominado "Stanford Car" [10] capaz de desplazarse sorteando obstáculos. Este robot utilizaba un algoritmo denominado Nearest Neighbor, algoritmo capaz de reconocer patrones. Esta se convertirá en la herramienta principal que provocará el resurgir del machine learning, revolucionando el procesamiento de datos de la década de los 80 y permitiendo el desarrollo de otros modelos importantes en este campo. Uno de ellos sería "Explanation Base Learning" en 1981 por Gerald Dejong, modelo que no solo permitía trabajar

con las variables ingresadas, sino que también permitía ingresar nuevas para la formulación de sus propias variables. Otro modelo es “NetTalk”, creado en 1985 de Terry Sejnowski, profesor e informático teórico, NetTalk se compondría de un algoritmo capaz de saber la pronunciación de las palabras a niveles escolares.

Tras esta primera etapa, el desarrollo del machine learning sufriría un estancamiento. En 1997, el ordenador Deep Blue, de IBM, ganó a Gary Kaspárov, campeón mundial de ajedrez, mostrando una gran capacidad de adaptación a los movimientos del humano.

En 2006 el machine learning iniciaría una etapa de avances en manejo y procesamiento de datos, impulsado por grandes empresas como IBM y Microsoft, expandiéndose a nivel global. En 2008 Microsoft desarrolló el programa Azure machine learning brindando un servicio en la nube. Años más tarde, IBM culminaría con su ordenador Watson que proporcionaría más fama a esta rama tecnológica gracias a su participación en un concurso norteamericano, Jeopardy, el cual consistía en responder preguntas, venció a sus rivales humanos consiguiendo una excelente puntuación.

En 2007 Jeff Dean, empleado de Google y Andrew N, profesor de la universidad de Stanford, llevaron a cabo un proyecto de Google denominado Google Brain, desarrollando así la primera red neuronal profunda capaz de detectar patrones en vídeos e imágenes consolidando así una de las grandes maravillas del machine learning. A raíz de este momento se crearon programas como DeepFace por Facebook en 2015, DeepMind (empresa que más tarde sería comprada por Google), que crearía un software capaz de jugar a la videoconsola Atari siendo capaz de vencer a profesionales del videojuego. Con el tiempo otras compañías invertirían en el machine learning, Microsoft innovó lanzando *“Distributed Machine Learning Toolkit”*, una herramienta para compartir programas y problemas de machine learning. En el mismo año, Elon Musk y Sam Altman entre otros fundarían OpenAI, incentivando la investigación y el avance en este campo.

1.3.1 Tipos de Aprendizaje en Machine Learning

En la era actual de la tecnología, el campo del Machine Learning (Aprendizaje Automático) se ha convertido en un componente clave para la resolución de problemas complejos y la toma de decisiones inteligentes en diversas industrias. El Machine Learning se enfoca en el desarrollo de algoritmos y modelos que dan pie a que las máquinas aprendan de datos y sean capaces de mejorar

su desempeño sin una programación explícita. En este fascinante mundo del aprendizaje automático, existen varios tipos de aprendizaje, cada uno con sus propias características y aplicaciones [11].

1. Aprendizaje supervisado (Supervised Learning): En el aprendizaje supervisado, se proporcionan a la máquina conjuntos de datos etiquetados, es decir, con información que indica la respuesta correcta para cada ejemplo de entrenamiento. El objetivo es entrenar un modelo que pueda predecir la etiqueta correcta para nuevos datos. El aprendizaje supervisado se utiliza para problemas de clasificación (en los que se deben predecir categorías discretas) y de regresión (en los que se deben predecir valores continuos).

2. Aprendizaje no supervisado (Unsupervised Learning): En el aprendizaje no supervisado, se proporcionan a la máquina conjuntos de datos no etiquetados, es decir, sin información que indique a que categoría corresponden. El objetivo es descubrir patrones o estructuras ocultas en los datos. El aprendizaje no supervisado se utiliza principalmente para problemas de clustering (en los que se deben agrupar los datos en conjuntos similares) y de reducción de dimensionalidad (en los que se deben encontrar las características más importantes de los datos).

3. Aprendizaje por refuerzo (Reinforcement Learning): En el aprendizaje por refuerzo, la máquina interactúa con un entorno dinámico y recibe una retroalimentación en forma de recompensas o castigos por las acciones que realiza. El objetivo es aprender una política que permita a la máquina maximizar la recompensa total a largo plazo. El aprendizaje por refuerzo se utiliza en problemas de toma de decisiones, como el control de robots y la optimización de sistemas.

A continuación, veremos un modelo muy concreto e importante del machine learning, que nos abrirá paso a entender la base de este trabajo, las redes neuronales.

1.4 Redes Neuronales

Las redes neuronales [12] son modelos computacionales que se inspiran en el cerebro humano, diseñados para procesar información y aprender de ella. Consiste en un conjunto interconectado

de nodos, llamados neuronas, que trabajan en conjunto para realizar una tarea específica, como reconocimiento de patrones, clasificación de datos o predicción de resultados. Son capaces de aprender de los datos de entrada y mejorar su rendimiento a medida que procesan más información. Están formadas por capas de neuronas interconectadas, y cada neurona se comunica con las demás a través de conexiones ponderadas. En su forma más simple cada neurona es un perceptrón [13].

El perceptrón simple puede utilizarse para clasificar datos que se pueden separar linealmente como podemos ver en la figura 2, en dos categorías.

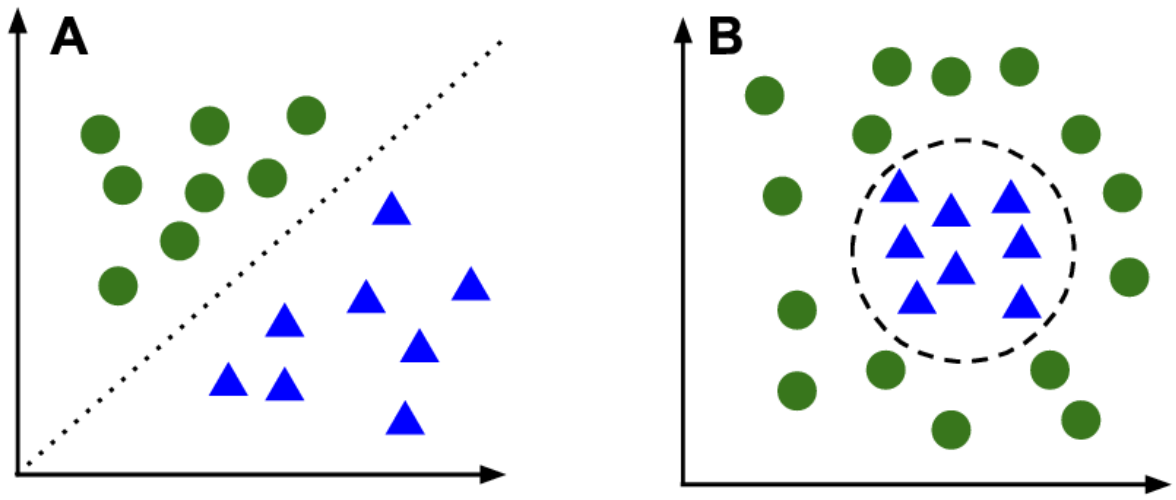

Figura 2. Comparación entre dos clasificadores, el clasificador A que es lineal y el clasificador B que no lo es, se aprecia la división que realizaría el algoritmo (líneas discontinuas) [14].

Por ejemplo, puede ser utilizado para separar imágenes en función de dos salidas, los círculos indicarían una clase como puede ser que haya cara y los triángulos otra clase como puede ser que no haya cara.

El perceptrón simple consiste en un vector de pesos, que se multiplica escalarmente por el vector de entrada (la entrada se vectoriza para que las operaciones sean mucho más rápidas y el coste computacional mucho menor), y un umbral que se decide con la función de activación determina si el resultado de la multiplicación es mayor o menor que un cierto valor. Como ejemplo, si el resultado es mayor que el umbral, el perceptrón devuelve un 1, lo que indica que el objeto

pertenece a una clase, y si es menor que el umbral, devuelve un 0, lo que indica que el objeto pertenece a la otra clase.

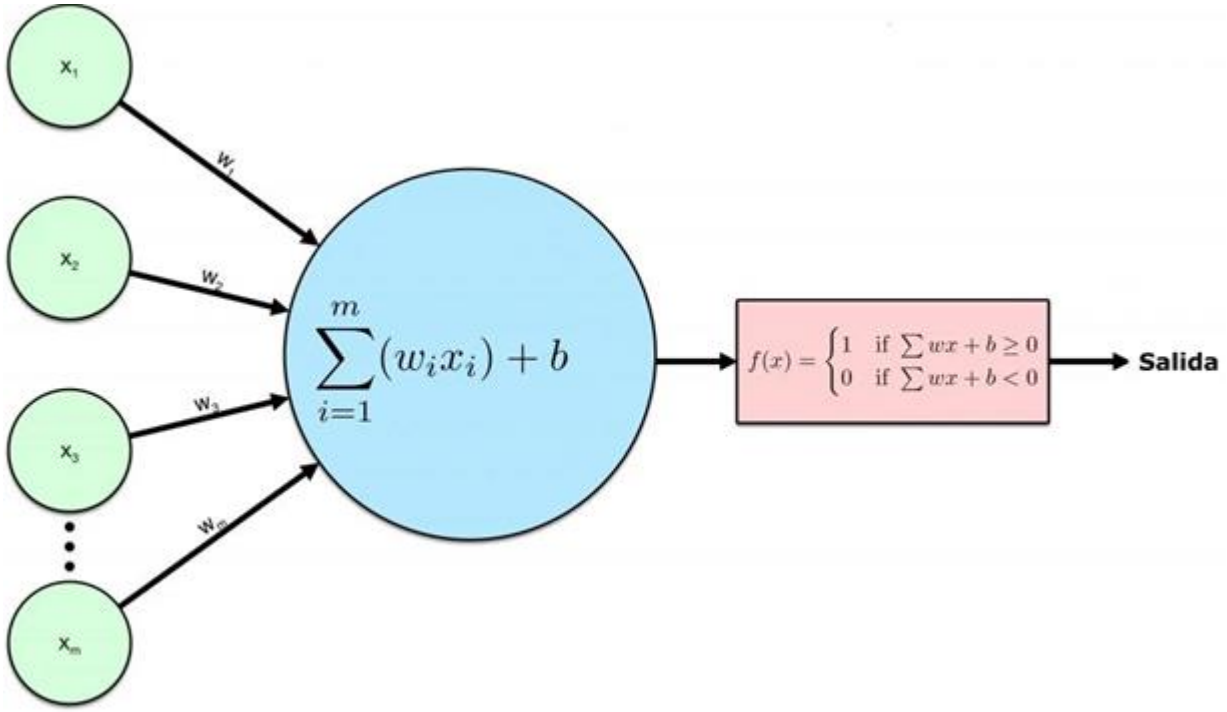

Figura 3. Estructura perceptrón simple junto con la operación que se lleva a cabo en la capa y su posterior procesamiento para la selección de clase [15].

Por otro lado, los perceptrones complejos son redes neuronales artificiales más avanzadas que constan de múltiples capas de neuronas interconectadas (perceptrones multicapa). Estas redes pueden utilizarse para resolver problemas más complejos, como pueden ser la clasificación de imágenes o el reconocimiento de voz. La red se interconecta mediante las entradas y salidas de las neuronas entre sí, la salida de una neurona será la entrada de las siguientes.

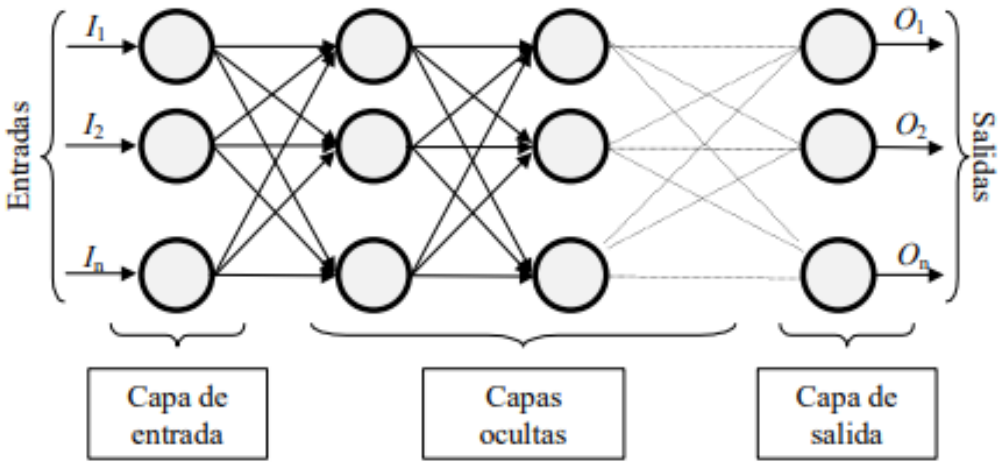

Figura 4: Estructura de un perceptrón multicapa con varias capas ocultas, la capa de entrada y la capa de salida, todas de tamaño variable [16].

Conociendo la estructura, lo datos ingresarán por la capa de entrada propagándose hacia delante mediante las capas ocultas y finalizarán el recorrido saliendo por la capa de salida, de manera similar a como trabajan las neuronas biológicas. Una de las mejores ventajas de las redes neuronales es que es posible cambiar el tamaño de la red a placer en función del problema que se busque resolver, redes neuronales grandes podrán resolver problemas mucho más complejos que redes neuronales más pequeñas, aunque exigirán mayor tiempo de cómputo.

Función de activación: Con ella se puede calcular el estado de actividad de una neurona mediante la transformación de la entrada. Existen varias funciones de activación, entre ellas las más importante son la función de activación Sigmoidal, la función de activación Hiperbólica (tanh) y la función de activación ReLu. La elección de la función de activación dependerá del problema que queramos resolver ya que cada una presenta ventajas diferentes [17]. A continuación, se explicará con más detalle algunos de las distintas funciones de activación:

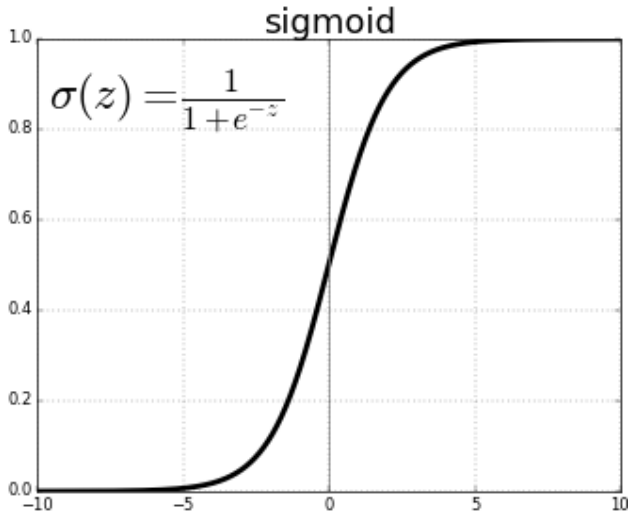

Figura 5: Función de activación Sigmoide [18].

La función de activación Sigmoidal es una función no lineal comprendida entre los valores verticales de 0 y 1(limitando los valores muy bajos y altos), es decir que se obtiene una salida de 2 clases, una salida binaria.

La función de activación ReLu (Rectified Linear Unit), es una función no lineal la cual convierte los valores negativos en 0 y los positivos en el mismo valor que el de la entrada evitando así el problema de saturación con un rango de valores entre 0 y el máximo valor de entrada aumentando la convergencia del algoritmo.

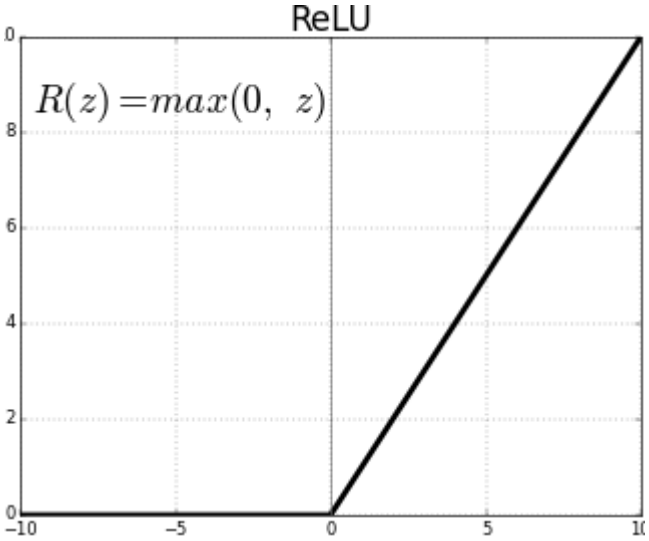

Figura 6. Función de activación ReLU [12].

Se suele utilizar la función de activación ReLu como función de activación de las capas ocultas mientras que la función Sigmoidal para la última capa, es decir para la capa de salida, en la que se lleva a cabo la asignación de clases.

En resumen, las redes neuronales se pueden entender como la similitud del perceptrón simple con una neurona, siendo una unidad que realiza una operación independiente que consta de una entrada y una salida. Y al perceptrón multicapa como el cerebro, que se forma a través de esa conexión múltiple de un conjunto de neuronas.

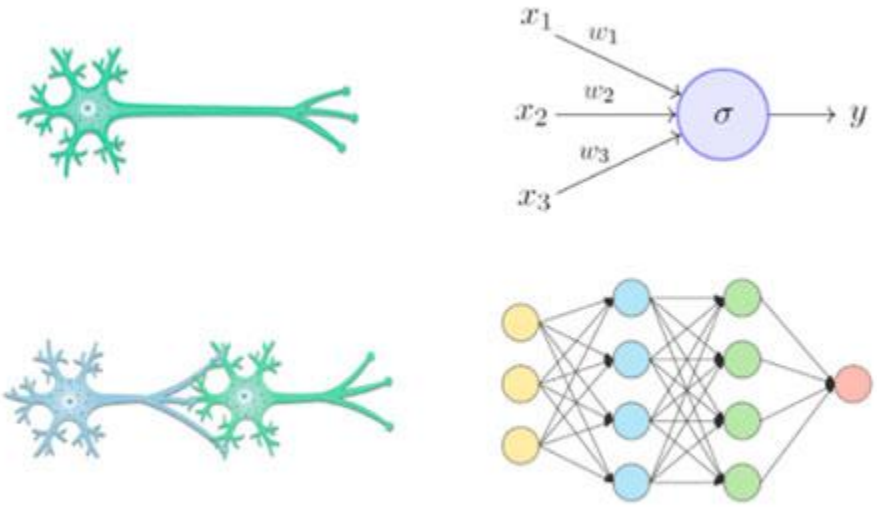

Figura 7. Comparación de la similitud entre los perceptrones simples y multicapas con las neuronas humanas [19].

1.4.1 Hiperparámetros

Otro parte importante que constituye el funcionamiento principal de las redes neuronales son los hiperparámetros [20]. A diferencia de los parámetros que son las variables de nuestro algoritmo que se van modificando durante el proceso de aprendizaje de manera automática (los pesos de las redes neuronales y el sesgo), los hiperparámetros son variables sensibles que podemos y que al hacerlo provocará una transformación importante en la red y en su funcionamiento. Algunos

ejemplos de hiperparámetros son: el número de capas ocultas por columna(filas), y el número de columnas o la selección de las funciones de activación.

A la hora de desarrollar una red neuronal, tenemos que saber concienzudamente que problema tenemos que resolver, y partir de ello plasmar una primera idea inicial en la cual elegiremos nuestra arquitectura y nuestra selección de hiperparámetros. A continuación, implementaremos esta idea en el código. Una vez se haya diseñado el programa se experimentará con el problema. Este proceso se visualiza en la figura 7. Cuando obtengamos los resultados valoraremos que estén dentro de nuestro objetivo óptimo, y si no es así, volveremos a cambiar los hiperparámetros hasta dar con un algoritmo que se ajuste de la mejor manera a nuestro problema en cuestión.

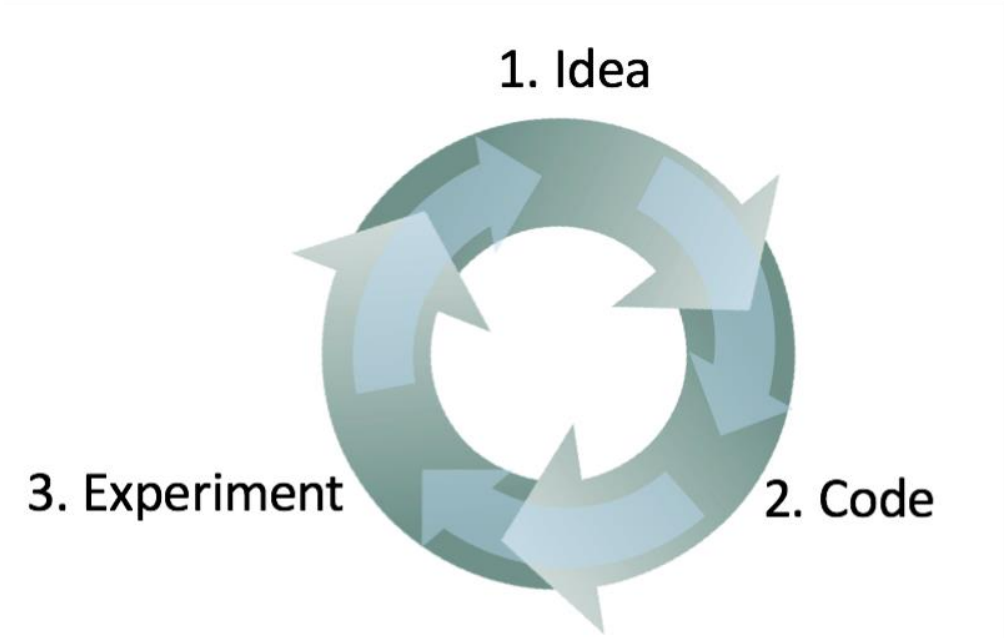

Figura 8. Estrategia para proyectos de machine learning y la selección de los hiperparámetros la cual consiste en tener una primera idea (valor), implementarla y experimentar con ella para después repetir el proceso hasta obtener los resultados óptimos [21].

Los hiperparámetros son valores que guían nuestro algoritmo, que guían el aprendizaje de nuestra red neuronal. Los hiperparámetros que vamos a tener en cuenta para nuestro algoritmo son:

1. Número de las capas ocultas. El número de capas ocultas que realizan las operaciones

2. Función de activación. Se trata de un rango de valores el cual nos puede devolver una neurona al realizarle una transformación a la entrada.

3. Inicialización de los pesos. Matriz que contiene los valores para genera los pesos que crea el algoritmo de aprendizaje.

4. Tasa de aprendizaje. Amortigua el cambio de pesos en cada actualización.

5. Algoritmo optimizador. Selección el método que optimizara el funcionamiento de la red.

6. Épocas. Parte del dataset que ha atravesado la red neuronal en ambas direcciones.

7. Iteraciones. Cantidad total de batchs (lotes en los que se divide el conjunto de entrenamiento) necesarios para completar una época.

8. Batch size. Lote de muestras de entrenamiento.

1.4.2 Sesgo y Varianza

Cuando desarrollamos un modelo nos esmeramos en que esta sea lo más preciso posible aun sabiendo que no se puede construir un modelo perfecto. Una de las mejores formas para conocer si nuestra red progresa adecuadamente son el sesgo y la varianza, que son errores que se pueden tratar para mejorar el rendimiento.

El sesgo o error de sesgo es la diferencia entre la predicción esperada de nuestro modelo y los valores correctos. Esto se deriva en que los algoritmos paramétricos (regresión lineal, la regresión logística y clasificador bayesiano entre otros) tienen un alto sesgo que los hace más rápidos en el aprendizaje, pero menos flexibles y con peor rendimiento predictivo en problemas complejos, este tipo de algoritmos pueden ser la regresión lineal, regresión logística, etc. En cambio, los árboles de decisión, el algoritmo k-vecinos entre otros serán algoritmos con bajo sesgo dado que son más flexibles a la hora de adaptarse al problema debido a su carácter no lineal.

La varianza por otro lado se refiere a la medida de la variabilidad o dispersión de los resultados de un modelo con respecto a diferentes conjuntos de datos de entrenamiento, e idealmente no debería cambiar demasiado de un conjunto de datos de entrenamiento a otro. Los algoritmos de machine learning que son sensibles a este tipo de error se debe a una gran susceptibilidad a los detalles de los datos de entrenamiento, en otras palabras, sucede que las características de la base de datos con la que entrenamos el algoritmo difieren mucho con las características que buscamos que el algoritmo detecte. Los algoritmos con baja varianza son la regresión lineal y la regresión logística mientras que los algoritmos que tiene alta varianza son los árboles de decisión, k-vecinos entre otros.

La compensación sesgo-varianza, el objetivo de todo algoritmo de machine learning es lograr un bajo sesgo y una baja varianza con un buen rendimiento. Sin embargo, hay que tener en cuenta que el sesgo y la varianza son errores inversamente proporcionales, si disminuyes el sesgo, aumentará la varianza, y si disminuyes la varianza, aumentará el sesgo. Por ello, es necesario encontrar un equilibrio entre ambos para desarrollar un buen modelo. Podemos diagnosticar ciertos modelos en base a cómo se comportan su varianza y sesgo:

1. Los algoritmos de baja varianza y alto sesgo tienden a ser menos complejos, entrenando modelos consistentes pero ineficaces en promedio, se da en algoritmos lineales.

2. Los algoritmos de bajo sesgo y alta varianza tienden a ser más complejos, entrenando modelos poco consistentes pero precisos en promedio con una estructura flexible, se da en algoritmos no lineales.

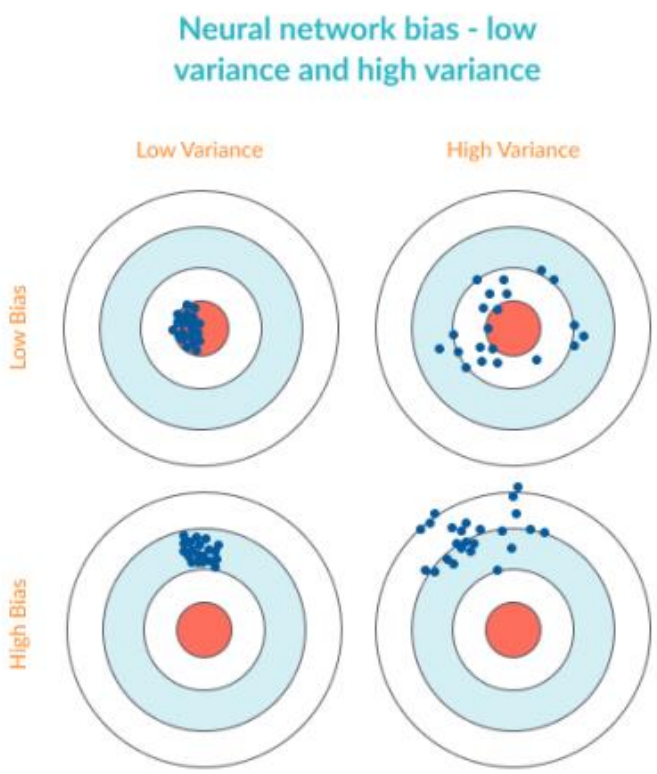

Figura 9. Ilustración del significado de los conceptos de sesgo y varianza además de mostrar como varían entre si el aumento o disminución entre el sesgo y la varianza [22].

1.4.3 Regularización

La regularización [21] en machine learning es una técnica muy útil que se utiliza para seleccionar el mejor modelo y evitar el sobreajuste (overfitting) que es cuando el modelo ha capturado ruido o detalles irrelevantes en los datos, en lugar de aprender las relaciones generales que se aplicarían a nuevos datos. Como resultado, el modelo tiene un excelente rendimiento en los datos de entrenamiento, pero se desempeña mal en datos no vistos, lo que reduce su capacidad de generalización. Hay diferentes enfoques para llevar a cabo la regularización, de los enfoques más típicos, podemos destacar:

1. L1 regularization. Minimiza las acciones de coeficientes del modelo con el objetivo de producir un modelo más interpretable.

2. L2 regularization. Mejora los errores de predicción al reducir los coeficientes de regresión restringiendo los de mayor tamaño para evitar el sobreajuste (overfitting).

3. Dropout. Elimina conexiones en la red neuronal con cierta probabilidad (esta probabilidad será otro hiperparámetro) llegando a aislar ciertas capas ocultas y dejándolas fuera de las propagaciones entrenando así una red más pequeña.

4. Data augmentation. Otros tipos de regularización consisten en el aumento de la base de datos a través de la transformación de las imágenes de esta, es decir, coger una imagen y por ejemplo voltearla, hacerle zoom, etc. Para tener una base de datos más grande sin tener que buscarla.

5. Early stopping. A medida que corres el descenso de gradiente llegará un punto en el que, si tu red esta sobre ajustada la función de coste que quieras minimizar empezará a aumentar otra vez, para evitar esto puede hacer una parada temprana evitando la actualización del gradiente y evitando el overfitting.

2. Marco teórico

2.1. CNN

Las CNN o Convolutional Neural Network [23] son un tipo de redes neuronales especializadas en el trabajo con imágenes, conocidas con este nombre por llevar a cabo en una de sus capas una operación matemática conocida como convolución. La convolución consiste en la aplicación de un filtro (matriz de valores) en la imagen obteniendo el valor del nuevo píxel a través de los valores de los pixeles vecinos como podemos ver en la figura 10.

Para explicar un poco más en profundo, las imágenes se componen de un conjunto de 3 matrices con unos valores de 0 a 255 donde cada celda de valor es un pixel. Los 3 canales se componen de una parte del espacio RGB y cuya unión forma la imagen a color normal. Dependiendo de los valores del filtro, nuestra nueva imagen tendrá diferente tipo de características, por ejemplo, si utilizamos un filtro 3x3.

Figura 10. Representación de una imagen con sus respectivos 3 canales RGB y su matriz de valores [24].

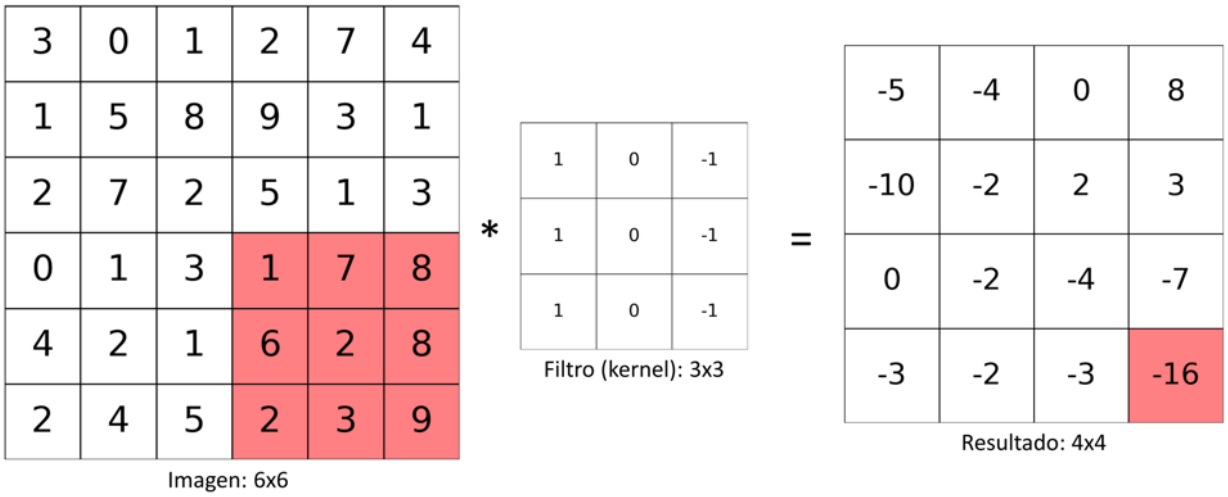

Figura 11. Ejemplo de una convolución. Aplicación de un filtro vertical 3x3 para detectar contrastes y resaltar bordes verticales. Se barre toda la matriz con el filtro obteniendo un nuevo valor en cada píxel exceptuando los de los bordes exteriores [25].

Las primeras capas convolucionales se centran en sacar pequeños patrones, en función del tipo de filtro que se aplique se podrán sacar características de la imagen como pueden ser bordes verticales, bordes horizontales y otro tipo de características. Y según se va profundizando en la red neuronal se van obteniendo objetos más grandes como podría ser en nuestro caso los ojos o la boca.

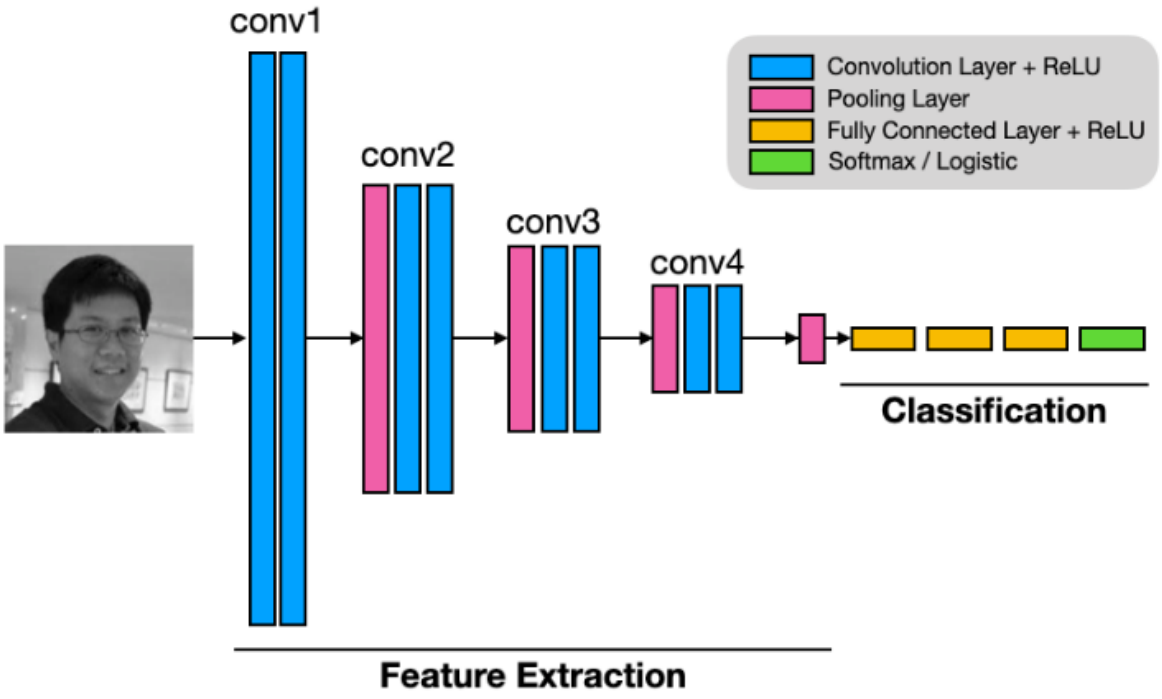

Figura 12. Arquitectura de una CNN típica. Con una imagen como input se atraviesa la red la cual contiene 4 capas convolucionales de las cuales las 3 ultimas tendrán Pooling layers para encontrarse más tarde con las capas Fully Connected y la capa Softmax [26].

La estructura general de las CNN consta de unos elementos básicos, los cuales varían en número y tamaño (número de filtros o tamaño de pixeles) en función de la búsqueda de optimización de nuestro programa. Las 4 partes más importante son:

1. La Pooling Layer opera sobre cada activación en la entrada, y escala la dimensión usando la función "Max" que es un tipo de operación que reduce el número de parámetros y conserva las características más importantes. Normalmente se utiliza un kernel de 2x2 aplicado con un stride (salto entre pixeles) de 2 pixeles. Esto genera una salida un cuarto más pequeño que la entrada manteniendo una buena relación entre la reducción de operaciones y la destrucción de información.

2. Las Fully Connected Layers También conocidas como capas densas o capas completamente conectadas son un tipo de capa en la red neuronal donde cada una de las

neuronas de una capa están conectadas a las neuronas de la siguiente capa permitiendo el aprendizaje de relaciones complejas entre las características de entrada y de salida.

3. La capa Softmax es una capa de activación que convierte las salidas de la red neuronal en una distribución de probabilidad lo cual resulta muy útil para la clasificación de múltiples clases determinando la probabilidad de que cierto objeto de entrada pertenezca a una de las clases posibles.

En resumen, las CNN son redes neuronales que se componen de una primera fase de obtención de patrones y características de las imágenes a través de operaciones convolucionales con diversos tipos de filtros. Dentro de esta primera fase podemos destacar dos componentes, las capas convolucionales encargadas de llevar a cabo las operaciones de extracción de características y las pooling layers que reducen significativamente el tamaño de las imágenes para reducir el coste computacional. Y la segunda fase es la que consta de las capas densas que se encargan de hacer la clasificación para que más tarde la capa Softmax determine la clase de la salida.

2.2. Introducción a reconocimiento facial

El reconocimiento facial [27] es una tecnología de inteligencia artificial que permite identificar a una persona a través de la captura y análisis de sus rasgos faciales. Esta tecnología ha experimentado un rápido desarrollo en los últimos años y ha encontrado aplicaciones en una amplia variedad de campos, desde la seguridad hasta el marketing y la identificación personal.

Uno de los usos más comunes del reconocimiento facial es en la seguridad, donde se utiliza para la identificación de individuos en aeropuertos, puertos y otras zonas críticas. Esta tecnología también se utiliza en la investigación criminal, para identificar a sospechosos y hacer seguimiento a sus movimientos.

Sin embargo, el uso del reconocimiento facial también ha generado controversia debido a cuestiones de privacidad y seguridad. Algunas organizaciones han expresado preocupaciones sobre el uso indebido de esta tecnología, como la recolección de datos biométricos sin consentimiento y el riesgo de discriminación. Un caso bastante reciente es la propuesta de los

senadores de Estados Unidos de estrechar el cerco y hasta prohibir Tiktok en su país ya que piensan que la red social puede estar almacenando demasiados datos sensibles de su población.

En el ámbito empresarial, el reconocimiento facial ha sido utilizado para mejorar la atención al cliente. Por ejemplo, se puede utilizar esta tecnología para reconocer a los clientes en una tienda y personalizar su experiencia de compra.

En la actualidad, el reconocimiento facial está siendo utilizado en nuevas aplicaciones, como la identificación de emociones en tiempo real, el reconocimiento de objetos y el control de acceso a instalaciones y dispositivos. El uso de filtros en las redes sociales también requiere de la aplicación del reconocimiento facial ya que el móvil debe detectar la cara y sus características principales para aplicar el filtro, por lo que su uso es habitual en las nuevas generaciones.

En el campo de la medicina, el reconocimiento facial se está utilizando para el diagnóstico y tratamiento de enfermedades. Por ejemplo, se ha desarrollado una aplicación que utiliza el reconocimiento facial para identificar síntomas de enfermedades neurológicas, como el Parkinson.

En este capítulo, se revisarán algunos de los avances más recientes en el campo del reconocimiento facial. Uno de los desarrollos más importantes ha sido el uso de algoritmos de aprendizaje profundo (deep learning) para mejorar la precisión del reconocimiento facial. Los algoritmos que usan aprendizaje profundo utilizan redes neuronales convolucionales (CNN) para identificar características faciales y compararlas con una base de datos de imágenes de referencia. Estos algoritmos han demostrado una precisión significativamente mayor que los métodos tradicionales de reconocimiento facial.

2.3. Haar Cascade

El algoritmo de Haar Cascade fue desarrollado por Paul Viola y Michael Jones en 2001. En el trabajo [28] desarrollaron una técnica para la detección de objetos la cual contenía filtros de Haar y un clasificador en cascada para identificar patrones específicos de características en una imagen. Es capaz de procesar la imagen en varias etapas, disminuyendo la complejidad computacional y mejorando la precisión. Se compone de dos partes fundamentales: los filtros de Haar y un clasificador en cascada.

Los filtros de Haar son filtros lineales capaces de detectar patrones de cambio de intensidad en una imagen, los cuales se aplican en una serie de subventanas de la imagen. Además de identificar características en la imagen, tales como bordes, esquinas, líneas y superficies planas. Cada clasificador se entrena con una serie de imágenes positivas que contienen el objeto a detectar y una serie de imágenes negativas que no contienen el objeto.

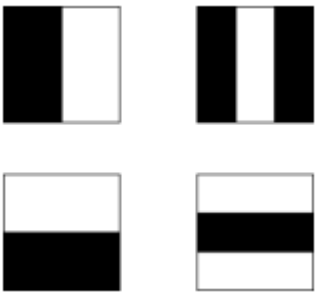

Figura 13. Ejemplo de formas tipo Haar para la extracción de características [29].

Mientras que el clasificador en cascada [29] es un modelo de aprendizaje supervisado que funciona evaluando cada característica de la imagen y descartando aquellas que no son relevantes para reducir el número de falsos positivos y aumentar la eficiencia del proceso de detección.

2.4. FaceNet

FaceNet [30] es un sistema de reconocimiento facial desarrollado por Florian Schroff, Dmitry Kalenichenko y James Philbin, empleados de la empresa Google en 2015, que utiliza técnicas de aprendizaje profundo para crear representaciones de alta dimensión de rostros humanos. El objetivo principal de FaceNet es crear una representación vectorial de un rostro humano que permita comparar rostros entre sí, independientemente de la pose, la iluminación o la expresión facial.

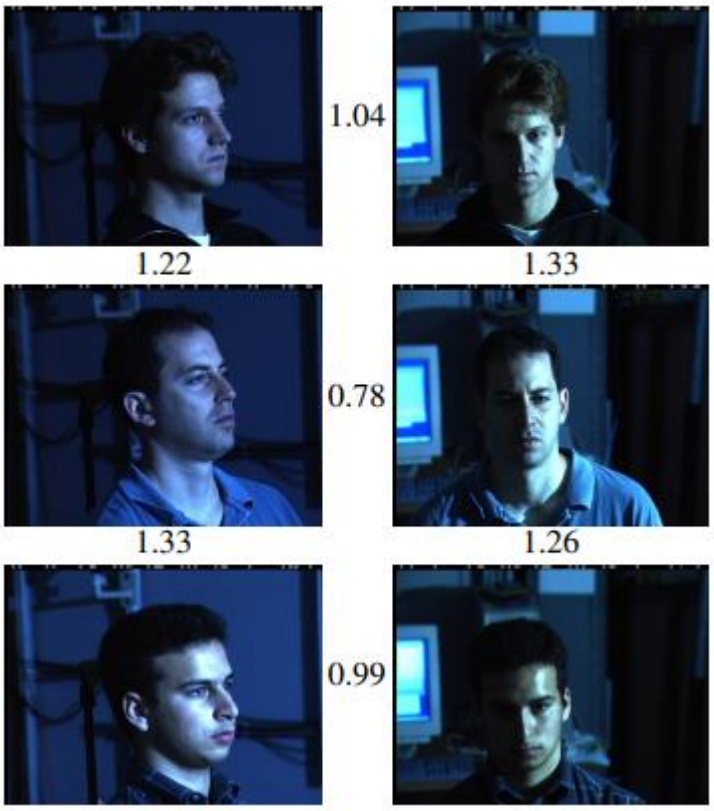

Figura 14. Cálculo de distancias entre imágenes con distinta pose e iluminación [30].

Para lograr este objetivo, FaceNet utiliza una técnica llamada Triplet Loss para entrenar las redes neuronales. Esta técnica compara tres imágenes: una imagen de una persona conocida -el Anchor-, otra imagen de la misma persona con una pose o iluminación diferente, el positivo, y una imagen de una persona desconocida, el negativo. El objetivo es que la representación de la persona conocida sea lo más cercana posible a sí misma y lo más lejana posible de la representación de la persona desconocida esto se lleva a cabo calculando las distancias entre imágenes.

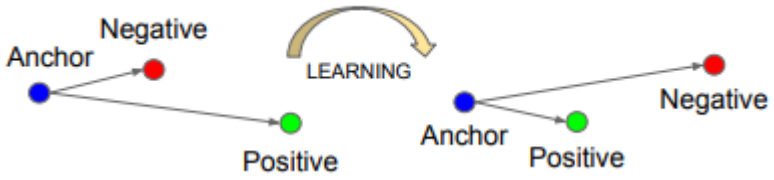

Figura 15: Técnica Triplet Loss en la que se pueden observar los tres elementos clave, el anchor o imagen de referencia, el positivo y el negativo. Además de mostrar como el entrenamiento logra reducir distancias con positivo y aumentarla con el negativo [31].

Una vez que se ha entrenado el sistema, se puede utilizar para identificar personas en imágenes y videos. El sistema utiliza una técnica llamada clasificación de pares para comparar la representación vectorial de una persona en una imagen con la representación vectorial de una persona en una base de datos. Si la distancia entre las dos representaciones es inferior a un umbral determinado, el sistema considera que son la misma persona.

Una de las ventajas de FaceNet es su capacidad para detectar la presencia de caras en imágenes, incluso en situaciones en las que hay varias caras en la imagen. El sistema es capaz de detectar la presencia de una cara y crear una representación vectorial de ella, incluso si la imagen es de baja calidad o está en una pose inusual. FaceNet también es capaz de identificar a personas en diferentes condiciones de iluminación, ángulos y poses faciales. El sistema utiliza una técnica llamada normalización de contraste adaptativo para normalizar las imágenes y hacerlas comparables entre sí. FaceNet ha sido utilizado en una variedad de aplicaciones, incluyendo la identificación de personas desaparecidas y la autenticación de usuarios en aplicaciones móviles. También se ha utilizado para crear bases de datos de rostros para aplicaciones de vigilancia y seguridad.

Sin embargo, FaceNet ha sido objeto de críticas y preocupaciones sobre la privacidad y la seguridad, especialmente en lo que respecta a la utilización de la tecnología en aplicaciones de vigilancia. También ha habido preocupaciones sobre la precisión del sistema y su capacidad para identificar a personas de diferentes grupos étnicos y raciales.

Para abordar estas preocupaciones, Google ha desarrollado una versión de FaceNet llamada FaceNet-Keypoints que es capaz de detectar y localizar puntos clave en los rostros, como los ojos, la nariz y la boca. Esto permite una mayor precisión en la identificación de rostros y también puede ayudar a mejorar la accesibilidad para personas con discapacidades visuales.

2.5. NIRFaceNet

NIRFaceNet [32] es un modelo de reconocimiento facial desarrollado por un equipo de investigación de la Universidad de Ciencia y Tecnología de China en 2018. Fue diseñado para operar en el rango del espectro infrarrojo cercano (NIR, por sus siglas en inglés). El modelo está

entrenado con una gran cantidad de datos de imágenes faciales en el rango NIR y utiliza una arquitectura de red neuronal convolucional para extraer características faciales distintivas.

El objetivo principal del modelo es abordar algunos de los desafíos del reconocimiento facial en condiciones de poca luz, como las que se encuentran en la vigilancia nocturna o en la seguridad de edificios y sistemas de control de acceso.

La arquitectura de red neuronal utilizada en NIRFaceNet está basada en una red de codificación de características profunda llamada ResNet-50 (arquitectura de red neuronal convolucional profunda utilizada en el procesamiento de imágenes y el reconocimiento de objetos), que ha demostrado ser efectiva en una variedad de tareas de visión por computadora. El modelo utiliza técnicas avanzadas, para mejorar su precisión y eficiencia. Además de su capacidad para operar en condiciones de poca luz, NIRFaceNet también se ha demostrado que tiene una alta precisión en la identificación de personas en una variedad de situaciones, incluyendo la detección de personas en multitudes y en ángulos de visión desafiantes. El modelo también es capaz de realizar la verificación de identidad en tiempo real, lo que lo hace adecuado para aplicaciones de seguridad en tiempo real.

Figura 16. Imágenes NIR de una persona en distintos escenarios [32].

Una de las características clave de NIRFaceNet es su capacidad para realizar la identificación facial utilizando imágenes NIR y visibles, lo que lo hace versátil en una variedad de entornos de iluminación. El modelo también ha sido optimizado para el rendimiento en dispositivos móviles, lo que lo hace adecuado para su uso en aplicaciones de seguridad en teléfonos inteligentes y tabletas.

2.6. Face-GCN

Face-GCN [33] es un modelo de reconocimiento facial desarrollado por un equipo de investigadores de la Universidad de Tsinghua en China en 2020. El modelo utiliza una arquitectura de red neuronal llamada GCN (Graph Convolutional Network) para realizar el reconocimiento facial y tiene una precisión excepcional en una variedad de tareas de reconocimiento facial.

La arquitectura de red neuronal GCN utilizada en Face-GCN se basa en una estructura de grafo, que representa las relaciones entre los elementos en un conjunto de datos. En el caso de Face-GCN, el grafo se construye a partir de las imágenes faciales de entrada y las etiquetas de identidad correspondientes. La red utiliza esta información para aprender a extraer características faciales únicas y distintivas.

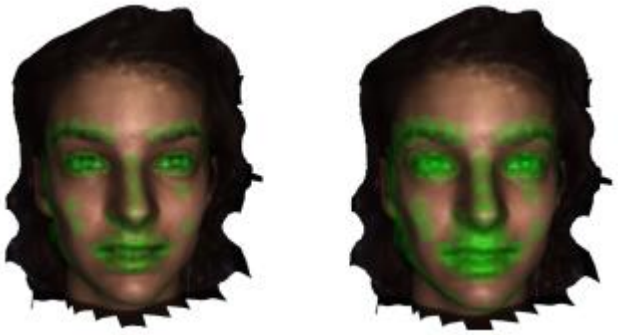

Figura 17. Extracción de características locales construidas alrededor de la selección de los puntos de referencia. Izquierda con 100 vecinos y derecha con 200 [33].

Face-GCN utiliza técnicas avanzadas de aprendizaje profundo, como el aprendizaje auto supervisado y el aprendizaje por transferencia, para mejorar su precisión y eficiencia. El modelo también utiliza técnicas de regularización para prevenir el sobreajuste durante el entrenamiento. El modelo ha demostrado ser altamente efectivo en una variedad de tareas de reconocimiento facial, incluyendo la identificación de personas en imágenes individuales y la comparación de imágenes faciales entre sí. Face-GCN también es capaz de realizar la verificación de identidad en tiempo real, lo que lo hace adecuado para aplicaciones de seguridad en tiempo real.

Una de las características clave de Face-GCN es su capacidad para realizar el reconocimiento facial en conjuntos de datos desequilibrados. Esto significa que el modelo puede funcionar bien incluso cuando hay una cantidad desigual de imágenes de diferentes personas en el conjunto de datos de entrenamiento. Además de su precisión y eficiencia, Face-GCN también es altamente escalable y puede manejar grandes conjuntos de datos de imágenes faciales. El modelo ha sido optimizado para su uso en sistemas de computación distribuida y ha demostrado ser efectivo en entornos de alta demanda.

2.7. VGG16

VGG16 [34] es un modelo de red neuronal convolucional (CNN) creado por un grupo de investigadores, Karen Simonyan y Andrew Zisserman de la Universidad de Oxford en 2014. El objetivo de estos investigadores era crear una red profunda con una estructura sencilla capaz de aprender características de las imágenes de una manera eficaz enfocando el estudio en la profundidad de la red neuronal para incrementar la precisión de la clasificación.

VGG16 tiene una arquitectura que se compone de 16 capas, de las cuales 13 son capas convolucionales y 3 son capas densas. Las capas convolucionales se encargan de extraer las características de las imágenes a través del uso de filtros. Más tarde, las representaciones resultantes pasan a través de 3 capas densas, estas capas se encargan de la clasificación, transformando las características extraídas de las imágenes en probabilidades de pertenencia a distintas categorías.

Lo que hace especial a VGG16 es que utiliza convoluciones 3x3 en todas las capas convolucionales. Dicha elección del tamaño se basa en la apreciación de que varias capas 3x3 poseen un campo receptivo equivalente a una convolución con un filtro de mayor tamaño, pero ahorrando parámetros, permitiendo así aumentar la profundidad de la red sin aumentar el número de operaciones.

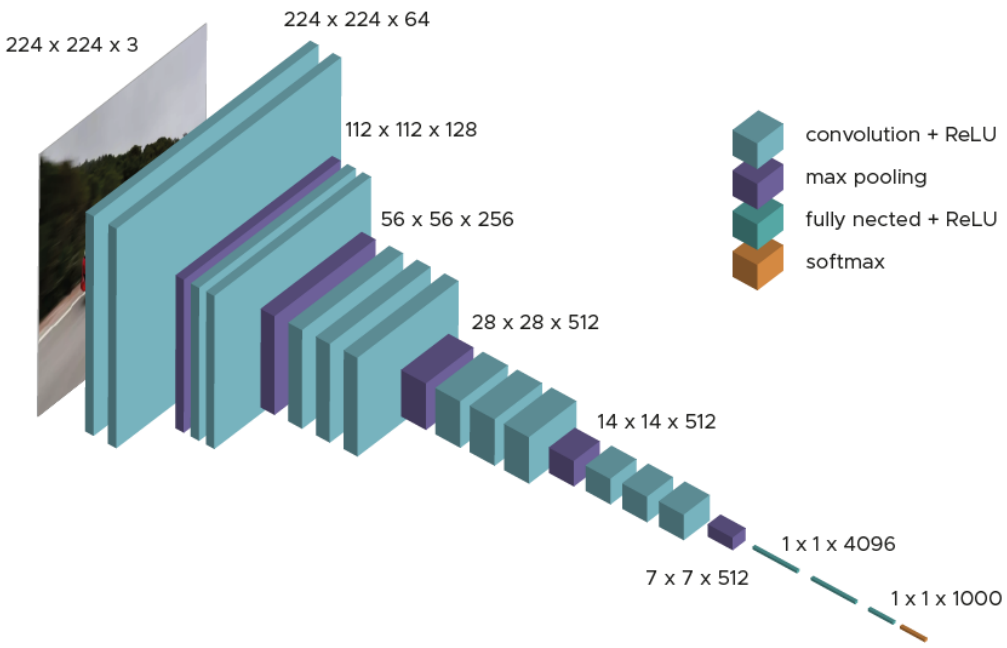

Ilustración 18. Ejemplo de la arquitectura de una red convolucional VGG16 compuesta por 16 capas [35].

VGG16 logró un gran éxito en la clasificación de imágenes al ser uno de los primeros modelos que permitió demostrar la importancia de las redes neuronales para su uso en tareas de visión por computadora, destacando su simplicidad y uniformidad, que han facilitado su comprensión y uso como punto de referencia en investigaciones posteriores.

En resumen, de los cuatro algoritmos que se acaban de comentar: Haar Cascade es una técnica de detección de objetos que utiliza filtros de Haar y un clasificador en cascada para identificar patrones específicos de características en una imagen. La técnica es eficiente y se utiliza comúnmente en aplicaciones de detección de objetos, incluyendo la detección facial, aunque en la actualidad se han desarrollado técnicas más avanzadas basadas en redes neuronales convolucionales, como FaceNet y Face-GCN. NIRFaceNet es un modelo de reconocimiento facial basado en redes neuronales convolucionales diseñado para operar en el rango NIR. El modelo utiliza técnicas avanzadas de aprendizaje profundo y ha demostrado ser efectivo en condiciones de poca luz y en la identificación de personas en una variedad de situaciones desafiantes. Con su capacidad para operar en entornos de iluminación variables y su rendimiento optimizado para dispositivos móviles, NIRFaceNet es una herramienta valiosa para aplicaciones de seguridad en tiempo real. FaceNet es un sistema de reconocimiento facial altamente avanzado compuesto por

una estructura que consta de dos modelos, el modelo de Zeiler&Fergus y el modelo de Inception de Szedegy [30] que utiliza técnicas de aprendizaje profundo para crear representaciones vectoriales de alta dimensión de rostros humanos. El sistema ha sido utilizado en una variedad de aplicaciones, incluyendo la identificación de personas desaparecidas y la autenticación de usuarios en aplicaciones móviles, aunque ha sido objeto de preocupaciones sobre la privacidad y la seguridad. Face-GCN es un modelo de reconocimiento facial altamente preciso y eficiente que utiliza técnicas avanzadas de aprendizaje profundo y una arquitectura de red neuronal basada en grafo para extraer características faciales únicas. El modelo es escalable y puede manejar grandes conjuntos de datos, y también es capaz de funcionar bien en conjuntos de datos desequilibrados. Con su capacidad para realizar el reconocimiento facial en tiempo real y su eficacia en entornos de alta demanda, Face-GCN es una herramienta valiosa para aplicaciones de seguridad y vigilancia. Y finalmente VGG16 es una arquitectura de CNN profunda que se utiliza como punto de referencia en el campo de visión de la computadora gracias a su enfoque de convoluciones en cascada.

3. Estado del arte

El reconocimiento facial [36] es una tecnología en constante evolución que ha experimentado una rápida expansión en los últimos años. Esta sección del arte tiene como objetivo proporcionar una visión general de los avances más importantes en esta área, abordando sus fundamentos, aplicaciones y desafíos. A medida que esta tecnología continua su desarrollo, es importante conocer su estado actual e implicaciones.

La cara es una parte esencial del cuerpo el humano, única para cada identidad, por ello, su reconocimiento está aceptado como el mejor método biométrico. Existen dos tipos de clasificadores biométricos: los fisiológicos, como son las huellas dactilares, reconocimiento facial, el iris de los ojos; y los clasificadores conductuales, que comprenden el comportamiento y la conducta de las personas. Las ventajas del reconocimiento facial con respecto a los otros clasificadores fisiológicos consisten en que, a diferencia de, por ejemplo, la verificación del iris se puede implantar con un sistema mucho menos costoso, aunque no llegue a ser tan preciso. Por otro lado, en comparación con las huellas dactilares, el reconocimiento facial puede usarse sin necesidad de la interacción directa de las personas con el sistema de reconocimiento.

El primer trabajo sobre la automatización de reconocimiento de expresiones faciales (AFER) fue publicado en 1978 [37]. Consistía en un modelo que seguía la moción de unos puntos de referencia en la imagen. Este modelo pasó desapercibido debido al limitado poder computacional de la época.
El trabajo de Mase and Pentland y Paul Ekman, dos reconocidos investigadores en el campo del reconocimiento facial y psicología de las emociones revivirían el proyecto de esta investigación a principios de los 90 [38]. En los 2000, se publicó el CK dataset, una base de datos creada por los investigadores Vicki Bruce, Jeffrey Cohn y Takeo Kanade con el objetivo de proporcionar una base de datos estándar para el estudio de las expresiones faciales y la comprensión de las emociones. Este suceso supuso el inicio del AFER moderno. Los primeros trabajos utilizaron representaciones geométricas, vectores para describir el movimiento de la cara, contornos activos para detectar la forma de la boca o las cejas o modelos de malla deformables 2D.

Otro trabajo importante que marcaría un antes y un después sería el BU-3DFE [39] publicado en 2011, que, como su propio nombre indica, iniciaría la extensión del reconocimiento facial hacia el mundo de las 3 dimensiones, utilizando para ello representaciones geométricas 3D de caras y distancias euclídeas normalizadas entre puntos de bordes en el espacio tridimensional.

Existen muchos métodos y enfoques en el amplio mundo del reconocimiento facial, los algoritmos clásicos como el análisis de componentes principales (PCA) y el análisis de discriminante lineal (LDA) [40]. Estos enfoques se basan en la extracción de características y la reducción de la dimensionalidad para representar las caras y llevar a cavo el reconocimiento. Sin embargo, se reconoce que estos métodos tienen limitaciones en términos de robustez y rendimiento en condiciones variables, como cambios en la iluminación y pose.

También nacen otros como el análisis de textura que se centra en las características locales de la superficie de la piel y los detalles finos de la cara. La textura se refiere a los patrones de pixeles que aparecen en una imagen y puede incluir detalles como arrugas, poros, marcas o lunares entre otros que sirven para distinguir a una persona. En el análisis de textura, se utilizan técnicas como el filtro de Gabor [41], que es capaz de capturar características texturales en diferentes escalas y orientaciones. Estos filtros analizan la distribución de las frecuencias espaciales en la imagen para identificar características texturales específicas de la cara.

Por otro lado, tenemos las técnicas de aprendizaje geométricas cuyo análisis se basa en la posición y la forma de las características faciales, como los ojos, la nariz, la boca, las cejas. Las características geométricas se obtienen utilizando puntos de referencia en la cara, también conocidos como landmarks. Al medir distancias y las proporciones entre estos puntos, se crea un grupo de características que es único para cada individuo.

Otro tipo son las técnicas de aprendizaje basadas en kernel, las cuales tienen como objetivo transformar los datos de entrada de un espacio de mayor dimensión para facilitar el reconocimiento. Estas técnicas se basan en el uso de una función de kernel, que permite realizar una transformación no lineal de los datos originales a un espacio donde sea más fácil separar y clasificar las muestras como se ve en la figura 19. Esto es potencialmente útil cuando los datos no son linealmente separables en su forma original. Al aplicar técnicas de aprendizaje basada en kernel, como el Support Vector Machine (SVM) o el principal Component Analysis (PCA) con

kernel, es posible obtener una representación mas discriminativa de las características faciales, lo que mejora la precisión del reconocimiento.

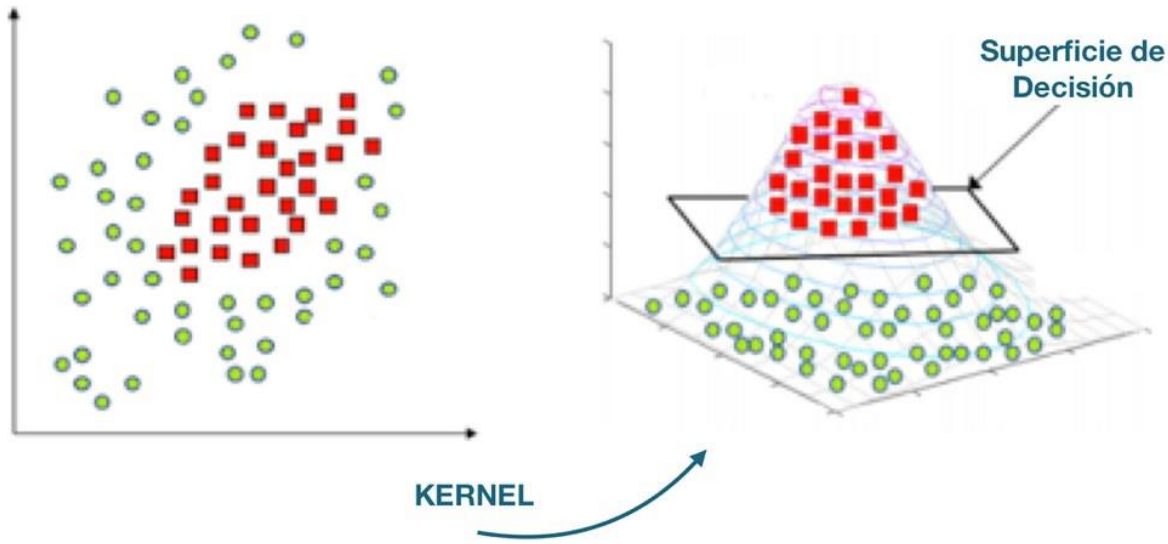

Ilustración 19. Muestra de una separación más sencilla gracias a aplicación de una transformación no lineal pasando de 2 dimensiones a 3 dimensiones [42].

Otra vertiente interesante del reconocimiento facial consiste en las técnicas de reconocimiento de emociones faciales [43]. El reconocimiento de emociones faciales es un área de investigación en alza debido a su amplio repertorio de aplicaciones en campos como la psicología, la inteligencia artificial, la interacción de humanos con computadoras y la medicina. Además de las técnicas mencionadas anteriormente hay que destacar una técnica llamada Random Forests [44], método de aprendizaje automático que se basa en la combinación de múltiples árboles de decisión para realizar predicciones más precisas. Su capacidad para manejar características correlacionadas y ruido en los datos, así como su resistencia al sobreajuste, los convierte en una opción popular en este campo de la investigación.

3.1. FaceID

El uso del reconocimiento facial ha tenido una tendencia exponencial en cuanto a su uso diario, sobre todo en los campos de seguridad. Uno de esos ejemplos es el FaceID [45] de los móviles IPhone, desarrollado por Apple, que consiste en un sistema de bloqueo de pantalla (seguridad)

cuya llave de desbloqueo es la cara del usuario (reconocimiento y verificación facial). Esto deriva en la línea de desarrollo del reconocimiento facial, pues para realizar la verificación primero hay que reconocer si hay caras en la imagen o vídeo. FaceID utiliza un sistema de escaneo facial 3D basado en la tecnología de infrarrojos para verificar a los usuarios. Es importante destacar que, si bien las implicaciones y aplicaciones de FaceID son prometedoras, también plantean desafíos y preocupaciones, como la privacidad de los datos biométricos, la posibilidad de falsificación o la vulnerabilidad a ataques cibernéticos. Por lo tanto, es esencial abordar estos aspectos y establecer salvaguardias adecuadas para garantizar un uso responsable y seguro de la tecnología de reconocimiento facial como FaceID.

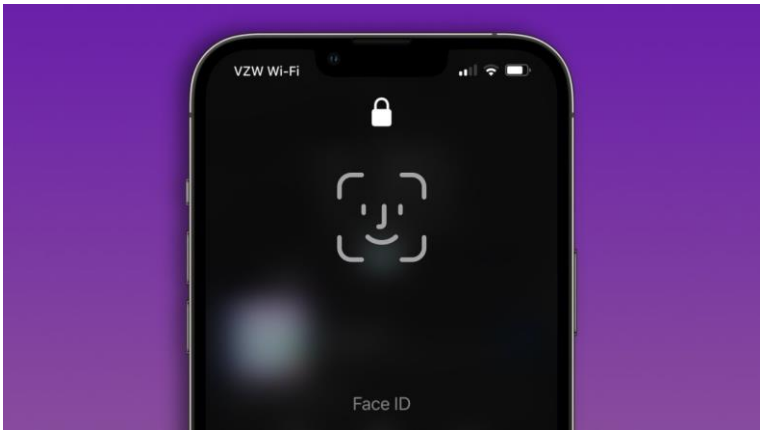

Figura 20. Sistema biométrico de IPhone FaceID, su presentación en un móvil [46].

3.2. Redes Sociales

Otro uso bastante expandido sobre todo en el espectro de gente joven es el uso de filtros en las redes sociales que requieren reconocimiento facial para ser aplicados. Hoy en día vivimos en una sociedad globalizada en la que la imagen puede marcar la diferencia ya que la tendencia es exponer en redes sociales tu día a día llegando a generar incluso marcas con tu propia imagen (influencers).

Los filtros son una herramienta para generar contenido de mayor calidad creando imágenes más comerciales y agradables al público. Si bien la mayoría de las empresas no revelan la información técnica de las redes convolucionales que usan para el reconocimiento facial, debido a su estrategia competitiva, pero podemos inferir en que la mayoría usan modelos de CNN muy optimizados y entrenado con largas bases de datos para lograr un reconocimiento facial eficiente y preciso. Lo

más probable es que este tipo de modelos se hayan desarrollado internamente o se hayan adaptado a partir de arquitecturas CNN como pueden ser VGG, ResNet o InceptionNet, por ejemplo.

3.3. Seguridad de empresas

El reconocimiento facial se ha convertido en una herramienta importante para la seguridad de las empresas [47] gracias a su capacidad para identificar y autenticar a las personas basándose en las características únicas de sus rostros.

Alguna de las técnicas que se usan en este ámbito son: extracción de características: Una vez que se ha detectado un rostro, se extraen características específicas que son únicas para cada individuo. Estas características pueden incluir la posición de los ojos, la forma de la nariz, la distancia entre los labios, entre otros. Se utilizan diferentes métodos, como el análisis de componentes principales (PCA) o las redes neuronales convolucionales (CNN), para extraer estas características y crear un patrón reconocible. Otra característica es la comparación y coincidencia: Después de extraer las características faciales, se compara el patrón obtenido con una base de datos preexistente para buscar posibles coincidencias. Esto implica comparar las características extraídas con los patrones almacenados en la base de datos para determinar si hay una coincidencia significativa.

Es importante destacar que la seguridad basada en el reconocimiento facial también debe considerar aspectos como la calidad de las imágenes, el rendimiento en tiempo real, la protección de datos y la privacidad de los usuarios. Cada empresa utiliza técnicas y algoritmos distintos en función de sus necesidades.

3.4. Reconocimiento facial en la medicina

Retornando al uso del reconocimiento facial para detectar emociones en las personas, el cual nos puede ayudar a analizar trastornos psicológicos, surgen enfoques que se centran en el desarrollo de un sistema automatizado que facilite comprenderlos e identificarlos mejor, como es el caso de LP Morency, A. Rizzo y J. Gratch [49] .

Su estudio se basa en la idea de que los trastornos psicológicos pueden manifestarse en patrones específicos de comportamiento no verbal y verbal. Por lo tanto, tratan de utilizar técnicas de reconocimiento facial y procesamiento de lenguaje natural para capturar y analizar estas señales.

Se utilizan algoritmos de visión por computadora para detectar y analizar expresiones faciales, gestos y otros aspectos relacionados. Estas señales visuales proporcionan información sobre el estado emocional y la respuesta afectiva de los individuos, lo que puede ser relevante para el diagnostico de ciertos trastornos psicológicos entre ellos destacando la depresión y el trastorno de estrés postraumático.

Figura 21. Mapeo de la cara con distintas emociones y sus características más representativas [50].

4. Diseño e implementación

Se ha usado la versión 3.9.6 de Python. Comenzaremos importando las primeras librerías que usaremos para llevar a cabo nuestro programa, estas son:

1. Os. Librería que nos ayudará a navegar por los archivos locales de nuestro sistema.
2. Cv2. Se utiliza para el procesamiento de imágenes y visión por computadora. Proporciona una amplia gama de funciones para leer, escribir, manipular y procesar imágenes y videos.
3. Time. Proporciona funciones relacionadas con el tiempo y la medición del tiempo de un programa.
4. Uuid. Nos añade funciones para generar identificadores únicos universales.

5. Matplotlib.pyplot. Biblioteca de visualización de datos en Python. “Pyplot” es un módulo dentro de matplotlib que proporciona una interfaz similar a MATLAB para crear gráficos y visualizaciones.

Estas bibliotecas nos serán útiles para llevar a cabo el primer tramo de este proyecto, crear y preparar la base de datos. Nuestra base de datos inicial se compondrá de 365 imágenes, en ella, se encontrarán las caras de familiares y amigos míos, aportando:

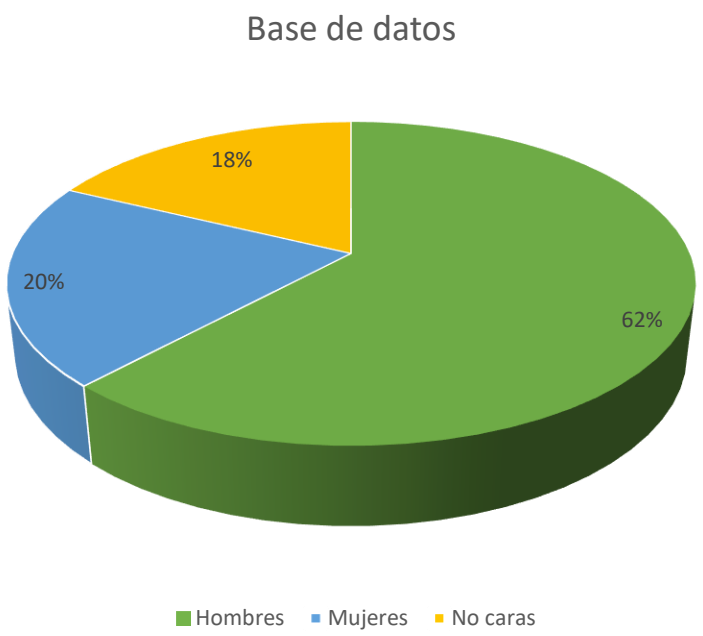

Figura 22. Gráfico con las proporciones de la cantidad de fotos de la base de datos para el proyecto y su procedencia.

- 226 fotos de varones.
- 74 fotos de mujeres.
- 65 fotos sin caras (serán fotos en las que no aparezca ninguna cara, en su mayoría fotos de mi habitación en la que salgo fuera de plano o mi cara cortada).

Todas las fotos están tomadas por mi webcam, en formato PNG y con una dimensión de 112x112 pixeles, a excepción de las fotos de mi hermano y amigos, las cuales están tomadas desde sus propios móviles o cámaras, las cuales he redimensionado y cambiado el formato para que estuvieran en concordancia con las tomadas por mi webcam.

Una vez que hemos recolectado esta base de datos inicial, pasaremos a la etiquetación manual de las fotos con caras con la etiqueta “face” en formato JSON en el cual también estarán las

coordenadas de las caras (Figura 24) utilizando la herramienta de anotación desarrollada por el MIT, LabelMe. Una vez etiquetadas las 300 fotos con caras, pasaremos a importar las bibliotecas para el siguiente caso.


```
{
  "version": "5.2.0.post4",
  "flags": {},
  "shapes": [
    {
      "label": "face",
      "points": [
        [
          22.57458563535912,
          12.132596685082879
        ],
        [
          57.71270718232045,
          70.80662983425414
        ]
      ],
      "group_id": null,
      "description": "",
      "shape_type": "rectangle",
      "flags": {}
    }
  ],
  "imagePath": "..\\images\\6d74d164-1295-11ee-8c28-fcaa14ca5db0.png",
```


Figura 23. Ejemplo visual de como es una etiqueta JSON creada a través de LabelMe.

1. TensorFlow. Utilizada para crear modelos de aprendizaje automático y realizar cálculos numéricos. Proporciona un conjunto de herramientas y API que permiten la construcción y entrenamiento de redes neuronales. Utilizamos la versión 2.12.0
2. Json. Proporciona funcionalidades para trabajar con el formato de datos JSON, intercambio de datos estructurado.
3. Numpy. Es una biblioteca fundamental en Python para el procesamiento numérico. Provee un objeto de matriz multidimensional eficiente y una amplia gama de funciones y operaciones que se pueden aplicar a esta matriz, agiliza los cálculos haciéndolos menos pesados y más rápidos gracias a este método.
4. Keras. Ejecutada sobre TensorFlow. Es capaz de proporcionar una interfaz de nivel avanzando y fácil de usar para crear y entrenar redes neuronales artificiales enfocado principalmente en la modularidad. Utilizamos la misma versión que TensorFlow, la 2.12.0.

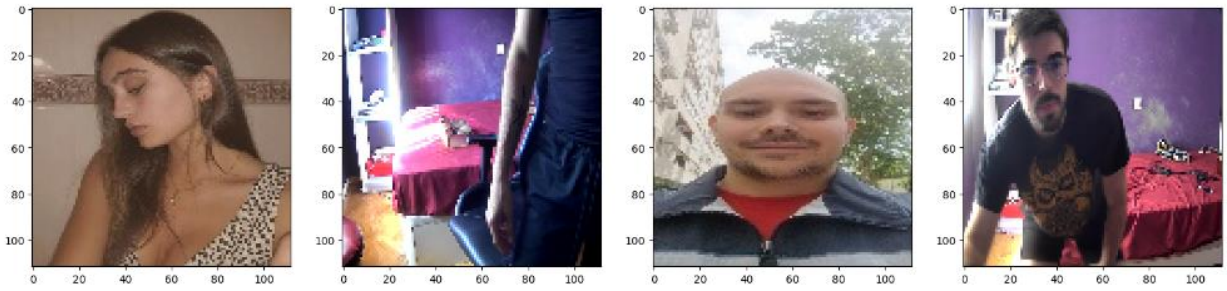

Figura 24. Ejemplo del display de las fotos ya redimensionadas gracias a la librería Matplotlib de nuestra base de datos en formato PNG.

A continuación, dividiéremos la base de datos en 3 partes de forma aleatoria, una para la parte del entrenamiento que será la más grande (90%, 326 fotos) y otras dos para test y validación (5%, 18 fotos cada una). Una vez divididas las fotos en sus respectivas partes, crearemos carpetas distintas también para las etiquetas.

Como bien sabemos, lo que les proporciona una base fuerte a los algoritmos de machine learning, son bases de datos muy amplias, partiendo de una base de datos pequeña como son 365 fotos llevaremos a cabo un método para incrementar el tamaño de la base. Utilizaremos data augmentation, importado desde la librería albumentations. El data augmentation consiste en coger las imágenes de nuestra base de datos y aplicarle transformación que simulen diversos escenarios. En nuestro caso las transformaciones que hemos decidido aplicar son:

1. RandomCrop. Esta transformación realiza un corte aleatorio de la imagen escogiendo una región aleatoria.
2. HorizontalFlip. Voltea la imagen horizontalmente, se refleja de izquierda a derecha.
3. RandomBrightnessContrast. Ajusta aleatoriamente el brillo y contraste de la imagen.
4. RandomGamma. Ajusta aleatoriamente el valor gamma de la imagen (controla la relación entre los valores de los pixeles).
5. RGBShift. Aplica un desplazamiento aleatorio a los canales rojo, verde y azul de la imagen.
6. VerticalFlip. Voltea verticalmente la imagen, se refleja de arriba abajo.

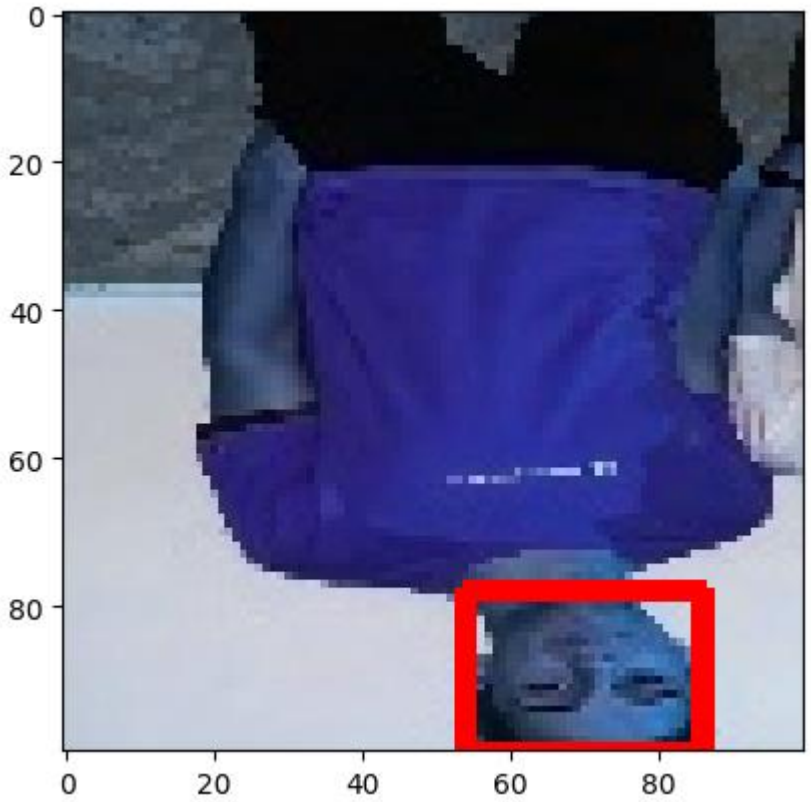

Figura 25. Ejemplo de una foto después de haberle aplicado data augmentation. Podemos ver que ha sufrido un RGBShift y un VerticalFlip.

Ahora correremos un bucle que nos aplique 60 transformaciones de manera aleatoria por cada foto, es decir, nuestra base de datos aumentaría de 365 fotos a 21900, teniendo un tamaño más considerable. Durante este proceso, se realizará una simplificación de las etiquetaciones y una creación de las etiquetas para las fotos que no contengan caras como se muestra en la figura 27. Además, se ajustarán las bounding box en función de las transformaciones aplicando la operación necesaria en cada caso y se normalizarán las coordenadas dejándolas en un número entre 1 y 0.

```
1   {"image": "0.png", "bbox": [0.33647640864727524, 0.0, 0.9316022099447513, 0.9188950276243093], "class": 1}
```

Figura 26. Etiqueta simplificada en formato JSON de una imagen después del data augmentation.

Una vez ya tenemos todas nuestras imágenes, procederemos a normalizarlas dejando los valores de los pixeles entre 0 y 1, para ello, dividiremos los valores de los pixeles por 255, esta

transformación se puede aplicar de manera sencilla utilizando la función “map”. Ahora cargaremos las etiquetas para finalizar la preparación de nuestra base de datos.

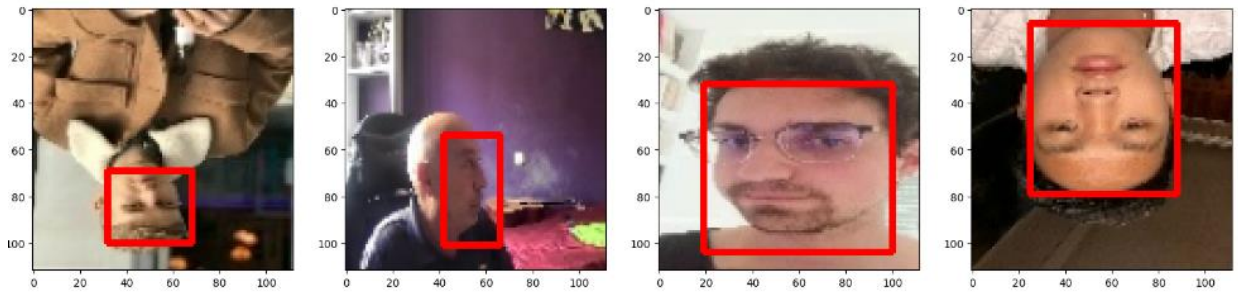

Figura 27. Ejemplo de fotos de nuestra base de datos final después de aplicar data augmentation.

Ahora pasaremos a la parte del modelo, para mi trabajo he decidido usar un modelo ya preestablecido, este modelo es VGG16 que recordemos que era un modelo de CNN compuesto de 16 capas utilizando filtros de tamaño 3x3. Para ello, lo importaremos a través de la librería de Tensorflow.

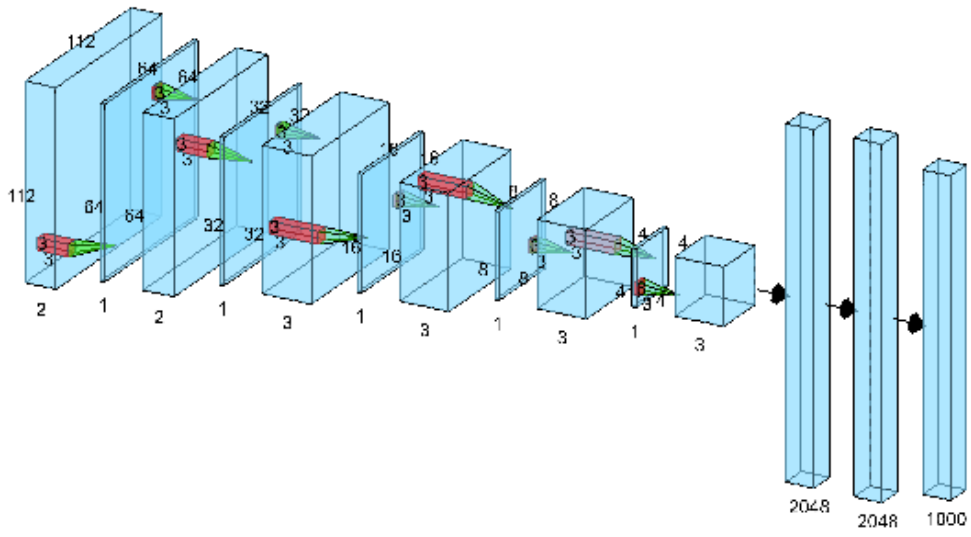

Figura 28. Ejemplo visual de la estructura en cuanto a capas de la red convolucional VGG-16.

Batches	Optimizador	Learning Rate	Capas	Tamaño de filtros
(1. /0.75 - 1) /len(train)	Adam	0.0001	16	3x3

Una vez tenemos la base de datos y el modelo entrenado escribimos el código que lleve a cabo la predicción, en mi caso el proceso a realizar es, a través de la webcam, ir cogiendo el video frame a frame y que a este se le vaya aplicando la predicción del modelo, se establece un umbral en mi caso he ido probando y el que mejor rendimiento ha sacado ha sido el 0.7 el cual si supera será considerado como cara. El paso siguiente es dibujar la bounding box, si se detecta cara se dibujará una bounding box de tamaño moldeable alrededor intentando delimitar el área de las caras.

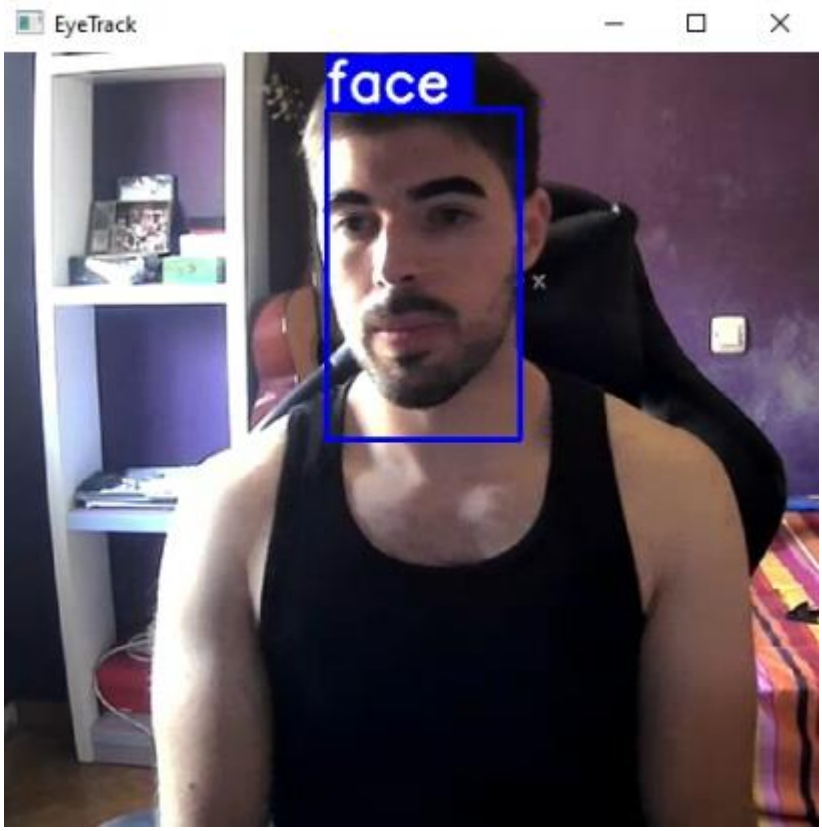

Figura 29. Resultado extraído de la webcam en la que se ve como el algoritmo realiza la predicción y al detectar cara dibuja una bounding box alrededor de esta.

5. Resultados

Para el apartado de resultados, hemos llevado a cabo una predicción de un conjunto de imágenes de test extraídas de la base de datos de DigiFace-1M [51], 20 imágenes y además 10 imágenes aleatorias que no contuvieran caras. Las imágenes han sido etiquetadas manualmente con LabelMe como se hizo para etiquetar el entrenamiento, generando archivo Json por cada imagen. He aplicado data augmentation a esta galería y he obtenido 600 imágenes para comprobar como de bien funciona mi algoritmo, 400 con caras y 200 sin presencia de ellas. Ahora sabiendo que tenemos 400 imágenes con caras y 200 sin caras procedemos a extraer el valor en el campo de clase que hemos introducido previamente de forma manual en el archivo Json. Estos valores los guardamos en un array. Recordemos que el valor 1 indica presencia de cara y el 0 su ausencia.

Ahora he recogido las 600 imágenes y las he pasado por el modelo para que este hiciera su predicción, de igual forma que en su diseño, al valor de la predicción que nos devuelve el modelo se le ha establecido un umbral el cual si pasa devolverá un 1 y si no llega a este valor devolverá un 0. Tras realizar la predicción del conjunto de test hemos obtenido una matriz.

Podríamos comparar manualmente los resultados. Aun así, se puede observar que como era de esperar, el modelo no tiene una precisión del 100%. Para saber que accuracy tiene con exactitud he importado accuracy_scores desde sklearn.metrics el cual me compara ambas matrices y me devuelve el valor. Para este caso nos da un accuracy de 0.945. Si bien hay que saber que este numero no es fijo, dependiendo del conjunto de entrenamiento este número cambiará, aunque nos permite hacernos una idea de lo bien que funciona nuestro modelo.

Para mostrar como funciona se ha decidido mostrar como capta un video real hecho por mi webcam, el proceso será guardar los frames del video y etiquetarlos manualmente a la vez que se guarda la predicción y como en el proceso anterior compararlo con accuracy_scores. El resultado final nos devuelve una comparación de los dos arrays, los de las etiquetas que han sido manualmente establecidas y los que ha devuelto la predicción. El video se compone de 200 frames en los que aparece primero mi padre, este se va y aparezco yo para más tarde salirme del ángulo y volver a entrar justo al final. El algoritmo reordena las imágenes y la etiquetas en función del Uuid por lo que no se aprecia lo que sería el comportamiento correcto.

Una vez hemos obtenido estos arrays, utilizando el comando accuracy_scores hemos obtenido un 0.995% de precisión en el algoritmo. Además, he decidido utilizar otras métricas para observar el rendimiento del algoritmo de manera más precisas. Con el mismo conjunto de test he obtenido los valores del F1 score [52], la matriz de confusión y del AUC [53]. Para entender más a fondo como funcionan estas métricas hay que tener claros 4 conceptos:

- Verdaderos positivos (True Positives, TP): Representa el número de muestras que fueron correctamente clasificadas como positivas por el modelo.
- Falsos positivos (False Positives, FP): Representa el número de muestras que fueron incorrectamente clasificadas como positivas por el modelo.
- Verdaderos negativos (True Negatives, TN): Representa el número de muestras que fueron correctamente clasificadas como negativas por el modelo.
- Falsos negativos (False Negatives, FN): Representa el número de muestras que fueron incorrectamente clasificadas como negativas por el modelo.

El F1 Score es una medida de evaluación del rendimiento de un modelo de clasificación que combina la precisión (precisión) y el recuerdo (recall) en un solo valor. Proporciona una medida equilibrada entre estas dos métricas y se calcula como la media de la precisión y el recuerdo.

Precisión: También conocida como valor predictivo positivo, indica la proporción de verdaderos positivos sobre el total de predicciones positivas. Mide la exactitud de las predicciones positivas del modelo. TP/ TN + FP es la fórmula que se utiliza para su cálculo.

Recuerdo (Recall): También conocido como sensibilidad o tasa de verdaderos positivos, indica la proporción de verdaderos positivos sobre el total de muestras positivas reales. Mide la capacidad del modelo para identificar correctamente las muestras positivas.

```
Informe de clasificación:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99        76
           1       1.00      0.99      1.00       124

    accuracy                           0.99       200
   macro avg       0.99      1.00      0.99       200
weighted avg       1.00      0.99      1.00       200
```

Figura 30. Resultados de la clasificación en función del F1 score con los valores de la precisión y el recuerdo.

La matriz de confusión es una tabla que muestra el desempeño en 4 apartados, los TP, FP, FN y TN respectivamente.

```
Matriz de confusión:
[[ 76   0]
 [  1 123]]
```

Figura 31. Matriz de confusión, que representa las métricas de TP, FP, TN y FN de la predicción de nuestro conjunto de test.

En nuestro caso observamos una predicción casi perfecta a excepción de un falso negativo.

El AUC representa el área bajo la curva ROC y provee una medida cuantitativa de la capacidad de discriminación del modelo. Un AUC más alto indica que el modelo tiene una mejor capacidad para distinguir entre las clases positiva y negativa. Un AUC de 1 indica un modelo perfecto, mientras que un AUC de 0.5 indica un rendimiento aleatorio, donde el modelo no tiene capacidad de discriminación y sus predicciones son equivalentes a una elección aleatoria.

La curva ROC es una representación gráfica que muestra la relación entre la tasa de verdaderos positivos (True Positive Rate, TPR) y la tasa de falsos positivos (False Positive Rate, FPR) para diferentes umbrales de clasificación. El AUC es una medida numérica que resume la curva ROC en un solo valor.

El valor de AUC que ha obtenido nuestro algoritmo en la clasificación ha sido de 0.9959677419354839. Un valor muy cercano al 1 por lo que tiene buen rendimiento.

6. Discusión y conclusiones

En este trabajo de fin de grado, se ha llevado a cabo un estudio sobre las técnicas de aprendizaje automático y usos del reconocimiento facial. El objetivo principal ha sido desarrollar un programa capaz de detectar caras y que dibujara bounding boxes alrededor de estas explorando la efectividad de distintas bases de datos y procedimientos como el data augmentation.

Durante el desarrollo de este proyecto, se han revisado y analizado diferentes enfoques y metodologías utilizadas en el reconocimiento facial en los diversos usos de su día a día. Se ha demostrado que el uso de algoritmos con redes convolucionales produce un rendimiento destacado en la tarea de detección de rostros.

Se ha investigado también la utilización del reconocimiento facial en aplicaciones de seguridad, como la autenticación biométrica y la vigilancia. Se ha encontrado que esta tecnología puede ofrecer un alto nivel de precisión y confiabilidad en la identificación de individuos, lo que la convierte en una herramienta valiosa para la prevención del fraude y la protección de la seguridad.

Además, se ha explorado el uso del reconocimiento facial en campos como la medicina y la psicología. Se ha observado que esta tecnología puede desempeñar un papel importante en el diagnóstico y detección temprana de enfermedades, así como en el análisis de comportamiento y emociones. Sin embargo, es necesario abordar las preocupaciones éticas y de privacidad asociadas con la recopilación y el uso de datos faciales en el ámbito médico.

En cuanto al programa, hemos encontrado diversos desafíos y limitaciones. La poca diversidad de la primera base de datos hacia el programa poco adaptable ya que se empezó con un conjunto de únicamente mi cara en mi cuarto y en posiciones muy similares entre unas y otras. Con la adición de rostros de más personas y diversos escenarios se solvento de manera efectiva este problema. Otro problema fue los escenarios con poca iluminación en los cuales bajaba el rendimiento, aunque las categorías de las imágenes de las caras están desproporcionadas, siendo la mayoría tomadas en mi cuarto (unos dos tercios del total). Esto se arregló aumentando el umbral en el cual el algoritmo detecta la predicción como un rostro.

Otro apartado importante de nuestro proyecto es el data augmentation, hay que destacar su rol ya que nos ha permitido obtener una base de datos de gran tamaño a partir de una cantidad de imágenes raíz pequeña. Como bien sabemos el poder de las redes neuronales reside en lo buena y amplia que sea la base de datos para que esta se entrene adecuadamente.

Para futuros proyectos se propone añadir la verificación facial de rostros, la identificación de personas según sus características más notables lo cual nos acercaría más a los campos del uso del reconocimiento facial en sistemas de seguridad.

7. Anexos

He subido todo el código a GitHub ya que me parece lo óptimo: https://github.com/iperezl2017/TFG.

8. Bibliografía

[1] L. Rouhiainen, "Inteligencia artificial," *Madrid: Alienta Editorial*, 2018.

[2] C. Rubio and M. , "Capítulo 1. El origen de la Inteligencia Artificial, sus caminos y cómo estudiarla," .

[3] F. B. Fitch, "Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133." *The Journal of Symbolic Logic*, vol. 9, *(2)*, pp. 49-50, 1944.

[4] A. M. Turing, "Computing machinery and intelligence," in *Parsing the Turing Test*Anonymous 2009, .

[5] J. Hernández-Orallo *et al*, "Turing Tests with Turing Machines." *Turing-100*, vol. 10, *(140-156)*, pp. 36, 2012.

[6] D. Hinestroza Ramírez, "El Machine Learning a través de los tiempos, y los aportes a la humanidad," 2018.

[7] J. A. Pighin *et al*, "Plant cuticular lipid export requires an ABC transporter," *Science*, vol. 306, *(5696)*, pp. 702-704, 2004.

[8] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, *(6)*, pp. 386-408, 1958. Available: https://www.ncbi.nlm.nih.gov/pubmed/13602029. DOI: 10.1037/h0042519.

[9] (). *¿Qué es el Perceptrón? Perceptrón Simple y Multicapa*. Available: https://aprendeia.com/que-es-el-perceptron-simple-y-multicapa/.

[10] D. Hinestroza Ramírez, "El Machine Learning a través de los tiempos, y los aportes a la humanidad," 2018.

[11] E. F. Franco and R. J. Ramos, "Aprendizaje de máquina y aprendizaje profundo en biotecnología: aplicaciones, impactos y desafíos," *Ciencia, Ambiente Y Clima*, vol. 2, *(2)*, pp. 7-26, 2019. . DOI: 10.22206/cac.2019.v2i2.pp7-26.

[12] (). *Redes neuronales*. Available: https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb.

[13] A. Morera Munt and J. T. Alcalá Nalvaiz, "No title," *Introducción a Los Modelos De Redes Neuronales Artificiales El Perceptrón Simple Y Multicapa*, 2018.

[14] (). *Predicción bursátil con inteligencia artificial*. Available: https://www.linkedin.com/pulse/predicción-bursátil-con-inteligencia-artificial-úbeda-camacho/?originalSubdomain=es.

[15] (). *El perceptrón como neurona artificial*. Available: https://blog.josemarianoalvarez.com/2018/06/10/el-perceptron-como-neurona-artificial/.

[16] (). *Desarrollo de un sistema inteligente para la clasificación de documentos ya digitalizados aplicando redes neuronales supervisadas*. Available: https://www.researchgate.net/figure/Ejemplo-de-una-red-totalmente-conectada_fig1_281380920.

[17] L. Llano *et al*, "Comparación del Desempeño de Funciones de Activación en Redes Feedforward para aproximar Funciones de Datos con y sin Ruido," *Avances En Sistemas E Informática*, vol. 4, *(2)*, 2007.

[18] Irene Alvarado, "Redes Neuronales," Available: https://ml4a.github.io/ml4a/es/neural_networks/.

[19] (). *Redes Neuronales y Deep Learning. Capítulo 2: La Neurona*. Available: https://www.futurespace.es/redes-neuronales-y-deep-learning-capitulo-2-la-neurona/.

[20] C. J. O. Echeverri, D. A. L. Cordoba and E. A. Quintero, "ajuste de hiperparámetros de una red neuronal convolucional para el reconocimiento de lengua de señas," *Con-Ciencia Y Técnica*, vol. 5, *(1)*, pp. 48-55, 2021.

[21] (). *Neural Network and Deep Learning*. Available: https://lrscy.github.io/2018/10/22/DeepLearningNotes-NNandDL/.

[22] (). *Conceptos generales de aprendizaje supervisado*. Available: https://albertotb.com/curso-ml-R/Rmd/02-supervised/02-supervised.html#31.

[23] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv Preprint arXiv:1511.08458*, 2015.

[24] (). *Convolución Matricial Aplicado al Procesamiento de Imágenes*. Available: https://www.tec.ac.cr/sites/default/files/media/uploads/presentacion_pablosoto.pdf.

[25] (). *La Convolución en las Redes Convolucionales*. Available: https://www.codificandobits.com/blog/convolucion-redes-convolucionales/.

[26] (). *Implementing Face Recognition Using Deep Learning and Support Vector Machines*. Available: https://www.codemag.com/Article/2205081/Implementing-Face-Recognition-Using-Deep-Learning-and-Support-Vector-Machines.

[27] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *Journal of Information Processing Systems*, vol. 5, *(2)*, pp. 41-68, 2009.

[28] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, .

[29] C. Choi *et al*, "Face Detection Using Haar Cascade Classifiers Based on Vertical Component Calibration," *Human-Centric Computing and Information Sciences*, vol. 12, 2022. . DOI: 10.22967/HCIS.2022.12.011.

[30] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Jun 2015, Available: https://ieeexplore.ieee.org/document/7298682. DOI: 10.1109/CVPR.2015.7298682.

[31] (). *The intuition of Triplet Loss*. Available: https://medium.com/analytics-vidhya/triplet-loss-b9da35be21b8.

[32] M. Peng *et al*, "NIRFaceNet: A Convolutional Neural Network for Near-Infrared Face Identification," *Information*, vol. 7, *(4)*, pp. 61, 2016. Available: https://search.proquest.com/docview/1858307898. DOI: 10.3390/info7040061.

[33] K. Papadopoulos *et al*, "Face-GCN: A graph convolutional network for 3D dynamic face recognition," in May 26, 2022, Available: https://ieeexplore.ieee.org/document/9847930. DOI: 10.1109/ICVR55215.2022.9847930.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Preprint arXiv:1409.1556*, 2014.

[35] Daniel, "VGG: ¿Qué es este modelo?" Available: https://datascientest.com/es/vgg-que-es-este-modelo-daniel-te-lo-cuenta-todo.

[36] C. A. Corneanu *et al*, "Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, *(8)*, pp. 1548-1568, 2016.

[37] M. Suwa, "A preliminary note on pattern recognition of human emotional expression," *Proc.IJPR*, pp. 408-410, 1978.

[38] K. Mase and A. Pentland, "Automatic lipreading by optical-flow analysis," *Syst. Comput. Jpn.*, vol. 22, *(6)*, pp. 67-76, 1991.

[39] L. Yin *et al*, "A 3D facial expression database for facial behavior research," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, 2006, .

[40] P. Kaur *et al*, "Facial-recognition algorithms: A literature review," *Medicine, Science and the Law*, vol. 60, *(2)*, pp. 131-139, 2020. Available: https://journals.sagepub.com/doi/full/10.1177/0025802419893168. DOI: 10.1177/0025802419893168.

[41] Cavas Martinez, "caracterización de los filtros de Gabor para el estudio de imágenes histológicas de la córnea obtenidas por microscopía con focal" .

[42] (). *Funciones de kernel*. Available: https://aprendeia.com/kernel-maquinas-vectores-de-soporte-clasificacion-regresion/.

[43] F. Z. Canal *et al*, "A survey on facial emotion recognition techniques: A state-of-the-art literature review," *Information Sciences*, vol. 582, pp. 593-617, 2022. Available: https://dx.doi.org/10.1016/j.ins.2021.10.005. DOI: 10.1016/j.ins.2021.10.005.

[44] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, *(1)*, pp. 217-222, 2005.

[45] A. Bud, "Facing the future: The impact of Apple FaceID," *Biometric Technology Today*, vol. 2018, *(1)*, pp. 5-7, 2018.

[46] (). *Face ID*. Available: https://appleinsider.com/inside/face-id.

[47] J. A. C. Osorio, F. A. M. Aguirre and J. A. M. Escobar, "Sistemas de seguridad basados en Biometría," *Scientia Et Technica*, vol. 17, *(46)*, pp. 98-102, 2010.

[48] (). *Reconocimiento facial: qué es y cómo se usa en España y en el mundo*. Available: https://www.diariodesevilla.es/tecnologia/reconocimiento-facial-uso-espana-mundo_0_1484851934.html.

[49] S. Scherer *et al*, "Automatic behavior descriptors for psychological disorder analysis," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2013, .

[50] H. Leal and Daniel, "SEP TNM instituto tecnológico de Culiacán," .

[51] G. Bae *et al*, "DigiFace-1M: 1 million digital face images for face recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, .

[52] (). *How to Calculate Precision, Recall, F1, and More for Deep Learning Models*. Available: https://machinelearningmastery.com/how-to-calculate-precision-recall-f1-and-more-for-deep-learning-models/.

[53] (). *Metrics to Evaluate your Machine Learning Algorithm*. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234.