



**Universitat**  
de les Illes Balears

## TRABAJO DE FIN DE GRADO

# EXPLICACIÓN LIME EN REDES NEURONALES PARA RECONOCIMIENTO FACIAL

**María Isabel Crespí Valero**

**Ingeniería Informática, especialidad en Inteligencia Artificial y computación**

**Escola Politècnica Superior**

**Año académico 2022-23**



# **EXPLICACIÓN LIME EN REDES NEURONALES PARA RECONOCIMIENTO FACIAL**

**María Isabel Crespí Valero**

**Trabajo de Fin de Grado**

**Escola Politècnica Superior**

**Universitat de les Illes Balears**

**Año académico 2022-23**

Palabras clave del trabajo: TFG, memoria, LIME, Inteligencia Artificial, red neuronal

*Tutor: José María Buades Rubio*

Autorizo la Universidad a incluir este trabajo en el repositorio institucional para consultarlos en acceso abierto y difundirlos en línea, con finalidad exclusivamente académicas y de investigación

Autor/a	Tutor/a
Sí <input checked="" type="checkbox"/>	No <input type="checkbox"/>
Sí <input checked="" type="checkbox"/>	No <input type="checkbox"/>



En agradecimiento a las personas más importantes de mi vida.  
A Devi y Carolina, por ser genuinamente geniales. A Odilo, por apoyarme siempre en  
todo.  
A mis hermanos, Jose y Jaime. Por ser un reflejo y mi inspiración diaria. Por ser los dos  
unos luchadores y por enseñarme el valor del esfuerzo.  
A mis padres, por estar siempre allí y hacerme el camino más fácil.  
Gracias a los profesores que me han ayudado con la realización de este TFG. José María  
Buades, Xavi Gayà y Fernando Alonso.  
Sin ninguna de estas personas, nada de esto habría sido posible.  
Mención especial a mis perritos Zarki y Xopi.



# ÍNDICE GENERAL

<b>Índice general</b>	<b>iii</b>
<b>Acrónimos</b>	<b>v</b>
<b>Resumen</b>	<b>vii</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Presentación del ámbito del tema . . . . .	1
1.1.1 Reconocimiento facial innato . . . . .	1
1.2 La inteligencia artificial . . . . .	2
1.3 Objetivos y planteamiento . . . . .	2
1.4 Estructura del documento . . . . .	3
<b>2 Herramientas y entorno</b>	<b>5</b>
2.1 Entorno y herramientas software . . . . .	5
2.1.1 Lenguaje de programación . . . . .	5
2.1.2 Entorno de programación . . . . .	6
2.1.3 Gestión de paquetes . . . . .	6
2.1.4 Módulos utilizados . . . . .	7
2.1.5 Documentación . . . . .	8
2.2 Hardware utilizado . . . . .	9
<b>3 Redes neuronales y Bases de datos</b>	<b>11</b>
3.1 Introducción redes neuronales . . . . .	11
3.2 Redes neuronales convolucionales . . . . .	13
3.3 Otras arquitecturas . . . . .	14
3.4 Redes neuronales residuales . . . . .	15
3.5 ArcFace . . . . .	17
3.6 Bases de datos . . . . .	18
<b>4 Implementación</b>	<b>21</b>
4.1 Planteamiento LIME . . . . .	21
4.2 Preparación modelos y bases de datos . . . . .	24
4.3 Explicaciones LIME . . . . .	24
4.3.1 Preparación formato imágenes . . . . .	24
4.3.2 Normalización vectores . . . . .	25
4.3.3 Aplicar LIME . . . . .	26

4.4	Normalización imágenes . . . . .	27
4.4.1	Detección facial . . . . .	28
4.4.2	Identificación landmarks . . . . .	28
4.4.3	Triangularización de landmarks . . . . .	29
4.5	Creación mapas de calor . . . . .	30
4.6	Creación dendrogramas . . . . .	31
<b>5</b>	<b>Experimentación</b>	<b>35</b>
5.1	Divergencia de Kullback-Leibler . . . . .	36
5.2	Experimentos . . . . .	38
5.2.1	Diferencias entre bases de datos . . . . .	38
5.2.2	Diferencias entre todos los modelos . . . . .	39
5.2.3	Proporción similitud personas con redes . . . . .	42
5.2.4	Particularización: diferencias de un sujeto para todos los modelos	42
5.2.5	Diferencias entre etnias . . . . .	45
5.2.6	Diferencias entre sexos . . . . .	48
<b>6</b>	<b>Conclusiones y aplicaciones futuras</b>	<b>51</b>
6.1	Conclusiones . . . . .	51
6.2	Aplicaciones futuras . . . . .	52
	<b>Bibliografía</b>	<b>55</b>

## ACRÓNIMOS

**CNN** Red Neuronal Convolucional

**ReLU** Rectified Lineal Unit

**LIME** Local Interpretable Model-agnostic Explanations

**KL** Divergencia de Kullback-Leibler

**R18** ResNet18

**R34** ResNet34

**R50** ResNet50

**R100** ResNet100

**IDE** Entorno de Desarrollo Integrado

**GUI** Interfaz Gráfica de Usuario

**NumPy** Numerical Python

**GPU** Unidad de Procesamiento Gráfico

**ResNet** Red Neuronal Residual

**Nats** Unidad natural de información

**FTC** Fallo De Captura

**FTE** Fallo de registro

**UPGMA** Unweighted Pair Group Method with Arithmetic mean

**WPGMA** Weighted Pair Group Method with Arithmetic Mean

## RESUMEN

Las redes neuronales especializadas en reconocimiento facial se están extendiendo en todas las tecnologías y ámbitos de la vida cotidiana. Sin embargo, los modelos son percibidos como cajas negras cuyo funcionamiento interno mantiene cierto misterio. Por lo tanto, ha cobrado un gran interés poder entender el por qué de sus resultados y cuál es el razonamiento que aplican.

En el presente proyecto se trabaja con una técnica de explicabilidad innovativa: Local Interpretable Model-agnostic Explanations (LIME). Ofrece explicaciones individuales para cada muestra que ayudan a entender por qué un modelo da una predicción. Este trabajo emplea un enfoque diferente para el ámbito de la biometría facial. Fundamentándose en el cálculo de distancias entre imágenes, se obtienen explicaciones de las regiones faciales relevantes sin necesidad de encorsetarse en tareas de clasificación.

Esto se consigue utilizando un planteamiento basado en la distancia del coseno entre los vectores de características obtenidos de cada imagen. Ahora se utilizan métricas de distancia, resultando en una puntuación que indica la similitud entre las imágenes. Se sigue un proceso de manipulación de los resultados hasta generar mapas de calor que resumen los rasgos faciales más identificativos.

Los resultados obtenidos muestran ciertas divergencias entre las redes, sobre todo aquellas con mayor diferencia en el número de capas. Por lo general todas indicaron gran fijación en la zona de la nariz. Algunos modelos remarcaron algunos rasgos más que otros y el área de reconocimiento también fluctuaba según la profundidad. Los modelos mostraron también ciertas diferencias por etnia y sexo.

Se podría aprovechar este nuevo enfoque basado en la distancia del coseno para impulsar nuevos estudios relativos a la biometría facial. Una idea podría ser forzar redes a fijarse en rasgos concretos y ver su desempeño. Otra, la importancia de la calidad de imagen para el éxito en la clasificación. También se pueden combinar redes para diferentes tipos de detección. Estas son algunas de las muchas ideas que se pueden explorar en un futuro.





# INTRODUCCIÓN

Este capítulo es un prólogo introductorio sobre los temas que cimientan las bases de trabajo para este proyecto. Se describen los conceptos clave relacionados, las herramientas utilizadas y la estructura de la documentación.

## 1.1 Presentación del ámbito del tema

Todos tenemos una cara, ojos, nariz y boca. La combinación de nuestros rasgos faciales nos convierten en individuos originales y completamente diferentes. De manera innata, los seres humanos tenemos la capacidad de reconocer a los demás en apenas fracciones de segundo, solo necesitamos un simple vistazo.

### 1.1.1 Reconocimiento facial innato

Desde recién nacidos ya aprendemos a reconocer la cara de nuestra madre[1]. Es una habilidad innata que ponemos en práctica constantemente, de manera inconsciente y que nos permite establecer conexiones con los demás. A medida que vamos creciendo desarrollamos esta cualidad hasta convertirnos en verdaderos expertos. Nuestro cerebro se especializa en extraer las características más importantes de cada cara. Pero, ¿Qué es lo que hace a cada persona distinguirse de las demás? ¿Existen propiedades de la cara que facilitan estas distinciones? Se han realizado experimentos para dar respuesta a estas cuestiones desde una perspectiva psicofísica de los seres humanos. Los estudios sugieren que las regiones de mayor fijación están en los ojos. Y de forma más secundaria, boca y nariz [2].

Estos problemas cognitivos de cariz humano también pueden ser desgranados por inteligencias artificiales. Se están realizando grandes avances en el análisis de imágenes para el reconocimiento facial y los resultados obtenidos en algunos casos, son ya incluso mejores[3].

## 1. INTRODUCCIÓN

---

### 1.2 La inteligencia artificial

La inteligencia artificial es la emulación de la inteligencia humana para resolver problemas de índole muy variada [4]. Es un concepto difícil de definir. Pero se podría describir como un sistema informático que trata de imitar las funciones cognitivas humanas, de modo que este adquiere la habilidad de aprender de forma autónoma[5]. Sin necesidad de programación adicional por parte del diseñador. Para ello, se aplican algoritmos y modelos matemáticos que procesan gran cantidad de datos e información y que posteriormente se transformará en conocimiento. Las tomas de decisiones se basan en reglas adquiridas que conceden a una red neuronal el razonamiento necesario para proporcionar resultados útiles[6].

Este sistema de aprendizaje se está utilizando dentro de la visión por computador. Consiste en el uso de algoritmos matemáticos para decodificar y procesar imágenes. Otorgándole al sistema la capacidad de “ver”[7]. Así, la red es capaz de inducir figuras, formas y atributos a través de los píxeles o conjuntos de píxeles de cada imagen.

Es aquí donde entra en juego la tecnología de reconocimiento facial. La cuál radica en la identificación de personas mediante contenido multimedia y audiovisual, como una imagen o vídeo. Se consigue mediante biometría facial, que analiza las características del rostro humano. El sistema tiene la capacidad de asociar la foto de una cara con un sujeto de una base de datos.

A raíz de esto sería lógico plantearse si efectivamente estas redes neuronales artificiales emulan de forma fidedigna a las humanas. Y por lo tanto, a la hora de reconocer caras se fijan en los mismos rasgos con los que lo haría una persona de verdad. También si estos patrones son consistentes entre redes neuronales. Si todos los modelos se fijan en lo mismo o si dependiendo de la profundidad podrían destacar más algunos rasgos que otros. Y también es interesante comprobar posibles sesgos según el sexo o la etnia.

### 1.3 Objetivos y planteamiento

Este trabajo consiste en el análisis de redes neuronales para poder entender en qué se fijan para reconocer caras. Para ello, se aplica el método LIME. La idea consiste en obtener explicaciones en varios ejemplos de sujetos individuales. Con una muestra lo suficientemente grande, se puede alcanzar una mejor comprensión del funcionamiento global del modelo.

El objetivo principal es llevar a cabo un estudio estadístico y analizar cómo funcionan las redes neuronales especializadas en el reconocimiento facial. Es sugerente explorar si las conclusiones a las que llegan estas redes tienen cierto criterio y no surgen del azar; si cada una se fija en los mismos rasgos, cuáles son los más importantes y los posibles sesgos de preentrenamiento que puedan subyacer.

Para poder llevar a cabo el estudio, se ha seguido un flujo de trabajo ordenado y estructurado. Lo primero es decidir qué modelos utilizar y qué base de datos elegir. Luego se procede a aplicar las explicaciones individuales de LIME de todos y cada uno de los sujetos. Se obtienen las máscaras mediante dicho método, utilizando el nuevo enfoque basado en la similitud del coseno y se consiguen las regiones de la cara con mayor importancia para cada persona. Se normalizan las máscaras obtenidas. Para ello, se aplican una serie de landmarks que detectan mismos puntos en cada cara;

posteriormente se conectan y alinean. En particular, se detectan los siguientes rasgos anatómicos[8]: cejas, ojos, nariz, labios y el contorno. Las transformaciones colocan los rasgos en las mismas regiones de píxeles para posteriormente obtener mapas de calor. En resumen, se obtienen las máscaras de mayor importancia, se normalizan y a continuación se realizan mapas de calor que muestren aquellas zonas de la cara con mayor influencia para las redes para el reconocimiento. Finalmente, se realizan varios experimentos que muestran estos mapas de calor según diferentes enfoques. Y a través de la divergencia Kullback–Leibler, se obtienen los dendrogramas relativos para sacar conclusiones de todo el conjunto de datos obtenidos.



Figura 1.1: Esquema que muestra el flujo de trabajo seguido durante todo el proyecto. Se puede dividir en tres fases principales: las explicaciones utilizando el método de LIME, la fase de normalización de las máscaras obtenidas de las explicaciones y una última fase de creación de mapas de calor para todos los sujetos, con su posterior agrupamiento según el caso de estudio particular. En el esquema vemos todos los ajustes que se aplican para una imagen de entrada dada. Al final del proceso se obtienen los mapas de calor y los dendrogramas resultantes de aplicar una métrica comparativa.

## 1.4 Estructura del documento

La estructura del documento se conforma por 5 capítulos más y se organiza de la siguiente manera:

- **Capítulo 2.** Se presenta tanto el hardware como el software. Se especifican las herramientas, librerías y paquetes, lenguaje de programación, el entorno utilizado y el soporte físico utilizado.
- **Capítulo 3.** Se detalla el modelo de explicabilidad LIME y cómo se ha adaptado a los requerimientos del análisis facial en el que se enmarca dicho trabajo. También se explica cómo funcionan las redes neuronales, las diferentes arquitecturas, planteamientos, modos de entrenamiento posibles y otros conceptos relacionados. También hay una sección clarificando los modelos y las bases de datos.
- **Capítulo 4.** Se profundiza en el proceso de la implementación de los datos, desde el método LIME, la normalización, la generación de los mapas de calor y dendrogramas. También se da una explicación de los conceptos relacionados en cada apartado.

## 1. INTRODUCCIÓN

---

- **Capítulo 5.** Se introduce el concepto de Divergencia de Kullback-Leibler (KL), que es la métrica de distancia que se utilizará para comparar los experimentos realizados. Se aplica un análisis a los resultados obtenidos.
- **Capítulo 6.** Por último, se exhiben las conclusiones de todo el estudio desde una perspectiva global y el enfoque futuro con el cuál se podría continuar dicha investigación.

CAPÍTULO



## HERRAMIENTAS Y ENTORNO

Para poder empezar este trabajo, es importante tener claro cuáles son las herramientas del proyecto. Esto es fundamental debido a que influyen directamente en la manera de trabajar, en los recursos y facilidades de los que se dispondrán para resolver los posibles retos que vayan surgiendo.

### 2.1 Entorno y herramientas software

El entorno y el lenguaje de programación suponen una elección importante para un desarrollo adecuado del trabajo. El ámbito en el cual se enmarca el proyecto es el aprendizaje automático, así que utilizar un lenguaje que presente librerías de calidad para manejar redes neuronales sería lo propio.

#### 2.1.1 Lenguaje de programación

Se han contemplado diversas opciones como Python, MATLAB, R o Julia. MATLAB es utilizado en investigaciones técnicas[9] y los docentes proporcionaron material y facilidades para este proyecto. R es un lenguaje muy útil en el ámbito de la estadística y el análisis de datos[10], lo cual también hubiera sido muy positivo. Julia también está especializado en aprendizaje automático y con una gran eficiencia operacional, su popularidad ha aumentado considerablemente los últimos años[11].

Finalmente, Python ha sido la elección final. Ya se cuenta con experiencia en el lenguaje y es el utilizado por el equipo de investigación de la universidad. Ofrece más facilidades y recursos de cara al proyecto.

Es un lenguaje de alto nivel, interpretado, orientado a objetos y tipado dinámicamente. Además es multiparadigma, por lo que se otorga cierta libertad al programador a la hora utilizar diferentes estilos de programación[12].

Es fácil de utilizar y es perfecto para trabajar con los modelos neuronales. Esto debido a que cuenta con una gran cantidad de librerías que ayudan a facilitar cálculos

## 2. HERRAMIENTAS Y ENTORNO

---

y algoritmos de aprendizaje automático. Es muy cómodo trabajar con las estructuras de datos, la sintaxis y semántica para escribir código.

### 2.1.2 Entorno de programación

La elección del Entorno de Desarrollo Integrado (IDE), es también importante. Una opción es Visual Studio Code, con el cual ya se tiene experiencia previa y cuenta con gran cantidad de extensiones, además de una gran adaptabilidad para sobrellevar diferentes flujos de trabajo[13]. Otra opción contemplada ha sido Jupyter Notebook, que aunque no es un IDE como tal, también es una herramienta que permite la programación y el análisis de datos de una forma más interactiva[14].

La opción final y con la cuál se trabaja es Pycharm. Es compatible con Windows, Linux y macOS. El proyecto se ha ido desarrollando paralelamente en dos sistemas operativos, por lo que esta condición es relevante. Uno de los sistemas es un ordenador de torre ubicado en los laboratorios de la universidad, cuyo entorno es Windows. Y adicionalmente también se han realizado múltiples pruebas en un portátil particular que utiliza macOS.

Es un entorno adecuado debido a que contiene módulos de aprendizaje automático que facilita el trabajo a la hora de programar, ofreciendo soluciones rápidas a ciertos desafíos y optimizando así el tiempo de desarrollo[15].

Además la edición de código es muy cómoda y la interfaz es intuitiva. De modo que la localización de errores se facilita bastante, existiendo una ubicuidad de control de código constante. Es fácil moverse por el programa escrito y por la estructura de carpetas asociada al proyecto. Además de las opciones de limpieza y depuración.

Otro punto importante es que es compatible con el sistema de gestión de paquetes de Conda. Favoreciendo así la integración de paquetes para trabajar con herramientas de modelos neuronales.

### 2.1.3 Gestión de paquetes

Para la gestión de paquetes se ha optado por utilizar Conda; se diseñó originalmente para programar en Python y lógicamente es compatible con Pycharm. Conda es un sistema de código abierto que facilita la instalación, gestión y actualización de paquetes. Por lo que es útil para manejar dependencias y módulos de inteligencia artificial. También es compatible con ambos sistemas operativos Windows y macOS[16]. Mediante Anaconda, que es la Interfaz Gráfica de Usuario (GUI), se maneja esta gestión del marco de trabajo.

Se crea un entorno personalizado dónde se instalan todos los paquetes necesarios. Este es un paso necesario para crear un espacio virtual donde aislar recursos que son independientes de otros proyectos.

Además como ventaja adicional, ya se tiene cierta experiencia de cursos anteriores con dicha herramienta.

Y también proporciona un sistema de línea de comandos que facilita la instalación de las extensiones con la ayuda de una terminal preinstalada en la propia GUI y en el IDE.

### 2.1.4 Módulos utilizados

Esta parte también es importante detallarla ya que ofrecen una visión global sobre las necesidades y las tareas a cumplir para el desarrollo adecuado del trabajo.

- **ONNX Runtime.** Es un motor de inferencia que permite utilizar modelos previamente entrenados de forma muy accesible[17]. Una de las ventajas del formato ONNX es que encapsula el marco de trabajo de creación de la red neuronal. Esto habilita la interoperabilidad y facilita el uso de estas redes para predicciones de nuevos datos. El caso particular que atañe a este trabajo, únicamente se cargan las redes ya entrenadas para poder aplicar nuevas predicciones. No es preciso ninguna manipulación ya que la salida proporciona los datos esperados.
- **Numpy.** Numerical Python (NumPy) es una biblioteca especializada en cálculos lógicos y matemáticos sobre vectores multidimensionales. Facilita enormemente las operaciones entre vectores y permite manejar grandes conjuntos de datos de una manera más eficaz que con las listas tradicionales del propio lenguaje[18]. En el caso de dicho proyecto, este paquete se utiliza para cálculos de datos en imágenes, detección de landmarks, entre otros.
- **Imutils.** Es una biblioteca de Python especializada en el procesamiento de imágenes. Facilita una gran cantidad de operaciones, como cambio de tamaño, rotación, clasificación de contornos, desplazamiento, entre otras[19]. En particular, se despliega el módulo *face\_utils* que ofrece funciones para trabajar con landmarks.
- **Dlib.** Es una librería que contiene algoritmos de visión por computadora para la detección facial y los puntos claves de un rostro humano. Es muy conocida por la detección de 68 puntos claves[20].
- **Cv2.** OpenCV es otra potente librería especializada en la manipulación de imágenes. Incluye la carga y visualización de imágenes, reconocimiento facial, detección de objetos, entre otras muchas cosas[21].
- **SciPy.** Es una librería enmarcada en cálculos matemáticos de diferente carácter. Destaca en la optimización, álgebra lineal, integración, interpolación, entre otros[22]. En particular, se utiliza el subpaquete *scipy.spatial* que se especializa en cálculos geométricos para estructuras de datos. Y concretamente, se usa el módulo *Delaunay*, cuya función es la de aplicar la técnica de triangulación de Delaunay para dividir un plano en triángulos no superpuestos y así juntarlos con los landmarks en los rostros de las personas.  
Por otro lado, también se usa del submódulo *Hierarchy*, a su vez incluido dentro del submódulo *Cluster*. Este último contiene herramientas para el tratamiento de agrupaciones de datos, mientras que el primero es para la gestión de dichas jerarquías. Estas funciones consisten en transformaciones para agrupamientos de datos. Realiza un aplanamiento de los clusters fusionando aquellos cuya distancia sea menor entre sí. En particular se utiliza la función *dendogram*, que lo que hace es crear un gráfico en forma de árbol de dichas agrupaciones. Y la

## 2. HERRAMIENTAS Y ENTORNO

---

función *linkage* que realiza el clustering jerárquico.

- **Os.** Es un módulo integrado de Python que permite acceder a funcionalidades y detalles relativos al Sistema Operativo y su entorno. Es muy útil para toda la gestión de directorios de manera más sencilla[23].
- **Functools.** Es un módulo estándar de Python que provee funciones que operan sobre otras funciones[24]. *Partial* es una función de orden superior que permite fijar los argumentos. Equivale al método original, pero dichos argumentos actúan como constantes.
- **Torch.** Torch es una biblioteca especializada en el aprendizaje profundo, para la tareas de diferenciación automática y cálculos de tensores. También permite la aceleración de cálculos en la Unidad de Procesamiento Gráfico (GPU) [25].
- **Skimage.** Skimage es una librería que proporciona una colección de algoritmos para el procesamiento de imágenes[26]. Del módulo *Segmentation* se utilizan dos funciones. *Slic* (Simple Linear Iterative Clustering) es una función para la segmentación de imágenes en superpíxeles. Esto es útil ya que contienen más información contextual para el análisis de regiones en imágenes. Utiliza un enfoque basado agrupamiento K-Medias en un espacio de color(x, y, z). *QuickShift* es otro módulo que ofrece un enfoque alternativo de identificación de superpíxeles, realizando el agrupamiento en un espacio de color (x, y). Produce una sobresegmentación en la imagen. El segundo módulo de esta librería utilizado es *Color*, que se especializa en las conversiones de espacios de color, de manera que no se vean alteradas las características inherentes de la imagen. La función *Label2RGB* convierte la imagen de CIE-LAB al espacio de color RGB[27].
- **Sklearn.** Es una biblioteca especializada en el aprendizaje automático que proporciona modelos de clasificación, regresión y análisis de grupos[28]. En particular, el submódulos *Metrics* lo que hace es ahondar en las métricas que evalúan el rendimiento de dichos algoritmos. La función *f1\_score* lo que hace es calcular la métrica con dicho nombre. El f1 score consiste en la combinación de las medidas de Recall y Precision en un solo valor. Básicamente se calcula la media armónica[29].

### 2.1.5 Documentación

Para la parte de documentación, se pueden encontrar una cantidad inmensa de opciones, tales como Microsoft Word, Google Docs, Sphinx, entre otros.

Al final se ha optado por el editor de texto en línea llamado Overleaf. Esta herramienta es muy útil ya que utiliza LaTeX[30], el cuál es un lenguaje de estructuración de texto mediante comandos y etiquetas. Facilita al usuario la estructuración lógica, permite realizar un control de versiones, la exportación en diferentes formatos, entre otras muchas ventajas[31]. Es ampliamente usado dentro el contexto de artículos de investigación, académicos y de cariz técnico en general. Además, la tipografía es equiparable a las de las editoriales científicas[32]. En particular, se está utilizando una

## 2.2. Hardware utilizado

---

plantilla proporcionada por la Escola Politècnica Superior de la Universitat de les Illes Balears.

### 2.2 Hardware utilizado

Se han utilizado dos sistemas físicos para el almacenamiento y procesamiento de los datos del proyecto. Uno de ellos es el ordenador principal, el cual ha sido prestado por la Universitat de les Illes Balears. Es aquí dónde la mayor parte del estudio ha sido realizado. Todos los cálculos relevantes se obtuvieron aquí. Por otro lado, también se ha contado con un portátil personal en el cual se han podido realizar pruebas y avanzar en otros aspectos relativos como la documentación o depuración de código. Las características de cada uno se resumen en la siguiente tabla:

	Ordenador laboratorio	Ordenador personal
Sistema Operativo	Microsoft Windows 10 Pro 22H2	MacBook Pro
Procesador	AMD Ryzen 7 5700G 3.80 GHz	Intel Core i5-8257U 1.40GHz
Gráficos pantalla	NVIDIA GeForce RTX3090	Intel Iris Plus Graphics 645
Memoria RAM	16 GB	16 GB

Cuadro 2.1: Resumen dispositivos hardware



CAPÍTULO



## REDES NEURONALES Y BASES DE DATOS

En este capítulo se profundiza en el concepto de red neuronal, cuáles son sus bases matemáticas y la lógica en la que se fundamenta. Se indagará en las redes neuronales convolucionales y residuales, que son los cimientos de los modelos de reconocimiento facial. También se exploran otras arquitecturas, conceptos como el método de entrenamiento de estas redes y de las bases de datos utilizadas. Finalmente, se muestra el resumen de los modelos que se utilizan para los experimentos posteriores.

### 3.1 Introducción redes neuronales

La idea de concebir el cerebro como una representación computacional para resolver problemas fue iniciada por Alan Turing en 1936[33]. Unos años más tarde se empezarían a modelar las neuronas asociándolas a circuitos eléctricos. La formulación de la neurona como concepto matemático requiere de ciertas hipótesis simplificadoras debido a que las células biológicas siguen siendo demasiado complejas para entender su total funcionamiento. Así que solo se tienen en cuenta aquellos parámetros que suponen de mayor relevancia para su representación[34].

La neurona artificial es la unidad de cálculo básica en un modelo de aprendizaje profundo. Se construye a través de una serie de entradas, provenientes de, o bien las uniones con otras neuronas[35], o de los datos que se quieren procesar. Estas conexiones tienen asociadas un peso que indica la contribución al resultado final; se aplica la suma ponderada de los valores de entrada junto a dichos pesos, además del sesgo. El resultado de esta suma se pasa a una función de activación que decidirá si la neurona se activa o no y por lo tanto, transmite la información[36].

El proceso de aprendizaje consiste en el ajuste de esos pesos hasta obtener la salida deseada. Para adaptar esos pesos se utilizan reglas de aprendizaje. La operación de una

### 3. REDES NEURONALES Y BASES DE DATOS

---

#### DISEÑO BÁSICO NEURONA ARTIFICIAL

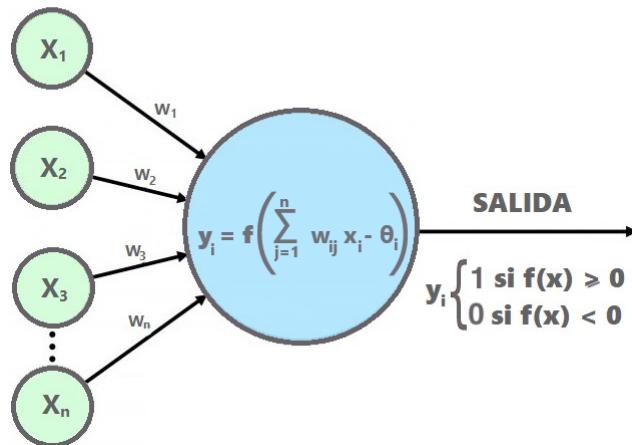


Figura 3.1: Modelo de una neurona artificial básica, también conocida como Perceptrón.

neurona simple puede definirse como:

$$y_i = f\left(\sum_{j=1}^n w_{ij} x_i - \theta_i\right)$$

Estas neuronas se organizan en capas que conforman una arquitectura compleja, conocida como red neuronal multicapa. Se puede definir como un modelo matemático, además de un método de inteligencia artificial. Como ya se ha mencionado, el concepto se basa en las estructuras neurobiológicas que conforman el cerebro.

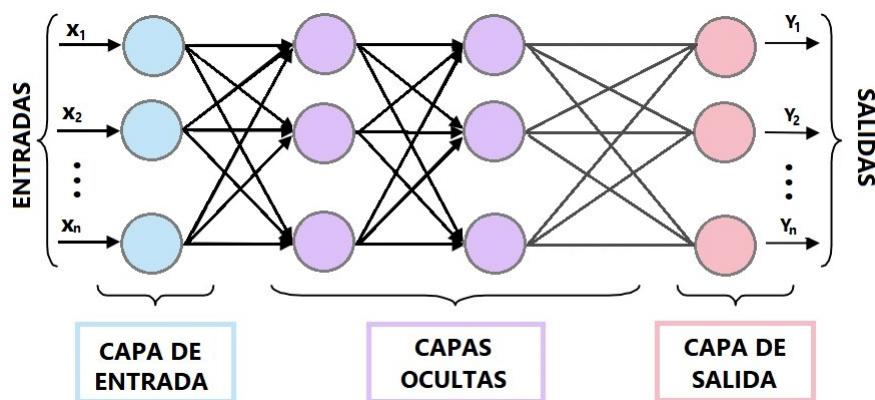


Figura 3.2: Estructura de una red neuronal multicapa. Básicamente consiste en un conjunto de neuronas conectadas entre si. Se puede distinguir la capa de entrada, que es la puerta de entrada donde se reciben los datos a procesar. A la salida de la red se encuentra la capa de salida, que proporciona los resultados y las predicciones. Y en medio, se encuentran las capas ocultas que son las que extraen parámetros y características que conforman el razonamiento de la arquitectura.

Las cuatro propiedades fundamentales de una red neuronal son las siguientes:

- **La topología.** Es la estructura de interconexión entre neuronas. Esto engloba la cantidad de capas, número, tipo de conexión entre neuronas y grado de conectividad de estas.
- **Los mecanismos de aprendizaje.** Es la regla que adapta los pesos de cada neurona. La finalidad es maximizar la velocidad del aprendizaje. O visto desde otra perspectiva, minimizar el error de la salida.
- **Tipo de asociación entre la información de entrada y de salida.** Las redes pueden aprender o por parejas de datos o por correlación entre esos datos almacenados.
- **Representación de la información.** Las salidas de las redes pueden proporcionar datos continuos o discretos.

## 3.2 Redes neuronales convolucionales

Existen distintos tipos de redes neuronales adaptadas según la tarea que se quiera solucionar. Este documento se focalizará en las redes neuronales convolucionales que son las más efectivas para el procesamiento de imágenes.

Una Red Neuronal Convolutacional (CNN) es una red neuronal jerárquica y ordenada por niveles, cuya inspiración proviene de las neuronas que se encuentran en la corteza visual del cerebro. Se especializa en el análisis y procesamiento de imágenes [37]. Los datos de entrada son matrices que representan las imágenes, los dos canales principales son la anchura, la altura y adicionalmente se suele añadir una dimensión más como canal de color[38].

En este tipo de redes solo una pequeña región de neuronas de entrada se conectan a las siguientes capas ocultas. En lugar de utilizar todas las conexiones, se utilizan múltiples filtros en forma de kernels que extraen las singularidades más relevantes de cada región. Esto se conoce como **convolución**. Estos kernels realizan una serie de cálculos matriciales donde se aplican multiplicaciones y sumas para finalmente obtener una matriz nueva que represente esas abstracciones en forma de mapa de características.

Otra de las cualidades fundamentales de este tipo de redes son las capas de **pooling** o agrupamiento. Las cuales se utilizan para reducir la dimensionalidad del mapa de características resultante de la capa de convolución. Esto se consigue mediante la aplicación de una ventana deslizante que almacena el valor máximo o la suma ponderada de los mapas de características[39]. De esta forma, acotamos a solo aquellas características verdaderamente relevantes para la imagen, por lo que reducimos el sobreajuste al minimizar el número de parámetros de la red. Lo que también supone un ahorro significativo del coste computacional, habilitando así poder construir arquitecturas mucho más eficientes y potentes. Además dichas transformaciones permiten a la red reconocer características globales de la imagen.

En las CNN, los pesos y el sesgo son los mismos para todas las neuronas de una capa oculta particular. Por lo que cada capa se especializa en la detección de una misma característica, patrón, textura o borde[40]. A medida que se avanza en las capas, la

### 3. REDES NEURONALES Y BASES DE DATOS

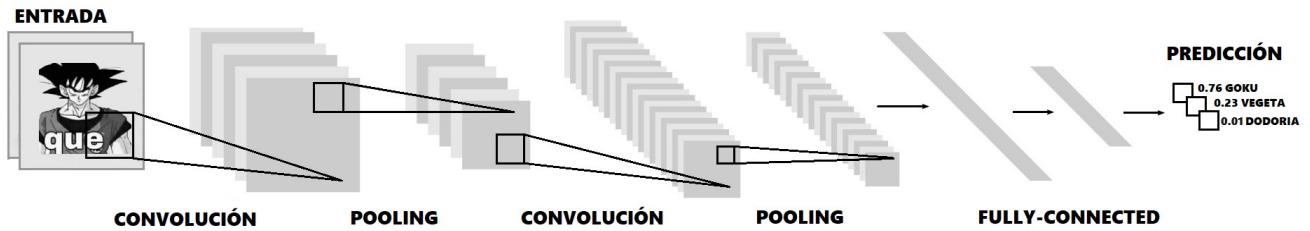


Figura 3.3: Idea básica de una red neuronal convolucional. Los datos de entrada son la imagen representada en formato matricial. En las siguientes capas se van aplicando filtros o kernels para la eliminación de redundancia y obtención de aquellas características más importantes. Finalmente, en las últimas capas se conectan todas las neuronas. En la penúltima el resultado es un vector con aquellas características más importantes de la imagen y en la última se obtiene un vector con las predicciones de las clases de estudio.

complejidad de los patrones va aumentando. En la última capa todas las neuronas se conectan entre sí.

También es importante remarcar la importancia de la función de activación de las neuronas. En el caso de las CNN, es habitual la función Rectified Lineal Unit (ReLU), que introduce no linealidad y habilita el aprendizaje. Básicamente consiste en que la neurona sólo se activará si la entrada es positiva y le asigna el valor más alto. En caso contrario, se asigna un cero.

Además, debido a su implementación sencilla, evita en cierta manera el problema de desvanecimiento del gradiente. Esto es, que se vuelven demasiado pequeños a medida que se avanza en las capas, por lo que la actualización de los pesos se vuelve prácticamente exigua. Es decir, que no los valores se mantienen prácticamente iguales[41].

### 3.3 Otras arquitecturas

Con la popularización de las CNN empezaron a surgir nuevas redes neuronales que han ido puliendo conceptos ya conocidos e introduciendo nuevas ideas y enfoques diferentes para el reconocimiento facial. Es clave explorar cuáles son y en qué destaca cada una, ya que ofrecen estructuras diferentes para tratar de mejorar el rendimiento lo máximo posible.

Las principales son las siguientes:

- **AlexNet.** Este modelo es el primero en aplicar con éxito una CNN para la clasificación de imágenes a gran escala. La red consta de 5 capas convolucionales, seguidas de 3 capas completamente conectadas[42]. Utiliza ReLU para la parte no lineal, en lugar de una función Tanh o Sigmoid que era el estándar anterior para las redes neuronales tradicionales[43]. Esta función de activación evita que las neuronas se saturén[44].
- **VGGNet.** Nació de la necesidad de reducir el número de parámetros en las capas y mejorar el tiempo de entrenamiento[42]. Esto lo consigue utilizando un kernel

3x3, permitiendo aumentar la profundidad de la red[44]. Consta de 16 capas convolucionales y presenta una arquitectura muy uniforme[45].

- **GoogLeNet/Inception.** Este modelo se basa en el uso de convoluciones de diferente tamaño, destacando aquellos kernels 1x1 que reducen drásticamente el número de características. Se basa en la idea de que el tamaño de las características del marco de las imágenes es variable. Se establecen kernels de diferente tamaño para poder aprender información a diferentes niveles de abstracción. Los núcleos más grandes adquieren características más globales mientras que los más pequeños se centran en patrones más específicos. Esta red plantea la utilización de núcleos de diferentes tamaños. Consta de 22 capas de profundidad y reduce los 60 millones de parámetros de AlexNet a tan solo 4 millones[46].
- **DenseNet.** Presenta una arquitectura muy profunda, donde conecta cada capa con todas las demás. Esto se llama Bloques Densos[45] y alivia el problema de desvanecimiento del gradiente, permite propagación, reutilización de características y reduce el numero de parámetros.
- **LightCNN.** Presenta un Max-Pooling que separa características ruidosas y señales informativas[46].

También mencionar otros modelos como DDML, Center-Loss, SphereFace, CosFace y UniformFace, entre otros. Aunque los más relevantes por su impacto en el desarrollo en la visión por computador son los detallados arriba.

Por último es fundamental ver qué métricas se utilizan para comprobar la fiabilidad de los modelos especializados en reconocimiento facial. Los más habituales son los siguientes[47]:

- Fallo De Captura (FTC). Es la proporción del número de veces que un sistema biométrico no logra capturar la muestra que se le presenta.
- Fallo de registro (FTE). Es la relación entre la cantidad de usuarios que no pueden registrarse positivamente y la cantidad total de usuarios presentados al sistema biométrico.

Existen otras métricas, que son la particularización de las típicamente utilizadas en clasificación: Precision, Recall, F1 y Accuracy. Cabe mencionar que algunas medidas son más adecuadas para la verificación, otras para reconocimiento. [48]

## 3.4 Redes neuronales residuales

Sería bastante razonable pensar que la profundidad de las redes neuronales afecta siempre positivamente en el aprendizaje y que cuánto mayor es esta, mayor es el nivel de abstracción del conocimiento debido al aumento del número de neuronas. Sin embargo y de forma contra intuitiva, esto no es completamente cierto. Se ha podido comprobar que llega un punto en el que se produce un estancamiento en la calidad del aprendizaje y comienza a degradarse. De hecho, aparecen problemas que hasta

### 3. REDES NEURONALES Y BASES DE DATOS

---

entonces permanecían latentes, como el desvanecimiento del gradiente y la maldición de la dimensionalidad. A raíz de esto, nacieron nuevos métodos para intentar resolver estos nuevas retos. Es aquí donde las redes neuronales residuales suponen una gran innovación.

Las redes neuronales residuales son una particularización de aquellas basadas en arquitecturas convolucionales. Se inspiran en las células de la pirámide de la corteza cerebral [49] (conocidas como excitatorias) y que reciben conexiones procedentes de diferentes núcleos localizados [50]. Comunican áreas del cerebro sin pasar por capas de neuronas intermedias.

Y este es básicamente el concepto esencial de estas redes, cuya característica fundamental son las conexiones atajo que conectan neuronas de distintas zonas, saltándose capas intermedias y transmitiendo directamente los parámetros sin necesidad de ningún filtro adicional [51]. Además de atenuar el problema de desaparición del gradiente, permitiendo que este viaje directamente hacia la entrada a gracias a estas conexiones [52].

Para poder desactivar capas se introdujo el concepto de bloque residual. Básicamente permite que la salida de una capa se combine con cualquier otra que no sea la adyacente. Este bloque está constituido por una secuencia de capas convolucionales y de normalización, seguida de una función de activación no lineal [53]. Estas alteraciones se anotan como:  $F(x)$ . Del otro lado le llega información sin procesar, como una identidad. Se anota como:  $x$ .

La fórmula queda así:

$$H(x) = F(x) + x$$

Al final de cada bloque residual se mezcla información que ha pasado por las capas de convolución con otra que ha atajado por las conexiones y se produce una mezcla de información semántica a diferentes niveles de abstracción [52]. Esto potencia el rendimiento de los sistemas de reconocimiento visual.

Estos saltos solo se aplican durante la etapa de entrenamiento, evitando complicaciones innecesarias, simplificando y acelerando el proceso de aprendizaje. Una vez finalizado el entrenamiento, se vuelven a habilitar todas las capas que componen la red.

Para entrenar una Red Neuronal Residual (ResNet) suele ser habitual aplicar lo que se conoce como *profundidad estocástica*. Consiste en desactivar para cada iteración un número de capas aleatorio; estas no se tendrán en cuenta y la profundidad de la red se vuelve variable. Según la iteración habrá más o menos capas de profundidad y esto es útil como técnica de regularización ya que evita el sobreajuste [52]. También permite que los gradientes tengan un recorrido menor hasta la entrada.

Para la realización de este trabajo se han escogido cuatro modelos, proporcionados por uno de los investigadores ayudantes. El objetivo es analizar qué diferencias en el aprendizaje y desempeño sugiere la profundidad de cada arquitectura. Estas son: ResNet18, ResNet34, ResNet50 y ResNet100.

Los modelos que se utilizan en el ámbito de este proyecto fueron entrenados con el método ArcFace.

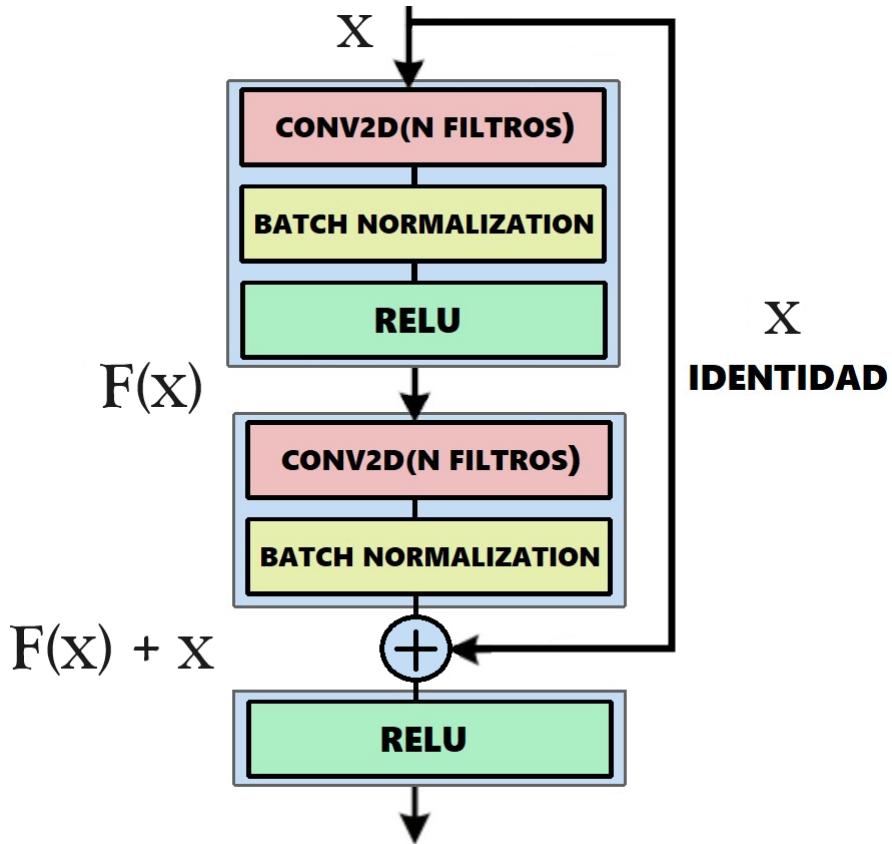


Figura 3.4: Un bloque residual se conforma por dos partes. A la izquierda se representa la secuencia de filtros aplicados para una información. A la derecha y de forma independiente, la información pasa sin ningún tipo de alteración.

### 3.5 ArcFace

ArcFace es un algoritmo de aprendizaje automático, basado en la interpretación geométrica del cálculo de la distancia coseno, para la discriminación de caras en imágenes[54]. Está basado en las CNN y se especializa en el reconocimiento facial. Es una técnica que ha ganado gran popularidad y se utiliza de forma muy extendida.

La similitud del coseno entre los vectores  $A$  y  $B$  se calcula utilizando la siguiente fórmula:

$$\text{Similitud del coseno} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Representar la distancia del coseno como un producto escalar en forma de tensor es muy práctico debido a que las GPU modernas están diseñadas para realizar operaciones intensivas con estructuras matriciales [55]. Los cálculos se optimizan y el proceso se vuelve más eficiente.

La distancia del coseno mide la similitud del ángulo entre dos vectores normalizados en un espacio euclíadiano y se calcula como el producto escalar de estos. El producto

### 3. REDES NEURONALES Y BASES DE DATOS

---

escalar mide la proyección de un vector sobre el otro y se puede obtener aplicando operaciones algebraicas básicas, permitiendo cuantificar la similitud. Si los vectores son iguales, la distancia será 0. Cuando sean completamente ortogonales, la distancia será 1.

Dada una imagen de una cara, la red neuronal construye un vector de las N características más importantes. Este podrá ser comparado con el vector de otra imagen mediante la función coseno, obteniendo un valor de distancia que ayudará a discernir si la persona es la misma.

ArcFace se puede entender como una función de pérdida de margen angular. Es una mejora de la función Softmax[56], sigue representando una distribución categórica en un vector de probabilidades que suman uno[57]. Pero ahora además permite a las CNN aprender características que son angularmente discriminatorias.

La función SoftMax es una generalización de la regresión logística, utilizada para la clasificación binaria. En este caso, soporta sistemas de clasificación para diversas clases, por lo que se suele utilizar como salida en los modelos neuronales de reconocimiento facial profundo[58]. La salida son las distribuciones de probabilidad de cada clase, con un rango acotado entre [0, 1] y sumando 1 en total. Así mismo, trata de minimizar la entropía cruzada, que es la diferencia entre las distribuciones de probabilidad de las clases verdaderas y las predichas por el modelo. Cuanto más pequeño sea el valor obtenido, más fidedignas a la realidad serán las predicciones.

Sin embargo, no es óptima para la tarea de reconocimiento facial ya que no incorpora representaciones vectoriales que faciliten la distinción de diferentes clases y la similitud entre clases iguales.

La diferencia de ArcFace respecto a SoftMax radica en que se añade un término angular a la función de pérdida. Se imponen restricciones como el margen angular mínimo que debe haber entre vectores de diferentes clases.

Este margen angular minimiza de forma adaptativa los ángulos entre características de la misma clase y aumenta para las clases diferentes. Estas restricciones mejoran la separación interclase y la compactación intrACLASE en un espacio hiperesférico. Consecuentemente, también mejora la capacidad de clasificación del modelo ya que las representaciones serán más distintivas. Esto aumenta la variabilidad y mejora significativamente la discriminación de los rasgos faciales. Ahora las predicciones sólo dependen del ángulo entre las características y pesos asociados[59].

La función es la siguiente:

$$L = -\log \left( \frac{e^{(s \cdot \cos(\theta_{y_i} + m))}}{e^{(s \cdot \cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^N e^{(s \cdot \cos(\theta_j))}} \right)$$

## 3.6 Bases de datos

Las bases de datos son la fuente a través de las cuales las redes neuronales comprenden el entorno en el que se van a especializar. Cuantos más casos de entrenamiento nutran el modelo, más puntos de referencia tendrá para aprender, mayor será el nivel de abstracción de la información y por lo tanto desarrollará una mejor inteligencia. Las decisiones sobre qué características, rasgos o formas deben tenerse en cuenta, provienen de esta fase de preentrenamiento. La importancia de proporcionar una cantidad

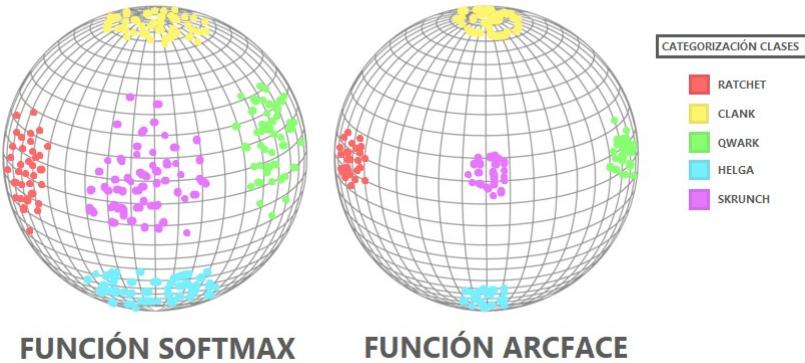


Figura 3.5: Representación de las características de cada clase en un espacio esferoidal. Cada color representa una categoría diferente, pertenecientes a las caras de los cinco individuos de la muestra. En la esfera de la izquierda vemos cómo la distribución de características está más espaciada que en la derecha. Por lo tanto, a la hora de categorizar entre las clases, será mucho más fácil hacerlo a través de la función ArcFace. El margen angular interclase aumenta y el de las características comunes, disminuye.

variable y adecuada de datos, es crucial para evitar posibles sesgos y aumentar así la confianza en las predicciones.

Las bases de datos para el preentrenamiento de las redes neuronales de este proyecto son MS1MV3 y Glint360K. Ambas contienen imágenes de rostros de personas y son ampliamente utilizadas dentro del campo de reconocimiento facial. Se han elegido estas porque son las utilizadas en los modelos ResNet, proporcionados por el equipo docente.

A continuación vamos a detallar cada una:

- **MS1MV3.** Es una base de datos masiva que forma parte del proyecto Microsoft-Celebrities[60]. En este caso, todas las imágenes han sido preprocesadas por el detector de caras Retina-Face. Tienen un tamaño de 122 x 122 y contiene 5,1 millones de imágenes de 93,400 identidades diferentes. Incluye en gran variedad de expresiones faciales a celebridades del mundo real, políticos, actores, músicos, deportistas entre otras figuras públicas. Cabe recalcar que la variación intraidentidad es limitada con un promedio de 81 imágenes por persona.
- **Glint360K.** Es el conjunto de datos de reconocimiento facial más grande, contiene 17,091,657 imágenes de 360,232 personas [61]. Lógicamente también cuenta con gran multitud de variedad de personajes públicos en diferentes situaciones y de diferentes etnias. Además las imágenes han pasado cierto control de calidad.

Estas redes han incluido de forma razonable y proporcional imágenes de identidades de razas variadas para intentar mitigar un posible sesgo discriminatorio al respecto.

Por otro lado, la base de datos utilizada para los experimentos de este proyecto es la conocida como VGGFace2, también proporcionada por el equipo docente. Aquí están los detalles:

### 3. REDES NEURONALES Y BASES DE DATOS

---

- **VGGFace2.** Es un conjunto de datos de 3,31 millones de imágenes de 9131 sujetos diferentes, con 363,6 imágenes para uno en promedio. Las imágenes se obtuvieron a través de Google Image y se diseñó pensando en cubrir una amplia gama de poses, edad, etnias y profesiones[62]. Conduce a un mejor rendimiento en los entrenamientos por edad y pose. Para los experimentos de este proyecto se define un subconjunto de VGGFace2-Pose, con 368 sujetos con 10 imágenes para cada uno. Solo se tendrá en cuenta la pose frontal.

La base de datos para el proceso de experimentación es distinta a las de preentrenamiento, por lo que no se pueden llevar a cabo tareas de clasificación. En vez de eso, se obtienen las máscaras de las explicaciones LIME y se crean los mapas de calor con los sujetos de VGGFace2.

Arquitectura	Dataset	Africanos	Caucásicos	Asiáticos sur	Asiáticos este
R18	MS1MV3	62.613	75.125	70.213	43.859
R18	GLINT360K	68.230	80.575	75.852	47.831
R34	MS1MV3	71.644	83.291	80.084	53.712
R34	GLINT360K	79.907	88.620	86.815	60.604
R50	MS1MV3	75.488	86.115	84.305	57.352
R50	GLINT360K	85.272	91.617	90.541	66.813
R100	MS1MV3	81.083	89.040	88.082	62.193
R100	GLINT360K	89.488	94.285	93.434	72.528

Cuadro 3.1: Resumen modelos ResNet utilizados para el trabajo

CAPÍTULO



# 4

## IMPLEMENTACIÓN

A continuación se detalla todo el proceso seguido hasta llegar a la parte de la experimentación. Primero se detalla cómo funciona la técnica LIME para obtener explicaciones. Se sigue por un proceso de normalización, revisión de landmarks, finalizando con la creación de los mapas de calor y dendrogramas, que muestran la comparación de las características más importantes para el reconocimiento facial.

### 4.1 Planteamiento LIME

Los modelos de aprendizaje automático siguen siendo en su mayoría procesos internos no observables y presentan dificultades para ser medidos[63]. LIME nace de la necesidad de entender el razonamiento detrás de las predicciones, es importante saber si los resultados atienden a una lógica para así confiar en los modelos.

Es una técnica de explicabilidad novedosa que apareció por primera vez en 2016 y cuya intención es esclarecer las predicciones de cualquier clasificador. La idea es construir un modelo local simple y fácil de interpretar sobre predicciones particulares. A partir de diversas muestras representativas, se extrapolan los resultados a un plano más general, dando así una explicación global del modelo [64].

Comprender el funcionamiento interno de las redes neuronales es necesario, ya que en algunas ocasiones se pueden producir ciertos errores que conducen a una mala evaluación de los resultados. Un caso es la fuga de datos, que ocurre en la fase de entrenamiento y donde se filtra involuntariamente información que no toca de los datos de validación al conjunto de entrenamiento. Esto puede aumentar de forma imperceptible la precisión del modelo o producir sobreajuste.

Otro problema difícil de detectar es el cambio en el conjunto de datos, donde cambia la distribución entre el conjunto de entrenamiento y el de prueba. Esto puede llevar a problemas en el rendimiento, la precisión y la generalización del modelo.

Analizar si el funcionamiento interno de los modelos es coherente ayuda a anticiparse a estos posibles problemas. Y además aumenta la confiabilidad de las predicciones,

#### 4. IMPLEMENTACIÓN

---

ya que se muestran los motivos en los que se basa el razonamiento del modelo.

En particular LIME cuenta con varios criterios que se deben cumplir:

- Debe ser **interpretable**. Para las explicaciones es un criterio esencial que haya una comprensión cualitativa entre las variables de entrada y respuesta. Es decir, se pueden observar y valorar sin necesidad de una representación matemática.
- Debe cumplir el principio de **fidelidad local**. Significa que el modelo se comporta de forma coherente con la muestra particular en la vecindad próxima a esta. Fidelidad local no implica fidelidad global: las características que son globalmente importantes pueden no serlo en el contexto local, y viceversa. Si bien la fidelidad global implicaría fidelidad local, identificar explicaciones fieles a nivel global que sean interpretables sigue siendo una tarea de mayor nivel de complejidad.
- Debe ser **agnóstico** al modelo. Separa la explicación del tipo de modelo de aprendizaje automático. Se debe poder explicar cualquier modelo y tratándolo así como una caja negra. No se tiene información de antemano de cómo son los procesos internos.
- Debe haber una perspectiva **global** del modelo. Además de explicar las predicciones individuales, es importante proporcionar una perspectiva global para poder determinar la confianza en el modelo. Sobre la base de las explicaciones de las predicciones individuales, se seleccionan aquellas que son consideradas representativas. Aunque una explicación de una sola predicción proporciona una cierta comprensión de la fiabilidad del clasificador para el usuario, no es suficiente para evaluar la confianza en el modelo en su conjunto. Es complicado estudiar el modelo como un todo debido a la gran cantidad de muestras individuales que habría que procesar. Sin embargo, se pueden tomar cierto número, donde se cumpla fidelidad local y la explicación sea significativa.

Por lo general, los espacios de características con los que los modelos de aprendizaje profundo hacen frente, presentan límites no lineales y de alta complejidad.

Para dar una explicación a una muestra particular, lo primero que se hace es generar varias muestras perturbadas a su alrededor. Estas muestras presentan ligeras variaciones respecto a la original, por lo que todas se ubican en un espacio dimensional muy cercano.

A continuación, se utiliza una función kernel que asigne pesos al subconjunto de muestras perturbadas, según la distancia respecto a la original. De esta manera, se pueden saber qué muestras son más similares a la instancia inicial.

El siguiente paso es entrenar un modelo simple. Este genera un resultado para el subconjunto de datos particular, dando una explicación interpretable, aunque muy específica. Entre las opciones del modelo local se encuentran los de regresión lineal, árboles de decisión o aquellos basados en reglas.

Finalmente, se analizan los coeficientes producidos por el modelo, mostrando aquellos factores que contribuyen más en la predicción. Por lo tanto, se obtienen las características más importantes que definen la instancia original[65].

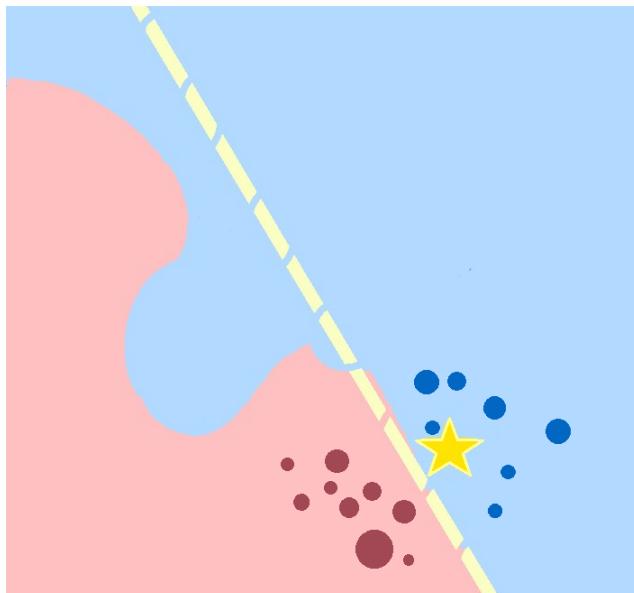


Figura 4.1: En este ejemplo, LIME busca dar una explicación de por qué la estrella amarilla se enmarca en la clase celeste en lugar de la rosada. Los puntos de datos que residen dentro de la curva rosada se clasifican como A y los de la curva celeste, como B. El objetivo es generar una explicación que prediga la clase de la estrella amarilla en la imagen de arriba[66]. En lugar de observar el comportamiento global, que sería dar una explicación a la curva, LIME explora el área cercana al punto de la estrella amarilla. De modo que pueda resolverse con un enfoque muy local, permitiendo que un clasificador lineal sea capaz de explicar la predicción[67].

Hay que tener en cuenta que con una sola muestra no se puede dar una explicación fidedigna de la realidad, puesto que estaríamos dando una explicación lineal para un problema de abstracción mucho mayor. Sin embargo, con la generación de explicaciones para un conjunto de datos suficientemente grande, se va construyendo poco a poco un esquema completo de la realidad.

En el contexto de la clasificación de imágenes, la porción de vecindad se genera a través de varias muestras con diferentes combinaciones de regiones ocluidas. Luego se analiza la importancia que tienen para la imagen original. De manera que si el parche tapaba una parte importante, esto afecta directamente a la predicción de la clase. Se utiliza un vector binario que indica la presencia o ausencia de esos parches en las nuevas imágenes distorsionadas. La representación original de la instancia explicada se denota como:  $x \in \mathbb{R}^d$ .

Se usa la anotación  $x' \in \{0, 1\}^{d'}$  para describir el vector binario de su representación interpretable.

El valor almacenado puede ser 1, indicando que el superpíxel original se mantiene y 0 si está tapado.

Para las predicciones de clasificación, se resaltan aquellos superpíxeles con un peso positivo y que dan una intuición de por qué el modelo pensaría que esa clase específica puede estar presente. También se pueden resaltar aquellos superpíxeles que podrían estar favoreciendo a una predicción contraria.

## 4. IMPLEMENTACIÓN

---

En el contexto de este proyecto se plantea un enfoque diferente, ya que se trabaja para dar explicaciones biométricas y no para entender una clasificación como tal. En el método original, el modelo de regresión lineal espera el vector de probabilidades obtenido al aplicar una función Softmax de la capa final del modelo. En vez de eso, se adapta LIME para poder utilizar los vectores de características de las imágenes. Se obtienen las similitudes del coseno entre el vector de características de la imagen original y los vectores de la vecindad generada, se calcula la medida de distancia y se le pasa al regresor lineal. Esto permite explicaciones pero sin limitarse al contexto de clasificación.

Aquellos parches que tapan regiones de la imagen más importantes, dan lugar a vectores de características más distintos que el original. De la misma manera ocurre con las probabilidades obtenidas de la función Softmax original. Si esto sucede, significa que esa región es importante para la identificación de la cara.

En consecuencia, se pueden comparar dos vectores mediante la distancia del coseno, la cual produce una puntuación que indica la similitud entre las imágenes. De este proceso se obtienen los parches más importantes de las regiones de la cara. Estos parches son las explicaciones que se quieren obtener.

### 4.2 Preparación modelos y bases de datos

Como ya se ha comentado en la sección anterior, se utilizan 8 modelos ResNet de diferente profundidad y preentrenadas con diferentes bases de datos. A modo de recordatorio, estas son ResNet18, ResNet34, ResNet50 y ResNet100. Las bases de datos son MS1MV3 y GLINT360K.

Y por otro lado, la base de datos para la parte de la experimentación, es un subconjunto de VGGFace2. El equipo docente proporcionó un archivo con imágenes de 368 personas, desde tres perspectivas diferentes(frontal, perfil y tres cuartos), con 10 muestras para cada una. Adicionalmente se ha añadido una persona más.

Se ha decidido trabajar únicamente con la perspectiva frontal de los sujetos debido a una cuestión de diseño inherente al propio proceso de normalización. La necesidad de estandarizar las caras para un posterior análisis de las regiones en un mapa de calor, excluyó las otras posturas.

### 4.3 Explanaciones LIME

La primera parte de este proyecto consiste en la aplicación del método LIME al subconjunto de imágenes de la base de datos VGGFace2 más una persona adicional.

#### 4.3.1 Preparación formato imágenes

La primera tarea es la carga de las imágenes desde memoria. Básicamente se obtiene la ruta completa del directorio donde se almacena. Se pasa a un modelo de color RGB y se le aplica una transformación que ajustará el tamaño a 112 x 112; las dimensiones esperadas por la entrada de todas las ResNet. Más tarde se transforma en tensor y se normalizan los valores para que el rango sea [-1, 1], que es el esperado por la entrada del modelo.

### 4.3.2 Normalización vectores

A continuación, se crea una sesión de inferencia con ONNXRuntime. Esta sesión establece una interfaz que permitirá correr modelos en tiempo de ejecución. Crear una sesión conlleva gran coste computacional, por lo que solo se realiza una vez para cada modelo. Con esto preparado, se pueden procesar todas las imágenes. Tanto las originales como las perturbadas, con el fin de obtener los vectores de características asociados.

En las redes neuronales, las últimas capas están totalmente conectadas. En la penúltima capa se obtiene un vector de características resultado de procesar la imagen a través de todas las etapas de convolución, pooling y max pooling que componen la arquitectura ResNet. Este vector contiene toda aquella información que se haya catalogado como importante para la descripción de la imagen. La capa final mapea este vector de características y utiliza una función de activación(típicamente SoftMax) para obtener otro vector, esta vez con la distribución de probabilidades de clasificación de cada clase.

Pero con el nuevo enfoque LIME, solo interesa el resultado de la penúltima capa, el vector con el resumen de las características. En los modelos del presente proyecto, no se ha tenido que realizar ningún cambio a las redes. Ya están adaptadas para devolver el vector de características directamente.

Una vez obtenido dicho vector, se procede a aplicar una normalización de los datos. En este caso se utiliza la norma Euclídea. Este proceso de normalización es fundamental debido a la importancia de evitar posibles diferencias en las magnitudes, donde aquellas más grandes obtendrían una ventaja favorable. Y con el posterior cálculo de distancias, es preciso eliminar inconsistencias en la información.

Además, esta transformación mueve la información de los vectores al mismo plano hiperesferoidal. Ahora las propiedades del vector se pueden representar fácilmente por la dirección, el ángulo y la longitud. Esto facilita el cálculo de distancias entre vectores, pudiendo obtener así una medida relativa de similitud. En este caso, se calcula la similitud del coseno y se obtienen valores en el rango [-1, +1]. Esta métrica refleja la orientación de los vectores, independientemente de su magnitud.

Posteriormente se convierte en una medida de distancia, acotando los datos a un rango de [0, 1]. Si la distancia del coseno está cerca de 0, entonces los vectores tienen orientaciones similares y están próximos uno del otro. Si es casi 1, entonces los vectores difieren; son ortogonales y no comparten relación.

Este paso es necesario porque LIME espera resultados provenientes de una función SoftMax. Es decir, trabaja con probabilidades que en total suman 1. Por lo tanto los datos deben ser acomodados al formato adecuado. El cálculo de la norma se puede realizar mediante la raíz cuadrada de la suma de todos los componentes al cuadrado.

La notación matemática es la siguiente:

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2 + \dots + v_n^2}$$

Finalmente, todos los elementos del vector de características se dividen por el valor obtenido tras aplicar dicha fórmula.

#### 4.3.3 Aplicar LIME

Una vez preparadas las imágenes en el formato adecuado, se aplica el método LIME para al reconocimiento facial. Los pasos se pueden apreciar en el diagrama 1.1.

Primero, se generan datos de vecindad perturbando algunas de las características de la instancia original. Básicamente se generan 1000 imágenes donde se crearán combinaciones aleatorias de regiones ocluidas.

La idea principal es comparar el vector de características de la instancia original con los vectores que se obtendrán después de generar esa muestra de vecindad. Se obtiene un valor de distancia entre esos vectores. Según si el valor es más o menos lejano, se vislumbra qué zonas ocluidas son más importantes para reconocer la cara. Si el valor difiere más, significa que esas regiones suponen un impacto mayor[68].

Para la construcción de estas imágenes alteradas, primero se aplica un algoritmo de segmentación a la imagen original. Hay dos posibles algoritmos.

- **QuickShift.** Hace un mapeo de las características según la información de color y la posición de los píxeles. Se utiliza la técnica de vecinos más cercanos para fusionar los píxeles, generando así las regiones.
- La otra opción es **SLIC**, cuyo enfoque se basa en una adaptación del método K-Medias. Se crean clusters a medida que avanzan las iteraciones del algoritmo, hasta conseguir una segmentación compacta y homogénea[69].

Los segmentos de SLIC son más uniformes, mientras que en QuickShift generan formas más irregulares. De manera predeterminada se opta por el algoritmo SLIC, a no ser que se indique lo contrario.

Así que el primer paso es establecer estas regiones, formadas por agrupaciones de píxeles. Se crea una matriz con las mismas dimensiones que la imagen original y cada espacio representa la región a la que pertenece cada píxel.

Más tarde se genera una matriz densa conformada por las 1000 imágenes perturbadas, con la combinación particular de regiones ocluidas para cada una. Los valores de estas regiones es binario, indicando si dicha región está ocluida con un 1. Si es visible, se representará con un 0.

También se obtiene un vector que almacena la distancia extraída de comparar la imagen alterada respecto a la instancia original.

Con estos datos se realiza una selección de características que se proporcionan más tarde a un modelo simple. En particular se utiliza Ridge; modelo de regresión lineal, donde la función de pérdida se modifica para considerar una mayor simplicidad[70].

Después de ajustar este modelo, se obtiene el coeficiente de importancia para cada característica de la imagen. Esto en base a si la región ocluida produce un cambio significativo en la métrica de distancia. A modo de recordatorio, esta distancia se obtiene de comparar los vectores de características entre dos imágenes. Si se obtiene un valor mayor, quiere decir que esas regiones presentan información de la cara importante. Los coeficientes se ordenan y se obtienen las características más significativas, que suponen una mayor importancia a la hora de reconocer a la persona.

Finalmente, se obtienen varios archivos de las explicaciones. Uno es un mapa de color RGB, con los superpíxeles identificados como importantes superpuestos en el

#### 4.4. Normalización imágenes

individuo y así poder ver claramente a que región de la cara hacen referencia. Aquellos con mayor saturación son los que suponen un mayor impacto para el reconocimiento de la persona 1.1 . También se obtienen las máscaras en formato color gris. Por último, también se guarda el mapa de regiones de los segmentos de la imagen.

Con lo que realmente se trabaja es con las máscaras. Los otros archivos permiten la interpretación de los datos de una forma más natural para los seres humanos.

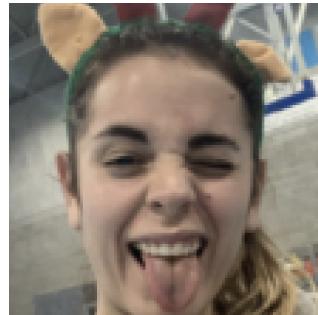


Figura 4.2: Ejemplo de una muestra de una persona que constituirá la imagen original con la que se compararán los vectores de las imágenes perturbadas y así obtener la medida de distancia.

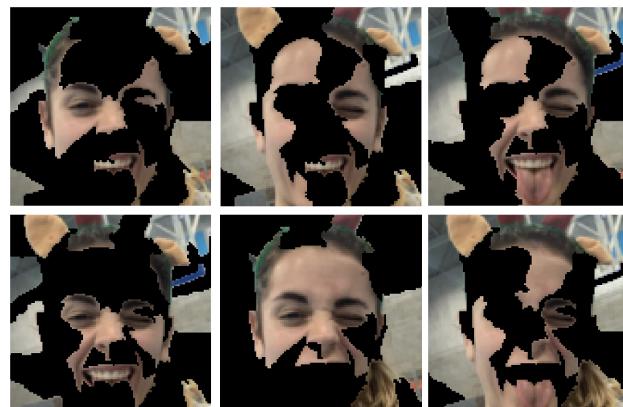


Figura 4.3: Conjunto de muestra de 6 imágenes perturbadas, donde se han ocluido algunas regiones de forma aleatoria.

#### 4.4 Normalización imágenes

Una vez obtenidas las explicaciones LIME, el siguiente paso es la normalización para alinear y estandarizar las caras. Se busca reducir la variabilidad en las poses y en las expresiones faciales, de manera que todas las imágenes mantengan una misma orientación y alineación. De esta forma se consigue representar la importancia de los superpíxeles de forma coherente para todos los individuos, creando un conjunto de datos que permita la comparación de forma más precisa y consistente.

La normalización estandariza la información geométrica de una imagen aplicando efectos de filtrados en la ubicación, escala y rotación. En el mundo real, las perso-

## 4. IMPLEMENTACIÓN

---

nas aparecen con distintas poses, expresiones o caras, por lo que es importante una normalización que permita un modo de medir todas estas características sin atribuir ningún tipo de peso adicional a ningún rasgo facial particular. Se aplican varios tipos de operaciones matemáticas para obtener dichas imágenes que servirán para que los análisis sean más robustos.

Es importante definir los siguientes conceptos:

- **Landmark.** Son puntos de referencia específicos en una imagen que se utilizan para identificar características anatómicas o puntos de interés.
- **Triangularización de Delaunay.** La triangulación de Delaunay es una técnica que se utiliza para dividir un conjunto de puntos en una red de triángulos conexos de un espacio bidimensional. Estos triángulos deben cumplir la condición de su regla homónima. Esta dice que la circunferencia circunscrita, la que se dibuja en el interior del triángulo, no debe contener ningún vértice de otro triángulo. Aunque sí que se pueden situar sobre el perímetro de la circunferencia [71].

### 4.4.1 Detección facial

El primer paso para la normalización es aplicar un detector de caras que focalice la persona e ignore objetos que puedan suponer una distracción a la hora de aplicar landmarks. En este caso se utiliza el detector de caras *Haarcascade* de la librería de OpenCV de Python.

Este algoritmo se basa en detectar características simples que representan propiedades en una imagen, como bordes, líneas y cambios de intensidad. El algoritmo consiste en unas ventanas rectangulares que se desplazan por la imagen. Cada característica calcula la diferencia de intensidad entre las regiones de dentro y fuera de la ventana, lo que permite capturar ciertas propiedades. Consta de diferentes etapas cuya estructura es similar a la de una cascada. En cada etapa se aplican clasificadores *AdaBoost*, que consisten en crear varios predictores sencillos en secuencia, de tal manera que iterativamente se ajustan las características que no se adaptaron bien en el anterior ciclo. A medida que se avanza en la profundidad se obtienen resultados de mayor precisión[72]. Básicamente se acota el encuadramiento de la cara, obteniendo un rectángulo en el que posteriormente se aplicarán los landmarks.

### 4.4.2 Identificación landmarks

La identificación exacta de puntos de referencia dentro de la cara es el segundo paso para normalizar. Se aplican detectores de referencias faciales que colocan los rasgos en una plantilla uniforme e idéntica para todos los demás rostros.

Se utiliza un enfoque que involucra un descriptor que detecta 68 landmarks faciales. Se adapta a cada cara distinta, centrándose en los puntos más representativos: boca, ceja derecha e izquierda, ojo derecho e izquierdo, nariz y mandíbula. A parte de generar un perímetro de puntos alrededor del borde exterior de la cara. Este descriptor face detector forma parte del conjunto de datos iBUG 300-W de 68 puntos[73].

Se ha escogido el descriptor mencionado debido a que es capaz de encontrar y adaptar imágenes con variaciones de diferente índole. Eso sí, partiendo de una base donde

las formas son perceptivamente similares, pero manteniendo la capacidad de adaptarse a caras que aparezcan giradas, trasladadas, escaladas o con cualquier otro tipo de transformación de naturaleza semejante. El descriptor es capaz de encontrar formas afectadas por el ruido, levemente distorsionadas o defectuosas en diferentes situaciones, que de manera natural también serían reconocidas por los seres humanos[74].

La forma de trabajar es haciendo coincidir una plantilla de puntos con una imagen de entrada con restricciones de suavidad en el campo de deformación. Los puntos de referencia se detectan automáticamente en función de las propiedades geométricas de la superficie, como la curvatura y la intensidad de la imagen.

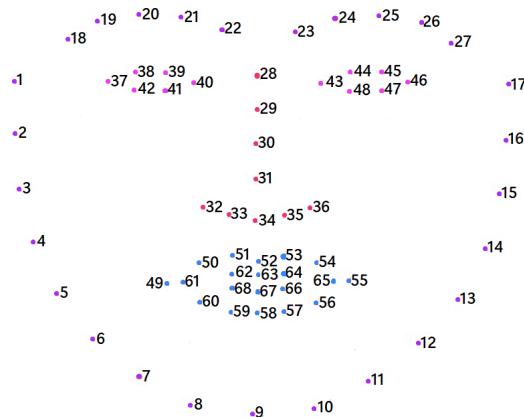


Figura 4.4: Conjunto de 68 puntos de referencia que se utiliza como plantilla para la detección de los rasgos faciales de mayor relevancia para el reconocimiento facial. El descriptor es el conjunto de valores numéricos que referencian cada punto de la cara.

#### 4.4.3 Triangularización de landmarks

El último paso es la creación de una malla formada por triángulos que cumplan la propiedad de Delaunay y que conecte los puntos de referencia de la cara. Los landmarks actúan como puntos de control, son los conectores de los vértices y es alrededor de estos donde se construyen los triángulos.

Se dibujan los puntos rojos y las conexiones azules, para formar los triángulos en la imagen. Esto ayuda a visualizar tanto la estructura triangular, como los landmarks en la imagen. Es útil para la comprobación de que el encaje de las mallas con los rostros es el adecuado. Así se puede asegurar perceptualmente que su aplicación ha sido satisfactoria.

Se hizo una comprobación manual de todas las imágenes para asegurar que las mallas fueron colocadas correctamente. En caso de que las mallas no se adapten bien, la representación de los rasgos faciales no sería equitativa. Después de la revisión, se tuvieron que descartar algunas imágenes, esto sencillamente significa que la representación para algunos individuos ha sido menor.

Finalmente se transforman las imágenes con la alteración definida por los puntos de referencia y la maya de triángulos. Cada píxel en la imagen final se calcula utilizando interpolación baricéntrica; copia píxeles de la imagen original a una imagen destino. La misma transformación aplica a las máscaras generadas por LIME.

#### 4. IMPLEMENTACIÓN

Todo el proceso conjunto de detección facial, detección de landmarks y cálculo de la triangularización, ayuda a crear una estructura en las imágenes que servirá de referencia estándar para la posterior construcción de mapas de calor. En el diagrama esto formaría parte de la fase de normalización 1.1.

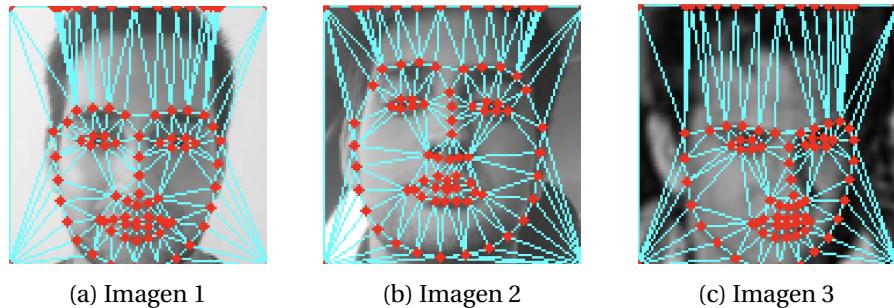


Figura 4.5: Landmarks aplicados a la cara de un sujeto, los puntos se adaptan a la cara y se identifican los rasgos faciales como ojos, nariz, cejas, boca y el contorno de la cara.

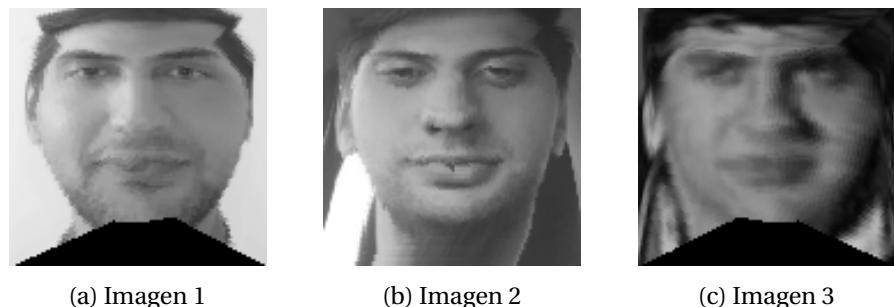


Figura 4.6: Resultado final de normalizar las caras para que todos los puntos faciales se encuentren estandarizados y enmarcados en una plantilla equivalente a todas las caras.,

### 4.5 Creación mapas de calor

Un mapa de calor es una representación que expone relaciones entre varias características a través de un mapa bidimensional de colores [75]. Se muestra la distribución de los valores, a través de la intensidad, tono, saturación o luminancia en esa zona concreta. Se manifiestan detalles visuales que facilitan el entendimiento. En resumen, resalta tendencias de una manera más sencilla de procesar para el cerebro humano, debido a que las imágenes siempre son más fáciles de interiorizar que cualquier tipo de datos o números[76].

Los mapas de calor son muy útiles ya que facilitan la comprensión de gran cantidad de datos, de manera que quedan resumidos en patrones que muestran relaciones, variaciones o perturbaciones. Solo se utilizan datos numéricos en las trazas de ambos ejes en la cuadrícula.

En el ámbito de este trabajo, se procesa la distribución de las máscaras obtenidas del conjunto de la base de datos. Habrá un eje que representará la altura y otro la

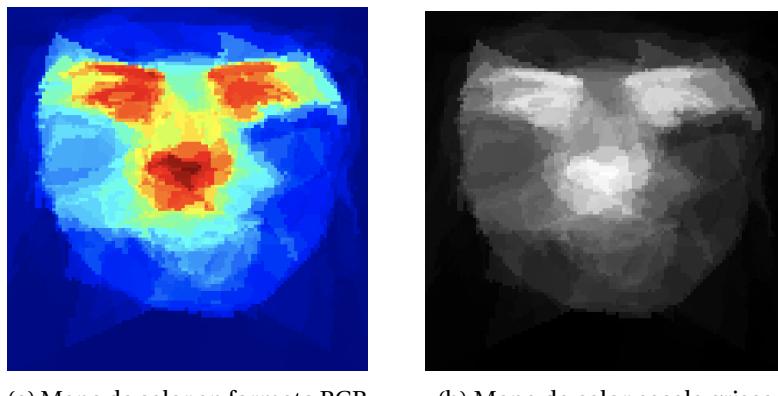
anchura. La naturaleza matricial del mapa de calor es equivalente a la de una imagen. En este caso, se asigna un rango de colores donde la paleta va desde tonos fríos para valores de baja intensidad, hasta tonos cálidos para los valores con mayor acumulación. Básicamente, aquellas regiones donde las explicaciones LIME suponen una mayor relevancia para el reconocimiento de la persona, se han coloreado más intensamente. Eso después se ve reflejado en los mapas de calor. Los pasos seguidos se muestran en este diagrama 1.1 y el procedimiento es el siguiente:

Primero se acumulan los valores por individuo de cada máscara normalizada, adquiridas del flujo de trabajo anterior. Se realiza una conversión a escala de grises que elimine la información de color; no supone ninguna ventaja conservar los detalles específicos, mientras que trabajar con un solo canal simplifica los cálculos. El hecho de mantener los datos en una sola escala, incide en un mayor enfoque en la intensidad de los valores obtenidos por la máscara. Esto mejora la interpretación visual de los resultados, pues resalta más el contraste.

El siguiente paso es la normalización de los píxeles para que estén ajustados al rango  $[0, 1]$ . Las imágenes suelen expresarse con valores que van del 0 a 255, sin embargo en algunos casos es útil realizar esta conversión. En esta coyuntura, tener los datos en un rango más acotado permite una mayor consistencia en la interpretación y en los análisis de resultados. Las imágenes pueden ser comparadas más fácilmente.

Los mapas de calor se crean acumulando los valores de los píxeles de cada sujeto, según una temática en particular. Se calcula la media de estos valores, dividiendo por la cantidad de sujetos implicados.

Finalmente, después de haber acumulado los valores en una matriz, se realizan algunos ajustes y esta se procesa para crear mapas de calor, tanto en escala de grises como en formato a color.



(a) Mapa de calor en formato RGB      (b) Mapa de calor escala grises

Figura 4.7: Ejemplo de los rasgos faciales más representativos después de aplicar las explicaciones de las 8 redes neuronales para un mismo sujeto.

## 4.6 Creación dendrogramas

Para la comparación de los mapas de calor se ha optado por la representación mediante dendrogramas.

#### 4. IMPLEMENTACIÓN

---

Esta es la fase final del flujo de acciones del proyecto, justo después de crear los mapas de calor según temática 1.1.

Para crear estos mapas se aplica un agrupamiento jerárquico en los datos. Este diagrama presenta aspecto de árbol, donde cada rama simboliza los grupos y el color la cercanía que estos muestran entre sí [77].

Son útiles para representar mapas de calor ya que se pueden visualizar cuáles son más similares entre sí en términos numéricos y en carácter de color. Comparan múltiples características y ayudan a identificar las relaciones subyacentes, resumiendo la información y permitiendo una mejor interpretación global.

Para la creación de los dendrogramas, se calcula la distancias entre mapas, en particular se adapta la divergencia de KL. El resultado es una matriz que define todas las relaciones.

Cabe mencionar que la métrica que calcula las relaciones entre mapas es distinta a la que calcula los agrupamientos.

El algoritmo va creando poco a poco una estructura de enlaces. Se combinan entre pares hasta que finalmente se fusionan todas las clases en un último grupo, que se convierte en la raíz del diagrama. Para crear los grupos, existen las siguientes opciones:

- **Single.** Se hacen los grupos según la menor distancia. También conocido como algoritmo del punto más cercano.
- **Complete.** Se calcula la mayor distancia entre grupos. También conocido como algoritmo del punto más lejano o algoritmo de Voor Hees.
- **Average.** Se calcula la media aritmética de los pares de grupos, sin ponderar. Asume una tasa de evolución constante. También conocido como algoritmo Unweighted Pair Group Method with Arithmetic mean (UPGMA).
- **Weighted.** Se calcula igual que la variante UPGMA, pero esta vez las distancias sí se ponderan. También se conoce como algoritmo Weighted Pair Group Method with Arithmetic Mean (WPGMA).
- **Centroid.** Se calcula a través de la distancia entre los centroides de cada grupo.
- **Median.** Igual que el anterior, pero en cada iteración se recalculan los centroides de las nuevas aglomeraciones.
- **Ward.** Se encuentra la media de cada grupo, se calculan los cuadrados de esas distancias y se suman. Finalmente se suma todo. Se consideran todas las combinaciones posibles y eligen los nuevos grupos minimizando la varianza[78].

En el presente proyecto se utiliza el método Ward ya que en investigaciones anteriores ha dado buenos resultados.

Finalmente, se muestran los enlaces creados anteriormente, en un dendrograma. Los grupos se ilustran formando vínculos en forma de U. Las dos patas indican las dos etiquetas que forman la unión, la parte superior simboliza la fusión resultante. La longitud de las dos patas es la distancia cofenética de las observaciones originales. La altura del vínculo en el que esas dos observaciones se unen por primera vez representa dicha altura.

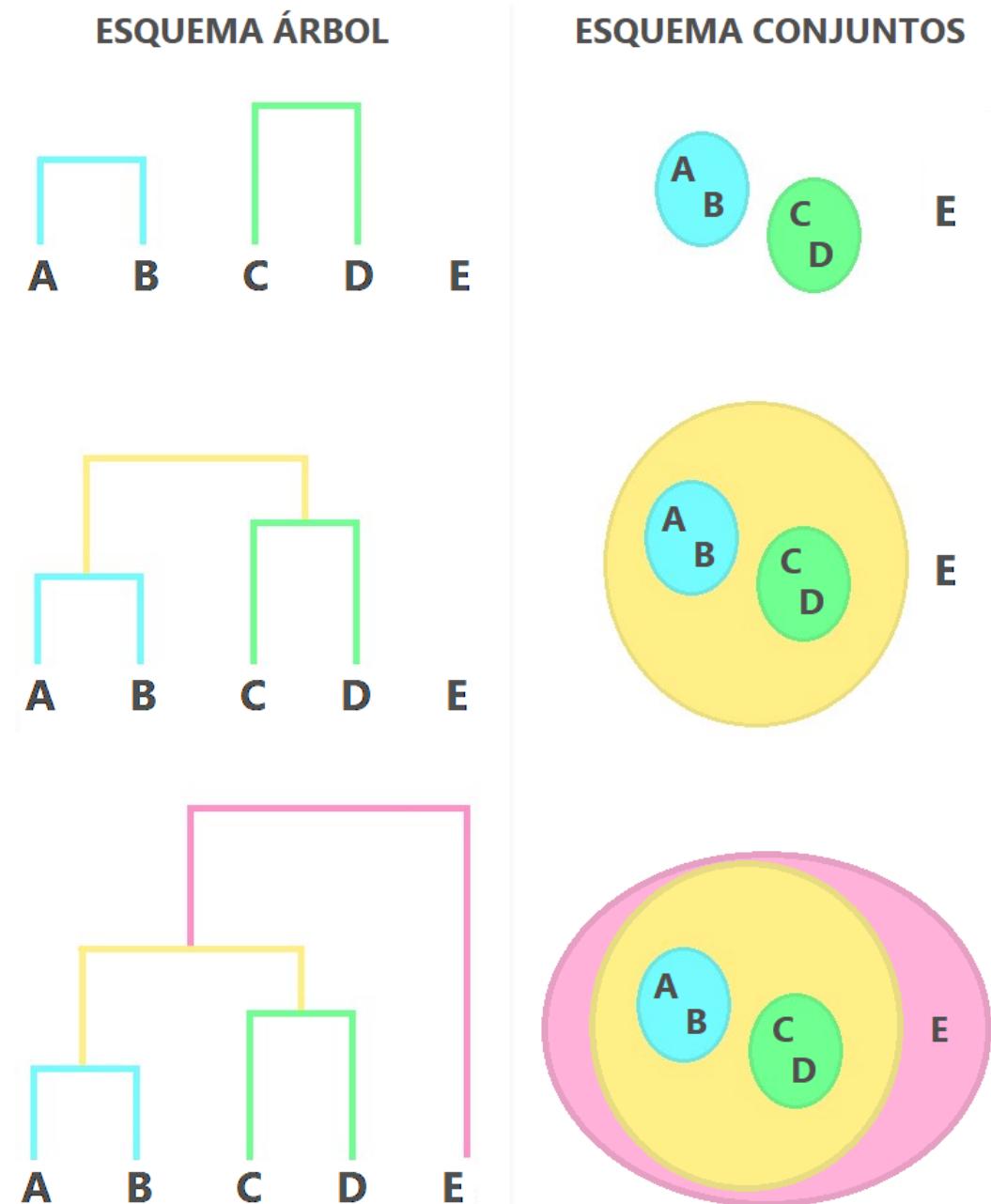


Figura 4.8: Así se crean los grupos para formar un dendrograma. A la izquierda, se muestra el dibujo del esquema en forma de árbol. A la derecha, la representación de cómo quedan los conjuntos. Se puede notar que a medida que se avanza en las iteraciones, los primeros conjuntos se convierten en subconjuntos de otros más grandes.



## EXPERIMENTACIÓN

En este capítulo de experimentación se realiza una colección de comparativas con su posterior análisis de los resultados.

Se generan diferentes mapas de calor para poder realizar mediciones entre ellos mediante el uso de una medida para así poder observar similitudes y diferencias a través de los diversos enfoques planteados. Los mapas de calor generados son los siguientes:

- **Según base de datos.** Se trabaja con redes neuronales preentrenadas con diferentes bases de datos. Como ya se mencionó anteriormente, estas son Glint360K y MS1MV3. Por lo que se realizan explicaciones repitiendo arquitectura ResNet. Se analiza si la base de datos en el preentreno inicial influye en la predicción, ya que una es bastante más profunda que la otra.
- **Según la red.** La idea es comparar entre todas las redes con las que se ha trabajado. Aquí se ven las medias de todas aquellas regiones de la cara que cada red ha considerado más relevante para reconocer a cada persona. Se comprueban las diferencias entre arquitectura y base de datos.
- **Según la persona.** Se ha hecho la media de las características en las que se fija cada red, por persona. Se puede entender como el resumen de aquellos rasgos que hacen más destacable a cada individuo. Se observa en qué se fija según el individuo.
- **Según la persona en cada red.** Este es el caso anterior particularizado por red. Se puede comparar los resultados de un individuo por cada red y ver si son consistentes a la hora de identificar a un mismo individuo.
- **Según la etnia.** Se revisa si por etnia hay rasgos más característicos para el reconocimiento facial. Se han distinguido tres clases: caucásicos, asiáticos y africanos.

## 5. EXPERIMENTACIÓN

---

- **Según la etnia por cada red.** Misma idea que el anterior enfoque, pero particularizado por tipo de red.
- **Según el sexo de las personas.** También es pertinente analizar si existen diferencias significativas entre los rasgos de las mujeres y de los hombres.
- **Según el sexo por cada red.** De la misma manera que con las etnias, para ver si cada red atiende a diferentes patrones según el género del individuo.
- **Según la arquitectura de la ResNet.** Se estudia si la profundidad de la red afecta directamente en el reconocimiento. Qué diferencias hay, si se fijan en unos rasgos más que otros, etc.
- **Total.** Finalmente, un único mapa de calor que resume las características de todas las redes neuronales juntas.

Para realizar el estudio estadístico se utiliza una medida basa en la divergencia de Kullback-Leibler.

### 5.1 Divergencia de Kullback-Leibler

La Divergencia de Kullback-Leibler es una métrica de distancia que mide la pérdida de información entre dos distribuciones probabilísticas. Se utiliza en el ámbito de la teoría de la información, la estadística, aprendizaje automático e inferencia.

KL sirve para ver cuánto se parecen dos funciones de distribución, donde penalizan más las grandes distancias. La medida de la distancia euclídea o el valor absoluto de la diferencia no tienen un significado probabilístico, así que quedan descartadas en el ámbito de este proyecto.

Esta medida tiene sus raíces en la hipótesis de los datos, cuyo objetivo esencial es evaluar cómo los datos están distribuidos en la información a través de la entropía, que es la medida más significativa.

La notación de KL sería la siguiente:  $KL(P \parallel Q)$ , donde P y Q son dos distribuciones de probabilidad. Esto se entendería como la divergencia de P de Q.

La fórmula es la siguiente:

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right)$$

Hay varias cosas acerca de esta métrica que deben ser tomadas en cuenta:

- Dos distribuciones son idénticas si la divergencia KL da como resultado 0.
- No es una medida simétrica, por lo que es bastante probable que si se calcula la distribución de Q a P y de P a Q, se obtenga valores diferentes.
- Si la información se mide en bits, los logaritmos de la fórmula son base 2. Si la información se mide en Unidad natural de información (Nats), se toma la base e.

- No satisface la desigualdad triangular. Es una de las cuatro características propias de una métrica de distancia[79].

Esta medida no cumple las condiciones necesarias para ser una función distancia: no es continua y no cumple el principio de similitud. Así que se utiliza una función distancia basada en KL, cuya idea principal consiste en calcular un punto R que será la media entre dos vectores:

$$R = \frac{1}{2}(P + Q)$$

Resultando en R como medida de probabilidad. P y Q son absolutamente continuos con respecto a R. Se puede considerar una distancia entre P y Q, todavía basada en la divergencia KL, pero usando R como intermediario[80]. Quedaría así:

$$\eta(P, Q) = \kappa(P|R) + \kappa(Q|R)$$

Esta función cumple las cuatro propiedades de una métrica de distancia:

- **Reflexividad.** Si dos puntos son iguales la distancia entre ellos es cero y si la distancia entre dos puntos es cero, son el mismo punto[81]. Notación:

$$d(x, y) = 0 \Leftrightarrow x = y$$

- **No negatividad.** Cualquier distancia entre dos puntos no puede ser un valor negativo[81]. Notación:

$$d(x, y) \geq 0$$

- **Simetría.** La distancia entre dos puntos es la misma sin importar el origen de medición[81]. Notación:

$$d(x, y) = d(y, x)$$

- **Desigualdad triangular.** La suma de las distancias entre dos rectas trazadas desde tres puntos no puede ser mayor que la tercera recta[81]. Notación:

$$d(x, y) + d(x, z) \geq d(y, z)$$

Con la medida de similitud y los mapas de calor, ya se puede proceder a la experimentación.

## 5. EXPERIMENTACIÓN

### 5.2 Experimentos

A continuación, se muestran los diferentes experimentos realizados y la discusión de los resultados obtenidos. Se han generado gráficos en forma de árbol para una mejor visualización de los agrupamientos, a través de aplicar la distancia basada en KL.

#### 5.2.1 Diferencias entre bases de datos

Las bases de datos de preentrenamiento inciden directamente en el desempeño posterior de las redes neuronales. A lo largo de los últimos años se ha comprobado que esta fase es fundamental, ya que impacta directamente en la aparición de posibles sesgos. Para tratar de paliar estos problemas, se le da más importancia a la variedad de caras, poses, escenas y etnias. Pueden aparecer diferencias al trabajar con redes de diferente potencia.

En el dendrograma se observa que los mapas de calor más cercanos son GLINT y TOTAL. Difieren mucho más de MS1MV3, que se agrupará en la siguiente iteración a una distancia mucho mayor.

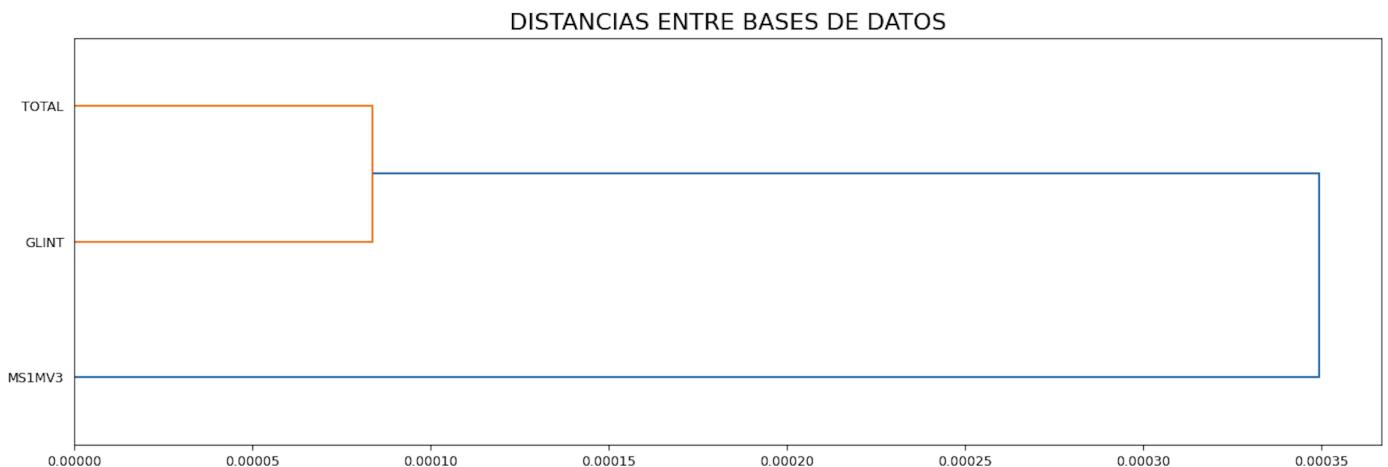
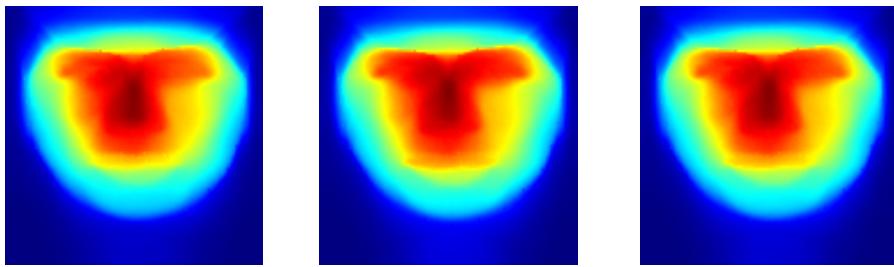


Figura 5.1: Dendrograma obtenido de aplicar la distancia de Kullback-Leibler a los mapas de calor de las bases de datos. TOTAL y GLINT forman un grupo. La siguiente iteración formará otro grupo con la adición de MS1MV3.

En cuanto a los mapas de calor, en general se repara una gran fijación en la parte de la nariz. MS1MV3 concede algo más de relevancia en la parte de los ojos, mostrando una sutil mayor saturación en ese rasgo, también se extiende más por la zona de la mandíbula y en general, la zona de reconocimiento es más amplia. Parece que GLINT acota algo más sobre la parte central de la cara.

Hay que tener en cuenta que la base de datos GLINT360K es mucho más extensa que MS1MV3. Mientras que la primera cuenta con 17 millones de imágenes, la segunda tan solo cuenta con 5,1 millones. Esto implica que las imágenes que forman parte de GLINT son mucho más diversas, se incluye mucha más variabilidad en edad, etnias, poses, situaciones, iluminación, entre otros. Además, las imágenes en MS1MV3 proceden de la recopilación en la web. La calidad puede variar bastante y en consecuencia ser algo peor en general. No están controladas al mismo nivel que en GLINT.



(a) Mapa de calor total de todas las arquitecturas ResNet preentrenadas con la base de datos GLINT.

(b) Mapa de calor de todas las arquitecturas ResNet preentrenadas con la base de datos MS1MV3.

(c) Mapa de calor total de todas las personas para todas las redes. Es la fusión de ambas bases de datos.

Figura 5.2: Resumen de los mapas de calor comparados desde el enfoque de las bases de datos. También se compara con el mapa de calor total para ver las diferencias.

De todos modos los mapas de calor son bastante similares entre sí. No se aprecian diferencias exageradas. Aunque parece que las redes neuronales entrenadas con GLINT son algo más robustas para el reconocimiento facial. Es razonable pensar que la media total es más cercana a la red preentrenada con la base de datos más potente, pues está más capacitada para interpretar mejor las caras humanas. Este mayor entrenamiento parece afectar en el área de reconocimiento de caras, sin necesitar resaltar tanto algunas facciones.

### 5.2.2 Diferencias entre todos los modelos

Una comparación que resulta casi intuitiva es el desempeño de cada red neuronal por separado. Se presentan 4 arquitecturas diferentes, con dos bases de datos para cada una. Así se puede observar si hay mucha discrepancia según el tipo, si cada una da más importancia a según qué rasgos, entre otras cosas.

Se ha realizado un dendrograma con los mapas de calor obtenidos para cada ResNet. En total, entre las cuatro posibles arquitecturas y las dos posibles bases de datos de preentrenamiento, hay 8 etiquetas.

En el diagrama se dibujan tres agrupaciones distintas, marcadas con sus respectivos colores.

- **Grupo naranja.** Compuesto por ambas ResNet18 (R18), cada una con su base de datos. Tiene sentido que la misma arquitectura diste menos entre sí.
- **Grupo verde.** Aquí se encuentran las demás redes. Parece ser que a partir de las 34 capas, estas no presentan demasiadas diferencias entre sus predicciones. Se mezclan entre todas sin ningún patrón destacable. Cabe mencionar que ResNet50 (R50) con MS1MV3 es la más cercana al mapa de calor total. Es lógico que una arquitectura con un número de capas promedio, esté más cerca de la media total de todas las redes.
- **Grupo azul.** Finalmente, el último agrupamiento es la fusión de los anteriores, verde y naranja. Parece que los modelos R18 son los que más se diferencian de

## 5. EXPERIMENTACIÓN

---

todos los demás, pues sus mapas de calor dan un mayor enfoque en algunas zonas más singulares. En este caso, la excesiva baja profundidad, crea más diferencias a la hora de reconocer caras ya que no generaliza tanto las regiones. En las arquitecturas medianas parece que la importancia converge hacia unos rasgos más definidos por el centro de la cara.

Es razonable que entre arquitecturas cercanas las distancias sean menores. Cabría esperar que entre una R18 y la ResNet100 (R100) hubiese más distancia que entre las arquitecturas medias. La profundidad afecta directamente en la calidad de las predicciones.

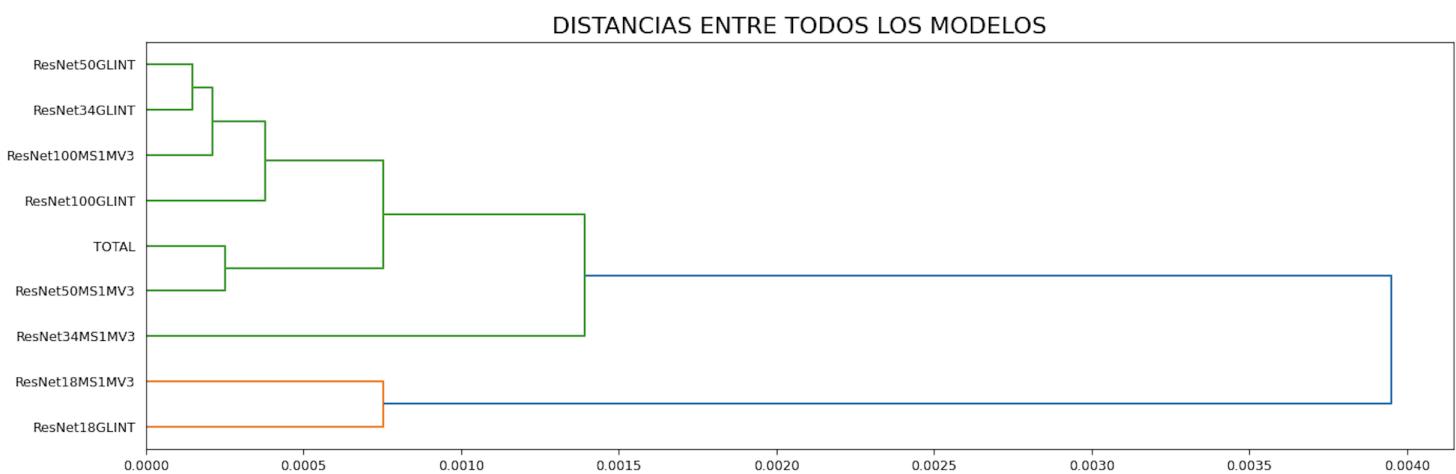


Figura 5.3: Dendrograma obtenido de aplicar la distancia de Kullback-Leibler a los mapas de calor de todos los modelos.

Por otro lado, en los mapas de calor se aprecian varias cosas.

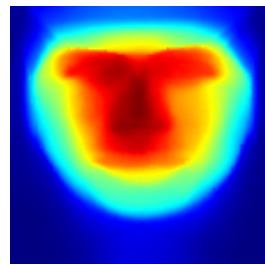
De forma general, parece que la nariz es una región bastante importante en casi todas las redes. Los ojos parecen no tener tanto peso. Se puede notar que a medida que la profundidad avanza, los modelos presentan mayor saturación en las zonas centrales, generalizando rasgos y focalizando menos la periferia. Las R100, dibujan áreas más reducidas que las que presentan menos neuronas.

Particularmente, destacan las R18, ya que ambas saturan diferentes zonas de la cara. MS1MV3, es la que más importancia da a los ojos. En cambio es curioso observar que con GLINT, prioriza mucho más la nariz.

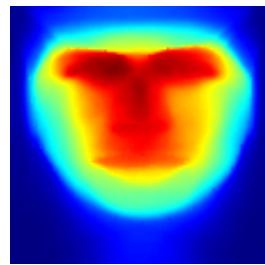
Todas las demás presentan rasgos bastante similares entre sí. La otra red que resalta debido a que es la que más se fija en la nariz y mandíbula, es ResNet34 (R34) entrenada con MS1MV3.

Y en cuanto a las bases de datos, se observa una menor variabilidad en aquellas redes que fueron entrenadas con GLINT. Presentan cierta consistencia y parecido, independientemente de la profundidad. De forma anecdótica, la que ofrece mayor distinción es R18. Es una base de datos mucho más extensa y potente, por lo que generaliza más los patrones que tiene en cuenta de la cara.

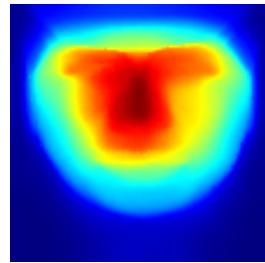
Las redes entrenadas con MS1MV3 difieren más entre ellas. La R18 se fija más en los ojos, la R34 en la nariz, R50 la zona del entrecejo y R100 prácticamente solo la nariz, pero de una forma mucho más sutil que las demás.



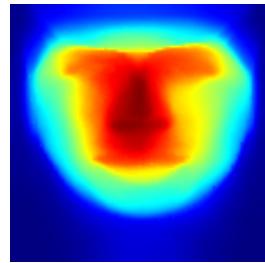
(a) Mapa de calor total de la ResNet18 entrenada con GLINT.



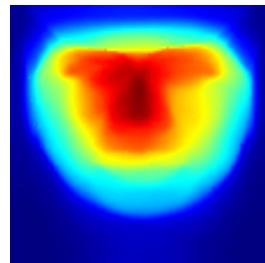
(b) Mapa de calor total de la ResNet18 entrenada con MS1MV3.



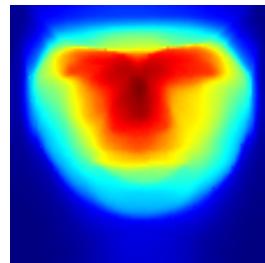
(c) Mapa de calor total de la ResNet34 entrenada con GLINT.



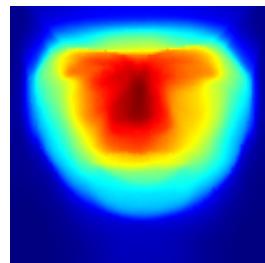
(d) Mapa de calor total de la ResNet34 entrenada con MS1MV3.



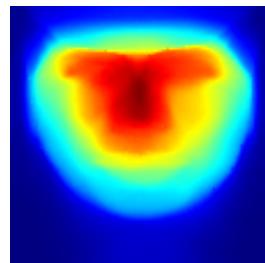
(e) Mapa de calor total de la ResNet50 entrenada con GLINT.



(f) Mapa de calor total de la ResNet50 entrenada con MS1MV3.



(g) Mapa de calor total de la ResNet100 entrenada con GLINT.



(h) Mapa de calor total de la ResNet100 entrenada con MS1MV3.

Figura 5.4: Resumen de los mapas de calor comparados desde el enfoque de las bases de datos.

### 5.2.3 Proporción similitud personas con redes

En principio las redes más potentes deben ser capaz de asimilar mejor los atributos en los rasgos faciales para el reconocimiento facial.

Se han creado los mapa de calor con la media total de las predicciones, para las redes y para los usuarios. Se ha calculado la distancia de KL de cada usuario para cada red.

La idea es comprobar a qué red tienden a parecerse más los mapas de las personas y ver si hay alguna tendencia.

Redes	Proporciones
ResNet100GLINT	39.30 %
ResNet34GLINT	19.78 %
ResNet100MS1MV3	15.45 %
ResNet50MS1MV3	9.76 %
ResNet34MS1MV3	9.49 %
ResNet18MS1MV3	2.71 %
ResNet50GLINT	2.44 %
ResNet18GLINT	1.08 %

Cuadro 5.1: Resumen de proporciones de semejanza global de los sujetos con las redes.

Analizando esta proporción se ve como R100GLINT ha salido claramente victoriosa. Esto es bastante lógico teniendo en cuenta que es la arquitectura más profunda y con la base de datos más potente de las dos posibles. La red contiene más capas con las que extraer características y debido a la mayor variedad de imágenes de GLINT, es más apta para obtener características biométricas.

Otras dos proporciones a tener en cuenta son las de R34GLINT y R100MS1MV3. Se vuelve a observar un impacto debido a la profundidad de la red y la base de datos de preentreno.

Las que peor proporción han obtenido son R18GLINT y R50GLINT. Las redes de menor profundidad no destacan demasiado. Sin embargo, es curioso ver como las R50 parecen tener siempre un rendimiento bastante comedido, dando la impresión de que las R34 son más potentes a pesar presentar una menor profundidad.

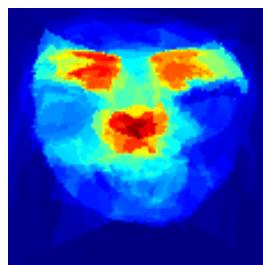
Las arquitecturas con menos neuronas siguen sin destacar tampoco en este ámbito. Ocupando la última y antepenúltima posición.

### 5.2.4 Particularización: diferencias de un sujeto para todos los modelos

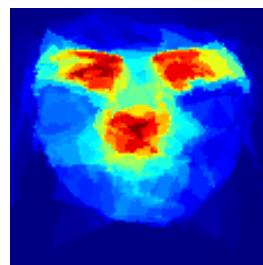
En los anteriores experimentos se ha comprobado que las redes presentan ciertas variaciones entre ellas, según profundidad y base de datos. Para cada sujeto ocurre lo mismo, se suelen fijar en los mismos rasgos, pero cada una dando importancia a detalles diferentes.

Se ha escogido un sujeto cuyos rasgos de la cara predominantes son los ojos, las cejas y la parte central de la nariz. Esta última es la que presenta mayor saturación en la imagen y por lo tanto aquella con más peso.

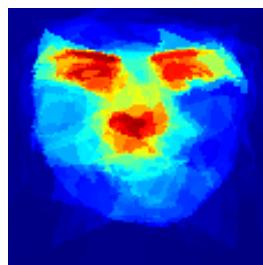
Se realiza la comparación del mapa de calor total, del sujeto con las redes.



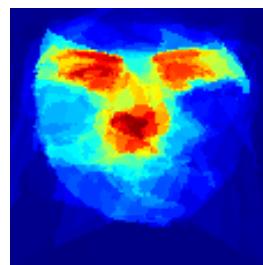
(a) Mapa de calor usuario de la ResNet18 entrenada con GLINT.



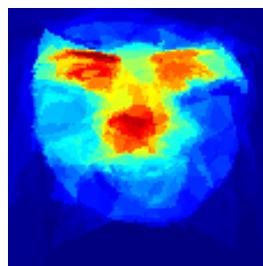
(b) Mapa de calor usuario de la ResNet18 entrenada con MS1MV3.



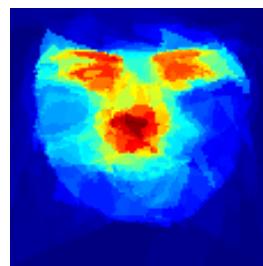
(c) Mapa de calor usuario de la ResNet34 entrenada con GLINT.



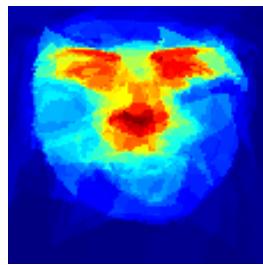
(d) Mapa de calor usuario de la ResNet34 entrenada con MS1MV3.



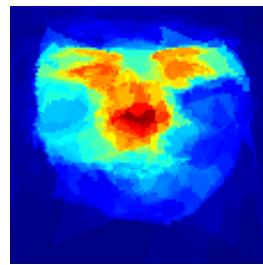
(e) Mapa de calor usuario de la ResNet50 entrenada con GLINT.



(f) Mapa de calor usuario de la ResNet50 entrenada con MS1MV3.



(g) Mapa de calor usuario de la ResNet100 entrenada con GLINT.



(h) Mapa de calor usuario de la ResNet100 entrenada con MS1MV3.

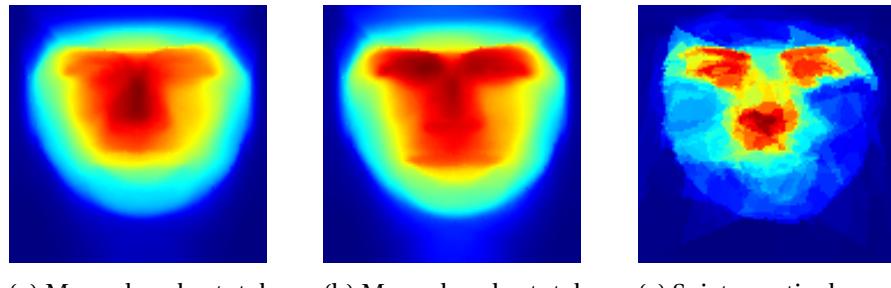
Figura 5.5: Resumen de los mapas de calor de un usuario para todas las redes.

## 5. EXPERIMENTACIÓN

---

Redes	Distancias
ResNet100GLINT	0.02335
ResNet34GLINT	0.02428
ResNet50GLINT	0.02431
ResNet100MS1MV3	0.02447
ResNet50MS1MV3	0.02753
ResNet34MS1MV3	0.02860
ResNet18GLINT	0.03566
ResNet18MS1MV3	0.03667

Cuadro 5.2: Resumen de distancias entre el mapa de calor de un sujeto y las redes neuronales. No tiene unidades específicas, ya que es una medida relativa entre dos distribuciones.



(a) Mapa de calor total de la ResNet100 entrenada con GLINT. Es la red de mayor semejanza con el sujeto.

(b) Mapa de calor total de la ResNet18 entrenada con MS1MV3. Es la red de menor semejanza con el sujeto.

(c) Sujeto particular cuyos rasgos principales son las cejas, ojos y nariz. Los otros rasgos no resaltan.

Hay varias cosas que se pueden observar. La red de menor divergencia es la R100 entrenada con GLINT. Es lo más esperable, ya se ha comprobado en la anterior sección que el 39,30 % de los sujetos suelen asemejarse más a este modelo.

De nuevo, las que más distan son las R18. En la media tabla volvemos a encontrar las arquitecturas intermedias.

R100GLINT y R18MS1MV3 constituyen la divergencia más cercana y la más lejana, respectivamente. De la primera, no hay mucho que decir. Ya se ha comentado anteriormente que al ser la red más potente, es bastante razonable que una mayor proporción de personas se asemenjen más. Sí que es más interesante indagar en los motivos que hacen a la imagen distar más del modelo R18MS1MV3. Se puede notar que su mapa de calor muestra una mayor fijación en la región de los ojos. Y de hecho, es algo que también aparece remarcado en el mapa del sujeto. Sin embargo, este modelo no presta tanta atención a la nariz, que también es muy importante. Además su área de incidencia es mucho más amplia de lo que realmente abarca el mapa del sujeto, que prácticamente solo remarca los ojos y la nariz de forma aislada. A pesar de la ventaja inicial que presentaba esa mayor fijación en los ojos, los otros detalles son los que en su conjunto, alejan el modelo del usuario.

### 5.2.5 Diferencias entre etnias

Los seres humanos presentamos una gran variabilidad en nuestros rasgos faciales. Somos animales y la necesidad de adaptarnos a entornos completamente diferentes ha conllevado a una gran diversidad. Aunque a día de hoy, se está evolucionando hacia una mayor globalización, categorizar por razas es cada vez más complicado. El aumento de integración y conexión entre todas las personas del mundo es una realidad.

Pero las redes neuronales no dejan de estar entrenadas con datos proporcionados por nosotros, que de forma inexorable están sesgados por años y años de cultura supeditada a muchos desequilibrios sociales. Han surgido variedad de temas de debate sobre si las redes son racistas, ya que hace no tanto aparecieron problemas para reconocer personas de etnias no caucásicas. Los investigadores se dieron cuenta de las profundas carencias en la representación de diversidad en las bases de datos[82]. A raíz de esto, se empezó a dar más importancia a la variedad de caras, poses y escenas.

Una idea sería estudiar sobre cómo estos posibles sesgos pueden actuar en los modelos de reconocimiento facial. Ver si las porciones de la cara en las que se suelen fijar son muy distintas a raíz de esto.

También si hay facciones que sean más características según la etnia. La literatura siempre apuesta por una mayor fijación en los ojos y sin embargo esto también podría tener cierto nivel de sesgo, según si los estudios científicos se han realizado en muestras lo suficientemente grandes y variadas de la población como para considerarse de rigor.

En este caso concreto, las redes neuronales han sido preentrenadas con extensas bases de datos y ya con cierta diversidad en sus contenidos.

Para el caso de estudio que ataña este proyecto, se ha optado por una categorización étnica de tres posibles razas, siguiendo un criterio basado en rasgos mas distintivos. La clasificación final es : caucásicos, africanos y asiáticos. Se compara entre los mapas de calor totales y el conjunto de todas las redes neuronales para cada una de las etnias. Para poder hacer esta clasificación, se ha hecho una revisión manual de los 369 individuos que componen la base de datos, obteniendo la siguiente tabla de proporciones:

Etnias	Personas	Proporción
Africanos	21 personas	5.69 %
Caucásicos	309 personas	83.74 %
Asiáticos	39 personas	10.57 %

Cuadro 5.3: Resumen de proporciones por etnias

Por lo que sería esperable una precisión mucho mayor en el caso de los caucásicos, a causa de contar con una mayor representación.

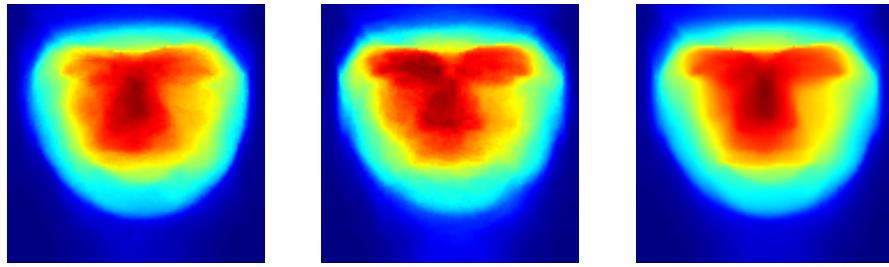
En sus mapas totales, se puede observar una distribución más uniforme del área de impacto en la raza caucásica. Esta tiene bastante fijación por la nariz, aunque no deja de tomar en cuenta tanto los ojos como la parte del mentón.

Por la parte de los asiáticos, el mapa de calor está algo más diluido. Sigue dando mucha importancia a la nariz. Es bastante similar al de los caucásicos, aunque la parte de las mejillas es algo más ancha.

El mapa de los africanos da mucha más relevancia a los ojos siendo esta área más amplia que en los otros mapas. No deja de lado la importancia de la nariz y también se

## 5. EXPERIMENTACIÓN

---



(a) Mapa de calor total de todas las redes para la etnia asiática.

(b) Mapa de calor total de todas las redes para la etnia africana.

(c) Mapa de calor total de todas las redes para la etnia caucásica.

Figura 5.7: Resumen de los mapas de calor comparados desde un enfoque basado en los rasgos más importantes por etnia.

fija en la parte de las mejillas.

También se calcula el mapa de calor total de las tres etnias, según cada red. Hay que tener en cuenta un detalle y es que la proporción de imágenes de africanos, asiáticos y caucásicos difiere mucho. Esto supone una distribución desigual, dándole más peso a los caucásicos. Por lo que no se puede calcular la media directamente con los 369 individuos de la base de datos.

Para evitar ese problema de sesgo, se aplica una normalización por grupo étnico, donde se calcula el promedio para los valores de cada mapa de calor relativo. Posteriormente, se combina las medias calculadas y ponderadas. De esta forma, se elimina el peso positivo hacia el grupo con mayor representación, los caucásicos. Este enfoque asegura que cada grupo étnico tenga la misma influencia en la creación del mapa de calor total, eliminando cualquier tipo de ventaja debido al desequilibrio en el tamaño de la muestra.

A continuación, se hace un análisis por secciones de la matriz de los mapas de calor obtenidos. Empezando por un enfoque según la perspectiva de los modelos. Y siguiendo por el de las razas, donde las observaciones se centran más en las diferencias entre etnias. Finalmente se hace la conclusión.

Empezando desde un análisis por redes. En general, para cada raza, se puede ver que a medida que la profundidad de la red aumenta, esta se fija menos en la parte más cercana del mentón. Las arquitecturas R18 obtienen resultados mucho más anchos de la cara, parece que tienen mucho más en cuenta las mejillas. En particular, la preentrenada con MS1MV3, se fija mucho en los ojos, ignorando más la parte de la nariz. Las arquitecturas medianas son muy parecidas entre ellas, obteniendo resultados prácticamente idénticos tanto en R34 y R50. Las más potentes convergen mucho más en zonas específicas de la cara, dándole mayor peso a la nariz y disminuyendo la relevancia de las otras áreas.

En cuanto al análisis por raza, ya se comentó anteriormente que los caucásicos se fijan más en la nariz, asiáticos igual aunque en menor grado y en los africanos es algo más disperso. Sin embargo, aquí se aparecen comportamientos bastante curiosos. Se puede observar que cada red presenta más variabilidad a la hora de detectar rasgos importantes de cada raza. Se marcan zonas de la cara con diferentes intensidades y en diferente proporción, para todas ellas. También se aprecian estos matices en el

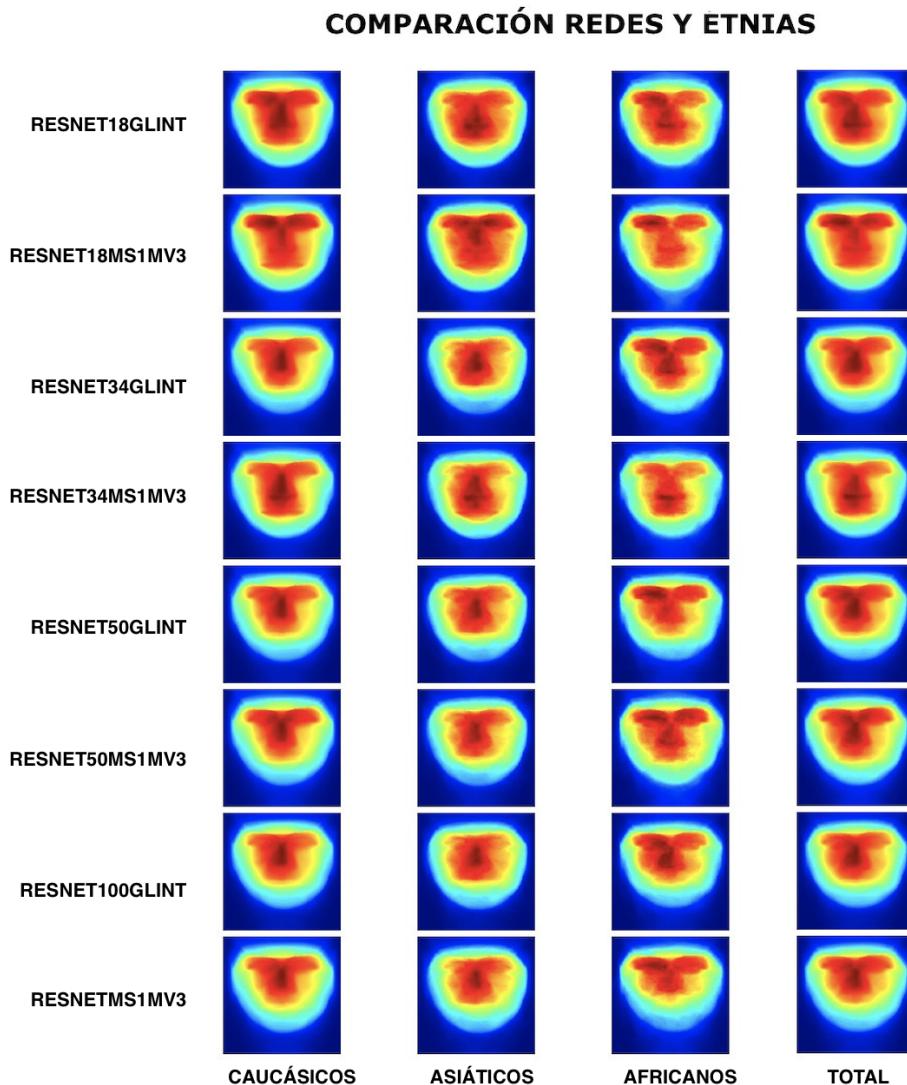
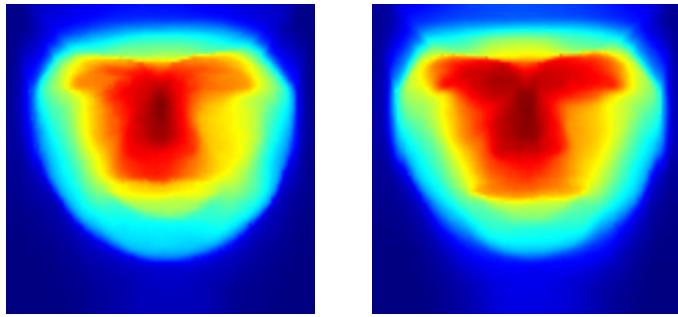


Figura 5.8: Comparación de todos los modelos entre las tres etnias y la media total.

conjunto del promedio total.

De forma global, las bases de datos no parecen ofrecer ningún tipo de sesgo hacia ninguna raza, ya que los mapas de calor son similares. En cualquier caso las diferencias vienen dadas por la proporción de muestras en cada etnia. Y es que tanto GLINT como MS1MV3 fueron entrenadas con una gran cantidad de sujetos y de extensa variedad. También se puede ver que por razas se suele fijar en lo mismo, presentando diferencias en cada una. De forma resumida, los caucásicos destacan más por el conjunto de nariz y ojos. Asiáticos nariz, restándole importancia a los ojos que a apenas son tenidos en cuenta. Y africanos, que sí toman más relevancia los ojos.



(a) Mapa de calor total de todas las redes para mujeres.  
 (b) Mapa de calor total de todas las redes para hombres

Figura 5.9: Resumen de los mapas de calor comparados desde un enfoque basado en los rasgos más importantes por sexo.

### 5.2.6 Diferencias entre sexos

Hombres y mujeres presentamos diferencias tanto físicas como faciales. De la misma manera que en el caso anterior, las redes neuronales pueden presentar variedad en el reconocimiento facial por sexo. En las bases de datos de entrenamiento, la proporción está bastante más compensada que con las etnias. Sin embargo, sigue habiendo un trato favorable hacia los hombres, que obtienen una mayor representación de imágenes.

En el caso de estudio de este proyecto, se trabaja con los siguientes números.

Sexo	Personas	Proporción
Mujeres	139 personas	37.67 %
Hombres	230 personas	62.33 %

Cuadro 5.4: Resumen de proporciones por sexos

Hay cierto desajuste. Sin embargo las muestras para las mujeres siguen siendo bastante amplias y se puede realizar un estudio bastante adecuado de sus respectivos mapas de calor. Aunque no hay que olvidar que puede haber cierto peso favorable para los hombres.

En sus mapas se pueden observar varios puntos a tener en cuenta. En los hombres sí que hay una mayor fijación en los ojos y la parte de las cejas. También tiene muy en cuenta la nariz, pero sí que se puede notar una mayor saturación en la parte de los ojos respecto a lo visto hasta ahora. En cambio, para las mujeres los ojos apenas sobresalen. El mapeo de calor se acota bastante respecto a los hombres, que parece que tiene en cuenta muchas más facciones de la cara, llegando hasta la parte de la mandíbula. Esto podría ser también porque los hombres suelen presentar cabezas más grandes y por lo tanto el rostro ocupa más espacio.

A continuación, se comparan los mapas de calor para ambos sexos, según cada red para ver qué rasgos son más importantes. Además, se ha organizado en formato matriz, por lo que se podrá apreciar la distinción entre sexos. Adicionalmente, se ha añadido una columna final donde se muestre el resultado de la media de mapas de calor para ambos sexos. El resumen de los rasgos más importantes en general para cada red.

## 5.2. Experimentos

---

A continuación, se divide el análisis por secciones. De la misma manera que con el resumen de las etnias, primero se estudiará la relación entre modelos. Qué es lo que destaca más y las diferencias que presentan por arquitectura y base de datos. Luego el foco recaerá en los rasgos más importantes por sexo. Se finaliza con las conclusiones globales.

En el análisis por redes se observan las mismas particularidades que en el estudio de las etnias. R18MS1MV3 tiene mucha fijación por la zona de los ojos, especialmente en los hombres. También abarca la parte de la mandíbula de forma muy prominente, en ambos sexos. La red que menos se fija en los ojos es R34. Y la que más R18, ambas entrenadas con MS1MV3. De nuevo, las redes más profundas son las que acotan más el área de calor. Es curioso ver cómo no necesitan resaltar tanto los rasgos para poder reconocer las caras.

En el análisis por sexo, los ojos suponen mayor fijación en el caso de los hombres. En las mujeres apenas constituyen relevancia, siendo más importantes las mejillas y otras partes del contorno de la cara. En ellas se focaliza más la nariz, de forma más sutil el puente. En los hombres abarca la zona nasal también.

Globalmente, si que se notan diferencias entre sexos. La parte de los ojos y mandíbula en hombres está más remarcada en los mapas. Biológicamente, los hombres tienen la cara más larga y más grande que las mujeres. Esto se refleja en una cara más rotunda y ancha, siendo pómulos, mandíbula y frente más prominentes que en el caso de las mujeres. El mapa está mucho más acotado por la parte central de la cara en las mujeres, debido a que por lo general, biológicamente ellas presentan rasgos mucho más suavizados que los de su contraparte masculina. Los reconocedores faciales no parecen que tengan una fijación particular hacia ningún rasgo. Las mujeres no presentan facciones tan distintivas, más no quiere decir que de forma global sí lo hagan. Pero si una persona presenta rasgos más particulares, será más fácil de reconocer. Y esto es lo que sucede con los hombres al tener rasgos que resaltan más, en general.

## 5. EXPERIMENTACIÓN

---

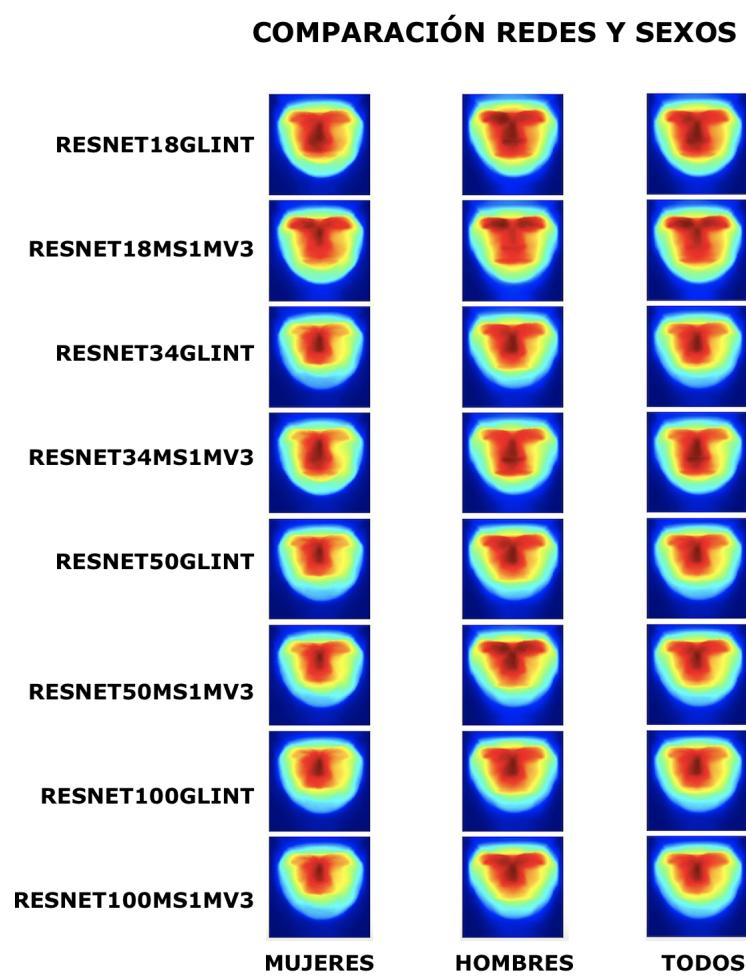


Figura 5.10: Comparación de todos los modelos entre los dos sexos y la media total.

## CONCLUSIONES Y APLICACIONES FUTURAS

En este capítulo se glosa el resumen del trabajo realizado y se exponen las conclusiones fruto del análisis aplicado a los resultados obtenidos. Se indaga también en las posibles utilidades futuras que pueda aportar este trabajo en el ámbito de la investigación.

### 6.1 Conclusiones

Entender cómo funcionan internamente los modelos de redes neuronales es una cuestión que está emergiendo simultáneamente con el avance del aprendizaje profundo. La naturaleza inexplorada en los modelos de inteligencia artificial abre las puertas a nuevas investigaciones.

Este proyecto es un estudio sobre qué razonamientos y procesos cognitivos siguen diferentes redes neuronales en el ámbito de la biometría facial. Se aplican una serie de técnicas para deducir cuáles son las características y rasgos de la cara más importantes para identificar las caras de las personas. El objetivo es observar qué tipo de variables podrían influir en las posibles predicciones, si estas son consistentes y si hay mucha diferencia entre ellas.

El flujo de trabajo consiste en dar una explicación mediante la técnica LIME, del razonamiento de unas redes especializadas para el reconocimiento facial como son las ResNet. Se aplica una adaptación en la cuál se utiliza la distancia del coseno para poder representar las diferencias entre los vectores de cada imagen.

Una vez obtenidas los superpíxeles de la cara más importantes de cada individuo, se aplica un proceso de normalización de estas para que todos los puntos de interés facial ocupen los mismos espacios. Para más adelante ponderar la acumulación de los rasgos más identificativos de los humanos, en forma de mapas de calor. Se exploraron varias perspectivas, desde unas más globales y que contemplan todo el conjunto de datos, hasta otras de mayor granularidad e individualidad.

Los resultados de los experimentos muestran diferencias sutiles entre los modelos y son útiles para comprender mejor su funcionamiento intrínseco. En primer lugar, de

## 6. CONCLUSIONES Y APLICACIONES FUTURAS

---

forma general se ha podido comprobar que el rasgo facial con mayor presencia es la nariz. En menor medida los ojos y la boca. En particular, la R18 entrenada con GLINT es la que más acusa a la nariz. R18MS1MV3, en los ojos. Se ha podido comprobar que a menos profundidad, las redes tienden a darle más importancia a algunos rasgos en específico, además de abarcar áreas de acción mayores. En cuanto de las arquitecturas intermedias, como R34 y R50, se puede ver que son bastante similares entre sí. A medida que la profundidad aumenta, la probabilidad de una mejor predicción también. Las redes más potentes son más capaces de captar los rasgos distintivos de las caras de los sujetos, sin presentar tanta variabilidad. Es decir, se centran más en los rasgos que importan para el sujeto. Únicamente hace falta reconocer un conjunto de características primordiales de cada cara. También se ha podido comprobar ciertas divergencias en cuanto a las bases de datos. GLINT es capaz de representar de forma general mejor a los conjuntos de personas. De la misma manera que con la profundidad de la red, el área de acción disminuye. Esto es razonable teniendo en cuenta que su escala de imágenes de preentrenamiento es mucho mayor que en MS1MV3. En cuanto al estudio por etnias, se encontró que los caucásicos tienen un área de incidencia más alargada, tomando importancia tanto los ojos como la mandíbula, además de la nariz. Los africanos presentan una mayor anchura en las zonas de reconocimiento y los asiáticos restan importancia a la zona de los ojos. En el estudio por sexos, en los hombres se observa una mayor anchura del mapa de calor, obteniendo una amplitud mucho mayor. Las mujeres por su parte, no presentan rasgos especialmente saturados, la intensidad en general disminuye.

### 6.2 Aplicaciones futuras

El terreno de la biometría facial abarca muchas vertientes que pueden ser exploradas en un futuro. En particular, con el estudio de este proyecto utilizando un enfoque adicional de LIME para el análisis de explicabilidad, se pueden tener en cuenta algunas direcciones futuras como las siguientes:

- Se puede usar la explicabilidad LIME para mejorar el reconocimiento en las imágenes cuyas distancias resultaron mayores. Al combinar dos redes que presten atención a diferentes regiones de la cara, pueden mejorar los resultados. Se minimiza el impacto de las occlusiones naturales que se pudiesen producir en según que regiones de la cara, como el uso de gafas, mascarillas, sombrero, gorros, entre otras prendas.
- Según los resultados obtenidos, las redes se fijan en la parte frontal de la cara, en los ojos, nariz y boca. En la literatura psicofísica sugiere que las personas suscitan una mayor fijación de los ojos para reconocer caras. Se puede forzar algún entrenamiento de la red a que se fije en alguna región en específico, como los ojos. Y ver los resultados que se obtienen.
- También se pueden realizar estudios que traten el tema de cómo la condición de las imágenes afecta directamente a la calidad de las predicciones en las redes

## 6.2. Aplicaciones futuras

---

neuronales. Es bastante posible que dos redes diverjan más si la imagen que se les ha proporcionado es de baja calidad. Se realizaron estudios que relacionan imágenes de baja calidad con una mayor variabilidad en las predicciones. Aquellas de mayor calidad, aglomeraba las regiones en la parte central de la cara. Se podría usar el método de explicabilidad LIME basado en la distancia del coseno para evaluar el nivel de calidad de las imágenes.



## BIBLIOGRAFÍA

- [1] H. Ichikawa, E. Nakato, Y. Igarashi, M. Okada, S. Kanazawa, M. K. Yamaguchi, and R. Kakigi, “A longitudinal study of infant view-invariant face processing during the first 3-8 months of life,” Febrero 2019. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/30529397/> 1.1.1
- [2] M. S. Keil, ““i look in your eyes, honey”: Internal face features induce spatial frequency preference for human face processing,” Marzo 2009. [Online]. Available: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000329> 1.1.1
- [3] M. D. D. CEREZO, “Reconocimiento facial, el superpoder que tenemos casi todos los humanos,” Octubre 2022. [Online]. Available: <https://www.rtve.es/noticias/20221018/reconocimiento-facial-superpoder-exclusivo-humanos/2399136.shtml> 1.1.1
- [4] D. J. Matich, “Redes neuronales: Conceptos basicos y aplicaciones.” Marzo 2001. [Online]. Available: [https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5\\_anio/orientadora1/monograias/matich-redesneuronales.pdf](https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf) 1.2
- [5] AWS, “¿que es una red neuronal?” [Online]. Available: <https://aws.amazon.com/es/what-is/neural-network/> 1.2
- [6] Coursera, “What is artificial intelligence? definition, uses, and types,” 2023. [Online]. Available: [https://www.coursera.org/articles/what-is-artificial-intelligence?utm\\_medium=sem&utm\\_source=gg&utm\\_campaign=B2C\\_EMEA\\_coursera\\_FTCOF\\_career-academy\\_pmax-nonNRL-within-14d-country-ES&campaignid=20425190789&adgroupid=&device=c&keyword=&matchtype=&network=x&devicemodel=&adposition=&creativeid=&hide\\_mobile\\_promo&gclid=Cj0KCQjwi7GnBhDXARIIsAFLvH4lRIJ-tOuF6V5wdfvPzjXjs2Fuo\\_1Wjp7MjUi1qAtW9Tke7ZG86dhcaAncFEALw\\_wcB](https://www.coursera.org/articles/what-is-artificial-intelligence?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA_coursera_FTCOF_career-academy_pmax-nonNRL-within-14d-country-ES&campaignid=20425190789&adgroupid=&device=c&keyword=&matchtype=&network=x&devicemodel=&adposition=&creativeid=&hide_mobile_promo&gclid=Cj0KCQjwi7GnBhDXARIIsAFLvH4lRIJ-tOuF6V5wdfvPzjXjs2Fuo_1Wjp7MjUi1qAtW9Tke7ZG86dhcaAncFEALw_wcB) 1.2
- [7] RecFaces, “¿que es la visión por computadora? principales funciones.” [Online]. Available: <https://recfaces.com/es/articles/vision-computador-soluciones> 1.2
- [8] M. B. Stegmann and D. D. Gomez, “A brief introduction to statistical shape analysis,” Marzo 2002. [Online]. Available: [https://graphics.stanford.edu/courses/cs164-09-spring/Handouts/paper\\_shape\\_spaces\\_imm403.pdf](https://graphics.stanford.edu/courses/cs164-09-spring/Handouts/paper_shape_spaces_imm403.pdf) 1.3
- [9] MATLAB, “Manual matlab.” [Online]. Available: <https://es.mathworks.com/products/matlab.html> 2.1.1

## BIBLIOGRAFÍA

---

- [10] Unir, "Lenguaje r, ¿qué es y por qué es tan usado en big data?" Noviembre 2019. [Online]. Available: <https://www.unir.net/ingenieria/revista/lenguaje-r-big-data/2.1.1>
- [11] T. School, "Lenguaje de programacion julia: ideal para machine learning," Diciembre 2020. [Online]. Available: <https://www.tokioschool.com/noticias/lenguaje-programacion-julia/2.1.1>
- [12] LUCA, "¿qué es python?" Febrero 2020. [Online]. Available: <https://web.archive.org/web/20200224120525/https://luca-d3.com/es/data-speaks/diccionario-tecnologico/python-lenguaje> 2.1.1
- [13] A. Staff, "Visual studio now supports debugging linux apps; code editor now open source," Noviembre 2015. [Online]. Available: <https://arstechnica.com/information-technology/2015/11/visual-studio-now-supports-debugging-linux-apps-code-editor-now-open-source/> 2.1.2
- [14] Jupyter, "Project jupyter documentation." [Online]. Available: <https://docs.jupyter.org/en/latest/> 2.1.2
- [15] JetBrains, "Ide de python para desarrolladores profesionales." [Online]. Available: <https://www.jetbrains.com/es-es/pycharm/> 2.1.2
- [16] Conda, "Conda documentation." [Online]. Available: <https://conda.io/en/latest/> 2.1.3
- [17] A. Omorogbe, E. Urban, L. Franks, S. Gilley, M. Akande, H. Arya, T. Takebayashi, C. Gronlund, "GiftA-MSFT", ".@tikmapari", "j martens", D. Coulter, F. Xu, and P. Lu, "Onnx y azure machine learning: Crear y acelerar modelos de ml," Junio 2023. [Online]. Available: <https://learn.microsoft.com/es-es/azure/machine-learning/concept-onnx?view=azureml-api-2> 2.1.4
- [18] R. KeepCoding, "¿qué es numpy y cómo funciona?" Noviembre 2022. [Online]. Available: <https://keepcoding.io/blog/que-es-numpy-y-como-funciona/> 2.1.4
- [19] L. C. Jiménez, S. P. Tituana, and A. R. Pincay, "Detección de mascarilla para covid-19 a través de aprendizaje profundo usando opencv y cascade trainer gui," Junio 2021. [Online]. Available: <https://incyt.upse.edu.ec/ciencia/revistas/index.php/rctu/article/view/572/509> 2.1.4
- [20] decodigo, "Mapeo facial con dlib y python." [Online]. Available: <https://decodigo.com/mapeo-facial-con-dlib-y-python> 2.1.4
- [21] D. de Luca, "Qué es opencv." [Online]. Available: <https://damiandeluca.com.ar/que-es-opencv> 2.1.4
- [22] L. A. O. García, "Scipy: El aliado de un matemático," Mayo 2023. [Online]. Available: <https://es.linkedin.com/pulse/scipy-el-aliado-de-un-matemÁtico-luis-alberto-oraa-garcia> 2.1.4

- [23] Uniwebsidad, “Módulos de sistema.” [Online]. Available: <https://uniwebsidad.com/libros/python/capitulo-10/modulos-de-sistema> 2.1.4
- [24] D. Python, “functools — funciones de orden superior y operaciones sobre objetos invocables.” [Online]. Available: <https://docs.python.org/es/3.8/library/functools.html> 2.1.4
- [25] J. Vieco, “Pytorch: ¿qué es y como se instala?” Diciembre 2017. [Online]. Available: <https://cleverpy.com/2017/12/04/que-es-pytorch-y-como-se-instala/> 2.1.4
- [26] P. J. D. L. Santos, “Introducción a scikit-image, procesamiento de imágenes en python,” Enero 2016. [Online]. Available: <https://numython.github.io/posts/2016/01/introduccion-scikit-image-procesamiento/> 2.1.4
- [27] DocumentaciónSkicit-Image, “skimage.segmentation documentation.” [Online]. Available: <https://scikit-image.org/docs/stable/api/skimage.segmentation.html> 2.1.4
- [28] U. de Alcalá, “Scikit-learn, herramienta básica para el data science en python.” [Online]. Available: <https://www.master-data-scientist.com/scikit-learn-data-science/> 2.1.4
- [29] D. Skicit-Learn, “Metrics and scoring: quantifying the quality of predictions.” [Online]. Available: [https://scikit-learn.org/stable/modules/model\\_evaluation.html](https://scikit-learn.org/stable/modules/model_evaluation.html) 2.1.4
- [30] Uc3m, “Overleaf - editor online latex.” [Online]. Available: <https://www.uc3m.es/sdic/servicios/overleaf> 2.1.5
- [31] ———, “Latex: redacción de documentos científicos,” Agosto 2023. [Online]. Available: <https://guiasbib.upo.es/latex> 2.1.5
- [32] A. R. Villalobos, “Latex en el mac os x,” Marzo 2009. [Online]. Available: <https://arodriguez.blogs.upv.es/latex-en-el-mac-os-x/> 2.1.5
- [33] N. Geographic, “Breve historia visual de la inteligencia artificial,” Diciembre 2020. [Online]. Available: [https://www.nationalgeographic.com.es/ciencia/breve-historia-visual-inteligencia-artificial\\_14419](https://www.nationalgeographic.com.es/ciencia/breve-historia-visual-inteligencia-artificial_14419) 3.1
- [34] A. Crawls, “¿que son las redes neuronales?” Marzo 2012. [Online]. Available: [https://web.archive.org/web/20141216214512/http://info fisica.uson.mx/arnulfo.castellanos/archivos\\_html/quesonredneu.htm](https://web.archive.org/web/20141216214512/http://info fisica.uson.mx/arnulfo.castellanos/archivos_html/quesonredneu.htm) 3.1
- [35] R. R. Abril, “Redes neuronales artificiales.” [Online]. Available: <https://lamaquinaoraculo.com/deep-learning/redes-neuronales-artificiales/> 3.1
- [36] J. C. Martin, “La importancia de las funciones de activación en una red neuronal,” Agosto 2022. [Online]. Available: <https://es.linkedin.com/pulse/la-importancia-de-las-funciones-activacion-en-una-red-calvo-martin> 3.1

## BIBLIOGRAFÍA

---

- [37] SIODEC, “¿las redes neuronales sueñan con ilusiones visuales?” 2020. [Online]. Available: <http://www.siodec.org/las-redes-neuronales-suenan-con-ilusiones-visuales/> 3.2
- [38] S. Silva and E. Freire, “Intro a las redes neuronales convolucionales,” Noviembre 2019. [Online]. Available: <https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8> 3.2
- [39] M. Quiroa, “Red neuronal convolucional,” Junio 2023. [Online]. Available: <https://economipedia.com/definiciones/red-neuronal-convolucional.html> 3.2
- [40] MathWorks, “¿qué son las redes neuronales convolucionales?” [Online]. Available: <https://es.mathworks.com/discovery/convolutional-neural-network-matlab.html> 3.2
- [41] A. Hernández, “Problema del desvanecimiento del gradiente (vanishing gradient problem),” Mayo 2018. [Online]. Available: <https://mlearninglab.com/2018/05/06/problema-de-desvanecimiento-del-gradiente-vanishing-gradient-problem/> 3.2
- [42] A. Anwar, “Difference between alexnet, vggnet, resnet, and inception,” Junio 2019. [Online]. Available: <https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecc96> 3.3
- [43] KOUSTUBH, “Resnet, alexnet, vggnet, inception: Understanding various architectures of convolutional networks,” 2017. [Online]. Available: <https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/> 3.3
- [44] S. Das, “Cnn architectures: Lenet, alexnet, vgg, googlenet, resnet and more...,” Noviembre 2017. [Online]. Available: <https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5> 3.3
- [45] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” Agosto 2016. [Online]. Available: <https://arxiv.org/abs/1608.06993v5> 3.3
- [46] Q. Wang and G. Guo, “Benchmarking deep learning techniques for face recognition,” Septiembre 2019. [Online]. Available: <https://par.nsf.gov/servlets/purl/10140793> 3.3
- [47] Javatpoint, “Biometric system functionality.” [Online]. Available: <https://www.javatpoint.com/biometric-system-functionality> 3.3
- [48] D. Riccio, “How do i determine the accuracy of face recognition?” Junio 2013. [Online]. Available: [https://www.researchgate.net/post/How\\_do\\_I\\_determine\\_the\\_accuracy\\_of\\_Face\\_Recognition](https://www.researchgate.net/post/How_do_I_determine_the_accuracy_of_Face_Recognition) 3.3
- [49] D. S. Team, “Redes neuronales residuales – lo que necesitas saber (resnet),” Mayo 2020. [Online]. Available: <https://datascience.eu/es/aprendizaje-automatico/una-vision-general-de-resnet-y-sus-variantes/> 3.4

- [50] H. J. P. J., "The cerebral cortex beyond the cortex," Octubre 2004. [Online]. Available: [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0034-74502004000500005](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0034-74502004000500005) 3.4
- [51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Diciembre 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385> 3.4
- [52] R. R. Abril, "Redes residuales: Resnet." [Online]. Available: <https://lamaquinaoraculo.com/deep-learning/redes-residuales/> 3.4, 3.4
- [53] P. Oommen, "Resnets — residual blocks and deep residual learning," Noviembre 2020. [Online]. Available: <https://towardsdatascience.com/resnets-residual-blocks-deep-residual-learning-a231a0ee73d2> 3.4
- [54] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," 2019. [Online]. Available: [10.1109/CVPR.2019.00482](https://doi.org/10.1109/CVPR.2019.00482) 3.5
- [55] R. Alonso, "Descubra qué es tensor core y cuál es su función," Marzo 2023. [Online]. Available: <https://cultura-informatica.com/conceptos/que-es-tensor-cores/> 3.5
- [56] D. Cochard, "Arcface : A machine learning model for face recognition," Mayo 2021. [Online]. Available: <https://medium.com/axinc-ai/arcface-a-machine-learning-model-for-face-recognition-5f743cdac6fa> 3.5
- [57] Javi, "Función softmax: Activación para la clasificación," Marzo 2023. [Online]. Available: <https://jacar.es/funcion-softmax-activacion-para-la-clasificacion/> 3.5
- [58] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," Agosto 2015. [Online]. Available: <https://arxiv.org/pdf/1801.07698.pdf> 3.5
- [59] Y. Wenming, J. Sun, R. Gao, J.-H. Xue, and Q. Liao, "Inter-class angular margin loss for face recognition," Septiembre 2019. [Online]. Available: [https://www.researchgate.net/publication/335805835\\_Inter-class.angular\\_margin\\_loss\\_for\\_face\\_recognition](https://www.researchgate.net/publication/335805835_Inter-class.angular_margin_loss_for_face_recognition) 3.5
- [60] X. An, X. Zhu, Y. Xiao, L. Wu, M. Zhang, Y. Gao, B. Qin, D. Zhang, and Y. Fu, "Partial fc: Training 10 million identities on a single machine," Enero 2021. [Online]. Available: <https://arxiv.org/pdf/2210.13664.pdf> 3.6
- [61] An, "Partial fc: Training 10 million identities on a single machine." [Online]. Available: <https://arxiv.org/pdf/2010.05222.pdf> 3.6
- [62] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," 2018. [Online]. Available: [10.1109/FG.2018.00020](https://doi.org/10.1109/FG.2018.00020) 3.6
- [63] J. B. Watson, "Psychology as the behaviorist views it." 2016. [Online]. Available: <https://psycnet.apa.org/record/1926-03227-001> 4.1

## BIBLIOGRAFÍA

---

- [64] M. T. Ribeiro, S. Singh, and C. Guestrin, “"why should i trust you?": Explaining the predictions of any classifier,” Febrero 2016. [Online]. Available: <https://arxiv.org/abs/1602.04938> 4.1
- [65] K. Safjan, “Lime - understanding how this method for explainable ai works,” Abril 2023. [Online]. Available: <https://safjan.com/how-the-lime-method-for-explainable-ai-works/> 4.1
- [66] R. Winastwan, “Interpreting image classification model with lime,” Enero 2021. [Online]. Available: <https://towardsdatascience.com/interpreting-image-classification-model-with-lime-1e7064a2f2e5> 4.1
- [67] M. T. Ribeiro, “Lime - local interpretable model-agnostic explanations.” [Online]. Available: <https://homes.cs.washington.edu/~marcotcr/blog/lime/> 4.1
- [68] A. Kundu, “Explaining face mask image classification model using lime,” Enero 2022. [Online]. Available: <https://towardsdatascience.com/explaining-face-mask-image-classification-model-using-lime-8f423c601ff9> 4.3.3
- [69] D. Jain, “Superpixels and slic,” Mayo 2019. [Online]. Available: <https://darshita1405.medium.com/superpixels-and-slic-6b2d8a6e4f08> 4.3.3
- [70] D. Singh, “Linear, lasso, and ridge regression with scikit-learn,” Mayo 2019. [Online]. Available: <https://www.pluralsight.com/guides/linear-lasso-ridge-regression-scikit-learn> 4.3.3
- [71] P. Cignoni, R. Scopigno, and C. Montani, “Dewall: A fast divide and conquer delaunay triangulation algorithm in ed,” Octubre 1997. [Online]. Available: <https://vcg.isti.cnr.it/publications/papers/dewall.pdf> 4.4
- [72] R. Fernandez and Felisa, “Deteccion de rostros, caras y ojos con haar cascad,” Abril 2018. [Online]. Available: [https://unipython.com/deteccion-rostros-caras-ojos-haar-cascad/?utm\\_content=cmp=true](https://unipython.com/deteccion-rostros-caras-ojos-haar-cascad/?utm_content=cmp=true) 4.4.1
- [73] A. Rosebrock, “Facial landmarks with dlib, opencv, and python,” Abril 2017. [Online]. Available: <https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/> 4.4.2
- [74] S. Pandey, “Dlib 68 points face landmark detection with opencv and python,” 2023. [Online]. Available: <https://www.studytonight.com/post/dlib-68-points-face-landmark-detection-with-opencv-and-python> 4.4.2
- [75] L. Wilkinson and M. Friendly, “The history of the cluster heat map,” Enero 2012. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1198/tas.2009.0033> 4.5
- [76] S. Singhal, “All about heatmaps,” Diciembre 2020. [Online]. Available: <https://towardsdatascience.com/all-about-heatmaps-bb7d97f099d7> 4.5
- [77] E. Fernandez, “Creadores de dendrograma en línea,” 2018. [Online]. Available: <https://www.neoteo.com/creadores-de-dendrograma-en-linea/> 4.6

- [78] S. H. To, “Ward’s method (minimum variance method).” [Online]. Available: <https://www.statisticshowto.com/wards-method/> 4.6
- [79] L. Benites, “Divergencia kullback-leibler kl,” Noviembre 2021. [Online]. Available: [https://statologos.com/kl-divergencia/?utm\\_content=cmp-true](https://statologos.com/kl-divergencia/?utm_content=cmp-true) 5.1
- [80] Did(<https://stats.stackexchange.com/users/2592/did>), “An adaptation of the kullback-leibler distance?” Agosto 2015. [Online]. Available: <https://stats.stackexchange.com/q/6937> 5.1
- [81] S. Gorthy, “Distance metrics,” Octubre 2021. [Online]. Available: <https://medium.com/mlearning-ai/distance-metrics-9f5830322dee> 5.1
- [82] J. Angileri, M. Brown, J. DiPalma, Z. Ma, and C. L. Dancy, “Ethical considerations of facial classification: Reducing racial bias in ai,” Diciembre 2019. [Online]. Available: [https://www.researchgate.net/publication/338225415\\_Ethical\\_Considerations\\_of\\_Facial\\_Classification\\_Reducing\\_Racial\\_Bias\\_in\\_AI?channel=doi&linkId=5e0916c6299bf10bc382bb88&showFulltext=true](https://www.researchgate.net/publication/338225415_Ethical_Considerations_of_Facial_Classification_Reducing_Racial_Bias_in_AI?channel=doi&linkId=5e0916c6299bf10bc382bb88&showFulltext=true) 5.2.5