



ugr | Universidad
de **Granada**

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

**Estimación de la distancia cámara-sujeto
en fotografías faciales mediante técnicas
de aprendizaje profundo**

Autor
Iván Salinas López

Directores
Enrique Bermejo Nievas
Pablo Mesejo Santiago



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, Junio de 2024

Estimación de distancia cámara-sujeto en fotografías faciales usando aprendizaje profundo

Iván Salinas López

Palabras clave: palabra_clave1, palabra_clave2, palabra_clave3,

Resumen

Poner aquí el resumen.

Camera-subject distance estimation in facial photographs using deep learning

Iván Salinas López

Keywords: Keyword1, Keyword2, Keyword3,

Abstract

Write here the abstract in English.

Yo, **Iván Salinas López**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 78026145W, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Iván Salinas López

Granada a X de Junio de 2024

D. **Enrique Bermejo Nievas**, Investigador Senior en Panacea Cooperative Research y miembro del Instituto Andaluz Interuniversitario en Ciencia de Datos e Inteligencia Computacional.

D. **Pablo Mesejo Santiago**, Profesor del Área de Ciencias de la Computación e Inteligencia Artificial del Departamento Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado *Estimación de la distancia cámara-sujeto en fotografías faciales mediante técnicas de aprendizaje profundo*, ha sido realizado bajo su supervisión por **Iván Salinas López**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de Junio de 2024.

Los directores:

Enrique Bermejo Nievas

Pablo Mesejo Santiago

Agradecimientos

Poner aquí agradecimientos...

Índice general

1. Introducción	1
1.1. Definición del problema	1
1.2. Motivación	4
1.3. Objetivos	5
1.4. Planificación del proyecto	5
2. Fundamentos teóricos	9
2.1. Aprendizaje automático	9
2.2. Aprendizaje profundo	10
2.2.1. Redes neuronales artificiales	10
2.2.2. Redes neuronales convolucionales	12
2.2.3. Transferencia de aprendizaje	16
2.2.4. Regularización	16
2.3. Parámetros de la cámara y perspectiva	18
3. Estado del Arte	21
3.1. Primeros enfoques	21
3.2. MediaPipe Iris	23
3.3. PerspectiveX	23
3.4. FacialSCDnet	24
4. Materiales y métodos	27
4.1. Materiales	27
4.1.1. Modelos 3D	27
4.1.2. Preparación del conjunto de datos	30
4.2. Métodos	32
4.2.1. FacialSCDnet+	33
5. Experimentos	39
5.1. Detalles técnicos de la implementación	39
5.1.1. Entorno de desarrollo	39
5.1.2. Organización del conjunto de datos	39
5.1.3. Entrenamiento del modelo	40
5.1.4. Gestión de experimentos	40

5.2.	Experimentos	41
5.2.1.	Protocolo de validación experimental	41
5.2.2.	Métricas	42
5.2.3.	Experimentos VGG16	43
6.	Conclusiones y trabajos futuros	45

Índice de figuras

1.1.	Número de publicaciones de imágenes faciales.	1
1.2.	Efectos de la distorsión de perspectiva en fotografías faciales.	2
1.3.	Relación entre parámetros de la cámara.	3
1.4.	Diagrama del proyecto.	6
2.1.	Esquema de red neuronal.	10
2.2.	Modelo neuronal.	11
2.3.	Ejemplo de red neuronal convolucional.	12
2.4.	Ejemplo de operación de convolución.	13
2.5.	Funciones de activación comunes.	14
2.6.	Tipos de <i>pooling</i> comunes.	15
2.7.	Infraajuste y sobreajuste en entrenamiento.	17
2.8.	Relación entre punto nodal y longitud focal.	18
2.9.	Tipos de tamaños de sensor.	19
2.10.	Ejemplo de longitud focal equivalente.	19
2.11.	Distancia desde la cámara al sujeto.	20
2.12.	Efectos de la distorsión según distancia.	20
3.1.	Número de publicaciones sobre la estimación de la SCD. . . .	22
4.1.	Ejemplos HeadSpace 3D.	28
4.2.	Ejemplos H3DS-net.	28
4.3.	Ejemplos HuMMan.	29
4.4.	Ejemplos People Snapshot	29
4.5.	Ejemplos Render People	30
4.6.	Ejemplos de imágenes rotadas verticalmente.	32
4.7.	Ejemplos de imágenes rotadas horizontalmente.	32
4.8.	Ejemplos de imágenes generadas para el conjunto de datos. .	33
4.9.	Ejemplos de imágenes utilizadas en FacialSCDnet.	34
4.10.	Transformaciones utilizadas en el aumento de datos.	35
4.11.	Arquitectura de la red VGG-16.	36
5.1.	Esquema de división del conjunto de datos.	41

Índice de cuadros

1.1. Planificación inicial del proyecto	7
1.2. Planificación final del proyecto	7
1.3. Estimación del coste del proyecto	8

Capítulo 1

Introducción

1.1. Definición del problema

En la era digital actual, las imágenes faciales han adquirido una relevancia significativa (véase Figura 1.1 ¹), dado su amplio uso en aplicaciones multimedia, redes sociales, sistemas de vigilancia y seguridad para la identificación de personas o control de accesos en edificios, así como en investigaciones criminales y forenses para la identificación de sospechosos o la reconstrucción de rostros. Esta expansión se debe en gran medida al continuo desarrollo tecnológico, que ha mejorado tanto la calidad como la ubicuidad de las fotografías faciales, permitiendo su presencia en una variedad cada vez mayor de contextos y aplicaciones.

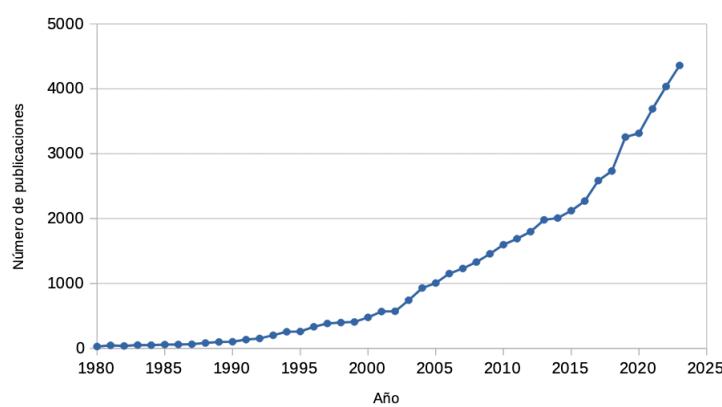


Figura 1.1: Número de publicaciones, en Scopus, relacionadas con imágenes faciales en los últimos 45 años.

¹Las búsquedas se pueden consultar en el apéndice

En este contexto, es importante resaltar el papel que tienen las imágenes faciales en campos como la biometría para la verificación de identidad, así como en la seguridad nacional, donde se pueden utilizar para la identificación facial. Para garantizar el correcto desempeño de estas aplicaciones, se debe tener en cuenta la calidad de las imágenes y todos los factores que afectan a la escena fotográfica. Por ello, existen numerosas herramientas y técnicas dirigidas a la extracción de metadatos, la detección facial o la estimación de la pose [52], todas ellas fundamentales para asegurar la fiabilidad y precisión de los sistemas de identificación facial.

En el ámbito forense, una de las técnicas más empleadas en la identificación facial es la comparación facial forense (CFF) [15]. Esta técnica, llevada a cabo por expertos manualmente o con la ayuda de sistemas automáticos, consiste en identificar similitudes y diferencias entre dos o más imágenes con el objetivo de determinar si representan a la misma persona. Para que este análisis sea confiable y concluyente, las imágenes faciales deben estar en unas condiciones adecuadas. Aspectos como la calidad, la resolución, el enfoque o la iluminación deben cumplir unos requisitos mínimos. Además, es importante que las características de la escena, como el ángulo de la cámara, la posición de la cabeza y la expresión facial, no varíen significativamente, con el fin de asegurar la similitud entre las imágenes y permitir una comparación precisa entre pares [14, 41].

Uno de los factores más importantes a tener en cuenta en las fotografías faciales es la distorsión de perspectiva, la cual puede provocar deformaciones en los rasgos faciales, como en las orejas, la nariz o la forma general del rostro, especialmente cuando la cámara está muy cerca del sujeto al momento de tomar la fotografía [33] (ver Figura 1.2). Esta alteración en la perspectiva repercute negativamente tanto en los sistemas de reconocimiento facial como en la CFF, complicando el análisis visual al alterar como se perciben ciertos rasgos.



Figura 1.2: Efectos de la distorsión de perspectiva en fotografías faciales realizadas a diferentes distancias: 0.3 m, 0.6 m y 1.5 m respectivamente.

La distorsión de perspectiva está estrechamente relacionada con la distancia cámara-sujeto (*subject-to-camera distance*, SCD en adelante). Esta

relación es de decrecimiento logarítmico, lo que significa que a distancias cortas se produce una mayor distorsión, la cual va disminuyendo a medida que la distancia entre la cámara y el sujeto aumenta [42]. Conocer la SCD en fotografías faciales permite cuantificar la cantidad de distorsión presente en una imagen, así como las diferencias en la distorsión entre dos pares de imágenes. Esta información puede ser determinante a la hora de evaluar la identidad de un individuo al aplicar la técnica de CFF. Además, conocer la SCD facilita el desarrollo de técnicas que permitan corregir con precisión dicha distorsión [49].

La SCD, a diferencia de otros parámetros de la cámara como la longitud focal o el tamaño del sensor, no puede obtenerse directamente desde los metadatos de la fotografía [45]. Por tanto, se necesita un método preciso para su estimación.

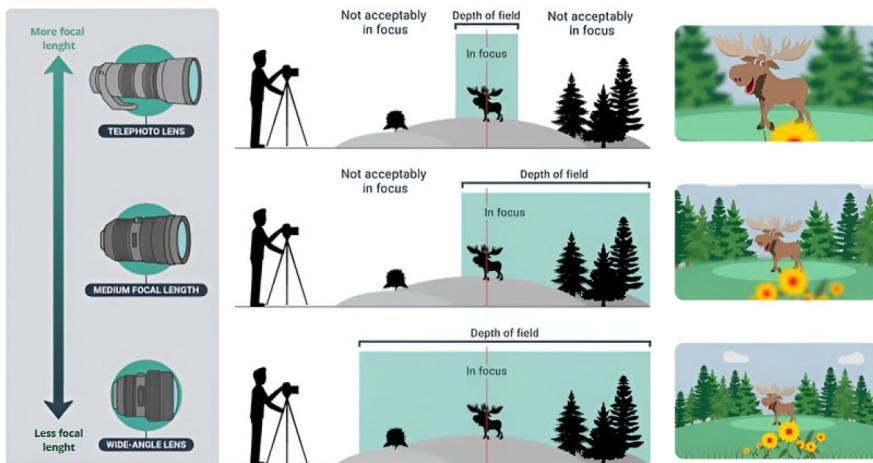


Figura 1.3: Relación entre longitud focal, distancia cámara-sujeto y la profundidad de campo.

En los últimos años se han utilizado varios métodos que combinan técnicas manuales y automatizadas basadas en puntos de referencia o en características anatómicas de la cara[16, 10]. Sin embargo, no se han obtenido resultados favorables debido a la dificultad para obtener estimaciones precisas en largas distancias por la diversa fisionomía de la cara, y a los problemas relacionados con los parámetros de la cámara, como el recorte de imágenes o la combinación de diferentes longitudes focales en el mismo conjunto de datos.

Hasta la fecha, uno de los métodos completamente automatizados para estimar la SCD en fotografías faciales es conocido como FacialSCDnet [4]. Este método emplea una arquitectura basada en aprendizaje profundo para procesar las imágenes faciales y calcular su SCD correspondiente. Sin embar-

go, FacialSCDnet presenta ciertas limitaciones. En primer lugar, el conjunto de datos utilizado es limitado, debido al número reducido de individuos y al hecho de que solo incluye adultos. Además, el conjunto de datos utilizado solo incluye modelos faciales. Estos factores aumentan el riesgo de sesgo en el modelo, lo que puede afectar a su capacidad para realizar estimaciones precisas en poblaciones más diversas.

Considerando todos estos aspectos, el presente Trabajo de Fin de Grado (TFG) pretende mejorar el método actual del estado del arte en la estimación automática de la distancia cámara-sujeto en fotografías faciales. Para ello, partimos de FacialSCDnet como una prueba de concepto sólida, reconociendo sus ventajas pero también identificando sus limitaciones inherentes. Nos centraremos en incorporar mejoras significativas que permitan solventar estas limitaciones con el objetivo de elevar el rendimiento y la precisión del sistema.

1.2. Motivación

En el campo del aprendizaje profundo, es común encontrar sesgos en los datos que pueden afectar la precisión y la capacidad de generalización de las soluciones. Estos sesgos pueden surgir debido a varios factores, como la falta de diversidad o el ruido excesivo en el conjunto de datos utilizado para entrenar los modelos. Este fenómeno también se observa en el método FacialSCDnet. Por lo tanto, para mejorar la calidad del conjunto de datos, sería beneficioso incorporar una mayor diversidad de sujetos en términos de edad, sexo biológico, ascendencia, expresiones faciales, condiciones de iluminación y fondos, así como la inclusión de modelos tanto faciales como de cuerpo completo.

Una estrategia para abordar este desafío consiste en integrar múltiples bases de datos con el objetivo de construir un conjunto de datos más completo y diverso. Esta mejora contribuiría a una mejor capacidad de generalización y adaptación del modelo, al mitigar los sesgos inherentes a los datos.

Por otro lado, obtener conjuntos de datos reales de alta calidad no siempre es una tarea sencilla. La recopilación y etiquetado de datos pueden resultar costosos y requerir bastante tiempo. En muchos casos, los conjuntos de datos reales disponibles pueden ser limitados en términos de tamaño y diversidad, como ocurre con FacialSCDnet. Una solución ampliamente utilizada consiste en emplear conjuntos de datos sintéticos en lugar de conjuntos de datos reales [44, 19, 48]. El uso de conjuntos de datos sintéticos puede ayudar a reducir costos y tiempo de recopilación de datos, manteniendo o mejorando el rendimiento de los modelos.

Dentro del contexto de FacialSCDnet, una restricción clave reside en la necesidad de conocer la longitud focal y desarrollar un modelo en función de esta. Utilizar imágenes sintéticas, facilita considerablemente la creación de conjuntos de datos según la focal deseada. Al emplear datos sintéticos, se tiene un mayor control sobre los parámetros de generación, lo que permite ajustar la focal de manera precisa y reproducible.

Con el propósito de aumentar la calidad de los conjuntos de datos empleados en el aprendizaje profundo, particularmente en el contexto de FacialSCDnet, este trabajo se enfoca en reducir sesgos al integrar una mayor diversidad de sujetos y condiciones, además de optimizar los recursos temporales y financieros mediante el uso de conjuntos de datos sintéticos. Este enfoque combinado busca mejorar la capacidad actual de generalización y adaptación de los modelos de FacialSCDnet.

1.3. Objetivos

El objetivo general de este TFG consiste en desarrollar un mejor modelo de aprendizaje profundo para mejorar la estimación de la distancia cámara-sujeto en fotografías faciales. Para el desarrollo del proyecto, dividiremos el objetivo general en una serie de objetivos parciales:

1. Realizar un análisis exhaustivo del estado del arte y de las bases de datos de modelos faciales y humanos 3D.
2. Desarrollar un protocolo de estandarización y generación de imágenes sintéticas fotorrealistas.
3. Realizar un estudio comparativo y analizar la viabilidad de la nueva aproximación propuesta.
4. Explorar el uso de arquitecturas y tecnologías alternativas que permitan mejorar el rendimiento y/o los resultados del método original.

En este contexto, es relevante citar el trabajo de referencia [4], el cual proporciona una base sólida para el desarrollo del TFG.

1.4. Planificación del proyecto

Para abordar el desarrollo de este proyecto, es esencial considerar que el TFG tiene asignados 12 créditos ECTS, lo que equivale a aproximadamente 300 horas de trabajo. Dada la distribución temporal del segundo cuatrimestre, con unas 20 semanas disponibles, se estima que se requerirá dedicar al

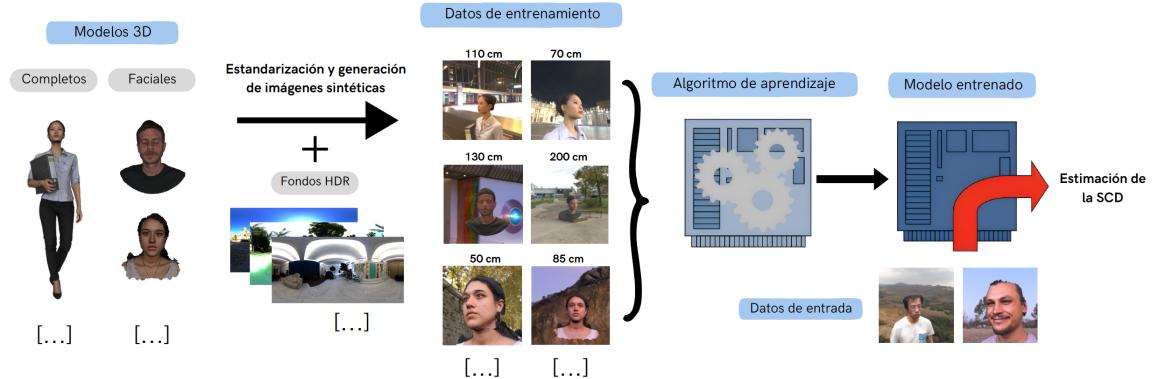


Figura 1.4: Diagrama del proceso de estimación automática de la SCD en este proyecto.

TFG unas 20 horas semanales, equivalentes a 4 horas diarias durante 5 días a la semana. Se reservan así 4 semanas como margen para posibles retrasos o imprevistos que puedan surgir durante el desarrollo del proyecto.

En cuanto a la metodología de desarrollo, se ha optado por seguir un enfoque basado en el ciclo de vida en cascada [34], aunque con una variante que permite retroalimentación. Aunque el proyecto presenta requisitos y objetivos claros, se reconoce la posibilidad de ajustes menores durante su desarrollo, especialmente a medida que se obtenga más información sobre el problema y los métodos. Esta flexibilidad se considera crucial para adaptarse a posibles cambios en el contexto o los requisitos del proyecto.

A continuación se describen las fases del ciclo de vida del proyecto:

- **Análisis de Requisitos:** Consiste en las reuniones iniciales con los clientes, en este caso los directores del TFG. Se realiza un análisis del problema y un estudio detallado de la bibliografía existente.
- **Diseño:** Consiste en la exploración y selección de los métodos apropiados así como de los conjuntos de datos basados en el análisis previo, tanto para la resolución como para la validación de la solución propuesta. Además, se llevarán a cabo pruebas preliminares y se elaborará el diseño del software experimental.
- **Implementación:** Consiste en la adaptación del código de los modelos investigados, la implementación de nuevas funcionalidades y la generación de un conjunto de datos sintético junto con su posterior preprocesado.
- **Pruebas:** Consiste en la realización de diversos experimentos para validar el funcionamiento del software desarrollado, utilizando los modelos

y datos previamente definidos.

Tarea	Semanas - Horas	Febrero	Marzo	Abril	Mayo	Junio
		5 12 19 26	4 11 18 25	1 8 15 22 29	6 13 20 27	3 10 17
Análisis de Requisitos	3 - 60					
Diseño	3 - 60					
Implementación	5 - 90					
Pruebas	5 - 90					

Tabla 1.1: Planificación inicial del proyecto

La planificación inicial se detalla en la tabla 1.1, sin embargo, experimentó varios retrasos, principalmente debido a que el autor también estaba trabajando en un proyecto en colaboración con la Universidad de Granada. Por otro lado, la obtención de los conjuntos de datos 3D no resultó ser una tarea sencilla, debido a su escasa disponibilidad y a los permisos necesarios para acceder a ellos. Además, el preprocesamiento de los datos 3D consumió más tiempo del previsto, ya que, aunque se automatizó en cierta medida, requirió ajustes manuales significativos. Estos contratiempos, junto con el aprendizaje de nuevas librerías por parte del autor, resultaron en modificaciones en la planificación original, tal como se ejemplifica en la tabla 1.2.

Para estimar los costos, comenzamos considerando un salario de 30 euros por hora para un responsable I+D en una empresa tecnológica o para un investigador senior. Además de esto, se contemplan los gastos asociados a los materiales, como el costo del portátil utilizado en el desarrollo del TFG y el uso de un servidor GPU de alto rendimiento. Estos costes se desglosan detalladamente en la Tabla 1.3.

En relación al servidor GPU, su valoración se estima en x euros.

Tarea	Semanas - Horas	Febrero	Marzo	Abril	Mayo	Junio
		5 12 19 26	4 11 18 25	1 8 15 22 29	6 13 20 27	3 10 17
Análisis de Requisitos	3 - 60					
Diseño	4 - 70					
Implementación	6 - 100					
Pruebas	6 - 100					

Tabla 1.2: Planificación final del proyecto

Fecha de inicio	05/02/2024
Fecha de fin	
Duración	

Item	Costo
Salario	
Portátil de Gama Alta	2.600 euros
Servidor GPU	
Total	

Tabla 1.3: Estimación del coste del proyecto

Capítulo 2

Fundamentos teóricos

2.1. Aprendizaje automático

El aprendizaje automático (*Machine Learning*, ML) [23, 7] es una rama de la inteligencia artificial y de las ciencias de la computación centrada en el uso de datos y algoritmos para imitar la forma en la que los humanos aprenden, detectando patrones o regularidades para realizar predicciones.

Existen 3 tipos de aprendizaje dentro del ML [1, 37]:

El **aprendizaje supervisado** consiste en entrenar un modelo con datos que tienen etiquetas conocidas, lo que indica la categoría a la que pertenece cada dato. Por ejemplo, si los datos de entrada son imágenes de animales, las etiquetas podrían ser 'perro' o 'gato'. A partir de estos datos etiquetados, el modelo aprende a predecir la etiqueta de nuevos datos. Es el tipo de aprendizaje más utilizado y los datos vienen ya 'preparados' para su uso. Es el tipo de aprendizaje que utilizaremos en este TFG.

En el **aprendizaje no supervisado**, el modelo analiza los datos de entrada sin etiquetas, buscando patrones y estructuras inherentes a los datos. El agrupamiento es una técnica común en este tipo de aprendizaje, ya que identifica posibles grupos dentro de los datos. Este enfoque suele requerir un gran volumen de datos para ser efectivo.

Por otro lado, en el **aprendizaje por refuerzo**, el modelo aprende a través de recompensas o penalizaciones en función de las acciones que realiza. El objetivo del agente es maximizar las recompensas a largo plazo, lo que lo hace especialmente útil en la enseñanza de estrategias en juegos y otras interacciones dinámicas.

2.2. Aprendizaje profundo

2.2.1. Redes neuronales artificiales

Las redes neuronales artificiales (*Artificial Neural Networks*, ANN) [17, 20, 5] son redes computacionales que intentan, a groso modo, simular el proceso de decisión de las neuronas del sistema nervioso central de animales y humano. Las ANN poseen unidades de procesamiento de información llamadas neuronas, las cuales están conectadas entre sí. La estructura básica de una ANN se compone de (ver Figura 2.1):

- Una capa de entrada, que tendrá tantos *inputs* como características o variables tenga el problema
- Una o varias capas ocultas, compuestas por neuronas. El número de capas ocultas define la profundidad de la red neuronal.
- Una capa de salida, la cual representa el valor o valores predichos

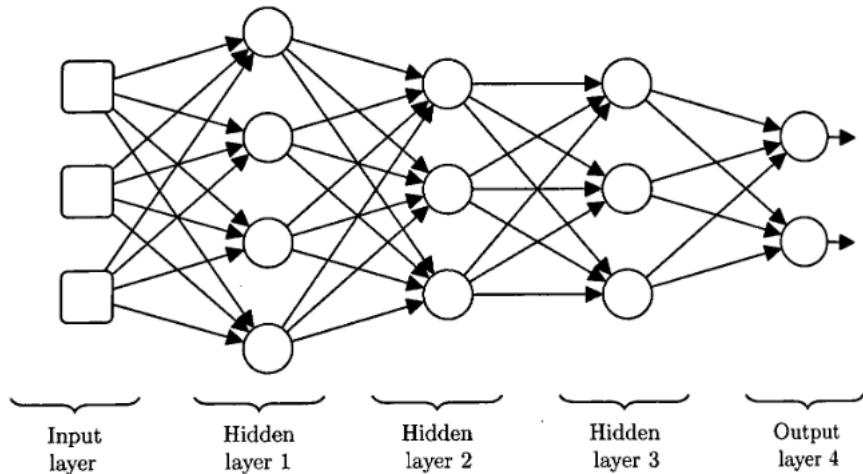


Figura 2.1: Esquema de una red neuronal [24].

Las neuronas son la unidad fundamental de cómputo, tienen varios valores de entrada y un valor de salida que se conecta con las neuronas de la siguiente capa. Los elementos básicos del modelo neuronal son (ver Figura 2.2):

- Un conjunto de conexiones con las señales de entrada. Cada conexión tiene su propio peso/fuerza.
- Una función de suma de las señales de entrada, ponderadas cada una con su peso. Estas operaciones constituyen una combinación lineal.

- Una función de activación, para limitar la amplitud de la salida de la neurona. Normalmente, el rango de salida está en el intervalo [0,1], o alternativamente en [-1,1]. Existen muchos tipos de funciones de activación pero, se suelen utilizar cuatro: la función signo, la función logística, la función arco-tangente o la función ReLU (*Rectified Linear Unit*).

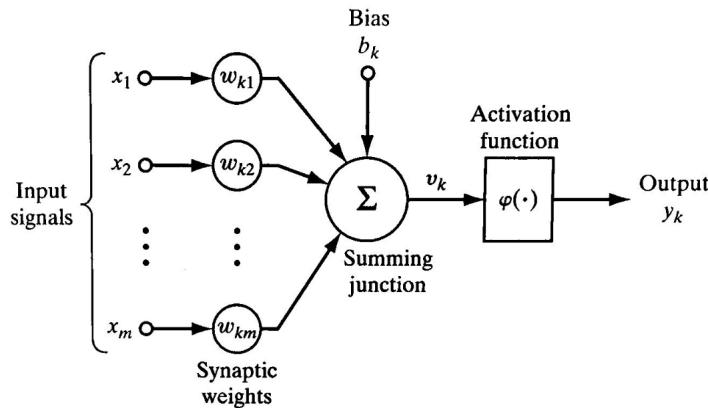


Figura 2.2: Modelo neuronal para una neurona k [20].

En términos matemáticos, podemos describir la salida de una neurona como:

$$y = \phi\left(\sum_{j=1}^m w_j x_j + b\right) \quad (2.1)$$

siendo ϕ la función de activación, m el número de señales de entrada, w_j el peso de cada entrada x_j , y b el sesgo.

El algoritmo de aprendizaje de la red neuronal consiste en ir modificando los pesos y el sesgo, iterativamente, hasta alcanzar el resultado deseado. Este proceso iterativo se conoce como entrenamiento, y permite, a través de las modificaciones de los pesos, reconocer y extraer las características más relevantes de los datos.

El objetivo del entrenamiento es minimizar el error de predicción de la salida de la red neuronal, para ello, se define una función de pérdida. Existen numerosas funciones de pérdida, algunas de las más conocidas son: el error cuadrático medio (MSE), el error absoluto medio (MAE) o la entropía cruzada. La información de la función de pérdida se transmite desde la salida a la capa inicial, con el fin de modificar adecuadamente los pesos para generar una mejor estimación de la predicción.

El sobreentrenamiento es un factor importante a evitar. Sobreentrenar el modelo de aprendizaje, significa, ajustarlo demasiado al conjunto de datos de entrenamiento, de manera que, al recibir nuevos datos no utilizados para entrenar, se estime un mal resultado debido a la poca capacidad de generalización ante nuevos datos.

2.2.2. Redes neuronales convolucionales

Las redes neuronales convolucionales (*Convolutional Neural Networks*, CNN) [26, 27, 51] son un tipo de red neuronal profunda que trabaja con patrones de cuadrícula, como pueden ser imágenes (ver Figura 2.3).

En estas redes neuronales, las capas convolucionales desempeñan un papel fundamental, y a menudo se complementan con capas de *pooling*. Dichas capas se encuentran en la primera parte de la red y son las encargadas de extraer las características relevantes de la entrada. Esto posibilita la automatización del proceso de extracción de características, mejorando simultáneamente tanto el tiempo como el rendimiento.

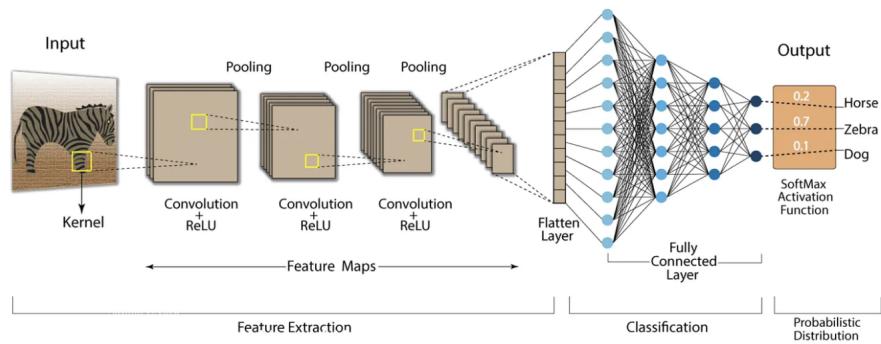


Figura 2.3: Ejemplo de red neuronal convolucional [38].

A continuación, se describen los posibles tipos de capas presentes en una CNN [46, 50]:

Capa de convolución

La capa convolucional es un componente fundamental de las CNN, utilizada para la extracción de características de una imagen o un conjunto de imágenes. Esta capa aplica una operación lineal especializada conocida como convolución, que consiste en aplicar un filtro o *kernel* a la imagen de entrada. El *kernel* es una matriz que se desliza a lo largo de la imagen, multiplicando sus valores con los píxeles correspondientes y sumándolos para producir un único valor en la imagen de salida. Este proceso se repite en

todas las posiciones de la imagen dando como resultado una nueva matriz denominada mapa de características (ver Figura 2.4).

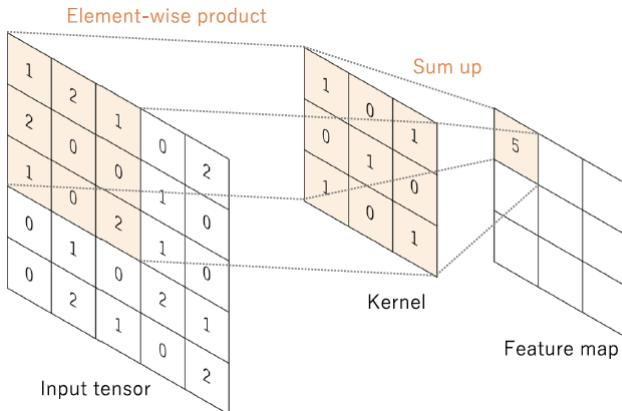


Figura 2.4: Ejemplo de operación de convolución en CNN [50].

Los pesos de los filtros se aprenden durante el proceso de entrenamiento de la red neuronal. Cada *kernel* tiene sus propios pesos que se ajustan iterativamente durante el entrenamiento para minimizar la función de pérdida y mejorar el rendimiento del modelo.

La característica clave de la operación de convolución es el *weight sharing*, que implica compartir los mismos *kernels* en toda la imagen. Esto permite que la red detecte patrones locales independientemente de su ubicación en la imagen. Además, contribuye a aprender jerarquías de características espaciales, lo que permite capturar una amplia gama de características en varios niveles de abstracción. Este enfoque también aumenta la eficiencia del modelo al reducir la cantidad de parámetros que necesita aprender en comparación con las redes totalmente conectadas.

Por otro lado, es importante la configuración de los hiperparámetros de cada capa convolucional, estos se definen antes de iniciar el entrenamiento de la red neuronal y afectan al comportamiento de la misma. Los más comunes son:

- Tamaño del *kernel*: se refiere a las dimensiones del filtro que se aplica a la imagen de entrada. Los tamaños comunes son 3x3, 5x5 o 7x7.
- Número de *kernels*: indica cuántos filtros se aplicarán a la imagen de entrada para extraer diferentes características. Cuantos más *kernels* se utilicen, mayor será la profundidad de los mapas de características de salida.
- *Padding*: esta técnica consiste en añadir píxeles alrededor de la imagen de entrada tras el proceso de convolución. Su propósito es mantener

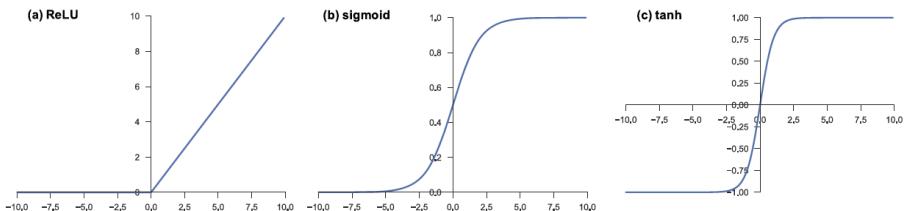
el tamaño de la salida, ya que al aplicar la convolución, las dimensiones del mapa de características se reducen con respecto a la imagen original.

- *Stride*: es el número de píxeles que se desplaza el *kernel* en cada paso durante la convolución. Un mayor *stride* reduce el tamaño del mapa de características y la cantidad de operaciones necesarias.

Sin embargo, es importante tener en cuenta que la operación de convolución por sí sola es lineal y puede no ser suficiente para aprender patrones complejos. En este contexto, entra en juego la capa de activación, que introduce no linealidades en la red y potencia su capacidad para capturar relaciones más complejas entre las características extraídas.

Capa de activación

La capa de activación en una CNN sigue a la capa convolucional y se encarga de introducir no linealidades en el modelo mediante una función de activación. Esta función aumenta la capacidad de la red para aprender relaciones no lineales en los datos, lo que es fundamental para capturar patrones más complejos. Algunas de las funciones de activación comunes utilizadas son la función ReLU, la función sigmoid y la función tangente hiperbólica (ver Figura 2.5).



Capa de *pooling*

La capa de *pooling* también es específica de las CNN y se encarga de reducir la dimensionalidad de las características conservando la información más relevante.

Esta capa resume la información en regiones locales mediante una operación de *downsampling* en las características de entrada. Al reducir la dimensionalidad de las características, la capa de *pooling* disminuye el número de parámetros aprendibles en la red, lo que puede ayudar a prevenir el sobreajuste y mejorar la eficiencia computacional del modelo. Además, esta

capa también ayuda a introducir invariancia a pequeñas traslaciones y distorsiones en los datos de entrada, permitiendo a la red reconocer patrones incluso si están ligeramente desplazados en la imagen.

Los dos tipos más comunes de *pooling* son el *max pooling*, que selecciona el valor máximo de una región local en las características de entrada, y el *average pooling*, que calcula el promedio de los valores en una región local (ver Figura 2.6).

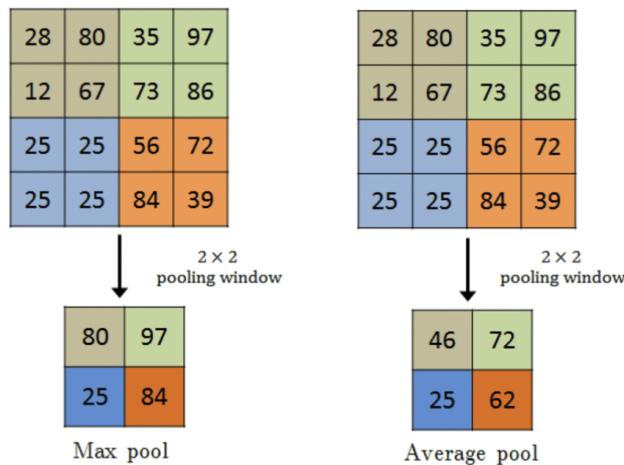


Figura 2.6: Tipos de *pooling* comúnmente utilizados en CNN [50].

Los hiperparámetros de la capa de *pooling* incluyen el tamaño del filtro, el *stride* y el tipo de *padding*. Estos hiperparámetros afectan la forma en que se realiza el *downsampling* en las características.

Capa totalmente conectada

La capa totalmente conectada sigue a las capas de convolución y *pooling*. En esta capa, las características extraídas por las capas anteriores se transforman en un formato unidimensional (vector) antes de conectarse a una o más capas totalmente conectadas, también conocidas como capas densas.

Esta capa, al igual que en las redes neuronales clásicas, tiene la responsabilidad de combinar y procesar las características extraídas para producir la salida final de la red.

Cada neurona en una capa totalmente conectada está conectada a todas las neuronas de la capa anterior a través de pesos aprendibles. Estos pesos determinan la contribución de cada neurona de entrada a la neurona de salida correspondiente en la capa totalmente conectada. Durante el entrenamiento, estos pesos se ajustan mediante algoritmos de optimización como *backpropagation* y descenso de gradiente para minimizar la diferencia entre

las salidas predichas y las etiquetas reales.

Es importante destacar que la capa totalmente conectada suele estar seguida por una función de activación no lineal, como ReLU, para introducir no linealidades en el modelo y permitir la representación de patrones complejos en los datos. Además, la última capa de activación de la CNN, generalmente se selecciona según la naturaleza de la tarea que se está abordando.

2.2.3. Transferencia de aprendizaje

La transferencia de aprendizaje [21], también conocida como *Transfer Learning* (TL), es una técnica fundamental en el campo del aprendizaje automático. Consiste en aprovechar el conocimiento adquirido al resolver un problema para mejorar el rendimiento en otro problema relacionado. En lugar de comenzar desde cero al entrenar un modelo para una tarea específica, el TL utiliza el aprendizaje previo en tareas similares, obteniendo múltiples beneficios, como una mayor eficiencia en el entrenamiento de modelos, una mejor generalización con conjuntos de datos limitados y una aceleración en el desarrollo de modelos.

En las arquitecturas convolucionales, la forma más común de llevar a cabo la transferencia de aprendizaje es mediante el *fine-tuning* [46], que implica utilizar pesos pre-entrenados, congelar todas las capas de la red excepto las superiores, y ajustar estas últimas para adaptarlas a nuestro problema específico, de manera que el entrenamiento se realice únicamente en esas capas superiores. Este enfoque aprovecha la capacidad de los modelos pre-entrenados para capturar características generales de los datos, lo cual es especialmente útil cuando se dispone de conjuntos de datos pequeños o limitados. Además, al congelar las capas iniciales se evita la pérdida de información importante aprendida durante el pre-entrenamiento, mientras que el *fine-tuning* en las capas superiores permite adaptar el modelo a la nueva tarea específica.

En este contexto, es común utilizar los pesos pre-entrenados en el conjunto de datos de ImageNet [22] debido a su gran tamaño, diversidad, representatividad y disponibilidad.

2.2.4. Regularización

Tanto en las redes neuronales clásicas como en las convolucionales, el sobreajuste a los datos de entrenamiento es una problema importante (ver Figura 2.7). Aunque la solución óptima sería adquirir más datos para el entrenamiento, esta opción no siempre está disponible. Por tanto, se recurre a técnicas de regularización para mitigar este problema. Entre las más destacadas se encuentran:

- *Dropout*: es una técnica de regularización donde se establecen aleatoriamente ciertas activaciones a 0 durante el entrenamiento, de modo que el modelo se vuelve menos sensible a pesos específicos en la red.
- *Weight decay*: también conocido como regularización L2, reduce el sobreajuste penalizando los pesos del modelo para que tomen solo valores pequeños.
- *Batch normalization*: es un tipo de capa suplementaria que normaliza adaptativamente los valores de entrada de la siguiente capa, mitigando el riesgo de sobreajuste, así como mejorando el flujo de gradiente a través de la red, permitiendo tasas de aprendizaje más altas y reduciendo la dependencia de la inicialización.
- *Data augmentation*: es un proceso de modificación de los datos de entrenamiento a través de transformaciones aleatorias, como volteo, traslación, recorte, rotación y borrado aleatorio, para que el modelo no vea exactamente las mismas entradas durante las iteraciones de entrenamiento. Esta técnica, además de reducir el sobreajuste, permite una mejor generalización del modelo.
- Elección del modelo: un modelo de una alta complejidad puede provocar sobreajuste ya que tiene la capacidad de ajustarse mucho mejor a los datos de entrenamiento. Es fundamental encontrar un modelo que tenga un equilibrio entre complejidad y generalización, es decir, que sea lo suficientemente complejo para captar las características importantes pero que a la vez sea capaz de generalizar sin sobreajustarse demasiado a los datos.

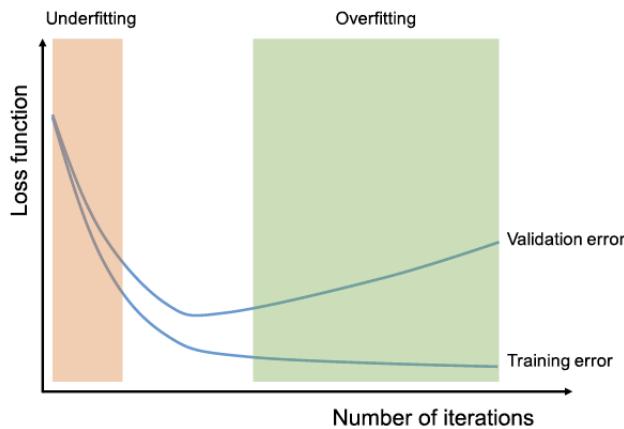


Figura 2.7: Zona de infraajuste y sobreajuste durante el entrenamiento [50].

A pesar de las técnicas anteriores, persiste la preocupación por el sobreajuste al conjunto de validación en lugar del conjunto de entrenamiento,

principalmente debido a la filtración de información durante el ajuste fino de hiperparámetros y el proceso de selección del modelo. Por tanto, es importante evaluar el rendimiento del modelo final en un conjunto de prueba separado, preferiblemente no visto previamente. Esto es fundamental para validar la capacidad de generalización del modelo y garantizar su fiabilidad.

2.3. Parámetros de la cámara y perspectiva

Dado que el conjunto de datos utilizado en este TFG son imágenes creadas sintéticamente, es importante conocer los parámetros de la cámara que influyen a la hora de sacar las fotografías.

Longitud focal

La longitud focal [18] mide la distancia, en milímetros, entre el 'punto nodal' (punto donde la luz converge en una lente) y el sensor de la cámara (ver Figura 2.8).

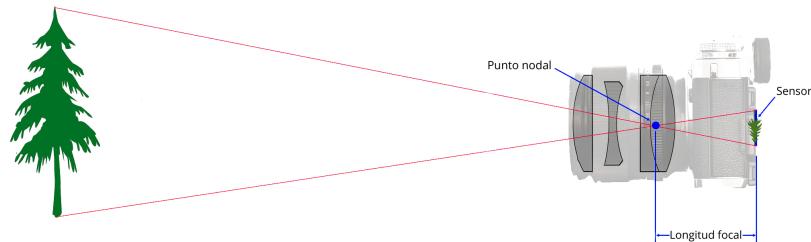


Figura 2.8: Relación entre punto nodal y longitud focal [18]

La longitud focal es importante porque se relaciona con el campo de visión de una lente, es decir, cuánta escena se captura. También explica qué tan grande o pequeño aparecerá un sujeto en la fotografía.

Valores más altos de longitud focal (como 500 mm) se ven más 'cerca', mientras que valores más bajos (como 20 mm) están más 'lejos'.

Sensor de la cámara

El sensor de la cámara [2, 29] es el componente que captura la luz y la convierte en una imagen digital. El tamaño del sensor afecta a la calidad de la imagen y a la cantidad de luz que puede capturar. El tamaño estándar es de 36mm x 24mm, también conocido como *full frame* o 35mm.

Otra forma equivalente de referirnos al tamaño del sensor es mediante

el factor de recorte, que se define como la relación existente entre el tamaño de un sensor de 35mm y el sensor de nuestra cámara (ver Figura 2.9).



Figura 2.9: Tamaños del sensor expresados según el factor de recorte [6]

Uno de los aspectos más importantes del factor de recorte es su influencia en la longitud focal, lo que nos lleva a hablar de 'longitud focal equivalente'. Por ejemplo, al tener una focal de 300 mm en un sensor con factor de recorte 1.6, estaríamos obteniendo un efecto equivalente al de una focal de 480 mm (300 mm x 1.6) en un sensor *full frame* (factor de recorte 1).

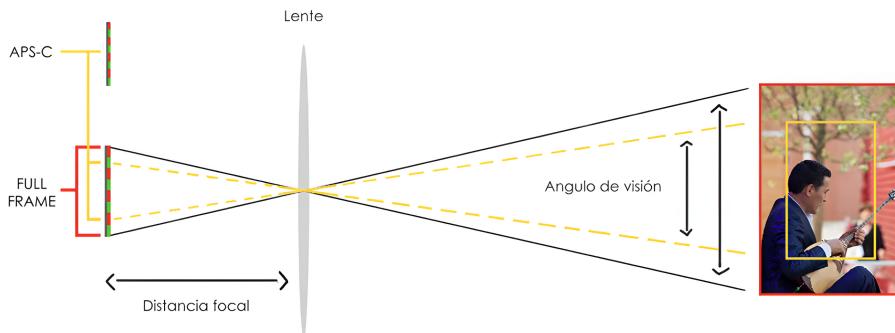


Figura 2.10: Ejemplo de longitud focal equivalente según el tamaño del sensor [43].

Distancia cámara-sujeto

La distancia cámara-sujeto se define como la separación física entre la cámara y el sujeto que está siendo fotografiado (ver Figura 2.11). Modifi-

ficar esta distancia provoca variaciones en la apariencia visual del rostro en la fotografía obtenida [32]. Este fenómeno se conoce como distorsión de perspectiva.



Figura 2.11: Distancia desde la cámara al sujeto.

Distorsión de perspectiva

La distorsión de perspectiva [45, 13] es la transformación que sufre un objeto y su entorno debido a la proximidad del mismo respecto al objetivo (ver Figura 2.12). En el caso de las fotografías faciales, cuanto menor es la distancia cámara-sujeto, mayor es la distorsión de perspectiva que afecta a la persona fotografiada. Esto afecta a rasgos de la cara que pueden aparecer más grandes, como la nariz, o más pequeños, como las orejas, de lo que realmente son.

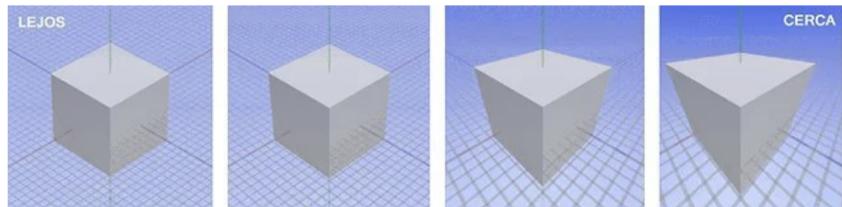


Figura 2.12: Efecto de la distorsión conforme se acerca la cámara al objeto [13].

Uno de los malentendidos comunes en fotografía es la creencia de que la longitud focal distorsiona los rasgos faciales, sin embargo, la longitud focal no tiene nada que ver con la distorsión del rostro de un sujeto, siendo esta únicamente provocada por la distancia de la cámara al sujeto [30].

Capítulo 3

Estado del Arte

En el campo del aprendizaje automático, el tema de la estimación de la distancia en fotografías faciales ha ganado recientemente mucha atención. Se puede observar en la Figura 3.1 la cantidad de publicaciones existentes en la base de datos Scopus¹ que hacen referencia a la estimación de la SCD. Hay 441 publicaciones registradas desde 1992.

El número de publicaciones relacionadas con este tema, va aumentando a lo largo del tiempo, llegando a obtener un mayor número de publicaciones en 2020. Pese al aumento de publicaciones en este ámbito, es a partir del 2015 cuando se empiezan a aplicar las técnicas de aprendizaje profundo. Este aumento está relacionado con los avances tecnológicos que permiten aplicar nuevas técnicas y conocimientos.

3.1. Primeros enfoques

El primer método utilizado para abordar la estimación métrica de la SCD fue propuesto por Flores et al. [16], quienes proponen utilizar un conjunto de puntos de referencia faciales para calcular la distancia y la posición respecto a la cámara, en un rango que va desde los 10 cm hasta los 3 m. Este método consiste en tomar una imagen 2D de una cara desconocida, identificar sus puntos de referencia faciales y compararlos con los puntos obtenidos de modelos faciales 3D conocidos. Luego, empleando el algoritmo EPnP [28], se determina la distancia entre la cámara y el sujeto. Esta técnica asume que los puntos de referencia no varían significativamente entre individuos, sino que tienden a agruparse en *clusters*.

Sin embargo, este primer enfoque presenta algunas limitaciones, como la dependencia de conjuntos de datos en 3D (los cuales no siempre están

¹Las búsquedas se pueden consultar en el apéndice

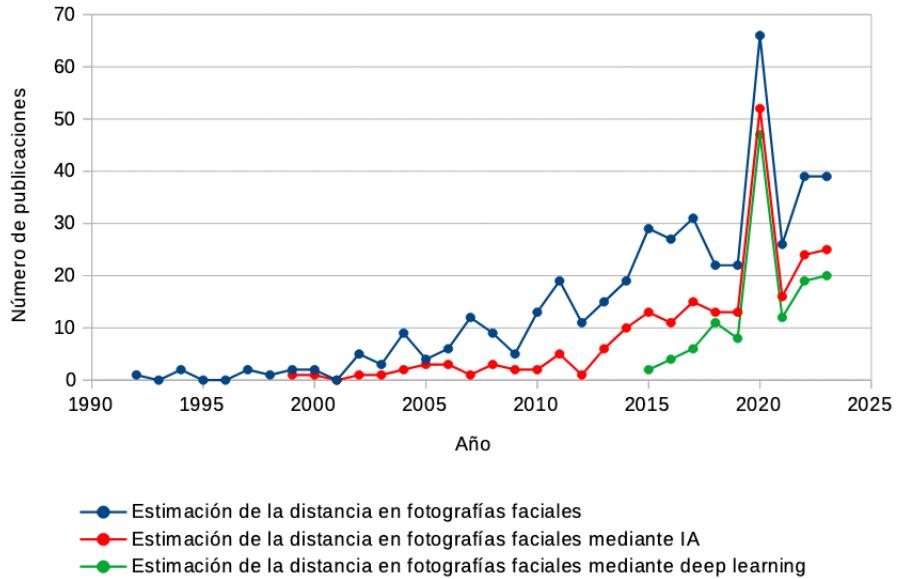


Figura 3.1: Número de publicaciones, en Scopus, relacionadas con la estimación de la distancia en fotografías faciales en función del año de publicación.

disponibles), la mezcla de diferentes longitudes focales en un mismo conjunto de datos y la necesidad de reconocimiento manual de los puntos de referencia faciales.

Posteriormente, Burgos-Artizzu et al. [10] introducen un método innovador que elimina la necesidad de anotación manual de los puntos de referencia de la imagen. En su lugar, estos puntos se estiman automáticamente mediante un enfoque de regresión conocido como *Robust Cascaded Pose Regression* (RCPR) [8]. Una vez que se han identificado los puntos de referencia faciales, se emplea un modelo automático de regresión para predecir la distancia entre la cámara y el sujeto en función de la posición relativa de estos puntos. Este regresor fue entrenado utilizando el conjunto de datos Caltech Multi-Distance Portraits (CMDP) [9], que consta de 53 retratos individuales tomados desde 7 distancias diferentes, que van desde 60 cm hasta 480 cm. Todos estos retratos están anotados manualmente con 55 marcas faciales.

Este método, sigue teniendo algunas limitaciones como el recorte de las imágenes (pérdida de resolución) o la única vista frontal.

Además de los métodos previamente citados, se han desarrollado otras técnicas para estimar la SCD basadas en características anatómicas como el tamaño facial [39], la separación entre los ojos [35], o una combinación de ambos factores [25].

3.2. MediaPipe Iris

MediaPipe Iris² es un modelo de aprendizaje automático desarrollado por investigadores de Google. Este modelo tiene la capacidad de rastrear puntos de referencia como el iris, la pupila y los contornos del ojo en tiempo real, utilizando únicamente una cámara RGB estándar y sin necesidad de utilizar ningún hardware especializado. Mediante el seguimiento de los puntos de referencia del iris, este modelo puede determinar la distancia métrica entre el sujeto y la cámara.

El modelo se basa en el diámetro horizontal del iris del ojo humano, el cual se mantiene relativamente constante en un rango de 11.7 ± 0.5 mm en una amplia población. Esta característica, combinada con argumentos geométricos simples, permite al modelo estimar la distancia SCD.

Sin embargo, es importante destacar que este modelo presenta ciertas condiciones y limitaciones. Es útil únicamente en situaciones donde existan datos EXIF disponibles, se capturen imágenes frontales donde el iris sea visible, y los individuos se encuentren a una distancia de menos de 2 metros de la posición de la cámara.

3.3. PerspectiveX

PerspectiveX, desarrollado por Stephan et al. [42], es un método para la estimación de la SCD en imágenes faciales, diseñado para mejorar el proceso de superposición craneofacial.

Este método se basa en la localización de una característica anatómica específica, la longitud de la fisura palpebral, definida por dos puntos de referencia fácilmente identificables.

Esta elección se justifica por varios aspectos: su clara visibilidad frontal, incluso cuando la cabeza experimenta un ligero giro hacia el lado más cercano a la cámara; su definición precisa, que garantiza una correcta medición; su mínima variabilidad, atribuible a restricciones evolutivas; su notable tamaño facial relativo, lo cual minimiza la probabilidad de errores en comparación con características más pequeñas, como el diámetro del iris; y su distribución normal, que contribuye a reducir el margen de error en las predicciones.

Además de la fisura palpebral, PerspectiveX requiere conocer el tipo de cámara, necesario para obtener las especificaciones de píxeles, así como la longitud focal de las lentes. Ambos datos pueden extraerse de las imágenes electrónicas mediante lectores EXIF disponibles en línea.

Finalmente, la estimación del SCD se realiza mediante la siguiente fórmu-

²<https://blog.research.google/2020/08/mediapipe-iris-real-time-iris-tracking.html>

la:

$$SCD = f \left(1 + \frac{A}{x \cdot y} \right) \quad (3.1)$$

donde: f , es la longitud focal de las lentes (mm); A , es la longitud real de la fisura palpebral (mm); x , es la longitud de la fisura palpebral en la foto (píxeles); y , son las especificaciones del tamaño del píxel del receptor de imagen (mm)

Dado que la longitud real de la fisura palpebral puede no estar disponible, se recurre al promedio de un grupo demográfico homogéneo en términos de sexo y edad, ya que se sabe que esta medida varía mínimamente debido a restricciones evolutivas.

Aunque PerspectiveX ofrece una estimación precisa de la SCD para una longitud focal conocida, presenta ciertas limitaciones, como la necesidad de intervención manual para marcar los puntos de referencia faciales y la falta de consideración de las rotaciones de cabeza superiores a 30°.

3.4. FacialSCDnet

FacialSCDnet, presentado por Bermejo et al. [4], es un método que estima la SCD directamente a partir de fotografías mediante el empleo de técnicas de aprendizaje profundo. La utilización de una arquitectura de redes neuronales profundas elimina una restricción crucial: la necesidad de detectar una característica anatómica específica para guiar el proceso de estimación. Esta capacidad permite que el método sea eficaz en la estimación de la SCD en cualquier posición de la cabeza, desde la frontal hasta el perfil lateral.

Para entrenar el modelo, se empleó un conjunto de datos compuesto por dos colecciones:

- Conjunto sintético: se generaron imágenes sintéticas 2D a partir de los modelos 3D de la base de datos Stirling ESRC 3D Face ³. En particular, se utilizaron 315 modelos faciales de 54 individuos diferentes para generar aproximadamente 150.000 fotografías sintéticas.
- Conjunto de fotografías digitales: se adquirieron fotografías de 28 individuos siguiendo un protocolo de adquisición específico. Se consideraron 4 longitudes focales diferentes (27 mm, 35 mm, 55 mm, 85 mm) en formato full frame y se capturaron 12 distancias diferentes de la

³Stirling ESRC 3D Face: <https://pics.stir.ac.uk/ESRC/index.htm>

cámara al sujeto, que oscilaron desde 50 cm hasta 6 m. Además, se fotografiaron 7 posiciones distintas de la cabeza, desde el perfil izquierdo hasta el perfil derecho, con intervalos de rotación de 30º.

La función de pérdida empleada en el modelo se basa en el error absoluto medio de la distorsión facial relativa, calculada mediante la fórmula:

$$Distorsion = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (3.2)$$

donde y_i son los valores reales (etiquetas) de la distorsión facial, y x_i son los valores predichos de la distorsión facial, calculados a partir del factor de distorsión (D_f):

$$D_f = \frac{1}{1 + \frac{SCD}{d}} \quad (3.3)$$

En la ecuación 3.3, $d = 12.6572$ cm corresponde a un valor derivado de cálculos geométricos [42] para obtener experimentalmente el factor de distorsión de una cabeza humana de tamaño promedio, según la SCD de la fotografía.

FacialSCDnet consta de 4 modelos de aprendizaje profundo, cada uno asociado a una longitud focal utilizada en el conjunto de datos. La estructura de cada CNN se basa en la arquitectura VGG-16, cuyos pesos se inicializan con los pre-entrenados en ImageNet ⁴. Para adaptar la arquitectura al problema de estimación de la SCD, se conservan los 5 bloques convolucionales, se elimina la cabeza superior de la red y se añaden 2 capas totalmente conectadas que se entranan desde cero. Finalmente, la última capa del modelo consiste en una activación lineal que realiza la tarea de regresión.

El proceso de entrenamiento de los modelos constó de dos fases. En primer lugar, los modelos se entrenaron con el conjunto de datos sintético para aprender las relaciones entre la SCD y las características faciales. Posteriormente, se realizó un ajuste fino utilizando el conjunto de datos reales.

Los resultados obtenidos indican que las cuatro redes de FacialSCDnet son capaces de predecir con precisión la SCD, con errores promedio por debajo de 5 cm (MAE) o 3% (MRE). Esta precisión en la predicción de la distancia métrica se traduce en un error promedio del 0.2% al considerar la métrica de distorsión facial relativa. Estos resultados muestran que FacialSCDnet logra una estimación precisa de la SCD en fotografías faciales, superando a otros métodos existentes y demostrando su robustez y eficacia en diversas situaciones y condiciones.

⁴ImageNet: <https://www.image-net.org/>

Capítulo 4

Materiales y métodos

4.1. Materiales

En este TFG se generará un conjunto de datos de imágenes sintéticas fotorrealistas a partir de varios modelos 3D seleccionados.

4.1.1. Modelos 3D

Se llevó a cabo un análisis exhaustivo de las bases de datos disponibles de modelos 3D de personas. Tras la búsqueda, se utilizaron diversos conjuntos de datos públicos con el objetivo de crear un conjunto de datos unificado, realista y diverso. En este conjunto, se incluyeron tanto modelos faciales como modelos de cuerpo completo, todos ellos con sus correspondientes texturas.

Modelos faciales

Se han seleccionado los siguientes conjuntos de datos: HeadSpace [12], H3DS-net [36] y DI4D_UGR_ANON¹.

El conjunto de datos de Headspace [12] es un conjunto de imágenes en 3D de la cabeza humana, que consta de 1519 sujetos que llevan gorros de látex ajustados para reducir el efecto de los peinados. Las ventajas de este conjunto son que tienen muy buena resolución y además incluye metadatos útiles para seleccionar un subconjunto de datos adecuado.

H3DS-net [36] contiene escaneos texturizados en 3D de la cabeza completa con una alta resolución. Este conjunto comprende un total de 23 modelos, todos ellos con los ojos cerrados, lo que añade una variabilidad adicional.

¹Conjunto de datos proporcionado por el tutor

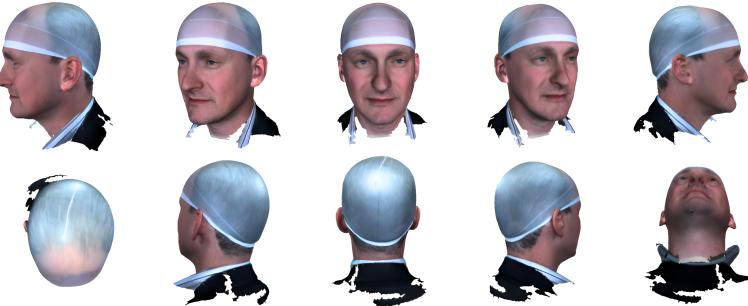


Figura 4.1: Ejemplos de modelos en HeadSpace 3D.



Figura 4.2: Ejemplos de modelos en H3DS-net.

DI4D_UGR_ANON es un conjunto de datos creado por la Universidad de Granada...

Si bien se investigaron otros conjuntos de datos como FaceVerse [47] o CASIA², estos fueron descartados debido a problemas como la baja calidad, formatos incompatibles y escalas no reales.

Modelos de cuerpo entero

Se han seleccionado los siguientes conjuntos de datos: HuMMan [11], People Snapshot [3] y Render People³

HuMMan [11] es un conjunto de datos 3D que consta de 153 sujetos humanos, con una amplia cobertura de sexos biológicos, edades, formas del cuerpo y poblaciones. Cada sujeto contiene 2-3 secuencias, y cada secuencia contiene aproximadamente 20 modelos.

People Snapshot [3] contiene 24 sujetos 3D en diferentes situaciones tales como casual, deporte y actividades al aire libre.

RenderPeople es una empresa privada especializada en la creación de modelos humanos en 3D. Existen modelos disponibles para comprar, pero dado su costo, hemos decidido emplear exclusivamente aquellos gratuitos que están disponibles. Aunque únicamente hay 2, estos son bastante realistas y

²<http://biometrics.idealtest.org/>

³<https://renderpeople.com/es/>



Figura 4.3: Ejemplos de modelos en HuMMan.



Figura 4.4: Ejemplos de modelos en People Snapshot.

presentan situaciones que no se contemplan en los anteriores modelos.

Selección del subconjunto de modelos

Ante la gran cantidad de modelos disponibles, surge la necesidad de elegir un subconjunto de modelos adecuado, combinando modelos faciales y de cuerpo completo. En total, se han seleccionado 176 modelos 3D, de los cuales 106 son modelos faciales y 70 son modelos de cuerpo completo.

En primer lugar, de H3DS-net se han seleccionado los 23 modelos, compuestos por 13 masculinos y 10 femeninos, todos ellos de ascendencia europea. De HeadSpace, se han elegido 73 modelos, de los cuales 36 son masculinos y 37 femeninos. Esta selección incluye modelos de ascendencia europea, asiática, africana o combinada. Además, todos los modelos son mayores de 14 años. Respecto a DI4D_UGR_ANON, se han seleccionado 10 modelos, compuestos por 6 femeninos y 4 masculinos, todos ellos de ascendencia europea.

Además, se han seleccionado los 2 modelos provenientes de RenderPeople, uno masculino y otro femenino, ambos de ascendencia europea. De People Snapshot se han seleccionado los 24 modelos, de los cuales 16 son mas-



Figura 4.5: Ejemplos de modelos en Render People.

culinos y 8 femeninos, todos de ascendencia europea. Finalmente, se han seleccionado 44 modelos de HuMMAn, distribuidos en 23 modelos masculinos y 21 modelos femeninos. En este conjunto, 28 modelos son de ascendencia asiática, 11 son de ascendencia africana y 5 son de ascendencia europea.

Estas selecciones se realizaron con el objetivo de incluir la mayor diversidad posible en términos de sexos biológicos y ascendencias, teniendo en cuenta las bases de datos disponibles.

4.1.2. Preparación del conjunto de datos

Procesamiento de modelos 3D

Al contar con un conjunto de datos compuesto por múltiples conjuntos, cada uno con una escala específica, ya sea en centímetros o metros, y dispuestos de manera diversa, surge la necesidad de normalizar la escala y alinear los modelos con respecto a un punto de referencia. Para esto, se emplearán los ojos como punto de origen (0, 0, 0), dado que son fácilmente visualizables, se ubican en una posición central y están presentes tanto en los modelos faciales como en los de cuerpo completo. Este procesamiento se produce para luego poder generar correctamente las imágenes a distintas distancias.

La normalización de la escala se lleva a cabo mediante transformaciones de escala, multiplicando o dividiendo por un factor de 100. Esta se realiza solo en algunos modelos, para que todos estén a la misma escala.

Para alinear los modelos, en primer lugar se aplicó un *script* de *Python* para realizar transformaciones (específicas para cada conjunto de datos) con el objetivo de posicionarlos de frente. Posteriormente, se desarrolló un pro-

grama en *Python* que utiliza librerías como *VTK*, *PyVista* y *Trimesh* para la manipulación de mallas 3D, junto con *Mediapipe* para la detección de rostros. El proceso implica tener un modelo de referencia ya alineado manualmente, junto con 13 puntos de referencia faciales 3D (incluyendo la nariz, los ojos y la boca) obtenidos mediante *Mediapipe*. Después, dado un nuevo modelo sin alinear, se calculan sus puntos de referencia correspondientes y se determina y aplica la matriz de transformación entre estos puntos y los del modelo de referencia.

A excepción de algunos ajustes manuales en los modelos de HeadSpace y HuMMAn, el proceso se automatizó de forma efectiva.

Generación de imágenes faciales sintéticas

Una vez procesados los modelos 3D, se procedió a generar el conjunto de imágenes faciales. Para ello, se utilizó un *script* en *Blender* que realiza las siguientes tareas para cada modelo:

1. Carga el modelo y lo posiciona a una distancia específica de la cámara. Se seleccionaron 27 distancias distintas, que van desde 50 cm hasta 6m, con incrementos graduales de 5 cm, 10 cm, 20 cm, 25 cm, 50 cm y 1 m.
2. Posteriormente, para cada distancia, se ajusta la longitud focal de la cámara. Se utilizaron 4 longitudes focales, incluyendo 27 mm, 35 mm, 53 mm y 83.6 mm.
3. A continuación, para cada longitud focal, se realizan 14 iteraciones, donde en cada iteración:
 - 1) Se aplica un fondo HDR seleccionado aleatoriamente de un conjunto de 95 fondos HDR descargados de Poly Haven⁴.
 - 2) Se aplican transformaciones aleatorias de rotación de la cámara con respecto al modelo para añadir variabilidad a las poses. Estas transformaciones incluyen tanto rotaciones horizontales (entre -70° y 70°) para mostrar los modelos desde diferentes perspectivas laterales como se muestra en la Figura 4.6, así como rotaciones verticales (entre -30° y 30°) para presentar perspectivas más altas o bajas como se muestra en la Figura 4.7.
 - 3) Se realizan pequeñas traslaciones de la cámara para evitar que todos los modelos aparezcan centrados en la imagen, añadiendo así una variabilidad extra.

⁴<https://polyhaven.com>

- 4) Se ajusta la iluminación y las sombras mediante una lámpara cuya intensidad y posición varían aleatoriamente dentro de unos rango determinados.
- 5) Por último, se genera la imagen con un tamaño de 224x224 píxeles.

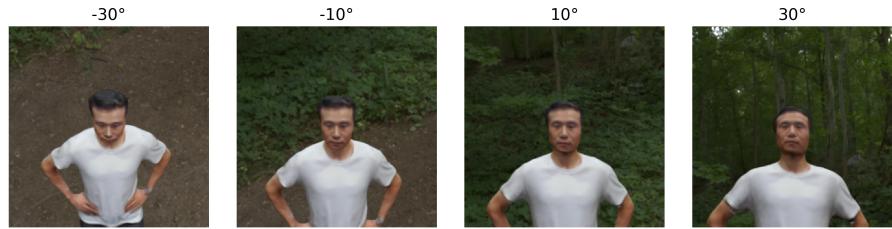


Figura 4.6: Imágenes generadas desde perspectivas más altas o más bajas.

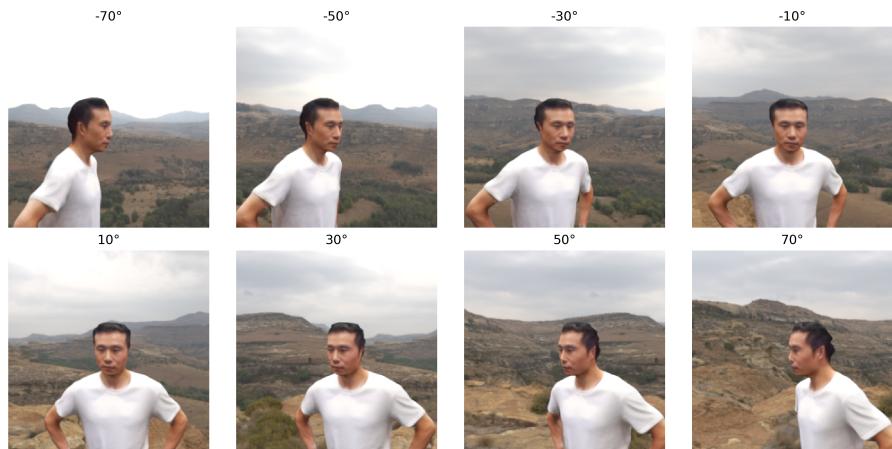


Figura 4.7: Imágenes generadas desde distintas perspectivas laterales.

Tras este proceso de generación de imágenes, y considerando que se contaba con 176 modelos 3D, el conjunto total de datos ascendió a 266112 imágenes, es decir, 66528 imágenes por cada longitud focal. Se pueden ver algunos ejemplos de imágenes en la Figura 4.8.

La mejora con respecto al conjunto de datos utilizado en FacialSCDnet [4] es evidente, como se observa claramente en la figura 4.9.

4.2. Métodos

En este apartado se describen las arquitecturas *deep learning* que se van a utilizar para realizar los experimentos.

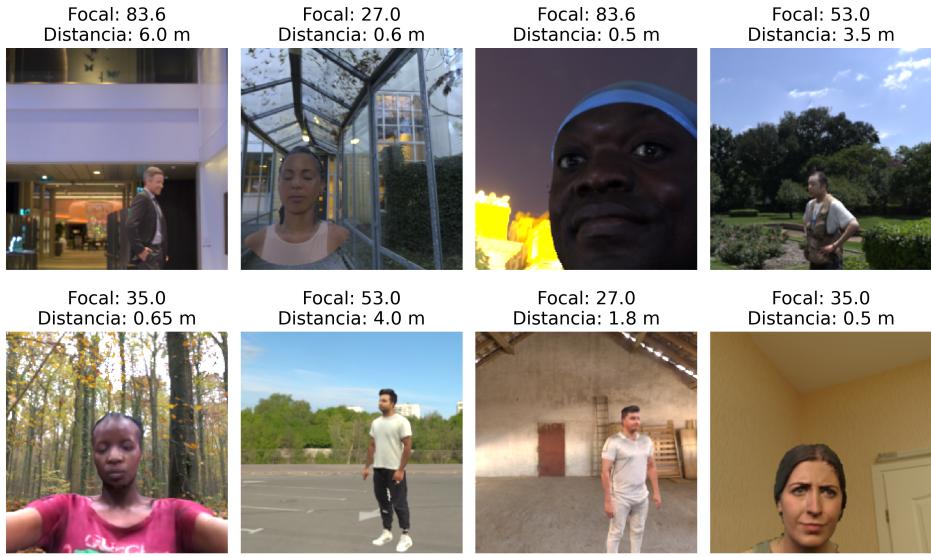


Figura 4.8: Imágenes generadas para el conjunto de datos sintético. Estos ejemplos contienen diferentes sujetos, longitudes focales y distancias.

4.2.1. FacialSCDnet+

El método FacialSCDnet+, basado en FacialSCDnet [4], es un enfoque de aprendizaje profundo para estimar la distancia entre el sujeto y la cámara en fotografías faciales. Se basa en una red neuronal convolucional (CNN) diseñada para regresar la distancia métrica de los individuos directamente desde las fotografías faciales. Este método tiene cuatro modelos de aprendizaje, uno por cada longitud focal presente en el conjunto de datos. A diferencia de FacialSCDnet, este método utiliza un conjunto de datos totalmente sintético, el cual es mucho más realista y diverso. Además, se ha rediseñado el sistema de aumento de imágenes aplicando las operaciones siguientes:

- Transformaciones afines: Se aplican rotaciones aleatorias de hasta 15 grados en sentido horario o antihorario, así como traslaciones horizontales y verticales de hasta el 20 % del tamaño de la imagen, con una probabilidad del 100 %. Esta transformación simula diferentes perspectivas de las imágenes.
- Emborronado Gaussiano: Se aplica un emborronado gaussiano con un *kernel* de tamaño 5 y un sigma aleatorio entre 0.1 y 2.0, lo que suaviza la imagen, con una probabilidad del 25 %. Esta transformación ayuda a introducir algo de ruido en las imágenes.
- Nitidez: Se ajusta la nitidez de la imagen aplicandole un factor de

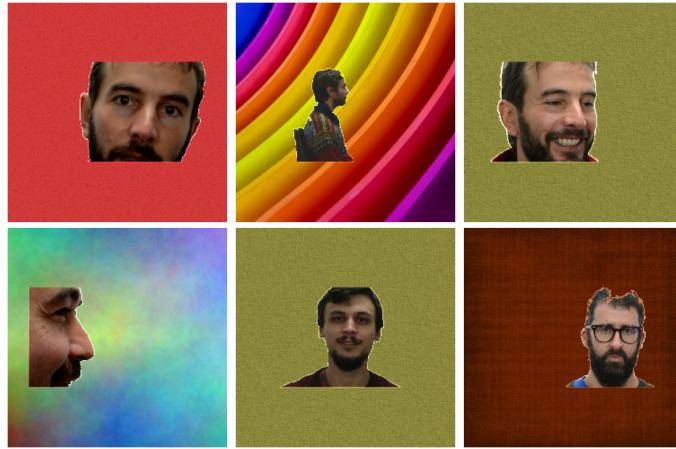


Figura 4.9: Imágenes utilizadas en FacialSCDnet [4].

nitidez de valor 2, con una probabilidad del 25 %. Esta transformación simula una mejor definición de las imágenes

- Alteraciones de color: Se aplican ajustes aleatorios en el brillo, contraste y tono de la imagen, con un rango de variación entre ± 0.1 en cada canal, con una probabilidad del 25 %. Esta transformación contribuye a aumentar la diversidad en la apariencia de las imágenes.
- Borrado de píxeles: Se borran zonas de píxeles aleatorias de la image. Estas zonas tienen un tamaño de entre el 2 % y el 5 % de la imagen, y la relación de aspecto está entre un 0.5 y un 1.5, dotando a estas zonas de un aspecto más rectangular. Esta transformación tiene una probabilidad del 25 %. Esta transformación introduce un grado adicional de variabilidad y robustez frente a la ocultación parcial de información.
- Escala de grises: La imagen se convierte a escala de grises, perdiendo la información de color, con una probabilidad del 25 %. Esta transformación permite al modelo mejorar su invarianza al color.

Estas transformaciones aumentan la calidad del conjunto de datos de entrenamiento, mejorando la robustez y la capacidad de generalización del modelo ante diferentes condiciones y variaciones en las imágenes de entrada.

FacialSCDnet+ también utiliza la arquitectura VGG-16 como *backend* para entrenar el modelo de aprendizaje.

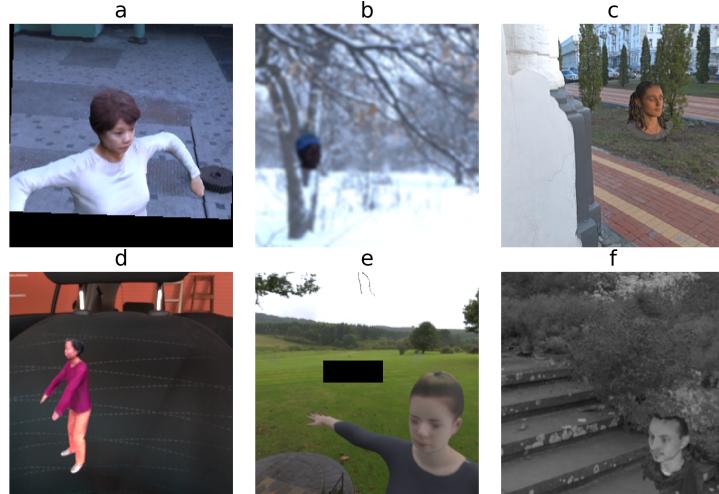


Figura 4.10: Transformaciones aplicadas a las imágenes: a) transformaciones afines, b) emborronado gaussiano, c) nitidez, d) alteraciones de color, e) borrado de píxeles, f) escala de grises.

VGG-16

VGG-16 es una red neuronal convolucional profunda basada en la arquitectura VGGNet [40]. Esta red contiene 16 capas entrenables, como su nombre indica, y destaca por su eficacia en la extracción de características en imágenes.

La red recibe como entrada una imagen RGB de tamaño fijo de 224x224. Esta imagen atraviesa una serie de capas convolucionales, donde se emplean filtros de tamaño 3x3. En estas capas, el *stride* se mantiene constante en 1 píxel y se utiliza un *padding* de 1 píxel para evitar la pérdida de dimensionalidad al aplicar los filtros de convolución 3x3. Cada capa convolucional contiene una capa de activación ReLU detrás. Tras algunas de las capas convolucionales, se realiza un *max pooling* con filtros de 2x2 y *stride* de 2 píxeles, esta operación se realizará para ir reduciendo el tamaño de los mapas de activación. En esta primera parte de la red es donde se extraen las características de la imagen.

Posteriormente, esta primera parte de la red es sucedida por tres capas totalmente conectadas: las dos primeras cuentan con 4096 neuronas cada una, mientras que la tercera realiza la clasificación con 1000 neuronas (una por cada clase). Tras cada una de estas capas, sigue una capa de activación

ReLU. La última capa corresponde a la capa de *soft-max* que calcula las probabilidades de pertenecer a cada clase. En esta parte final de la red se realiza la clasificación final de las características extraídas para la tarea de reconocimiento de imágenes.

La arquitectura VGG-16 se puede ver en la Figura 4.11.

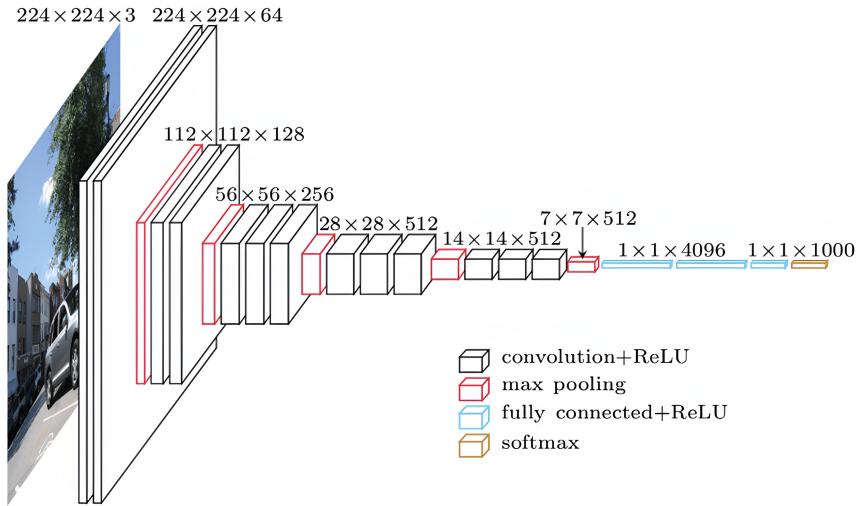


Figura 4.11: Arquitectura de la red VGG-16. Las dimensiones se muestran en formato: Columnas x Filas x Canales [31].

El uso de convoluciones 3x3 tiene dos ventajas:

- Permite aplicar más convoluciones e incrementar la profundidad de la red al tener menos parámetros entrenables.
- Combinando las pequeñas convoluciones y la profundidad de la red, se pueden detectar características a pequeña escala, mientras que la agregación de escalas mayores va implícita al pasar de capa.

El uso de esta red se justifica por su sencillez y su capacidad para aprender características significativas de las imágenes mediante el bloque de capas convolucionales.

En FacialSCDnet+, se ha modificado la estructura de la red de manera que se conserva la extracción de características a través de los bloques convolucionales, aprovechando los pesos preentrenados en ImageNet, mientras que en la parte final de la red se incluyen dos capas fully connected con 4096 neuronas cada una, junto con una capa de salida con una sola neurona para abordar el problema de regresión. Se realiza un proceso de fine-tuning, es decir, los pesos de los bloques convolucionales permanecen congelados,

mientras que la parte final de la red es entrenada desde cero. En total, la red cuenta con 119,545,857 parámetros para el entrenamiento.

Dado su alto número de parámetros a entrenar, VGG-16 demanda recursos computacionales significativos, sin embargo, su gran rendimiento lo convierte en una opción atractiva para este trabajo.

Capítulo 5

Experimentos

5.1. Detalles técnicos de la implementación

5.1.1. Entorno de desarrollo

Para llevar a cabo este proyecto, se ha utilizado exclusivamente *Python* como lenguaje de programación. Esta elección se fundamenta en la versatilidad y la eficacia que ofrece para trabajar tanto en *Blender* como en el desarrollo de modelos de aprendizaje profundo. Se emplearon diversas bibliotecas para distintas tareas: *Trimesh* y *PyVista* junto con *VTK* para manipulación de modelos 3D; *Mediapipe* para la extracción de puntos de referencia faciales; *NumPy* y *pandas* para el procesamiento de datos; *PIL* para el trabajo con imágenes; *matplotlib* y *Seaborn* para la generación de gráficos; *Optuna* para la optimización de hiperparámetros; *MLflow* para la gestión de experimentos; y finalmente, *PyTorch* para el entrenamiento de modelos de aprendizaje profundo.

5.1.2. Organización del conjunto de datos

Como se ha mencionado anteriormente, se utiliza un *script* de *Blender* para generar las imágenes. La organización de las imágenes es la siguiente: para cada sujeto, se crean carpetas denominadas S1 hasta S176. Dentro de cada una de estas carpetas, se encuentran subcarpetas correspondientes a diferentes distancias, que van desde d05 hasta d6 (en metros, por ejemplo d05 representa distancia 0.5 metros). Estas subcarpetas, a su vez, contienen cuatro carpetas adicionales, representando cada una las diferentes longitudes focales, desde f27 hasta f83.6. Cada una de estas carpetas contiene 14 imágenes del sujeto a esa distancia y con esa longitud focal específica. Por ejemplo, la identificación de una imagen sería S144d1f27_i0.png, donde 'S144' representa el sujeto número 144, 'd1' indica una distancia de 1 metro, 'f27' indica

una longitud focal de 27, y '`_i0`' señala que es la imagen número 0.

A su vez, se genera un archivo CSV que incluye los nombres de las imágenes, sus rutas absolutas, las distancias y las longitudes focales correspondientes. Este archivo facilita el proceso de carga del conjunto de datos en *PyTorch*.

5.1.3. Entrenamiento del modelo

El proceso de entrenamiento del modelo ha sido organizado en distintos archivos de *Python* para mejorar su modularidad y claridad. En primer lugar, el archivo *SCD_Dataset.py* se encarga de cargar el conjunto de datos y aplicar las transformaciones necesarias para el aumento de datos. Por otro lado, la implementación del modelo de aprendizaje, junto con sus métodos de entrenamiento y evaluación, se encuentra en el archivo *SCD_Model.py*. La configuración de los hiperparámetros se almacena en un archivo llamado *config.json*, el cual es cargado en el archivo *SCD_Main.py*, donde se lleva a cabo el proceso completo de carga de datos y entrenamiento utilizando los archivos mencionados anteriormente.

Para la optimización de los hiperparámetros, se dispone del archivo *SCD_MainTuner.py*, que contiene los rangos utilizados durante la optimización de cada parámetro. Por otra parte, para generar gráficos del rendimiento del modelo, se emplea el archivo *SCD_MainGraficas.py*. Además, se ha creado un archivo denominado *metrics.py* para definir las métricas que serán utilizadas en la evaluación del rendimiento del modelo.

Los archivos main son ejecutados a través de un *script* de bash que inicia un proceso utilizando SLURM en una de las GPUs disponibles en la Universidad de Granada, utilizando para ello la conexión remota SSH. Dentro de este *script*, se activa el entorno Conda y se selecciona la cola y el servidor específico donde se llevará a cabo el proceso. Posteriormente, se ejecuta el archivo *Python* correspondiente.

5.1.4. Gestión de experimentos

Para mejorar la gestión de experimentos, se ha puesto en práctica la metodología MLOps, con el objetivo de automatizar y monitorizar los procesos, garantizando al mismo tiempo la reproducibilidad de los resultados. Esta implementación se ha realizado utilizando la librería *MLflow*.

5.2. Experimentos

5.2.1. Protocolo de validación experimental

La técnica empleada para llevar a cabo el entrenamiento se conoce como *hold-out* (veáse Figura 5.1). Esta metodología implica dividir el conjunto de datos en dos partes distintas: el conjunto de entrenamiento y el conjunto de test. A su vez, dentro del conjunto de entrenamiento, se realiza una subdivisión adicional para crear un conjunto de validación. Este conjunto se utiliza durante el proceso de entrenamiento del modelo para evaluar periódicamente la calidad del mismo mediante comparaciones con las métricas obtenidas en el conjunto de entrenamiento. Una vez finalizado el entrenamiento, se evalúa el modelo utilizando el conjunto de test, que contiene datos que no se han visto nunca durante el entrenamiento, con el propósito de obtener una evaluación definitiva sobre la calidad del aprendizaje.

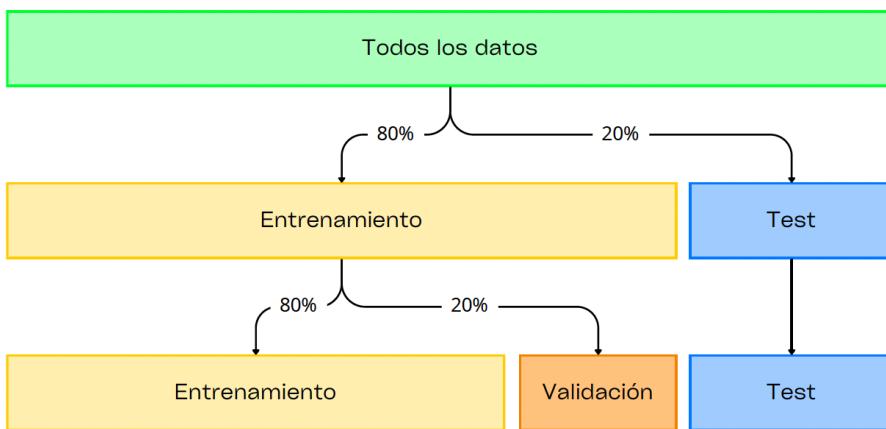


Figura 5.1: Esquema de división del conjunto de datos total en los subconjuntos de entrenamiento, validación y test.

Considerando que el conjunto completo incluye los datos de los 4 modelos a utilizar (F27, F35, F53, F83.6), el primer paso consiste en reservar todas las imágenes de 30 sujetos de forma aleatoria para el conjunto de test. Posteriormente, se añaden imágenes adicionales de manera aleatoria, manteniendo las mismas distribuciones de focales, para constituir el 20 % del conjunto total de test, mientras que el 80 % restante se asigna al conjunto de entrenamiento.

Finalmente, durante el proceso de entrenamiento de los modelos, se aparta un 20 % aleatorio del conjunto de entrenamiento como conjunto de validación, dejando el 80 % restante para el entrenamiento propiamente dicho. Por tanto, para cada modelo focal contamos con 66528 imágenes, de las cuales 42577 se destinan al entrenamiento, 10645 a la validación y 13306 a test.

5.2.2. Métricas

Dado que se aborde un problema de regresión en el que la importancia de la distorsión en distancias cercanas es crucial, hemos optado por emplear la distorsión como función de pérdida. Esta medida se calcula mediante la siguiente fórmula:

$$\text{Distorsion} = \frac{\sum_{i=1}^n \left| \frac{1}{1 + \frac{y_i}{d}} - \frac{1}{1 + \frac{x_i}{d}} \right|}{n} \quad (5.1)$$

siendo y_i la distancia verdadera en la imagen i , x_i la distancia predicha en la imagen i , y $d = 12.6572$ cm, que corresponde a un valor derivado de cálculos geométricos [42] para obtener experimentalmente el factor de distorsión de una cabeza humana de tamaño promedio, según la SCD de la fotografía.

Esta función de pérdida asegura que el modelo aprenda a predecir distancias cercanas con mayor precisión, mitigando así la posibilidad de una distorsión significativa.

Aunque la distorsión se considera la medida principal de rendimiento, también se han empleado otras métricas como el error absoluto medio (MAE) y el error relativo medio (MRE) para evaluar el desempeño del modelo:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad (5.2)$$

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - x_i|}{y_i} \quad (5.3)$$

El MAE, al calcular la diferencia absoluta promedio entre las predicciones del modelo y los valores reales, mide cuánto se desvían las predicciones del modelo en términos absolutos. Esto es útil para entender la magnitud de los errores de predicción sin considerar su distorsión. Por otro lado, el MRE, al calcular la diferencia relativa promedio entre las predicciones del modelo y los valores reales, proporciona una medida de cuánto se desvían las predicciones del modelo en relación con la distancia real (normalmente en porcentaje). Esto es fundamental cuando se necesita evaluar el rendimiento del modelo en términos de precisión relativa.

5.2.3. Experimentos VGG16

Tuneo de hiperparámetros

VGG16

Capítulo 6

Conclusiones y trabajos futuros

Bibliografía

- [1] Yaser S. Abu-Mostafa, M. Magdon-Ismail y H.T. Lin. *Learning from Data: A Short Course*. AMLBook, 2012.
- [2] Adorama. *What Is Crop Factor And How Do You Calculate It?* Accedido el 17 de Marzo de 2024. 2022. URL: <https://www.adorama.com/alc/what-is-crop-factor-everything-you-need-to-know/>.
- [3] Thiemo Alldieck et al. «Video Based Reconstruction of 3D People Models». En: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, págs. 8387-8397.
- [4] Enrique Bermejo et al. «FacialSCDnet: A deep learning approach for the estimation of subject-to-camera distance in facial photographs». En: *Expert Systems with Applications* 210 (2022), pág. 118457.
- [5] C.M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.
- [6] Blas. *Factor de recorte de un sensor*. Accedido el 18 de Marzo de 2024. 2018. URL: <https://blasfotografia.com/factor-de-recorte-de-un-sensor/>.
- [7] Sara Brown. *Machine learning, explained*. Accedido el 11 de Marzo de 2024. 2021. URL: <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>.
- [8] Xavier Burgos-Artizzu, Pietro Perona y Piotr Dollar. «Robust Face Landmark Estimation under Occlusion». En: *IEEE International Conference on Computer Vision* (2013), págs. 1513-1520.
- [9] Xavier Burgos-Artizzu, Matteo Ruggero Ronchi y Pietro Perona. *Caltech Multi-Distance Portraits (CMDP)*. 2022.
- [10] Xavier P. Burgos-Artizzu, Matteo Ruggero Ronchi y Pietro Perona. «Distance Estimation of an Unknown Person from a Portrait». En: *European Conference on Computer Vision*. Vol. 8689. 2014, págs. 313-327.
- [11] Zhongang Cai et al. «HuMMan: Multi-modal 4d human dataset for versatile sensing and modeling». En: *European Conference on Computer Vision*. 2022, págs. 557-577.

- [12] Hang Dai et al. «Statistical Modeling of Craniofacial Shape and Texture». En: *International Journal of Computer Vision* 128.2 (2019), págs. 547-571.
- [13] Alfonso Domínguez. *Distorsión de lente vs Distorsión de la perspectiva*. Accedido el 23 de Marzo de 2024. 2011. URL: <https://www.xatakafoto.com/guias/distorsion-de-lente-vs-distorsion-de-la-perspectiva>.
- [14] Gary Edmond et al. «Law's Looking Glass: Expert Identification Evidence Derived from Photographic and Video Images». En: *Current Issues in Criminal Justice* 20 (2009), págs. 337-377.
- [15] FISWG. *Facial Comparison Overview and Methodology Guidelines V2.0*. Accedido el 7 de Febrero de 2024. 2022. URL: https://fiswg.org/fiswg_facial_comparison_overview_and_methodology_guidelines_V2.0_2022.11.04.pdf.
- [16] Arturo Flores et al. «Camera Distance from Face Images». En: *Advances in Visual Computing*. Vol. 8034. 2013, págs. 513-522.
- [17] Daniel Graupe. *Principles of artificial neural networks 3rd edition*. World Scientific, 2007.
- [18] Elizabeth Gray. *What Is Focal Length in Photography? A Beginner's Guide*. Accedido el 17 de Marzo de 2024. 2023. URL: <https://photographylife.com/what-is-focal-length-in-photography>.
- [19] Ankush Gupta, Andrea Vedaldi y Andrew Zisserman. «Synthetic Data for Text Localisation in Natural Images». En: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2016), págs. 2315-2324.
- [20] Simon Haykin. *Neural Networks and Learning Machines 3rd edition*. Pearson, 2009.
- [21] Asmaul Hosna et al. «Transfer learning: a friendly introduction». En: *Journal of Big Data* 9 (2022).
- [22] Minyoung Huh, Pulkit Agrawal y Alexei Efros. «What makes ImageNet good for transfer learning?» En: (2016).
- [23] IBM. *What is machine learning?* Accedido el 11 de Marzo de 2024. 2019. URL: <https://www.ibm.com/topics/machine-learning>.
- [24] John D. Kelleher. *Deep learning*. MIT Press, 2019.
- [25] M.S. Shashi Kumar, K.S. Vimala y N. Avinash. «Face distance estimation from a monocular camera». En: *IEEE International Conference on Image Processing*. 2013, págs. 3532-3536.
- [26] Y. LeCun et al. «Backpropagation Applied to Handwritten Zip Code Recognition». En: *Neural Computation* 1 (1989), págs. 541-551.

- [27] Y. Lecun et al. «Gradient-based learning applied to document recognition». En: *Proceedings of the IEEE* 86 (1998), págs. 2278-2324.
- [28] Vincent Lepetit, Francesc Moreno-Noguer y Pascal Fua. «EPnP: An accurate O(n) solution to the PnP problem». En: *International Journal of Computer Vision* 81 (2009).
- [29] Javier Lucas. *Qué Es El Factor de Recorte de tu Sensor y Cómo Influye en la Focal de tus Objetivos*. Accedido el 17 de Marzo de 2024. 2022. URL: <https://www.dzoom.org.es/que-es-el-factor-de-recorte-de-tu-sensor-y-como-influye-en-la-focal-de-tus-objetivos/>.
- [30] Nasim Mansurov. *Does Focal Length Distort Subjects?* Accedido el 23 de Marzo de 2024. 2020. URL: <https://photographylife.com/does-focal-length-distort-subjects>.
- [31] Will Nash, Tom Drummond y Nick Birbilis. «A review of deep learning in the study of materials degradation». En: *npj Materials Degradation* 2 (2018).
- [32] Eilidh Noyes y Rob Jenkins. «Camera-to-subject distance affects face configuration and perceived identity». En: *Cognition* 165 (2017), págs. 97-104.
- [33] Bo Peng et al. «Position Determines Perspective: Investigating Perspective Distortion for Image Forensics of Faces». En: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, págs. 1813-1821.
- [34] R.S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 2005.
- [35] Khandaker Abir Rahman et al. «Person to Camera Distance Measurement Based on Eye-Distance». En: *International Conference on Multimedia and Ubiquitous Engineering*. 2009, págs. 137-141.
- [36] Eduard Ramon et al. «H3D-Net: Few-Shot High-Fidelity 3D Head Reconstruction». En: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, págs. 5620-5629.
- [37] Stuart Russell y Peter Norvig. *Artificial Intelligence: A Modern Approach 4º edition*. Pearson, 2021.
- [38] Nafiz Shahriar. *What is Convolutional Neural Network — CNN (Deep Learning)*. Accedido el 12 de Marzo de 2024. 2023. URL: <https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5>.
- [39] Mohamed Tahir Ahmed Shoani, Shamsudin H. M. Amin e Ibrahim M. H. Sanhoury. «Determining subject distance based on face size». En: *Asian Control Conference*. 2015, págs. 1-6.

- [40] Karen Simonyan y Andrew Zisserman. «Very deep convolutional networks for large-scale image recognition». En: 2015.
- [41] Nicole Spaun. «Facial Comparisons by Subject Matter Experts: Their Role in Biometrics and Their Training». En: *International Conference on Biometrics*. 2009, págs. 161-168.
- [42] Carl N. Stephan. «Estimating the Skull-to-Camera Distance from Facial Photographs for Craniofacial Superimposition». En: *Journal of Forensic Sciences* 62 (2017), págs. 850-860.
- [43] Nicholas Tinelli. *Sensores y factor de recorte: en palabras fáciles*. Accedido el 17 de Marzo de 2024. 2020. URL: <https://nicholastinelli.com/es/sensores-y-factor-de-recorte-en-palabras-faciles/>.
- [44] Jonathan Tremblay et al. «Training deep networks with synthetic data: Bridging the reality gap by domain randomization». En: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2018), págs. 1082-1090.
- [45] Andrea Valsecchi. *Comprendión de los parámetros de la cámara*. Accedido el 17 de Febrero de 2024. 2019. URL: <https://skeleton-id.com/investigaciones/comprehension-de-los-parametros-de-la-camara>.
- [46] Gal Vardi. «On the Implicit Bias in Deep-Learning Algorithms». En: *Communications of the Association for Computing Machinery* 66 (2022), págs. 86-93.
- [47] Lizhen Wang et al. «FaceVerse: a Fine-grained and Detail-controllable 3D Face Morphable Model from a Hybrid Dataset». En: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022.
- [48] Qi Wang et al. «Learning from synthetic data for crowd counting in the wild». En: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2019), págs. 8190-8199.
- [49] Zhixiang Wang et al. «DisCO: Portrait Distortion Correction with Perspective-Aware 3D GANs». En: (2023).
- [50] Rikiya Yamashita et al. «Convolutional neural networks: an overview and application in radiology». En: *Insights into Imaging* 9 (2018), págs. 611-629.
- [51] Guangle Yao, Tao Lei y Jiandan Zhong. «A review of Convolutional-Neural-Network-based action recognition». En: *Pattern Recognition Letters* 118 (2019), págs. 14-22.
- [52] Xiangxin Zhu y Deva Ramanan. «Face Detection, Pose Estimation, and Landmark Localization in the Wild». En: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2012), págs. 2879-2886.

Búsquedas en Scopus

Imágenes faciales (2937)

TITLE-ABS-KEY (facial AND (images OR photographs))

Estimación de la distancia en fotografías faciales (441)

TITLE-ABS-KEY ((distance OR depth) AND estimation AND (photographs OR images) AND facial) AND (LIMIT-TO (SUBJAREA , ÇOMP") OR LIMIT-TO (SUBJAREA , .^{ENGI"}))

Estimación de la distancia en fotografías faciales mediante IA (224)

TITLE-ABS-KEY ((deep AND learning) OR (machine AND learning) OR (artificial AND intelligence) OR (computer AND vision) OR (soft AND computing) AND ((distance OR depth) AND estimation AND (photographs OR images) AND facial)) AND (LIMIT-TO (SUBJAREA , ÇOMP") OR LIMIT-TO (SUBJAREA , .^{ENGI"))}

Estimación de la distancia en fotografías faciales mediante deep learning (129)

TITLE-ABS-KEY ((deep AND learning) AND ((distance OR depth) AND estimation AND (photographs OR images) AND facial)) AND (LIMIT-TO (SUBJAREA , ÇOMP") OR LIMIT-TO (SUBJAREA , .^{ENGI"))}