



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: aprendizaje automático, aprendizaje profundo, distorsión de perspectiva, estimación de distancia cámara-sujeto, regresión, visión por computador.

Resumen

La relevancia de las imágenes faciales ha aumentado significativamente a lo largo de los años, debido a la creciente disponibilidad de los dispositivos fotográficos y su uso en el contexto digital. Esta amplia disponibilidad ha permitido su uso en sectores como la biometría o la seguridad, donde el análisis de la calidad y las condiciones de adquisición de las imágenes es especialmente relevante. En particular, la estimación de la distancia entre la cámara y el sujeto es un factor crucial para dicho análisis. Esta predicción permite calcular la distorsión de perspectiva y, en consecuencia, desarrollar avances tanto en la corrección de dichas distorsiones como en ámbitos de identificación o reconocimiento facial.

En este TFG se ha diseñado un método para la estimación de la distancia cámara-sujeto a partir de una fotografía con una longitud focal conocida. El objetivo principal es mejorar la única propuesta basada en aprendizaje profundo conocida hasta la fecha. Para ello, se ha generado un conjunto de 135730 imágenes completamente sintéticas, diversas y realistas, utilizando una variedad de 277 modelos 3D tanto faciales como de cuerpo completo. Con este *dataset*, se pretende abordar diversos sesgos identificados en propuestas anteriores.

La propuesta de este trabajo, denominada FacialSCDnet+, analiza el comportamiento de dos modelos diferentes de aprendizaje profundo para estimar automáticamente la distancia. En concreto, se han empleado las arquitecturas VGG-16 y ResNet-50, adaptadas para el problema de regresión. Además, se ha diseñado un *benchmark* para evaluar el rendimiento de ambas arquitecturas y del método en el que se basa este trabajo, empleando tanto conjuntos de imágenes sintéticas como reales. Los experimentos demuestran cómo las modificaciones propuestas superan la precisión en la estimación de distancias mejorando el error medio absoluto en 5.5 cm y obteniendo un error de distorsión menor de 1% tanto en imágenes reales como sintéticas.

Camera-subject distance estimation in facial photographs using deep learning

Iván Salinas López

Keywords: machine learning, deep learning, perspective distortion, camera-subject distance estimation, regression, computer vision.

Abstract

The relevance of facial images has significantly increased over the years due to the growing availability of photographic devices and their use in the digital context. This wide availability has enabled their use in sectors such as biometrics or security, where the analysis of the quality and acquisition conditions of the images is particularly relevant. Specifically, estimating the distance between the camera and the subject is a crucial factor for such analysis. This prediction allows calculating the perspective distortion and, consequently, developing advancements both in correcting these distortions and in the fields of identification or facial recognition.

In this Final Degree Project, a method has been designed to estimate the subject-to-camera distance from a photograph with a known focal length. The main objective is to improve the only proposal based on deep learning known to date. To achieve this, a set of 135730 fully synthetic, diverse, and realistic images has been generated, using a variety of 277 3D models, including both facial and full-body models. With this dataset, the aim is to address various biases identified in previous proposals.

The proposal of this work, called FacialSCDnet+, analyzes the behavior of two different deep learning models to automatically estimate the distance. Specifically, the VGG-16 and ResNet-50 architectures, adapted for the regression problem, have been employed. Additionally, a benchmark has been designed to evaluate the performance of both architectures and the method on which this work is based, using both synthetic and real image sets. The experiments demonstrate how the proposed modifications improve the accuracy of distance estimation, enhancing the mean absolute error by 5.5 cm and achieving a distortion error of less than 1% in both real and synthetic images.

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	3
1.3. Objetivos	5
1.4. Planificación del proyecto	5
2. Fundamentos teóricos	9
2.1. Aprendizaje automático y aprendizaje profundo	9
2.1.1. Fundamentos	9
2.1.2. Redes neuronales artificiales	10
2.1.3. Redes neuronales convolucionales	12
2.1.4. Transferencia de aprendizaje	16
2.1.5. Regularización	17
2.2. Parámetros de la cámara	19
3. Estado del Arte	23
3.1. Estimación métrica de la SCD	24
3.1.1. Estimación basada en características anatómicas . . .	25
3.1.2. Técnicas <i>deep learning</i>	26
4. Materiales y métodos	29
4.1. Materiales	29
4.1.1. Modelos 3D	29
4.1.2. Preparación del conjunto de datos	33
4.2. Métodos	36
4.2.1. FacialSCDnet	36
4.2.2. FacialSCDnet+	38
4.2.3. <i>Backend</i>	44
5. Experimentos	49
5.1. Entorno de desarrollo	49
5.2. Entorno de ejecución	49
5.3. Resultados	50
5.3.1. Ajuste de hiperparámetros	50

5.3.2. Comparativa Arquitecturas	50
5.3.3. Comparativa con el estado del arte: FacialSCDnet . .	52
6. Conclusiones y trabajos futuros	59
Apéndice	61
Bibliografía	63

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.	Diagrama del proyecto.	5
2.1.	Esquema de red neuronal.	11
2.2.	Digrama del modelo neuronal.	12
2.3.	Arquitectura típica de red neuronal convolucional.	13
2.4.	Ejemplo de operación de convolución.	14
2.5.	Funciones de activación comunes.	15
2.6.	Tipos de <i>pooling</i> comunes.	16
2.7.	Infraajuste y sobreajuste en entrenamiento.	18
2.8.	Relación entre punto nodal y distancia focal.	19
2.9.	Relación entre distancia focal y campo de visión.	20
2.10.	Tipos de tamaños de sensor.	21
2.11.	Ejemplo de distancia focal equivalente.	22
2.12.	Distancia desde la cámara al sujeto.	22
2.13.	Efectos de la distorsión según distancia.	22
3.1.	Número de publicaciones sobre la estimación de la SCD. . . .	23
4.1.	Ejemplos HeadSpace 3D.	30
4.2.	Ejemplos H3DS-net.	30
4.3.	Ejemplos Stirling ESRC 3D Face.	30
4.4.	Ejemplos DI4D_UGR.	31
4.5.	Ejemplos HuMMan.	31
4.6.	Ejemplos People Snapshot	32
4.7.	Ejemplos Render People	32
4.8.	Ejemplos de imágenes con diferentes fondos.	35
4.9.	Ejemplos de imágenes rotadas verticalmente.	36
4.10.	Ejemplos de imágenes rotadas horizontalmente.	36
4.11.	Ejemplos de imágenes generadas para el conjunto de datos. .	37
4.12.	Ejemplos de imágenes reales con fondos FacialSCDnet. . . .	38
4.13.	Transformaciones utilizadas en el aumento de datos.	40
4.14.	Esquema de división del conjunto de datos.	43

4.15. Arquitectura de la red VGG-16.	45
4.16. Bloque residual ResNet.	47
4.17. Arquitectura de la red ResNet-50.	47
5.1. Gráfica de pérdida en entrenamiento y validación.	51
5.2. Comparación predicciones de error test sintético.	53
5.3. Ejemplos de predicciones en imágenes sintéticas.	54
5.4. Comparación predicciones de error test real.	55
5.5. Ejemplos de predicciones en imágenes reales.	56
5.6. Comparación etiquetas de test.	57

Índice de cuadros

1.1.	Planificación inicial del proyecto.	6
1.2.	Planificación final del proyecto.	7
1.3.	Estimación del coste del proyecto.	7
5.1.	Selección de parámetros de entrenamiento FacialSCDnet+. .	50
5.2.	Métricas en validación VGG-16 y ResNet-50.	52
5.3.	Métricas en el conjunto de test sintético.	53
5.4.	Métricas en el conjunto de test real.	55

Capítulo 1

Introducción

1.1. Definición del problema

En la era digital actual, las imágenes faciales han adquirido una relevancia significativa (Figura 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 [1, 2, 3, 4].

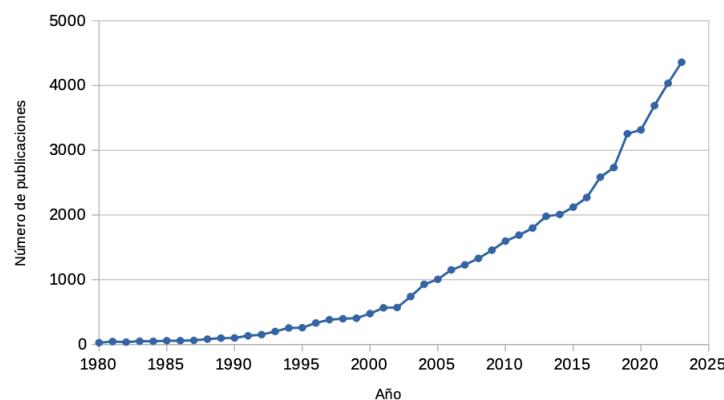


Figura 1.1: Número de publicaciones, en Scopus, relacionadas con imágenes faciales en los últimos 45 años ¹.

En este contexto, es importante resaltar el papel que tienen las imáge-

¹Las búsquedas se pueden consultar en el Apéndice.

nes faciales en campos como la biometría para la verificación de identidad, así como en la seguridad nacional y las ciencias forenses, donde se pueden utilizar para la identificación facial [5, 6, 7]. A fin de 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 [8, 9]. Aspectos como la resolución, el enfoque o la iluminación deben cumplir unos requisitos mínimos. Por ello, desde hace años existen numerosas herramientas y técnicas dirigidas a la extracción de metadatos, la detección facial [10, 11] o la estimación de la pose [12, 13], todas ellas fundamentales para asegurar la fiabilidad y precisión de los sistemas de identificación facial.

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 [14] (Figura 1.2). Esta alteración en la perspectiva repercute negativamente tanto en los sistemas de reconocimiento facial como en el análisis de imágenes, complicando la interpretación visual al alterar cómo se perciben ciertos rasgos.

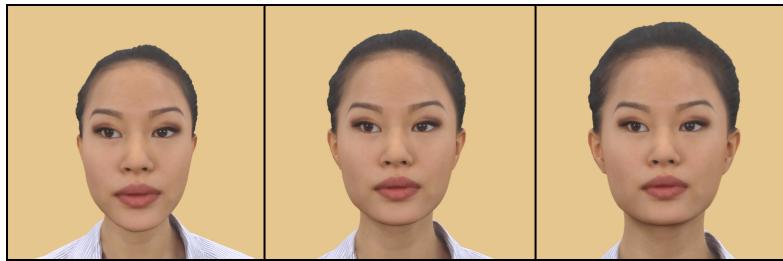


Figura 1.2: Efectos de la distorsión de perspectiva en fotografías faciales realizadas a diferentes distancias: 0.3 m, 0.6 m y 1.5 m respectivamente.

La distorsión de perspectiva está estrechamente relacionada con la **distancia cámara-sujeto** (*subject-to-camera distance*, SCD en adelante), que se define como la separación física entre la posición de la cámara y el sujeto fotografiado. Esta relación entre la distorsión de perspectiva y la SCD sigue un patrón de decremento logarítmico, lo que significa que a distancias cortas se produce una mayor distorsión, la cual disminuye gradualmente a medida que la distancia entre la cámara y el sujeto aumenta [15]. Conocer la SCD en fotografías faciales permite cuantificar la cantidad de distorsión de perspectiva presente en una imagen, así como las diferencias en la distorsión entre dos pares de imágenes. Esta información puede ser determinante al evaluar la identidad de un individuo mediante técnicas forenses de identificación facial, además de facilitar el desarrollo de métodos precisos para corregir dichas distorsiones [16].

La SCD, a diferencia de otros parámetros de la cámara 2.2 como la distancia focal o el tamaño del sensor, no se puede obtener directamente desde los metadatos de la fotografía [17]. Por tanto, es necesario un método preciso para su estimación.

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[18, 19, 20]. 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 distancias focales en el mismo conjunto de datos.

Recientemente, se han comenzado a aplicar técnicas de aprendizaje profundo para la estimación de la SCD, como el método FacialSCDnet [21], que utiliza una arquitectura basada en aprendizaje profundo para procesar imágenes faciales y calcular su distancia correspondiente. Estas técnicas, aunque prometedoras, enfrentan limitaciones debido a la calidad y cantidad insuficiente de datos para el entrenamiento, lo que puede resultar en modelos sesgados y con capacidad limitada para generalizar. Además, la variabilidad en la calidad de las imágenes y las condiciones de captura presentan desafíos significativos. Estas cuestiones subrayan la necesidad de seguir desarrollando y refinando estas técnicas para mejorar su precisión y aplicabilidad en un amplio rango de contextos y poblaciones.

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 de la biometría, cuando se comparan rasgos faciales para la identificación humana en fotografías o videos, es crucial tener en cuenta varios factores, como la iluminación [22, 23], la pose [24] y la expresión [25, 26]. Además de ellos, uno de los desafíos clave es la distorsión de perspectiva, que afecta los atributos faciales según la distancia entre el sujeto y la cámara en el momento de la fotografía (Figura 1.2). Esta distorsión puede afectar significativamente el análisis comparativo y la precisión de las herramientas de reconocimiento asistido por computadora. Por tanto, conocer la SCD facilita el desarrollo de técnicas que permitan analizar y corregir con precisión

dicha distorsión [16, 27], con el fin de realizar un análisis facial mucho más fiable.

En el ámbito forense, también es importante determinar la distancia entre el sujeto y la cámara, ya que esto permite evaluar la distorsión de la imagen en la escena original donde se tomó la fotografía. Además, puede ayudar a recrear las condiciones bajo las cuales se capturó la escena, lo que facilita una comparación más precisa de los individuos [28]. De este modo, se optimiza la confiabilidad y precisión de técnicas como la **comparación facial forense** (CFF) [29], la cual consiste en identificar similitudes y diferencias entre imágenes faciales con el objetivo de determinar si corresponden a la misma persona.

En el campo del aprendizaje automático, es común encontrar sesgos en los datos que pueden afectar la precisión y la capacidad de generalización de las soluciones [30]. 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, entrenado con una sola base de datos facial compuesta únicamente por individuos femeninos. 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 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. Esta mejora contribuiría a una mayor 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 [31, 32, 33]. 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. Además, al emplear datos sintéticos, se tiene un mayor control sobre los parámetros de generación de la imagen, lo que permite ajustar la escena de manera precisa y reproducible.

En resumen, la distancia entre la cámara y el sujeto desempeña un papel esencial en diversas disciplinas debido a su impacto en la distorsión de los sujetos en las imágenes. Estimar con precisión la SCD es vital para mejorar el análisis de las fotografías faciales, por lo que este trabajo pretende abordar algunas limitaciones y sesgos identificados en los métodos actuales.

1.3. Objetivos

El objetivo general de este TFG consiste en desarrollar un modelo de aprendizaje profundo para mejorar la estimación de la distancia cámara-sujeto en fotografías faciales (Figura 1.3). Para el desarrollo del proyecto, dividiremos el objetivo general en una serie de objetivos parciales:

1. Realizar un análisis exhaustivo del estado del arte, en concreto, examinar con pausa y de forma crítica el modelo de referencia FacialSCDnet.
2. Examinar detenidamente las bases de datos existentes de modelos faciales y humanos en 3D, y diseñar un protocolo de estandarización para generar imágenes sintéticas fotorrealistas a partir de estos modelos.
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.

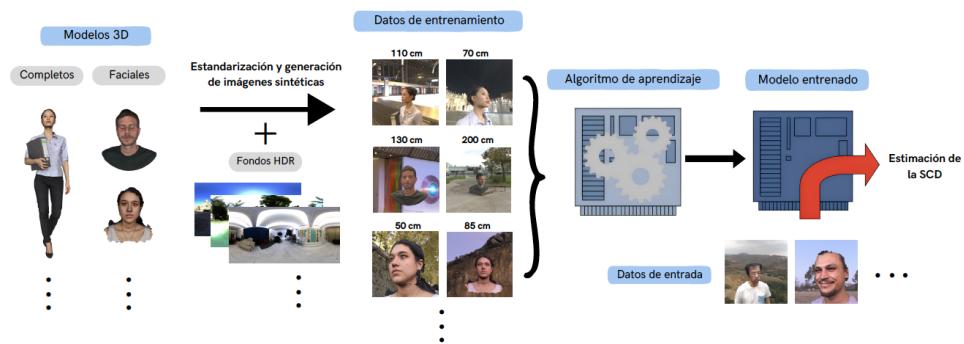


Figura 1.3: Diagrama del proceso de estimación automática de la SCD en este proyecto. El punto de partida es una serie de modelos 3D, tanto completos como faciales, a los cuales se les aplicará un proceso de estandarización y normalización. Posteriormente, se añadirán fondos HDR para generar imágenes fotorrealistas a diferentes distancias, las cuales se utilizarán para entrenar el modelo de aprendizaje con el objetivo de estimar la SCD ante nuevos datos de entrada.

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 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.
- Experimentación: Consiste en la realización de diversos experimentos para validar el funcionamiento del software desarrollado, utilizando los modelos y datos previamente definidos.
- Documentación: De forma paralela a las cuatro fases anteriores, se realiza un proceso continuo de documentación de la memoria del proyecto. Este proceso asegura un registro de todas las actividades realizadas, detallando cada paso y decisión tomada durante el desarrollo del proyecto, facilitando así la comprensión y la replicación del trabajo realizado.

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					
Experimentación	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.

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					
Experimentación	7 - 120					

Tabla 1.2: Planificación final del proyecto.

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 16000 euros. Con una amortización proyectada a dos años, lo que equivale a un pago diario de 21.92 euros, su contribución al costo total del proyecto se estima en 3090.72 euros.

Fecha de inicio	05/02/2024
Fecha de fin	24/06/2024
Duración	141 días, 101 laborables

Item	Costo
Salario	12120 euros
Portátil de Gama Alta	2600 euros
Servidor GPU	3090.72 euros
Total	17810.72 euros

Tabla 1.3: Estimación del coste del proyecto.

Capítulo 2

Fundamentos teóricos

2.1. Aprendizaje automático y aprendizaje profundo

2.1.1. Fundamentos

El aprendizaje automático (*Machine Learning*, ML) [35, 36] es una rama de la inteligencia artificial 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:

- 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.

El aprendizaje profundo (*Deep Learning*, DL) [37, 38] constituye una subárea del ML que se enfoca en algoritmos basados en redes neuronales profundas, capaces de aprender representaciones jerárquicas de los datos. Para ello, el DL suele requerir grandes cantidades de datos y poder computacional para entrenar modelos efectivos.

2.1.2. Redes neuronales artificiales

Las redes neuronales artificiales (*Artificial Neural Networks*, ANN) [39, 40, 41] son redes computacionales que intentan, a grosor 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 (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

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 (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*).

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

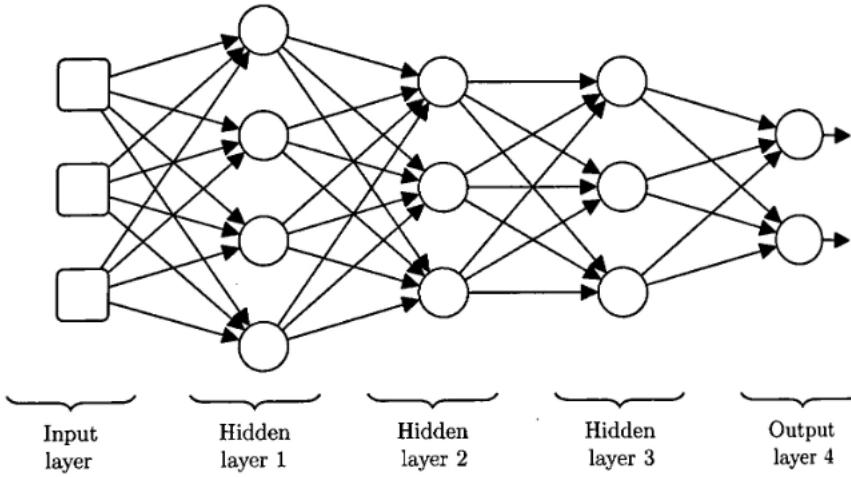


Figura 2.1: Esquema general de una red neuronal [37], que consta de una capa de unidades de entrada, una capa de unidades de salida y tres capas de unidades ocultas. Cada círculo representa una unidad de procesamiento simple, la cual incluye el producto escalar de pesos por entradas y una función de activación (generalmente no lineal), tal como se muestra en la Figura 2.2.

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

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.

Uno de los aspectos más importantes al entrenar un modelo es el sobreajuste. Este fenómeno ocurre cuando el modelo se adapta excesivamente al conjunto de datos de entrenamiento, lo que resulta en un rendimiento defi-

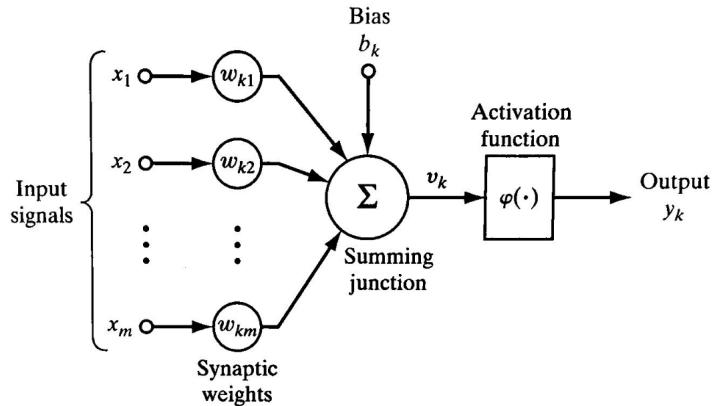


Figura 2.2: Diagrama detallado del modelo neuronal para una neurona k [40]. El diagrama muestra las señales de entrada que se multiplican por sus respectivos pesos. Estas señales ponderadas se suman junto con el sesgo, y se les aplica una función de activación para producir la salida de la neurona.

ciente al enfrentarse a nuevos datos no incluidos en el entrenamiento. Esto se debe a una limitada capacidad de generalización, que puede mitigarse mediante técnicas de regularización (Sección 2.1.5).

2.1.3. Redes neuronales convolucionales

Las redes neuronales convolucionales (*Convolutional Neural Networks*, CNN) [42, 43, 44] son un tipo de red neuronal profunda que trabaja con patrones de cuadrícula, como pueden ser imágenes (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.

A continuación, se describen los posibles tipos de capas presentes en una CNN [46, 47]:

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

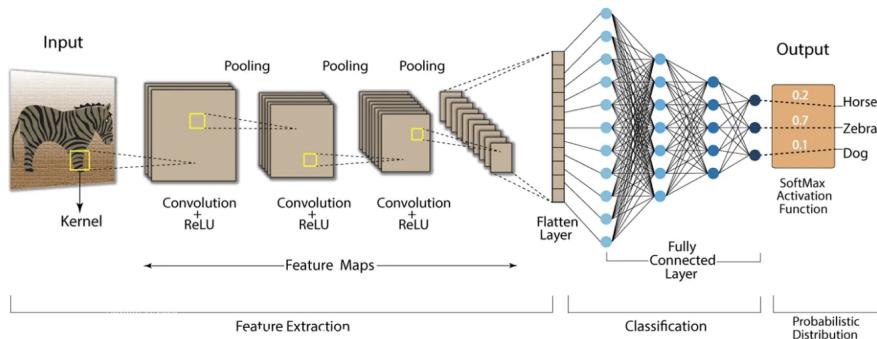


Figura 2.3: Arquitectura típica de red neuronal convolucional [45], utilizada principalmente para la tarea de clasificación de imágenes. La entrada es una imagen de un objeto. La red consta de varias capas de convolución seguidas por funciones de activación y capas de *pooling* que reducen la dimensionalidad de los mapas de características. Finalmente, las características extraídas se aplanan y se pasan a través de capas totalmente conectadas que terminan en una capa de clasificación *softmax*, proporcionando una distribución de probabilidad sobre las posibles clases de salida.

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 (Figura 2.4).

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.

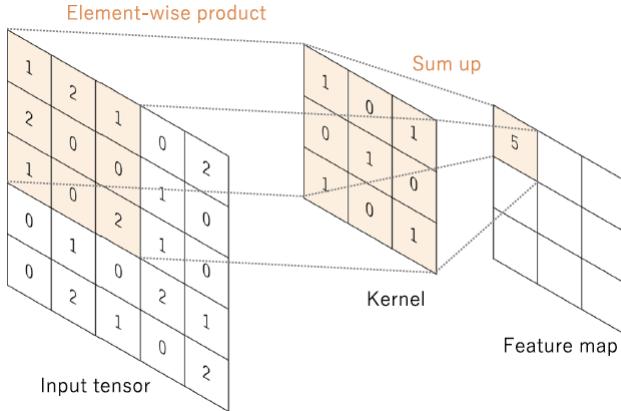


Figura 2.4: Ejemplo de operación de convolución en CNN [47]. Se muestra cómo un *kernel* se aplica sobre un tensor de entrada (una matriz de píxeles de la imagen), realizando un producto elemento a elemento. Los productos resultantes se suman para obtener un único valor en el mapa de características. Este proceso se repite para cada posición del *kernel* sobre la imagen de entrada, generando así un nuevo mapa de características.

- 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 (Figura 2.5).

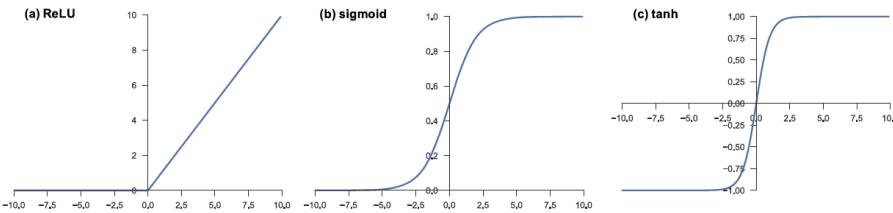


Figura 2.5: Funciones de activación comúnmente aplicadas en CNN [47]. (a) La función ReLU activa los valores positivos, estableciendo los negativos en cero. (b) La función sigmoid, que mapea los valores de entrada a un rango entre 0 y 1. (c) La función tangente hiperbólica, que mapea los valores de entrada a un rango entre -1 y 1.

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 (Figura 2.6).

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 trans-

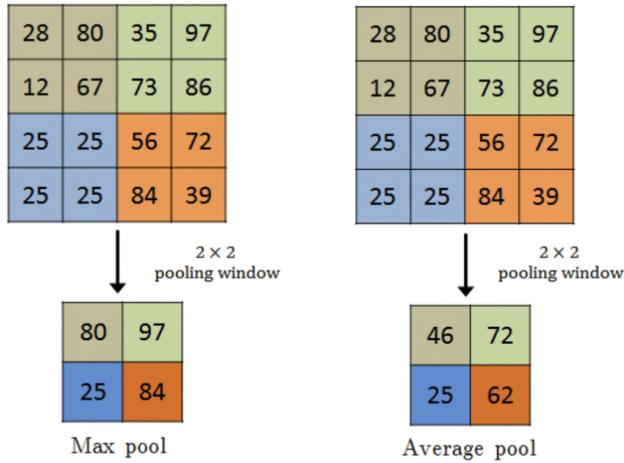


Figura 2.6: Tipos de *pooling* comúnmente utilizados en CNN [48]. En el *max pooling*, se selecciona el valor máximo de una región local de la matriz de características de entrada, mientras que en el *average pooling*, se calcula el promedio de los valores en una región local.

forman 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.

2.1.4. Transferencia de aprendizaje

La transferencia de aprendizaje [49], 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 lu-

gar 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 [50] debido a su gran tamaño, diversidad, representatividad y disponibilidad.

2.1.5. Regularización

Tanto en las redes neuronales clásicas como en las convolucionales, el sobreajuste a los datos de entrenamiento es un problema importante (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.

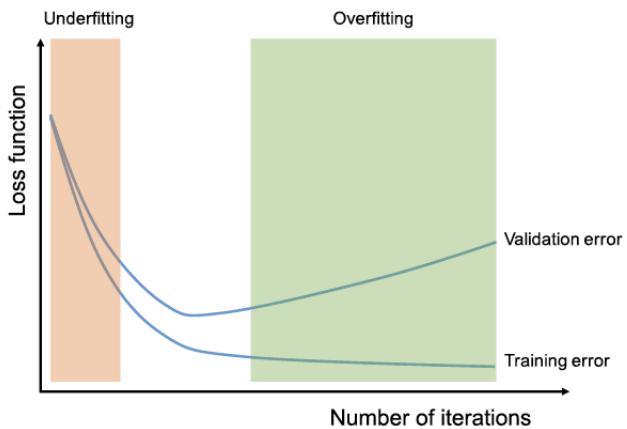


Figura 2.7: Zona de infraajuste y sobreajuste del modelodurante el entrenamiento [47]. El eje vertical representa la función de pérdida, mientras que el eje horizontal muestra el número de iteraciones durante el entrenamiento. Al inicio del proceso, tanto el error de entrenamiento como el de validación son altos, indicando que el modelo no ha aprendido lo suficiente de los datos, situación conocida como infraajuste. A medida que el entrenamiento avanza, ambos errores disminuyen hasta alcanzar un punto óptimo donde el modelo generaliza bien en datos no vistos. Sin embargo, si el entrenamiento continúa, el error de validación comienza a aumentar mientras que el error de entrenamiento sigue disminuyendo, fenómeno conocido como sobreajuste.

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.

2.2. Parámetros de la cámara

A la hora de trabajar con imágenes faciales y, particularmente, para comprender todos los factores que intervienen en el proceso de la simulación de imágenes faciales, es esencial introducir una serie de conceptos relativos a los parámetros de la cámara [51, 52, 53, 54]. Estos parámetros, como la distancia focal, el sensor de la cámara y la distancia cámara-sujeto, entre otros, están estrechamente relacionados entre sí y ejercen una influencia significativa tanto en la configuración de la escena fotográfica como en la percepción visual de los sujetos retratados en ella.

Distancia focal

La distancia focal mide la distancia, en milímetros, entre el *punto nodal* (punto donde la luz converge en una lente) y el sensor de la cámara (Figura 2.8).

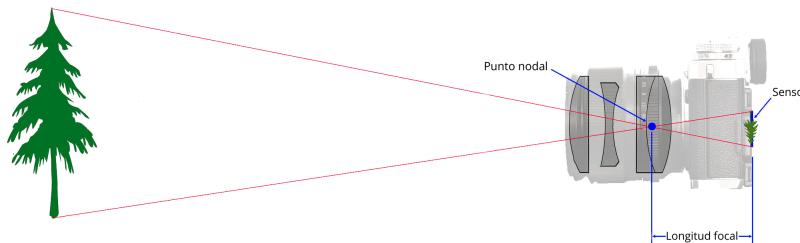


Figura 2.8: Relación entre el punto nodal y la distancia focal en un sistema óptico [55]. La imagen muestra cómo la luz que pasa por el punto nodal de la lente converge en el sensor de la cámara.

La distancia focal es un factor importante, ya que determina el campo de visión de una lente, es decir, la cantidad de escena que se captura (Figura 2.9). En distancias focales más largas, los objetos parecen estar más cerca del objetivo de la cámara, lo que puede hacer que parezcan más grandes en la imagen. Por el contrario, con distancias focales más cortas, los objetos aparentan estar más distantes en la fotografía.

Sensor de la cámara

El sensor de la cámara es el componente encargado de capturar la luz y transformarla en una imagen digital. Su tamaño incide directamente en la calidad de la imagen y en su capacidad para capturar la luz. El estándar comúnmente utilizado es de 36mm x 24mm, conocido como *full frame* o 35mm, siendo este tamaño una referencia debido a su similitud con el for-

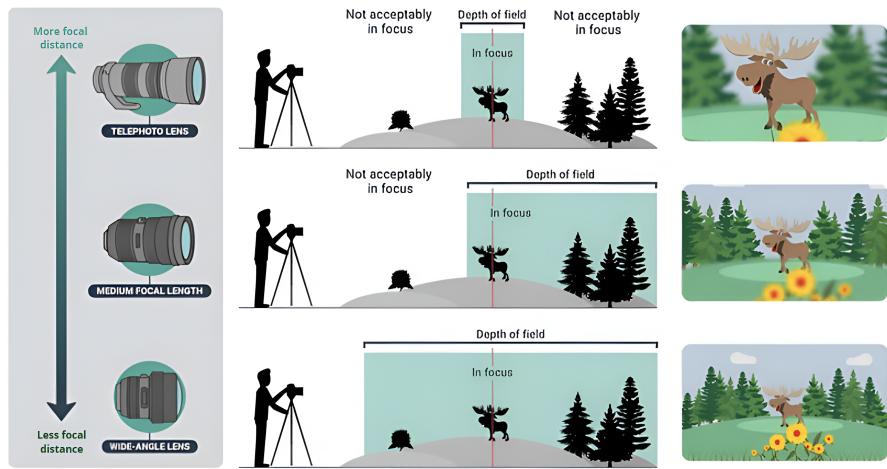


Figura 2.9: Relación entre distancia focal y campo de visión [56]. La distancia focal afecta al tamaño aparente de los objetos y a la cantidad de escena que aparece en la imagen.

mato de película fotográfica analógica utilizado en el pasado.

La necesidad de establecer un estándar es esencial para comparar equitativamente imágenes capturadas por diferentes dispositivos. Al referirnos a un estándar de 35 mm, disponemos de un sistema de referencia común que nos permite convertir las imágenes, de manera que los objetos visibles en la escena tengan dimensiones similares, facilitando así su comparación y análisis.

Otra manera de expresar el tamaño del sensor es a través del factor de recorte, que se calcula como la proporción entre el tamaño del sensor de 35 mm y el de nuestra cámara (Figura 2.10). El factor de recorte se emplea a menudo para comprender la dimensión del sensor de la cámara en relación con el estándar de 35 mm. Esta medida facilita una comparación directa entre el tamaño del sensor de 35 mm y el de nuestra cámara, lo que permite entender mejor su capacidad para capturar imágenes.

Uno de los aspectos más importantes del factor de recorte es su impacto en la distancia focal, lo que nos lleva al concepto de *distancia focal equivalente* (Figura 2.11). 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 ($300 \text{ mm} \times 1.6$) en un sensor *full frame* con factor de recorte 1.

Distancia cámara-sujeto

La distancia cámara-sujeto se define como la separación física entre la cámara y el sujeto que está siendo fotografiado (Figura 2.12). Modificar



Figura 2.10: Comparación de tamaños de sensores fotográficos expresados según el factor de recorte [57]. Los tamaños van desde medio formato hasta 1/2,3”.

esta distancia provoca variaciones en la apariencia visual del rostro en la fotografía obtenida [59]. Este fenómeno se conoce como distorsión de perspectiva.

La distorsión de perspectiva [17, 51] es la transformación que sufre un objeto y su entorno debido a la proximidad del mismo respecto al objetivo (Figura 2.13). 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 1.2).

Uno de los malentendidos comunes en fotografía es la creencia de que la distancia focal distorsiona los rasgos faciales, sin embargo, la distancia 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 [60].

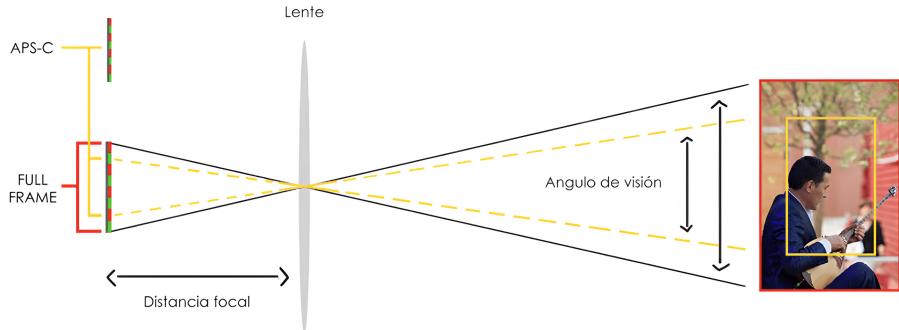


Figura 2.11: Diagrama que muestra el concepto de distancia focal equivalente en función del tamaño del sensor [58]. Se observa cómo un lente proyecta la luz en sensores de diferentes tamaños (APS-C y Full Frame), lo que modifica el ángulo de visión y, en consecuencia, la cantidad de escena capturada en la imagen resultante.

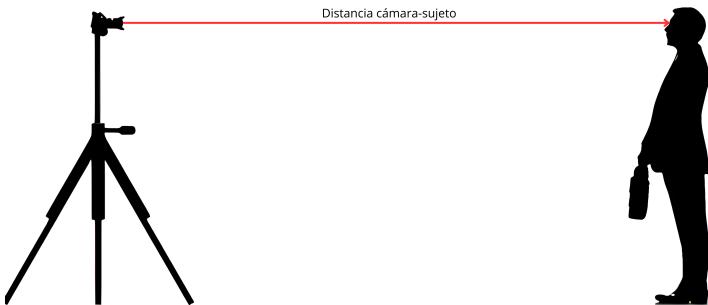


Figura 2.12: Distancia desde la cámara al sujeto. El punto de referencia utilizado para el sujeto es el centro de los ojos.

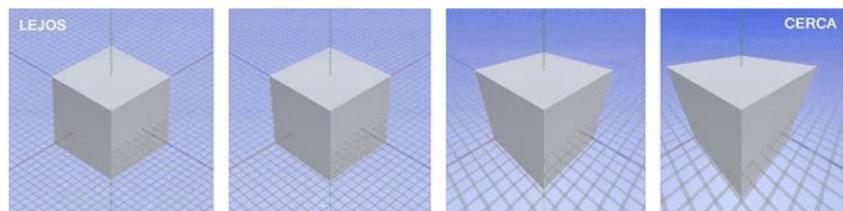


Figura 2.13: Efecto de la distorsión según la distancia al objeto [51]. De izquierda a derecha, la distancia entre la cámara y el objeto disminuye, lo que provoca un aumento progresivo de la distorsión.

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. La Figura 3.1 muestra la tendencia ascendente de publicaciones que hacen referencia a la SCD. El número alcanza 441 artículos desde 1992 indexados en la base de datos Scopus¹.

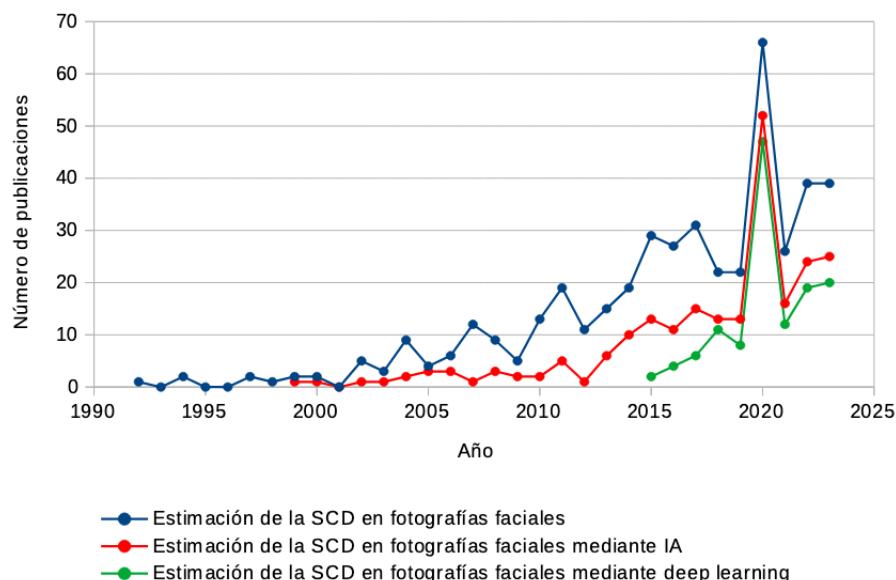


Figura 3.1: Número de publicaciones, en Scopus, relacionadas con la estimación de la distancia en fotografías faciales en función del año de publicación.

El número de publicaciones relacionadas con este tema, ha ido en aumento a lo largo del tiempo, alcanzando un mayor número de publicaciones

¹Las búsquedas se pueden consultar en el Apéndice.

en 2020. En particular, a partir de 2015, se empezaron a aplicar técnicas de aprendizaje profundo. Este cambio se debe a los avances tecnológicos que han permitido la implementación de nuevas estrategias y conocimientos.

Sin embargo, a pesar de existir 411 publicaciones relacionadas con la SCD, son escasas las investigaciones que tratan el problema específico que aborda este trabajo. La mayoría de ellas se enfocan en el desarrollo de técnicas para inferir la profundidad de los objetos capturados en la imagen, analizando meticulosamente la disposición y las relaciones entre los elementos de la escena fotográfica. Estos enfoques buscan identificar patrones visuales que revelen información sobre la distancia relativa de los objetos o individuos a la cámara, sin considerar necesariamente la relación directa con esta última.

3.1. Estimación métrica de la SCD

Uno de los primeros métodos utilizados para abordar la estimación métrica de la SCD a partir de una imagen facial, fue propuesto por Flores et al. [18], 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 [61], 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 disponibles), la mezcla de diferentes distancias focales en un mismo conjunto de datos y la necesidad de reconocimiento manual de los puntos de referencia faciales.

Posteriormente, Burgos-Artizzu et al. [19] 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) [62]. 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) [63], 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 [64], la separación entre los ojos [65], o una combinación de ambos factores [66].

3.1.1. Estimación basada en características anatómicas

En 2017, Stephan et al. [20] desarrollaron un método para la estimación de la SCD en imágenes faciales llamado PerspectiveX. Este método fue diseñado para mejorar el proceso de superposición craneofacial y 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. Sin embargo, 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.

Junto con la longitud de la fisura palpebral, PerspectiveX requiere conocer el tipo de cámara, necesario para obtener las especificaciones de píxeles, así como la distancia 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 fórmula:

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

donde: f , es la distancia 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).

A pesar de que PerspectiveX ofrece una estimación precisa de la SCD para una distancia focal conocida, también presenta ciertas limitaciones. Estas incluyen la necesidad de intervención manual para marcar los puntos de referencia faciales y la incapacidad para considerar las rotaciones de cabeza que superen los 30°. Además, se llevó a cabo un estudio para validar el algo-

ritmo, utilizando fotografías tanto de vista frontal como de perfil, tomadas con cámaras DSLR y *smartphones* [67]. Si bien los resultados fueron satisfactorios para las fotografías obtenidas con cámaras DSLR, se observaron imprecisiones notables en las tomadas con *smartphones*, tanto en la vista frontal como en la de perfil.

Posteriormente, en 2020, surge MediaPipe Iris², 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.

3.1.2. Técnicas *deep learning*

A finales de 2022, Bermejo et al. [21] presentan un novedoso 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.

Este método se compone de cuatro modelos de aprendizaje profundo basados en la arquitectura VGG-16, cada uno asociado a una distancia focal específica: 27 mm, 35 mm, 55 mm y 85 mm, respectivamente. Para entrenar estos modelos, se empleó un conjunto de datos híbrido que incluye dos colecciones: una colección sintética de aproximadamente 150.000 imágenes generadas a partir de los modelos 3D disponibles en la base de datos Stirling ESRC 3D Face³; y una colección de fotografías digitales de 28 individuos tomadas a diversas distancias, desde 50 cm hasta 6 m, y en siete posiciones

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

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

diferentes de la cabeza, desde el perfil izquierdo hasta el perfil derecho.

Este enfoque destacó por varias características clave, entre las cuales se incluye la utilización de pesos preentrenados en el conjunto de datos ImageNet [68] como punto de partida para la inicialización de los modelos, acelerando así el proceso de entrenamiento. Además, se empleó el error absoluto medio de la distorsión facial relativa como medida principal de rendimiento, lo que contribuyó a mejorar la precisión en la estimación de distancias cortas, que suelen presentar una mayor distorsión.

Los resultados obtenidos por FacialSCDnet son prometedores. En la Sección 4.2.1 analizaremos con más detalle las características de este método así como las limitaciones que justifican el desarrollo del presente TFG.

Capítulo 4

Materiales y métodos

4.1. Materiales

Dada la exigencia de grandes volúmenes de datos en los modelos de aprendizaje profundo, optaremos por la utilización de un conjunto de datos sintéticos. Para ello, se simularán imágenes fotorrealistas a partir de la selección de diversos modelos 3D, que abarcan desde modelos faciales hasta modelos de cuerpo entero. Además, se implementará un *pipeline* específico para garantizar que todos los modelos se encuentren de forma estandarizada.

4.1.1. Modelos 3D

Tras una análisis exhaustivo de las bases de datos disponibles de modelos 3D de personas, se optó por la combinación de múltiples conjuntos de datos públicos. El objetivo fue crear un conjunto de datos unificado que fuera a la vez realista y diverso. Este conjunto final incluye tanto modelos faciales como de cuerpo completo, todos ellos asociados con sus correspondientes texturas.

Modelos faciales

Se seleccionaron los siguientes conjuntos de datos: HeadSpace [69], H3DS-net [70], Stirling ESRC 3D Face ¹ y DI4D_UGR ².

El conjunto de datos de Headspace [69] 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 (Figura 4.1). Este conjunto presenta múltiples ventajas, entre las cuales destacan su excelente resolución y la inclusión de metadatos útiles que facilitan la selección de un subconjunto de datos adecuado. Sin embargo, debido a que este tipo de modelos pueden

¹Stirling ESRC 3D Face disponible en <https://pics.stir.ac.uk/ESRC/>

²Conjunto de datos proporcionado por el tutor.

introducir sesgos al no representar casos realistas, se utilizará un número limitado de ellos para complementar el conjunto de modelos final.

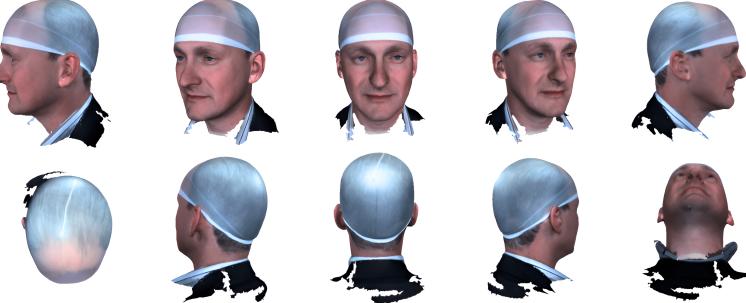


Figura 4.1: Ejemplos de modelos en HeadSpace 3D.

H3DS-net [70] contiene escaneos texturizados en 3D de la cabeza completa con una alta resolución (Figura 4.2). Este conjunto comprende un total de 23 modelos, todos ellos con los ojos cerrados, lo que proporciona una variabilidad adicional en los datos.



Figura 4.2: Ejemplos de modelos en H3DS-net.

Stirling ESRC 3D Face es una colección que contiene 99 sujetos con redecilla en el pelo (Figura 4.3). Cada uno de ellos contiene múltiples modelos faciales en 3D que capturan una variedad de expresiones faciales.



Figura 4.3: Ejemplos de modelos en Stirling ESRC 3D Face.

DI4D_UGR es un conjunto de datos adquiridos en la Universidad de Granada, concretamente en el laboratorio A!4HumanID Lab del instituto Dasci. Los modelos fueron adquiridos utilizando un dispositivo de gran calidad basado en fotoantropometría, llamado DI4D Pro. El conjunto de datos consta

de 40 sujetos, cada uno de los cuales fue escaneado mostrando diferentes expresiones faciales, tales como sonrisa, enfado, tristeza, sorpresa o neutral (Figura 4.4).



Figura 4.4: Ejemplos de modelos en DI4D-UGR.

Si bien se investigaron otros conjuntos de datos como FaceVerse [71] o CASIA³, 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 [72], People Snapshot [73] y Render People⁴.

HuMMan [72] 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 (Figura 4.5). Cada sujeto contiene 2-3 secuencias, y cada secuencia contiene aproximadamente 20 modelos. Este *dataset* destaca por la gran cantidad de poses disponibles para cada modelo.



Figura 4.5: Ejemplos de modelos en HuMMan.

People Snapshot [73] contiene 24 sujetos 3D con texturas adaptadas a diferentes situaciones tales como casual, deporte y actividades al aire libre (Figura 4.6).

³CASIA disponible en <http://biometrics.idealtest.org/>

⁴RenderPeople disponible en <https://renderpeople.com/es/>



Figura 4.6: Ejemplos de modelos en People Snapshot.

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 presentan situaciones que no se contemplan en los anteriores modelos (Figura 4.7).



Figura 4.7: Ejemplos de modelos en Render People.

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. Inicialmente, se seleccionaron 300 modelos 3D, de los cuales 206 son modelos faciales y 94 son modelos de cuerpo completo.

En cuanto a los modelos faciales, se seleccionaron 23 modelos de H3DS-net, compuestos por 13 masculinos y 10 femeninos, todos de ascendencia europea. De HeadSpace, se seleccionaron 93 modelos, con una distribución de 46 masculinos y 47 femeninos, cubriendo una amplia gama de edades y

representando diversas ascendencias, incluyendo europea, asiática, africana o mixta. Se seleccionaron 50 modelos de la base de datos Stirling ESRC 3D Face, con 43 femeninos y 7 masculinos, mayoritariamente de ascendencia europea. Por último, se seleccionaron 40 modelos de DI4D_UGR, con una distribución de 17 femeninos y 23 masculinos, todos de ascendencia europea.

Por otro lado, en cuanto a los modelos completos, se seleccionaron 2 modelos de RenderPeople, uno masculino y otro femenino, ambos de ascendencia europea. De People Snapshot, se eligieron 24 modelos, siendo 16 masculinos y 8 femeninos, todos de ascendencia europea. Finalmente, se seleccionaron 68 modelos de HuMMan, distribuidos equitativamente en 34 masculinos y 34 femeninos, con una variedad de ascendencias incluyendo asiática, africana y europea.

Estas selecciones se hicieron con el propósito de garantizar una amplia diversidad en términos de sexos biológicos, ascendencias y poses, teniendo en cuenta las bases de datos disponibles. Sin embargo, a pesar de que inicialmente la muestra era de 300 modelos 3D, después de algunas pruebas preliminares, se descartaron 23 modelos debido a su potencial impacto negativo en el aprendizaje, lo que resultó en un conjunto final de 277 modelos.

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 empleará el centro de 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 programa 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* cuyo pseudocódigo está en el Algoritmo 1.

Algoritmo 1 Generación de imágenes

```

1: for each model in models do
2:   for each d in distances do
3:     for each f in focals do
4:       for i = 0 to model_iters - 1 do
5:         background  $\leftarrow$  SELECT_RANDOM_BACKGROUND()
6:         rx  $\leftarrow$  UNIFORM(-30, 30)
7:         rz  $\leftarrow$  UNIFORM(-70, 70)
8:         APPLY_ROTATION(rx, 0, rz)
9:         value = (d/10.0) * (83.6/f)5
10:        tx  $\leftarrow$  UNIFORM(-value, value)
11:        tz  $\leftarrow$  UNIFORM(-value, value)
12:        ty  $\leftarrow$   $\sqrt{d^2 - tx^2 - tz^2}$ 6
13:        APPLY_TRANSLATION(tx, ty, tz)
14:        SET_RANDOM_ILLUMINATION()
15:      end for
16:    end for
17:  end for
18: end for
```

A continuación, se detallan las tareas específicas llevadas a cabo para cada modelo del conjunto de datos 3D:

1. Carga el modelo y lo posiciona a una distancia específica de la cámara. Se seleccionaron 35 distancias diferentes, que van desde 50 cm hasta 6 m, con incrementos graduales de 5 cm, 10 cm, 20 cm y 25 cm.
2. Posteriormente, para cada distancia, se ajusta la distancia focal de la cámara. En este caso, solo se utilizó la focal de 35 mm.

⁵Fórmula aplicada para evitar traslaciones que provoquen que el sujeto salga del cuadro de la imagen. Esta fórmula se desarrolló mediante un proceso de prueba y error.

⁶Ajuste realizado para mantener invariante la distancia cámara-sujeto tras la traslación en los ejes X y Z.

3. A continuación, para cada distancia focal, se realizan 14 iteraciones, donde en cada iteración:
- 1) Se aplica un fondo seleccionado aleatoriamente de un conjunto de 95 fondos HDR descargados de Poly Haven⁷. Algunos de estos fondos se pueden observar en la Figura 4.8.
 - 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 (Figura 4.9), así como rotaciones verticales, entre -30° y 30°, para presentar perspectivas más altas o bajas (Figura 4.10).
 - 3) Se realizan pequeñas traslaciones de la cámara para evitar que todos los modelos aparezcan centrados en la imagen, añadiendo así una variabilidad extra. Sin embargo, estas transformaciones provocan una ligera alteración de la distancia entre la cámara y el sujeto. Por ello, tras aplicar las transformaciones, se lleva a cabo un ajuste para mantener la distancia invariante.
 - 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. Este tamaño fue elegido específicamente para ajustarse a las dimensiones de entrada de los modelos de aprendizaje empleados.



Figura 4.8: Imágenes generadas con distintos fondos HDR.

⁷Poly Haven disponible en <https://polyhaven.com>



Figura 4.9: Imágenes generadas desde perspectivas más altas o más bajas. La secuencia se sigue de izquierda a derecha, mostrando rotaciones verticales que abarcan desde -30° hasta 30° en intervalos de 20° .



Figura 4.10: Imágenes generadas desde distintas perspectivas laterales. La secuencia se sigue de arriba hacia abajo y de izquierda a derecha, mostrando rotaciones horizontales que abarcan desde -70° hasta 70° en intervalos de 20° .

Tras este proceso de generación de imágenes, y considerando que se contaba con 277 modelos 3D, el conjunto total de datos ascendió a 135730 imágenes. La gran diversidad de poses, expresiones faciales, modelos y fondos del conjunto de datos se pueden observar en la Figura 4.11.

4.2. Métodos

4.2.1. FacialSCDnet

El método FacialSCDnet [21], 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, en concreto VGG-16, adaptada para regresar la distancia métrica de los individuos directamente desde las fotografías faciales. Este método consta de cuatro modelos de aprendizaje,

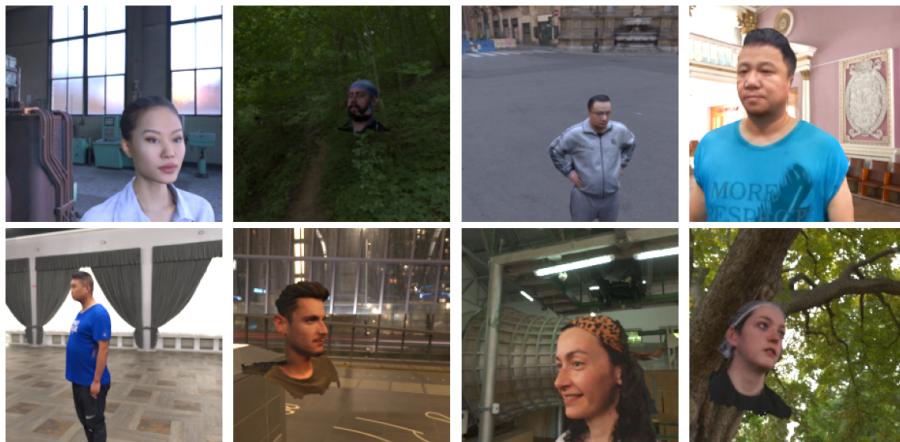


Figura 4.11: Imágenes generadas para el conjunto de datos sintético. Estos ejemplos contienen diferentes sujetos y distancias.

uno por cada distancia focal presente en el conjunto de datos: 27 mm, 35 mm, 53 mm y 83.6 mm.

Para entrenar estos modelos, 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 Face⁸. En particular, se utilizaron 315 modelos faciales de 54 individuos femeninos 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 distancias focales diferentes (27 mm, 35 mm, 55 mm, 85 mm) en formato full frame y se capturaron 12 distancias diferentes de la 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 30°.

El proceso de entrenamiento de los modelos consta de dos fases. En primer lugar, se entrena los modelos con el conjunto de datos sintéticos para capturar las relaciones entre la SCD y las características faciales. Posteriormente, se realiza un ajuste fino utilizando el conjunto de datos reales. Además, se aplica un proceso de aumento de datos que incluye la adición de diferentes fondos aleatorios a las imágenes (Figura 4.12), así como rotación,

⁸Stirling ESRC 3D Face: <https://pics.stir.ac.uk/ESRC/>

desenfoque, ruido, saturación, cambios de color e iluminación a las imágenes de entrenamiento.

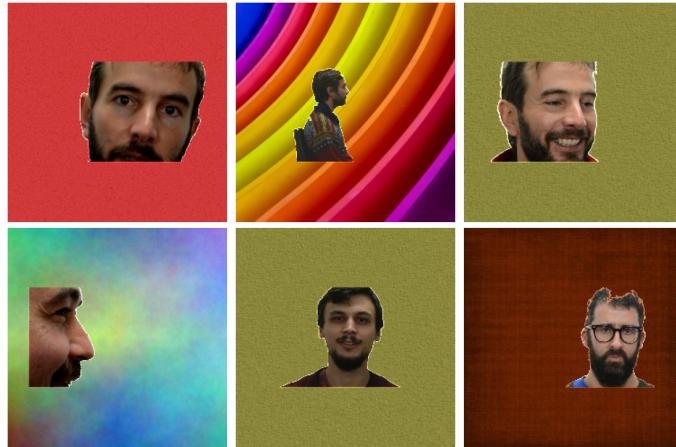


Figura 4.12: Imágenes reales con fondos utilizados en FacialSCDnet [21]. Los rostros se integraron en los fondos aplicando una máscara con la silueta facial.

Durante el análisis preliminar del método publicado en [21], se detectaron dos limitaciones de gran relevancia que impiden su aplicabilidad real:

- **Limitación del conjunto de datos:** La colección sintética consta de una única base de datos facial que está compuesta exclusivamente por individuos femeninos. Además, la colección real incluye un número reducido de individuos.
- **Sesgo sobre el preprocessamiento original:** Como hemos visto en la Figura 4.12, el preprocessamiento aplicado en [21] suponía la aplicación de una máscara de transparencia a la imagen facial con el fin de dotar al método de mayor robustez gracias a la variabilidad de fondos simulados. Sin embargo, una batería de pruebas preliminares han permitido detectar que, gracias a esta máscara, el modelo estaría aprendiendo a localizar más fácilmente la posición de los individuos en las fotografías y, por ende, sesgando los resultados. Así, FacialSCDnet tendría una capacidad limitada de generalización en un entorno real.

Estas limitaciones motivan los desarrollos propuestos en el presente TFG, como se describe en la siguiente sección.

4.2.2. FacialSCDnet+

En este trabajo se propone un enfoque alternativo que permite superar las limitaciones identificadas en el principal método del estado del arte

para la estimación de la distancia cámara-sujeto en fotografías faciales: FacialSCDnet+. Entre las mejoras se presentan dos modelos de aprendizaje diferentes, basados en las arquitecturas VGG-16 y ResNet-50, respectivamente. Ambos modelos se entrenaron con el propósito de estimar la SCD en imágenes con una distancia focal aproximada de 35mm. Además de integrar nuevas arquitecturas, FacialSCDnet+ se ha desarrollado completamente desde cero para optimizar el rendimiento y la eficiencia empleando un framework alternativo como se describirá más adelante.

Otra de las principales diferencias es el uso exclusivo de un dataset sintético (siguiendo el procedimiento descrito en la Sección 4.1.2), considerablemente más realista, diverso y extenso que el empleado en [21]. Eliminando la necesidad de realizar una segunda etapa de fine-tuning con datos reales para el entrenamiento. 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 1 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 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 imagen. 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 nuevas 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.

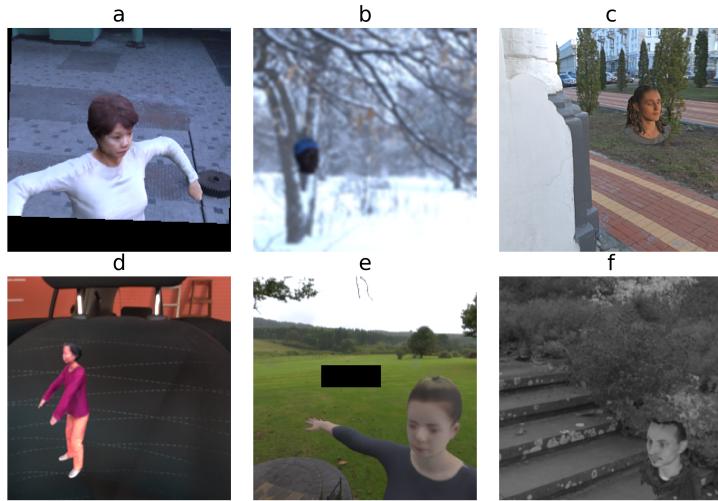


Figura 4.13: Transformaciones aplicadas a las imágenes en el conjunto de datos de FacialSCDnet+: a) transformaciones afines, b) emborronado gaussiano, c) nitidez, d) alteraciones de color, e) borrado de píxeles, f) escala de grises.

A continuación, se detallaran los aspectos técnicos de implementación más relevantes, junto con el diseño experimental del método.

Gestión de experimentos

En el método FacialSCDnet+ se ha puesto en práctica la metodología MLOps. Esta metodología ha surgido como una respuesta efectiva para mejorar la gestión de experimentos en el campo del aprendizaje automático. Al implementar MLOps, se persigue un doble objetivo: automatizar y monitorizar de manera eficiente los procesos relacionados con el desarrollo, entrenamiento y despliegue de modelos de ML, a la vez que se garantiza la coherencia y reproducibilidad de los resultados obtenidos en todo momento. Este enfoque integral no solo aumenta la eficiencia y la fiabilidad de los sistemas de IA, sino que también fomenta una cultura de colaboración, transparencia y mejora continua en los equipos de desarrollo y operaciones.

Para llevar a cabo esta metodología, se ha hecho uso de la librería *MLflow*, una herramienta que facilita la gestión y el seguimiento del proceso

de desarrollo de modelos de aprendizaje automático. Esta librería no solo proporciona un entorno unificado para organizar experimentos y compartir resultados, sino que también ofrece capacidades avanzadas de seguimiento y visualización de métricas, parámetros y artefactos asociados con cada iteración del proceso de desarrollo.

Optimización de hiperparámetros

El ajuste de hiperparámetros es esencial en el desarrollo de modelos de ML, ya que éstos controlan el comportamiento y la complejidad del algoritmo de aprendizaje. La optimización adecuada puede mejorar significativamente el rendimiento del modelo en términos de precisión, velocidad de entrenamiento y capacidad de generalización. Además, el proceso de ajuste de hiperparámetros permite explorar el espacio de búsqueda de manera sistemática, lo que proporciona una comprensión más profunda del comportamiento del modelo y sus interacciones con los datos.

Con el objetivo de realizar la optimización de hiperparámetros, se ha optado por utilizar *Optuna*, una librería de optimización de hiperparámetros automatizada. Esta herramienta ofrece una manera eficiente de encontrar los mejores valores para los hiperparámetros de un modelo de aprendizaje automático. En particular, se emplea el algoritmo conocido como *Tree-structured Parzen Estimator* (TPE) [74], el cual es una técnica de muestreo que ajusta una distribución probabilística a los datos recopilados durante la búsqueda para dirigir la exploración hacia regiones prometedoras del espacio de búsqueda. Además, se utiliza una técnica de poda para descartar de manera temprana las configuraciones de hiperparámetros menos prometedoras, lo que contribuye a acelerar el proceso de búsqueda y mejorar su eficiencia.

Cambio de *framework*

El método FacialSCDnet fue desarrollado utilizando *Keras*, uno de los principales *frameworks* en *deep learning*. *Keras* es una biblioteca de código abierto que fue adoptada e integrada en *Tensorflow* a mediados de 2017. A pesar de su popularidad, *Keras* está escrita en alto nivel, lo que implica una mayor facilidad de uso pero un menor rendimiento en cuanto a velocidad. Esta biblioteca es una excelente opción para conjuntos de datos pequeños o prototipos rápidos, ya que permite construir, entrenar y evaluar modelos de manera rápida. Sin embargo, debido al gran volumen de datos y experimentos en este proyecto, *Keras* puede quedarse rezagado en cuanto a rendimiento.

Por esta razón, en FacialSCDnet+ se modifica el *framework* utilizado. En concreto, se realiza una transición desde la versión 1.15 de *TensorFlow*, desarrollada en 2019, a la versión 2.0.1 de *PyTorch*, lanzada en 2023. Esta librería, relativamente reciente, cuenta con un excelente soporte comunitario

y un desarrollo activo. Entre sus principales ventajas se incluyen:

- Flexibilidad: *PyTorch* ofrece un control más granular sobre cada aspecto del modelo gracias a su API de bajo nivel.
- Simplicidad: A pesar de su bajo nivel de operación, *PyTorch* se siente natural, lo que facilita la programación y la hace más intuitiva.
- Depuración: *PyTorch* facilita la depuración de modelos gracias a su estructura dinámica de grafos computacionales, lo que permite una mejor visualización y seguimiento de errores.
- Eficiencia en el uso de memoria: *PyTorch* optimiza el uso de memoria a través de técnicas como la gestión de tensores y el cálculo diferencial automático, permitiendo un uso más eficiente de los recursos disponibles.

Protocolo de validación experimental

La técnica empleada para llevar a cabo el entrenamiento se conoce como *hold-out* (Figura 4.14). 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.

El uso de la metodología *hold-out* en lugar de *cross-validation* se debe a dos razones principales: el alto costo de tiempo asociado al entrenamiento de los modelos y el hecho de ser el protocolo empleado en el trabajo de referencia.

El primer paso de esta metodología 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, 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, contamos con 135730 imágenes, de las cuales 85435 se destinan al entrenamiento, 21360 a la validación y 28935 a test.

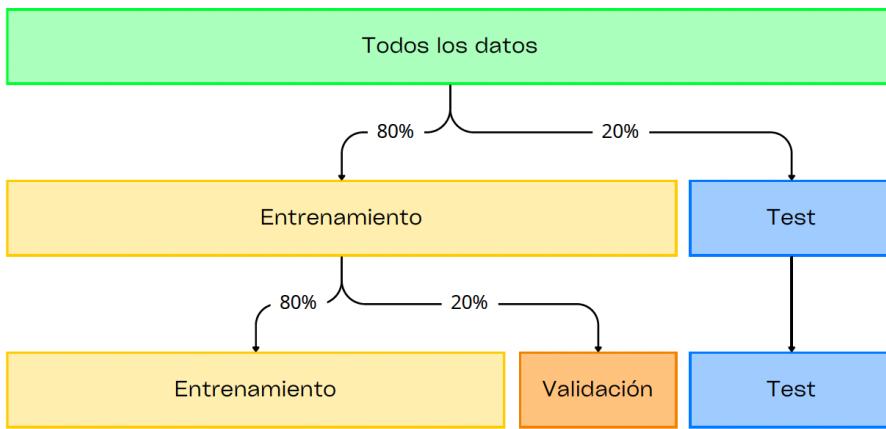


Figura 4.14: Esquema de división del conjunto de datos total en los subconjuntos de entrenamiento, validación y test.

Métricas

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

$$\text{Distorsion} = \frac{\sum_{i=1}^n \left| \frac{1}{1 + \frac{y_i}{d}} - \frac{1}{1 + \frac{x_i}{d}} \right|}{n} \quad (4.1)$$

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 [15] para obtener experimentalmente el factor de distorsión de una cabeza humana de tamaño promedio, según la SCD de la fotografía.

En cuanto a la interpretabilidad de esta métrica, podremos considerar que la predicción de la distancia obtenida es aceptable cuando baje del 1% de error. Este umbral se ha definido como el máximo admisible para realizar comparaciones en contextos forenses [15].

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:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad (4.2)$$

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - x_i|}{y_i} \quad (4.3)$$

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.

4.2.3. Backend

VGG-16

VGG-16 es una red neuronal convolucional profunda basada en la arquitectura VGGNet [75]. Esta red contiene 16 capas entrenables, como su nombre indica, y destaca por su eficacia en la extracción de características en imágenes.

La arquitectura VGG-16 se puede ver en la Figura 4.15. Inicialmente, 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 realizar 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.

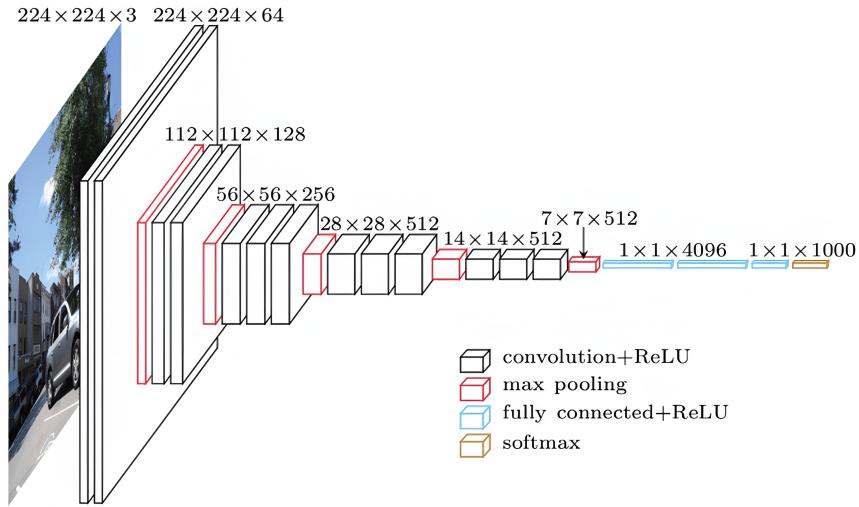


Figura 4.15: Arquitectura de la red VGG-16. Las dimensiones se muestran en formato: Columnas x Filas x Canales [76].

Las arquitecturas VGGnet destacan por emplear convoluciones únicamente de tamaño 3x3. Este enfoque representó un importante avance con respecto a las arquitecturas predecesoras, ofreciendo diversas ventajas significativas:

- Mayor profundidad: Permite aplicar más convoluciones e incrementar la profundidad de la red al tener menos parámetros entrenables.
- Agregación implícita de escalas: 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.

En FacialSCDnet y FacialSCDnet+, se realizó una modificación estructural en la red originalmente diseñada para clasificación. Se mantuvo la extracción de características a través de los 5 bloques convolucionales, aprovechando los pesos preentrenados en ImageNet [68]. Sin embargo, se eliminó la parte superior de la red y se agregaron 2 capas totalmente conectadas de 4096 neuronas, junto con una capa final de una neurona configurada para llevar a cabo la tarea de regresión.

Tras realizar estos ajustes en la estructura de la red, se congelaron los pesos de los bloques convolucionales, mientras que la parte final de la red se entrenó desde cero. En total, la red cuenta con 119,545,857 parámetros para el entrenamiento.

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

ResNet-50

ResNet-50 es una red neuronal convolucional profunda que pertenece a la familia de las redes residuales [77] (ResNets). Estas redes destacan por la introducción de conexiones “residuales”, diseñadas para evitar el desvanecimiento del gradiente, uno de los principales problemas de las CNN profundas. Este problema surge durante el entrenamiento de las redes neuronales cuando se emplean métodos basados en descenso estocástico de gradientes y retropropagación. En concreto, ocurre cuando los gradientes de la función de error con respecto a los pesos de la red se vuelven excesivamente pequeños, lo que dificulta la actualización de dichos pesos durante el proceso de aprendizaje. Esta situación puede interrumpir el aprendizaje, especialmente en redes profundas con múltiples capas.

En las ResNets, en lugar de simplemente apilar capas una sobre otra, se añaden conexiones directas que saltan una o más capas (Figura 4.16). Estas conexiones de atajo permiten que la red aprenda las diferencias entre la representación deseada y la representación actual, en lugar de tener que aprender la representación completa en cada capa. Este enfoque permite aumentar la profundidad de la red sin que su rendimiento se vea afectado.

La arquitectura ResNet-50 se puede observar en la Figura 4.17. Esta consta de 50 capas, incluyendo capas convolucionales, capas de *pooling* y capas totalmente conectadas. Inicialmente, la imagen de entrada se procesa a través de una capa de convolución seguida de una capa de agrupamiento. A esta capa le siguen varias capas de bloques residuales, cada uno de ellos consta de múltiples capas de convolución

- Entrada: La imagen de entrada se procesa inicialmente a través de una capa de convolución seguida de una capa de agrupamiento promedio.
- Bloques de convolución: ResNet-50 tiene varias capas de bloques residuales. Estos bloques se repiten varias veces, y cada uno consta de múltiples capas de convolución. Aquí es donde se incorporan las conexiones “residuales”, la entrada de cada bloque se añade a la salida del bloque.
- Capa de *pooling*: Tras los bloques convolucionales hay una capa de agrupamiento máximo.

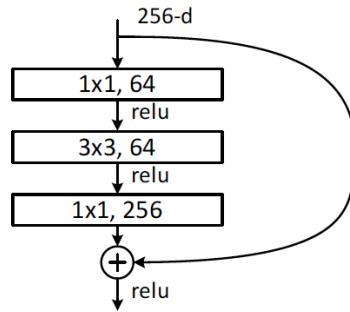


Figura 4.16: Bloque *bottleneck* construido para las ResNet-50/101/152 [78]. Este bloque consiste en tres capas de convolución: una convolución 1x1 que reduce la dimensionalidad a 64 canales, seguida de una convolución 3x3, que aprende las características espaciales, y otra convolución 1x1 que restaura la dimensionalidad original de 256 canales. Todas las capas de convolución usan activaciones ReLU. La conexión de salto agrega la entrada del bloque directamente a la salida, antes de pasar por una activación ReLU final.

- Capas totalmente conectadas: Por último, las características se pasan a un vector unidimensional y se conectan a través de una capa totalmente conectada para generar las predicciones finales.

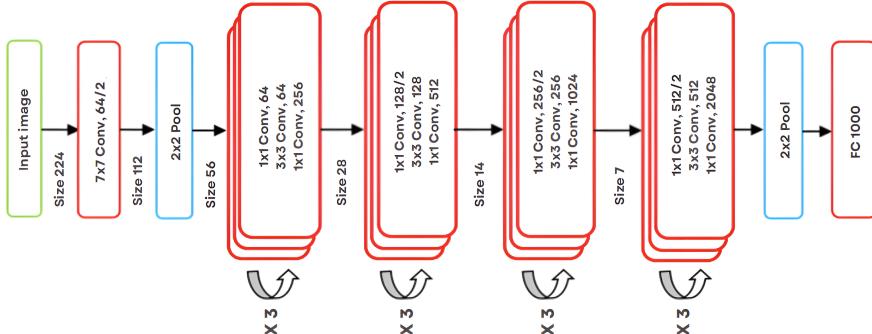


Figura 4.17: Arquitectura de la red ResNet-50 [78].

En FacialSCDnet+, al igual que con VGG-16, se realizó una modificación estructural en la red originalmente diseñada para clasificación. Se mantuvo la extracción de características a través de los bloques residuales, aprovechando los pesos preentrenados en ImageNet. Sin embargo, se prescindió de la última capa de la red y se agregaron 2 capas totalmente conectadas de 4096 neuronas, junto con una capa final de una neurona configurada para llevar a cabo la tarea de regresión.

Tras realizar estos ajustes en la estructura de la red, se congelaron los pesos de los bloques convolucionales, mientras que la parte final de la red se entrenó desde cero. En total, la red cuenta con 25,178,113 parámetros para el entrenamiento.

La implementación de conexiones residuales en ResNet-50 posibilita la construcción de una red más profunda, con un gran conjunto de capas convolucionales. Esta estructura aumenta su capacidad para aprender características complejas en las imágenes. A pesar de no alcanzar la cantidad de parámetros de VGG-16, su profundidad implica una demanda notable de recursos computacionales. No obstante, su alto rendimiento en tareas de visión la posicionan como una opción atractiva para este proyecto.

Capítulo 5

Experimentos

5.1. Entorno de desarrollo

El proyecto se llevó a cabo utilizando exclusivamente el lenguaje de programación *Python* en su versión 3.11.8, debido a su versatilidad y eficacia en diversas áreas, desde el renderizado de modelos 3D en *Blender* hasta el desarrollo de modelos de aprendizaje profundo. Se emplearon diversas bibliotecas para diferentes tareas: *Trimesh* y *PyVista* junto con *VTK* para la 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* en su versión 2.0.1 junto con las librerías CUDA para el entrenamiento de modelos de aprendizaje profundo.

Con el objetivo de realizar un control de versiones durante el desarrollo del proyecto, se utilizó Git junto con GitHub. El código generado se puede encontrar en el siguiente enlace <https://github.com/ivansalinasugr/TFG>. Para más información, consultar el Readme del repositorio.

5.2. Entorno de ejecución

El proceso de ejecución se lleva a cabo en un entorno de alto rendimiento ubicado en la Universidad de Granada, al que se accede de forma remota a través de SSH. Se emplea un script de Shell para configurar los parámetros esenciales de los archivos. Por un lado, se utiliza SLURM para reservar recursos en la partición “dios” del clúster, asignando una GPU del nodo “dionisio”. Este nodo cuenta con dos Quadro RTX 8000, una memoria RAM de 512 GB DDR4 y dos procesadores Intel Xeon Silver 4216. Por otro lado, Conda se encarga de gestionar el entorno de software, garantizando la disponibilidad de las bibliotecas necesarias durante el proceso.

5.3. Resultados

En este trabajo se realizarán varios experimentos para entrenar y validar los dos modelos de aprendizaje presentados en FacialSCDnet+. Antes de los entrenamientos, se llevará a cabo un ajuste de parámetros para optimizar el rendimiento de los modelos. Finalmente, se compararán los resultados obtenidos con los obtenidos por FacialSCDnet. Estas comparaciones se efectuarán mediante el rendimiento en dos conjuntos de imágenes: el conjunto de test sintético de FacialSCDnet+, y el conjunto real de FacialSCDnet. Esto nos permitirá evaluar la robustez y la capacidad de generalización de los modelos propuestos.

El objetivo de estos experimentos es validar la hipótesis de que los modelos presentados en este trabajo pueden igualar o superar el rendimiento del modelo de referencia y por tanto, estimar mejor la SCD en fotografías faciales. A continuación, se presentan los detalles específicos y los resultados de estos experimentos.

5.3.1. Ajuste de hiperparámetros

Inicialmente, se llevó a cabo una fase de ajuste de hiperparámetros con el fin de determinar una configuración óptima para las arquitecturas propuestas. Los rangos de valores y los parámetros finales se detallan en la Tabla 5.1. Además de los hiperparámetros mencionados, se estableció un entrenamiento a lo largo de 300 épocas, con una tasa de aprendizaje mínima de 10^{-12} y una reducción de la tasa de aprendizaje del 20% cada 3 épocas consecutivas sin mejoras (pacienza), hasta un mínimo de 10^{-12} .

Parámetros	Opciones	Mejor VGG-16	Mejor ResNet-50
Optimizador	[adam, sgd]	adam	adam
Tasa de aprendizaje	$[10^{-6}, 10^{-3}]$	$4.63 \cdot 10^{-5}$	0.00042
Tamaño del lote	[16, 32, 64, 128]	32	32
Paciencia	[2, 3, 4]	3	3
<i>Early Stopping</i>	[4, 6, 8]	6	6
<i>Dropout (%)</i>	[0, 10, 20, 30]	0	0

Tabla 5.1: Parámetros de entrenamiento seleccionados para las redes VGG-16 y ResNet-50, junto a los rangos de valores utilizados durante el proceso de optimización de hiperparámetros.

Los detalles de implementación de este tuneo de hiperparámetros se pueden observar en la Sección 4.2.2.

5.3.2. Comparativa Arquitecturas

A continuación, para valorar el comportamiento de las diferentes arquitecturas propuestas, compararemos los resultados obtenidos para Fa-

cialSCDnet+ usando el conjunto de validación. La Figura 5.1 muestra la gráfica de la función de pérdida durante el entrenamiento de ambas arquitecturas, mientras que la Tabla 5.2 muestra los valores finales de las métricas en el conjunto de entrenamiento de validación tras finalizar el entrenamiento.

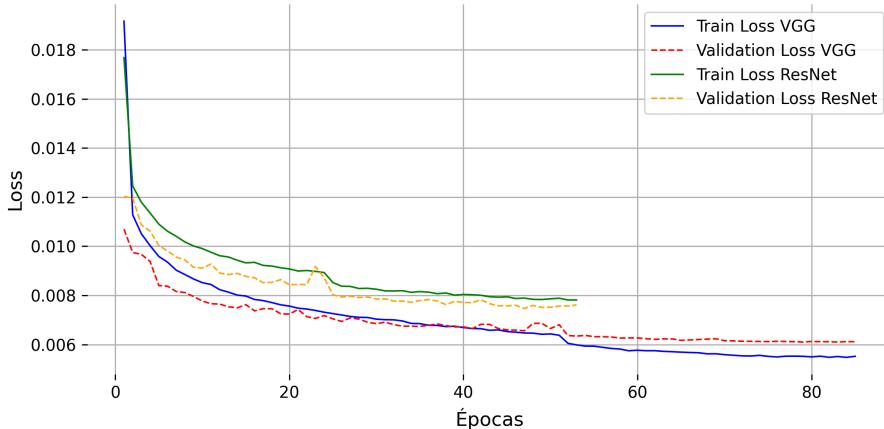


Figura 5.1: Gráfica de la función de pérdida durante el entrenamiento de las redes VGG-16 y ResNet-50. Para la red VGG se representa en azul la pérdida en el conjunto de entrenamiento mientras que en rojo se representa la pérdida en el conjunto de validación. Para la red ResNet se representa en verde la pérdida en el conjunto de entrenamiento mientras que en naranja se representa la pérdida en el conjunto de validación.

Ambas arquitecturas comienzan con una pérdida alta que disminuye rápidamente durante las primeras épocas, lo cual es esperado ya que los modelos empiezan a ajustarse a los datos de entrenamiento. A medida que avanza el entrenamiento, ambos modelos van aprendiendo de los datos y por tanto, las pérdidas de entrenamiento y de validación siguen disminuyendo.

VGG-16 muestra una convergencia más suave y una pérdida más baja tanto en entrenamiento como en validación en comparación con ResNet-50. Aunque ResNet-50 comienza bien, alcanza una meseta en la reducción de la pérdida de entrenamiento más rápidamente que VGG-16. Por tanto, la arquitectura VGG-16 tiene un mejor rendimiento en términos de pérdida tanto en entrenamiento como en validación en comparación con ResNet-50.

Además, ambas arquitecturas finalizan el entrenamiento debido al *early stopping*, tras seis épocas sin mejorar. La aplicación de esta técnica parece haber sido efectiva, ya que no se observa una gran divergencia entre las pérdidas de entrenamiento y validación hacia el final de la gráfica. Esto indica que los modelos no están sobreajustando significativamente y mantienen un buen equilibrio entre el ajuste a los datos de entrenamiento y la capacidad de generalización.

Métricas de evaluación		Validación VGG-16	Validación ResNet-50
Distorsión (%)	Media (std)	0.61 (0.614)	0.746 (0.722)
	Mediana	0.434	0.532
	Perc 90, 95, 99	[1.356, 1.747, 2.742]	[1.692, 2.176, 3.266]
MAE (cm)	Media (std)	31.425 (44.048)	34.072 (43.447)
	Mediana	12.194	15.597
	Perc 90, 95, 99	[89.95, 125.106 , 204.385]	[93.849, 126.878, 194.479]
MRE (%)	Media (std)	0.111 (0.138)	0.128 (0.137)
	Mediana	0.075	0.093
	Perc 90, 95, 99	[0.233, 0.328, 0.647]	[0.266, 0.356, 0.659]
R ²		0.897	0.894

Tabla 5.2: Métricas en el conjunto de validación tras el proceso de entrenamiento de las redes VGG-16 y ResNet-50.

A pesar de que los valores del MAE son relativamente altos, ambos modelos muestran un error medio de distorsión menor al 1% y un error relativo medio muy bajo, lo cual es un valor aceptable para que la predicción sea fiable. En este contexto, VGG-16 presenta una distorsión ligeramente más baja que ResNet-50, lo que indica una mejor predicción de las distancias.

En general, aunque ambos modelos tienen buenos resultados, sobre todo en términos de distorsión, VGG-16 destaca con mejor rendimiento que ResNet-50 en todas las métricas.

5.3.3. Comparativa con el estado del arte: FacialSCDnet

En esta sección se lleva a cabo una comparativa entre los tres modelos utilizados para la estimación de la SCD: VGG-16 y ResNet-50 de FacialSCDnet+ y VGG-16 de FacialSCDnet. Para ello, se evalúan los modelos en dos conjuntos de datos: el conjunto de test sintético propuesto en este trabajo, previamente definido en la Sección 4.2.2 y un conjunto de test de imágenes reales propuesto en [21]. Así, se pretende proporcionar una comparativa en igualdad de condiciones para ambos métodos.

Aunque no se deberían utilizar los conjuntos de test para tomar decisiones metodológicas debido al riesgo de *data snooping* (lo que puede llevar a un sobreajuste a los datos de test), en este trabajo seguiremos el protocolo del estudio de referencia [21] y de muchos otros estudios, utilizando los conjuntos de test para evaluar y comparar el rendimiento de los modelos.

Para hacer la comparación más visual, se complementarán las métricas con figuras que muestren las predicciones comparadas con los valores objetivo, así como los errores de predicción representados en diagramas de caja.

Test imágenes sintéticas

Este conjunto de datos contiene 28935 imágenes sintéticas en 35 distancias desde 50 a 600 cm. Los resultados de evaluar los modelos en este

conjunto de datos se muestran en la Tabla 5.3.

Métricas de evaluación	VGG-16	VGG-16 FSCDnet+	ResNet-50	
Distorsión (%)	Media (std) Mediana Perc 90, 95, 99	1.218 (1.395) 0.764 [2.878, 3.893, 6.826]	0.624 (0.596) 0.453 [1.381, 1.758 , 2.639]	0.781 (0.754) 0.55 [1.782, 2.265, 3.445]
MAE (cm)	Media (std) Mediana Perc 90, 95, 99	53.237 (73.434) 21.594 [147.604, 220.045, 338.481]	32.165 (44.045) 12.771 [92.037, 127.188 , 200.073]	35.095 (43.954) 16.498 [96.793, 130.566, 193.916]
MRE (%)	Media (std) Mediana Perc 90, 95, 99	0.215 (0.299) 0.132 [0.455, 0.681, 1.567]	0.113 (0.135) 0.078 [0.24, 0.327 , 0.64]	0.133 (0.14) 0.099 [0.277, 0.362, 0.669]
R ²	0.673	0.895	0.89	

Tabla 5.3: Métricas en el conjunto de test sintético de FacialSCDnet+, comparando los modelos VGG-16 y ResNet-50 de FacialSCDnet+ contra el modelo de FacialSCDnet.

Esta tabla refleja cómo VGG-16 FacialSCDnet+ presenta mejores resultados en la mayoría de las métricas, seguido por ResNet-50 FacialSCDnet+ y VGG-16 FacialSCDnet. Estos resultados sugieren que el método FacialSCDnet+ proporciona una mejora significativa en el rendimiento de la predicción en comparación con el método FacialSCDnet, y se reafirma la conclusión obtenida en la comparativa anterior, donde la arquitectura de VGG-16 se comporta mejor que ResNet-50 en este contexto específico.

Ambos modelos de FacialSCDnet+ presentan una distorsión inferior al 1 %, lo cual se considera un margen de error aceptable, incluso a pesar de que muestran un MAE elevado. Visualmente, podemos analizar la distribución del error en la Figura 5.2, donde se compara el error en las predicciones mediante gráficos de caja.

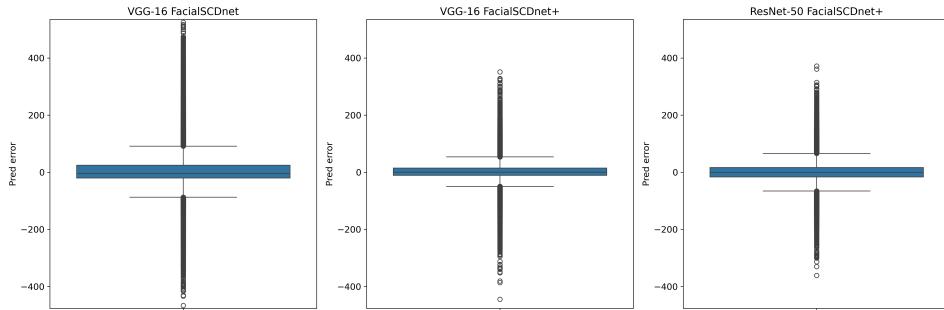


Figura 5.2: Gráfica que muestra la comparación de las predicciones de error para cada uno de los 3 modelos utilizados.

En estas gráficas se puede observar cómo VGG-16 FacialSCDnet muestra la mayor variabilidad y la mayor cantidad de outliers, lo que sugiere que este modelo enfrenta más dificultades para hacer predicciones precisas en muchos casos. Los dos modelos de FacialSCDnet+ mejoran en comparación con FacialSCDnet, reduciendo tanto la variabilidad como la cantidad

de outliers, lo que indica un desempeño más consistente. En particular, el modelo VGG-16 FacialSCDnet+ parece tener menor variabilidad, mientras que ResNet-50 presenta la menor cantidad de outliers.

En la Figura 5.3 se pueden observar predicciones de la distancia para algunas de las imágenes del conjunto sintético.



Figura 5.3: Ejemplos de predicciones en imágenes sintéticas para los distintos modelos VGG-16 y ResNet-50 de FacialSCDnet+ y VGG-16 de FacialSCDnet.

Es importante destacar que los modelos de FacialSCDnet+ parten con cierta ventaja en esta comparativa, ya que han sido entrenados en un conjunto similar al empleado en este test. Por esta razón, a continuación se evaluarán los modelos en un conjunto de test compuesto únicamente de imágenes reales.

Test imágenes reales

Este conjunto de datos contiene 1134 imágenes reales en 29 distancias desde 50 a 500 cm. Los resultados de evaluar los modelos en este conjunto de datos se muestran en la Tabla 5.4.

Esta tabla refleja cómo VGG-16 FacialSCDnet+ presenta mejores resultados en todas las métricas, seguido por ResNet-50 FacialSCDnet+ y VGG-16 FacialSCDnet. Esto sugiere que el método FacialSCDnet+ proporciona una mejora significativa en el rendimiento de la predicción en comparación con el método FacialSCDnet estándar, y que el modelo VGG-16 es más

Métricas de evaluación	VGG-16 FSCDnet	VGG-16 FSCDnet+	ResNet-50 FSCDnet+	
Distorsión (%)	Media (std) Mediana Perc 90, 95, 99	1.233 (1.615) 0.77 [3.005, 3.869, 8.389]	0.702 (0.727) 0.477 [1.637, 2.195, 3.459]	1.257 (0.993) 1.085 [2.59, 3.136, 4.108]
MAE (cm)	Media (std) Mediana Perc 90, 95, 99	31.594 (55.366) 9.994 [98.136, 154.061, 288.525]	26.05 (48.391) 8.69 [71.072, 117.413, 261.572]	39.852 (58.581) 16.233 [130.143, 179.453, 257.784]
MRE (%)	Media (std) Mediana Perc 90, 95, 99	0.187 (0.33) 0.103 [0.409, 0.58, 1.61]	0.098 (0.101) 0.067 [0.212, 0.3, 0.523]	0.161 (0.126) 0.137 [0.342, 0.409, 0.523]
R ²		0.786	0.829	0.624

Tabla 5.4: Métricas en el conjunto de test real de FacialSCDnet, comparando los modelos VGG-16 y ResNet-50 de FacialSCDnet+ contra el modelo de FacialSCDnet.

efectivo que el ResNet-50 en este contexto específico.

El modelo VGG-16 de FacialSCDnet+ es el único que presenta una distorsión inferior al 1 %. Aunque todavía tiene un MAE ligeramente alto, este nivel de error en la distorsión se considera aceptable.

En este contexto, es notable destacar un error cometido en el método FacialSCDnet. A la hora de evaluar el modelo con las imágenes reales, se les aplicaba una máscara para añadirles un fondo sintético. Esto provocaba que el rendimiento del modelo aumentara artificialmente, ya que la red neuronal internamente estaba aprendiendo a localizar a los individuos en fotografías lejanas gracias al preprocesamiento. Así, se estaba cometiendo un sesgo significativo debido a la aplicación de máscaras. Los resultados pueden comprobarse en la publicación original [21], donde se muestran resultados que reducen una sexta parte del error real sin preprocesamiento.

La Figura 5.4 muestra una comparación de las predicciones de error para los tres modelos utilizando gráficos de caja.

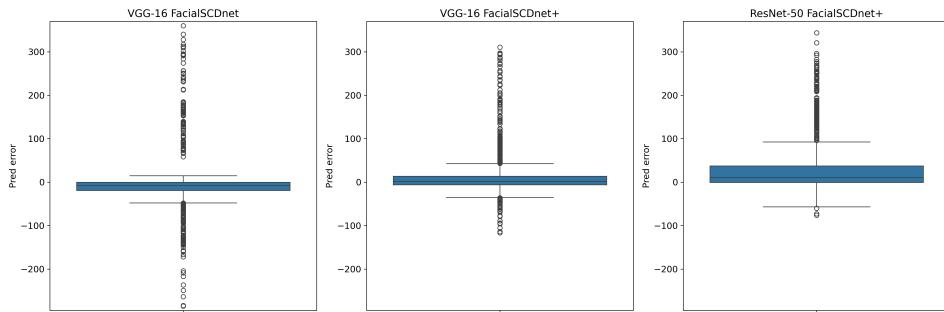


Figura 5.4: Gráfica que muestra la comparación de las predicciones de error para cada uno de los 3 modelos utilizados.

Se observa que VGG-16 FacialSCDnet presenta la mayor cantidad de outliers, lo que sugiere que este modelo tiene más dificultades para hacer

predicciones precisas en ciertos casos. Además, este modelo tiende tanto a sobreestimar como a infraestimar las predicciones. Por otro lado, ambos modelos de FacialSCDnet+ muestran una menor cantidad de outliers, aunque tienen una mayor variabilidad, especialmente el ResNet-50 FacialSCDnet+. El modelo VGG-16 FacialSCDnet+, a pesar de tener una variabilidad ligeramente mayor que el VGG-16 FacialSCDnet, presenta menos outliers, lo cual sugiere que es el modelo más consistente de los presentados.

En la Figura 5.5 se pueden observar predicciones de la distancia para algunas de las imágenes del conjunto real.

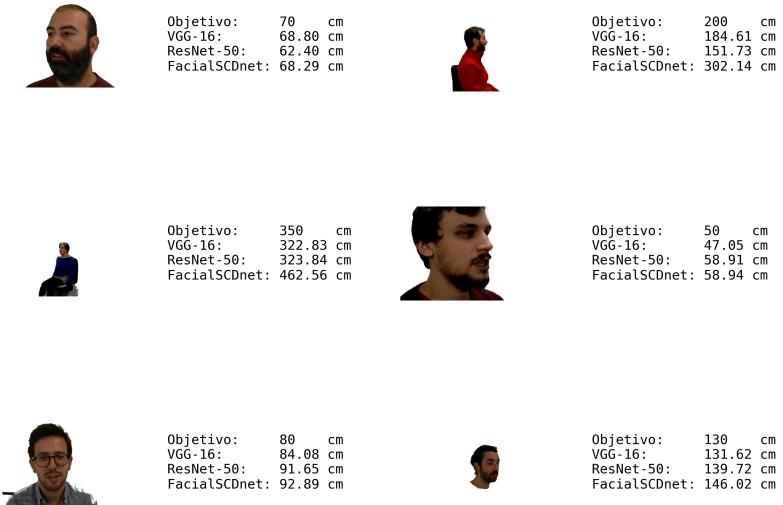


Figura 5.5: Ejemplos de predicciones en imágenes reales para los distintos modelos VGG-16 y ResNet-50 de FacialSCDnet+ y VGG-16 de FacialSCDnet.

La Figura 5.6 representa una comparativa del rendimiento de los métodos a la hora de predecir la SCD, de forma que podemos observar la dispersión real de las estimaciones con respecto a la SCD real. La primera fila muestra los resultados en test de los tres métodos comparados para el conjunto de imágenes sintéticas, mientras que la segunda fila muestra los resultados sobre el conjunto de imágenes reales. La principal diferencia entre ambos conjuntos de datos es evidente, el número de imágenes sintéticas que podemos generar permite cubrir un amplio rango de distancias en comparación con las imágenes reales. Esto nos permite identificar el rendimiento real de los modelos.

En línea con los resultados cuantitativos mostrados en las Tablas 5.3 y

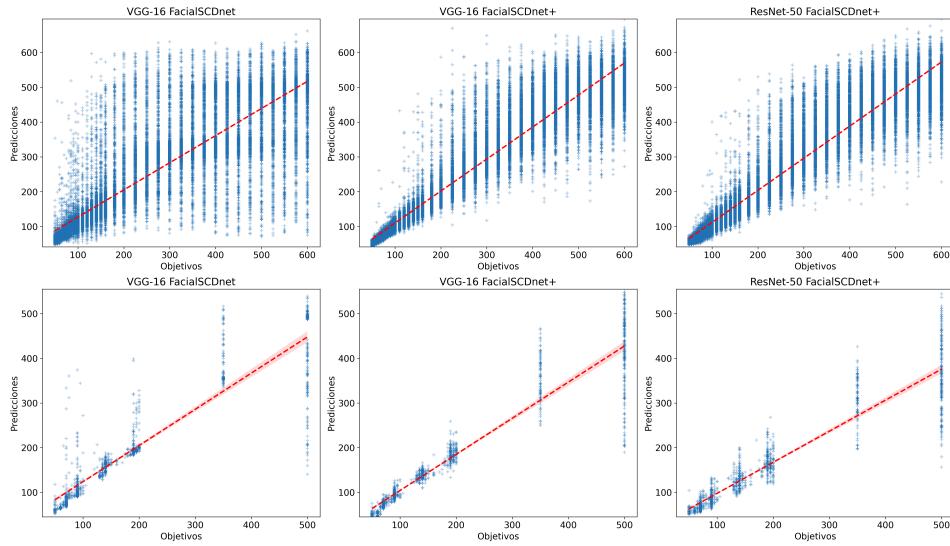


Figura 5.6: Gráfica comparativa de predicciones vs objetivos para los tres modelos utilizados. La primera fila presenta los resultados en el conjunto de test sintético, mientras que la segunda fila muestra los resultados en el conjunto de test real.

5.4, ambas arquitecturas de FacialSCDnet+ tienen un comportamiento similar, realizando estimaciones mucho más precisas a distancias cortas (en torno a 1.5 metros) y mostrando más variabilidad a distancias mayores. Es precisamente en esos casos donde el impacto de la distorsión de perspectiva es menos relevante, con lo que los resultados se explican al emplear la metrica de distorsión para guiar el entrenamiento del modelo. Para el conjunto de datos real se puede extraer una conclusión similar, destacando cómo el modelo basado en VGG-16 de FacialSCDnet+ parece minimizar la dispersión en la estimación de SCD, reduciendo los outliers y mostrando un comportamiento más consistente.

Además, los coeficientes de determinación R^2 (Tablas 5.3 y 5.4) proporcionan una medida cuantitativa adicional del rendimiento de los modelos, indicando qué tan bien se ajustan las líneas de regresión de la Figura 5.6 a los datos. En concreto, el modelo VGG-16 de FacialSCDnet+ alcanza un valor de 0.895 para el conjunto de datos sintéticos, mientras que para el conjunto de datos reales se reduce a 0.829. Esto indica que VGG-16 tiene una buena capacidad (mayor del 80 %) para explicar la variabilidad en los datos de ambos conjuntos.

Capítulo 6

Conclusiones y trabajos futuros

La distancia entre la cámara y el sujeto en fotografías faciales es un tema relevante en diversas disciplinas, debido a los cambios en la apariencia de los sujetos fruto del impacto de la distorsión de perspectiva. Para facilitar el análisis de estas imágenes, en el TFG desarrollado, nos enfocamos en crear un método basado en técnicas *deep learning* para la estimación precisa de la distancia cámara-sujeto (SCD) en dichas fotografías faciales. Para ello, se cumplieron satisfactoriamente una serie de objetivos fundamentales: se realizó un análisis del estado del arte y del método de referencia FacialSCDnet [21]; se examinaron las bases de datos existentes de modelos faciales y humanos en 3D; se diseñó un protocolo de estandarización para generar imágenes sintéticas fotorrealistas a partir de estos modelos; se llevó a cabo un estudio comparativo de la nueva aproximación propuesta; y se exploraron nuevas arquitecturas en busca de un mejor rendimiento respecto al método original.

La propuesta desarrollada en este trabajo, FacialSCDnet+, aborda las limitaciones identificadas sobre el método principal del estado del arte [21]. Esta propuesta reduce aproximadamente a la mitad el error en la estimación, además de reducir los sesgos producidos en FacialSCDnet por el preprocesamiento original de las imágenes. Para ello, se empleó un conjunto de imágenes sintéticas generadas a partir de modelos 3D, que sirvieron para entrenar dos modelos de aprendizaje: VGG-16 y ResNet-50. Ambos modelos se adaptaron a la tarea de regresión de la distancia, mostrando buenos resultados en cuanto a predicciones tanto en conjuntos de test sintéticos como reales. En concreto, el modelo VGG-16 superó el rendimiento del modelo ResNet-50 e incluso del modelo FacialSCDnet, consiguiendo un error medio de distorsión menor al 1 % en ambos conjuntos sintético y real. Este hecho se observa tanto a distancias cortas como largas, sin embargo, el impacto de la métrica de

distorsión es evidente a distancias mayores, ya que sigue presente cierta dispersión en las estimaciones. Este comportamiento es esperable ya que según avanza la distancia, el impacto de la distorsión de perspectiva se reduce exponencialmente. Así, podemos concluir que todas las propuestas realizadas en este trabajo han contribuido a superar el rendimiento del estado del arte para la estimación de la SCD con técnicas de *deep learning*. Además, ha permitido identificar algunos de los factores que deben tenerse en cuenta a la hora de trabajar con datos y modelos de aprendizaje automático, con el fin de evitar posibles sesgos en el proceso de entrenamiento.

En general, para el desarrollo de este TFG, se utilizaron conocimientos tanto de la asignatura de Aprendizaje Automático como de la de Visión por Computador. Además, se adquirieron nuevos conocimientos sobre modelado 3D con Blender, y el uso de herramientas desconocidas para el autor como Keras.

De cara a los trabajos futuros, se presentan varios desafíos y oportunidades de mejora. Uno de los principales desafíos es la predicción robusta de distancias más lejanas, donde las actuales arquitecturas pueden no ser suficientemente precisas, para ello se explorarían métricas alternativas que permitan tener en cuenta tanto la distancia métrica como el error de distorsión. Además, sería interesante explorar el impacto del uso de distintas focales en las predicciones, así como un diseño alternativo que integre el valor de la distancia focal para evitar la necesidad de entrenar varios modelos diferentes. Este es un factor importante a tener en cuenta, ya que las variaciones en la distancia focal pueden influir significativamente en la exactitud de la estimación de la SCD. Por último, se propone ampliar el uso de modelos 3D de cuerpo completo para aumentar la variabilidad de poses de forma que el conjunto de datos sea más diverso y realista aún. Esto podría proporcionar una mejor representación de las variaciones anatómicas y mejorar la robustez del modelo ante diferentes condiciones y tipos de sujetos.

Apéndice

Búsquedas en Scopus

Imágenes faciales

Encontradas 2937 publicaciones a fecha 27 de Marzo de 2024 mediante el comando de consulta: TITLE-ABS-KEY (facial AND (images OR photographs))

Estimación de la distancia en fotografías faciales

Encontradas 441 publicaciones a fecha 28 de Marzo de 2024 mediante el comando de consulta: TITLE-ABS-KEY ((distance OR depth) AND estimation AND (photographs OR images) AND facial) AND (LIMIT-TO (SUBJAREA , “COMP”) OR LIMIT-TO (SUBJAREA , “ENGI”))

Estimación de la distancia en fotografías faciales mediante IA

Encontradas 224 publicaciones a fecha 28 de Marzo de 2024 mediante el comando de consulta: 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 , “COMP”) OR LIMIT-TO (SUBJAREA , “ENGI”))

Estimación de la distancia en fotografías faciales mediante deep learning

Encontradas 129 publicaciones a fecha 28 de Marzo de 2024 mediante el comando de consulta: TITLE-ABS-KEY ((deep AND learning) AND ((distance OR depth) AND estimation AND (photographs OR images) AND facial)) AND (LIMIT-TO (SUBJAREA , “COMP”) OR LIMIT-TO (SUBJAREA , “ENGI”))

Bibliografía

- [1] Carl N. Stephan et al. «An overview of the latest developments in facial imaging». En: *Forensic Sciences Research* 4 (2018), págs. 10-28.
- [2] Shan Li y Weihong Deng. «Deep Facial Expression Recognition: A Survey». En: *IEEE Transactions on Affective Computing* 13 (2022), págs. 1195-1215.
- [3] Lixiang Li et al. «A Review of Face Recognition Technology». En: *IEEE Access* 8 (2020), págs. 139110-139120.
- [4] Ashu Kumar, Amandeep Kaur y Munish Kumar. «Face Detection Techniques: A Review». En: *Artificial Intelligence Review* 52 (2019).
- [5] Jiankang Deng et al. «ArcFace: Additive Angular Margin Loss for Deep Face Recognition». En: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, págs. 4685-4694.
- [6] Xiaobo Wang et al. «Co-Mining: Deep Face Recognition With Noisy Labels». En: *IEEE International Conference on Computer Vision*. 2019, págs. 9357-9366.
- [7] Yaoyao Zhong et al. «Unequal-Training for Deep Face Recognition With Long-Tailed Noisy Data». En: *IEEEConference on Computer Vision and Pattern Recognition*. 2019, págs. 7804-7813.
- [8] Keyan Ding et al. «Comparison of Full-Reference Image Quality Models for Optimization of Image Processing Systems». En: *International Journal of Computer Vision* 129 (2021), págs. 1258-1281.
- [9] Guangtao Zhai y Xiongkuo Min. «Perceptual image quality assessment: a survey». En: *Science China Information Sciences* 63 (2020).
- [10] Ashu Kumar, Amandeep Kaur y Munish Kumar. «Face Detection Techniques: A Review». En: *Artificial Intelligence Review* 52 (2019).
- [11] Xudong Sun, Pengcheng Wu y Steven C.H. Hoi. «Face detection using deep learning: An improved faster RCNN approach». En: *Neurocomputing* 299 (2018), págs. 42-50.

- [12] 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.
- [13] Rajeev Ranjan, Vishal M. Patel y Rama Chellappa. «HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2019), págs. 121-135.
- [14] 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.
- [15] Carl N. Stephan. «Perspective distortion in craniofacial superimposition: Logarithmic decay curves mapped mathematically and by practical experiment». En: *Forensic Science International* 257 (2015), 520.e1-520.e8.
- [16] Zhixiang Wang et al. «DisCO: Portrait Distortion Correction with Perspective-Aware 3D GANs». En: (2023).
- [17] Andrea Valsecchi. *Comprensión de los parámetros de la cámara*. Accedido el 17 de Febrero de 2024. 2019. URL: <https://skeleton-id.com/investigaciones/comprension-de-los-parametros-de-la-camara>.
- [18] Arturo Flores et al. «Camera Distance from Face Images». En: *Advances in Visual Computing*. Vol. 8034. 2013, págs. 513-522.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] Ridha Ilyas Bendjillali et al. «Illumination-robust face recognition based on deep convolutional neural networks architectures». En: *Indonesian Journal of Electrical Engineering and Computer Science* 18 (2020), págs. 1015-1027.
- [23] Wuming Zhang et al. «Improving Shadow Suppression for Illumination Robust Face Recognition». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2019), págs. 611-624.

- [24] Xi Yin y Xiaoming Liu. «Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition». En: *IEEE Transactions on Image Processing* 27 (2018), págs. 964-975.
- [25] Feifei Zhang et al. «Joint Pose and Expression Modeling for Facial Expression Recognition». En: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), págs. 3359-3368.
- [26] Kai Wang et al. «Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition». En: *IEEE Transactions on Image Processing* 29 (2020), págs. 4057-4069.
- [27] Yajie Zhao et al. «Learning Perspective Undistortion of Portraits». En: (2019), págs. 7848-7858.
- [28] Owen Mayer y Matthew C. Stamm. «Forensic Similarity for Digital Images». En: *IEEE Transactions on Information Forensics and Security* 15 (2020), págs. 1331-1346.
- [29] 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.
- [30] Ninareh Mehrabi et al. «A Survey on Bias and Fairness in Machine Learning». En: *ACM Computing Surveys* 54 (2021).
- [31] 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.
- [32] Ankush Gupta, Andrea Vedaldi y Andrew Zisserman. «Synthetic Data for Text Localisation in Natural Images». En: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2016), págs. 2315-2324.
- [33] Qi Wang et al. «Learning from synthetic data for crowd counting in the wild». En: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2019), págs. 8190-8199.
- [34] R.S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 2005.
- [35] Yaser S. Abu-Mostafa, M. Magdon-Ismail y H.T. Lin. *Learning from Data: A Short Course*. AMLBook, 2012.
- [36] Stuart Russell y Peter Norvig. *Artificial Intelligence: A Modern Approach 4th edition*. Pearson, 2021.
- [37] John D. Kelleher. *Deep learning*. MIT Press, 2019.
- [38] Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep Learning*. MIT Press, 2016.

- [39] Daniel Graupe. *Principles of artificial neural networks 3rd edition.* World Scientific, 2007.
- [40] Simon Haykin. *Neural Networks and Learning Machines 3rd edition.* Pearson, 2009.
- [41] C.M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, 1995.
- [42] Y. LeCun et al. «Backpropagation Applied to Handwritten Zip Code Recognition». En: *Neural Computation* 1 (1989), págs. 541-551.
- [43] Y. Lecun et al. «Gradient-based learning applied to document recognition». En: *Proceedings of the IEEE* 86 (1998), págs. 2278-2324.
- [44] 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.
- [45] 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>.
- [46] Anamika Dhillon y Gyanendra Verma. «Convolutional neural network: a review of models, methodologies and applications to object detection». En: *Progress in Artificial Intelligence* 9 (2019), págs. 85-112.
- [47] Rikiya Yamashita et al. «Convolutional neural networks: an overview and application in radiology». En: *Insights into Imaging* 9 (2018), págs. 611-629.
- [48] Pratik Ahamed et al. «Handwritten Arabic numerals recognition using convolutional neural network». En: *Journal of Ambient Intelligence and Humanized Computing* 11 (2020).
- [49] Asmaul Hosna et al. «Transfer learning: a friendly introduction». En: *Journal of Big Data* 9 (2022).
- [50] Minyoung Huh, Pulkit Agrawal y Alexei Efros. «What makes ImageNet good for transfer learning?» En: (2016).
- [51] 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>.
- [52] J. Igual. *Óptica y Fotografía: Libros 1 y 2.* Independently Published, 2017.
- [53] Michael John Langford y Francesc Rosés. *Fotografía básica, Guía para fotógrafos.* Omega, 2007.
- [54] T. Ang et al. *Fundamentos de la fotografía.* Blume, 2017.

- [55] 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>.
- [56] Dan Zafra. *Does Focal Length Distort Subjects?* Accedido el 12 de Junio de 2024. 2021. URL: <https://capturetheatlas.com/what-is-focal-length/>.
- [57] 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/>.
- [58] 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/>.
- [59] Eilidh Noyes y Rob Jenkins. «Camera-to-subject distance affects face configuration and perceived identity». En: *Cognition* 165 (2017), págs. 97-104.
- [60] Nasim Mansurov. *Does Focal Length Distort Subjects?* Accedido el 23 de Marzo de 2024. 2020. URL: <https://photographylife.com/does-focal-length-distort-subjects>.
- [61] 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).
- [62] 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.
- [63] Xavier Burgos-Artizzu, Matteo Ruggero Ronchi y Pietro Perona. *Caltech Multi-Distance Portraits*. 2022.
- [64] 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.
- [65] 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.
- [66] 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.
- [67] Sean Healy y Carl Stephan. «Focus distance estimation from photographed faces: a test of PerspectiveX using 1709 frontal and profile photographs from DSLR and smartphone cameras». En: *International Journal of Legal Medicine* 137 (2023).

- [68] Olga Russakovsky et al. «ImageNet Large Scale Visual Recognition Challenge». En: *International Journal of Computer Vision* 115 (2014), págs. 211-252.
- [69] Hang Dai et al. «Statistical Modeling of Craniofacial Shape and Texture». En: *International Journal of Computer Vision* 128.2 (2019), págs. 547-571.
- [70] Eduard Ramon et al. «H3D-Net: Few-Shot High-Fidelity 3D Head Reconstruction». En: *IEEE International Conference on Computer Vision*. 2021, págs. 5620-5629.
- [71] 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.
- [72] 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.
- [73] Thiemo Alldieck et al. «Video Based Reconstruction of 3D People Models». En: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018, págs. 8387-8397.
- [74] Shuhei Watanabe. «Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance». En: *ArXiv* (2023).
- [75] Karen Simonyan y Andrew Zisserman. «Very deep convolutional networks for large-scale image recognition». En: 2015.
- [76] Will Nash, Tom Drummond y Nick Birbilis. «A review of deep learning in the study of materials degradation». En: *npj Materials Degradation* 2 (2018).
- [77] Kaiming He et al. «Deep Residual Learning for Image Recognition». En: *IEEE Conference on Computer Vision and Pattern Recognition* (2015), págs. 770-778.
- [78] Ridha Ilyas Bendjillali et al. «Illumination-robust face recognition based on deep convolutional neural networks architectures». En: *Indonesian Journal of Electrical Engineering and Computer Science* 18 (2020), págs. 1015-1027.