

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Generación de datos artificiales para el procesamiento de imágenes mediante técnicas de aprendizaje profundo

Autor: Jorge Huete Fernández-Bravo

Tutor: Ana María González Marcos

Mayo 2025

Agradecimientos

Quisiera agradecer a mis profesores por su dedicación y por haber contribuido a mi formación a lo largo de la carrera. Agradezco especialmente a mi tutora, Ana, por su orientación y apoyo continuo durante el desarrollo de este trabajo. También quiero dar las gracias a mis amigos y compañeros por su ayuda, motivación y por haber hecho más llevadero este camino académico.

Resumen

Este Trabajo de Fin de Grado estudia el impacto de la generación de datos artificiales mediante técnicas de *data augmentation* en tareas de clasificación de imágenes utilizando aprendizaje profundo. Para hacer este estudio se han utilizado dos conjuntos de datos: uno binario (COVID vs No-COVID) y otro multiclase (CUB-200-2011), representando escenarios reales donde la cantidad de datos disponibles para el entrenamiento es limitada. Esta limitación, común en muchos contextos reales, se ve agravada por la conocida maldición de la dimensionalidad, que dificulta la generalización de los modelos cuando el espacio de entrada es amplio y el número de muestras insuficiente.

Los modelos preentrenados han sido VGG16 y ResNet50, aplicando *transfer learning* para adaptarlos a ambos casos, comparando su rendimiento con y sin técnicas de *data augmentation*. Las métricas que se han utilizado para el análisis de los resultados han sido precisión, pérdida, matriz de confusión y curva ROC, según el tipo de problema.

Los resultados obtenidos muestran que el uso de *data augmentation* mejora significativamente la capacidad de generalización de los modelos, especialmente en el *dataset* multiclase, donde se ha observado que ayuda a mitigar el sobreajuste. En el caso binario, también se observa una mejora en la precisión y el área bajo la curva ROC. El trabajo concluye que el uso de datos artificiales es una herramienta eficaz para mejorar el rendimiento de modelos convolucionales en contextos donde los datos disponibles son limitados, y se proponen futuras líneas de investigación como el *fine-tuning* avanzado o el uso de técnicas de explicabilidad para comprender mejor el comportamiento del modelo.

Palabras clave

aprendizaje profundo, *data augmentation*, clasificación de imágenes, *transfer learning*, redes convolucionales, datos artificiales, VGG16, ResNet50, maldición de la dimensionalidad, *fine-tuning*, CUB-200-2011, COVID-19, curva ROC, generalización, sobreajuste.

Abstract

This Final Degree Project studies the impact of employing artificial data generated with data augmentation techniques in image classification problems using deep learning. For this study, two datasets have been used: a binary dataset (COVID vs No-COVID) and a multiclass dataset (CUB-200-2011), representing real life problems where the amount of available data is limited. This limitation, commonly encountered in many day-to-day problems gets aggravated because of a known problem in deep learning: the curse of dimensionality. This increases the problems of generalization of models when the entry space is wide, and the number of samples is insufficient.

The pre-trained models used in this project are VGG16 and ResNet50, both adjusted with transfer learning for each case, comparing the performance with and without applying data augmentation techniques. The metrics used for the analysis are accuracy, loss, confusion matrix and ROC curve, depending on the type of problem.

The results obtained demonstrate that the usage of data augmentation improves the model generalization significantly, especially in the multiclass dataset, where it has been observed that it helps mitigate the overfitting problem. In the binary case it also shows a noticeable improvement in the accuracy and area under the ROC curve. The study concludes that artificial data generation is an efficient tool that can be used to improve the performance of convolutional models in contexts where the available data is limited. It proposes future research lines such as advanced fine-tuning and explainability methods to better understand model behavior.

Keywords

deep learning, data augmentation, image classification, transfer learning, convolutional neural networks, artificial data, VGG16, ResNet50, curse of dimensionality, fine-tuning, CUB-200-2011, COVID-19, ROC curve, generalization, overfitting.

Índice

Agradecimientos	III
Resumen	V
Abstract	VII
Índice	IX
Índice de figuras	XII
Índice de tablas	XIII
1 Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
2 Estado del arte	4
2.1. Redes Neuronales Convolucionales (CNN)	4
2.2. Data augmentation	4
2.3. Transfer learning	5
2.4. Maldición de la dimensionalidad y escasez de datos	5
3 Diseño del sistema	7
3.1. Planteamiento del problema	7
3.1.1. Maldición de la dimensionalidad	7
3.1.2. Limitaciones en la obtención de datos reales	8
3.2. Estrategia de solución adoptada	9
3.2.1. Aplicación de data augmentation	9
3.2.2. Uso de modelos preentrenados con <i>transfer learning</i>	12
4 Implementación	15
4.1. Selección del conjunto de datos	16
4.1.1. COVID-19 Radiography Database	16
4.1.2. Caltech-UCSD Birds 200 (CUB-200-2011)	16
4.2. Configuración del entorno de trabajo	17
4.3. Preparación de los datos	17
4.3.1. Dataset binario: COVID-19 Radiography Database	18
4.3.2. Dataset multiclase: CUB-200-2011	18
4.3.3. Consideraciones comunes	18
4.4. Configuración de los generadores de datos	19
4.5. Construcción del modelo	19

4.6. Entrenamiento de los modelos.....	20
4.7. Evaluación y análisis de los resultados.....	21
4.8. Almacenamiento y documentación	22
5 Desarrollo, Experimentación y Análisis de Resultados	24
5.1. Experimento sobre el <i>dataset</i> binario: COVID vs. No-COVID	24
5.1.1. Resultados sin <i>data augmentation</i>	25
5.1.2. Resultados con <i>data augmentation</i>	25
5.1.3. Análisis comparativo	25
5.2. Experimentos sobre el <i>dataset</i> multiclase: CUB-200-2011.....	26
5.2.1. Resultados sin <i>data augmentation</i>	26
5.2.2. Resultados con <i>data augmentation</i>	27
5.2.3. Análisis comparativo	27
5.3. Análisis de métricas y visualizaciones	28
5.3.1. Precisión y pérdida	28
5.3.2. Matriz de confusión	30
5.3.3. Curva ROC y AUC (solo <i>dataset</i> binario).....	33
6 Análisis Comparativo, Discusión y Limitaciones del Estudio.....	36
6.1. Comparación entre modelos	36
6.2. Efecto de aplicar <i>data augmentation</i>	36
6.3. Discusión de los resultados	37
7 Conclusiones y futuros trabajos	39
7.1. Conclusiones.....	39
7.2. Líneas de mejora y futuros trabajos	40
<i>Bibliografía</i>	41
<i>Apéndices</i>.....	43
<i>Apéndice A: Recursos</i>.....	45

Índice de figuras

Figura 5.1: Resultados Dataset Binario - Modelo ResNet50 sin data augmentation

Figura 5.2: Resultados Dataset Binario - Modelo ResNet50 & data augmentation

Figura 5.3: Resultados Dataset Binario - Modelo VGG16 sin data augmentation

Figura 5.4: Resultados Dataset binario - Modelo VGG16 & data augmentation

Figura 5.5: Resultados Dataset - Modelo VGG16 sin data augmentation

Figura 5.6: Resultados Dataset - Modelo VGG16 & data augmentation

Figura 5.7: Matriz de confusión Dataset binario - ResNet50 sin data augmentation

Figura 5.8: Matriz de confusión Dataset binario - ResNet50 & data augmentation

Figura 5.9: Matriz de confusión Dataset binario - VGG16 sin data augmentation

Figura 5.10: Matriz de confusión Dataset binario - VGG16 & data augmentation

Figura 5.11: Matriz de confusión Dataset multiclase - VGG16 sin data augmentation

Figura 5.12: Matriz de confusión Dataset multiclase - VGG16 & data augmentation

Figura 5.13: Curva ROC - ResNet50

Figura 5.14: Curva ROC – VGG16

Índice de tablas

Tabla 5.1: Resultados Dataset Binario

Tabla 5.2: Resultados Dataset Multiclase

Introducción

Con el avance de las técnicas de aprendizaje profundo en los últimos años, se han obtenido grandes progresos en el campo del procesamiento de imágenes. Sin embargo, para que estas técnicas sean efectivas es necesario disponer de una gran cantidad de datos bien etiquetados. En muchos escenarios reales reunir estos grandes volúmenes de datos es, en sí mismo, un reto lento y costoso. Esta limitación compromete el funcionamiento de estos modelos de clasificación y es el problema que se va a intentar resolver en este trabajo.

Existen numerosas estrategias para resolver este problema. En este trabajo se ha optado por explorar el concepto de *data augmentation*. Estas técnicas permiten, a partir de un conjunto de datos inicial bien etiquetados, crear variantes artificiales aplicando transformaciones geométricas (rotaciones, escalados, ...) o alteraciones de color, entre otras técnicas más complejas. Aunque no se generan datos nuevos en el sentido estricto de la palabra, aumenta la variabilidad de los datos de entrada, fortaleciendo el modelo y favoreciendo la generalización.

Esta escasez de datos no solo compromete la capacidad de entrenamiento, sino que acentúa lo que en el ámbito del aprendizaje automático se conoce como la maldición de la dimensionalidad: a medida que aumentan las dimensiones de entrada (como ocurre con los píxeles de las imágenes), el espacio de características crece exponencialmente y se requieren muchas más muestras para cubrirlo adecuadamente.

1.1. Motivación

Los últimos avances en tareas de clasificación de imágenes han demostrado el alto potencial de usar redes neuronales profundas en este tipo de problemas. No obstante, su aplicación de forma efectiva en contextos donde los datos disponibles son escasos todavía es un reto abierto y por explorar. Esta situación es habitual y se da en situaciones reales a diario como en medicina, en industria o en biodiversidad; donde obtener datos y más aún, clasificarlos correctamente, es costoso y complejo. Lo que se suele observar en estos ejemplos es que los modelos tienden a sobreajustarse y por lo tanto perdiendo la capacidad de generalización.

El estudio de técnicas que permiten mitigar esta limitación sin recurrir a la opción más costosa (adquirir nuevos datos) es un campo de estudio muy interesante y de gran relevancia práctica. Entre ellas, *data augmentation* ofrece una solución flexible y computacionalmente eficiente. Este trabajo surge del interés por investigar y demostrar, desde una perspectiva experimental y escalada a las limitaciones de computación y datos de uso libre, el impacto real de estas técnicas

en el rendimiento de modelos convolucionales preentrenados cuando se utilizan conjuntos de datos limitados.

El concepto conocido como la maldición de la dimensionalidad (Bellman, 1961), implica que, para representar correctamente un espacio de entrada con muchas dimensiones, se requiere una cantidad de datos proporcionalmente mayor. En tareas de clasificación de imágenes, donde cada píxel puede considerarse una dimensión, este problema se agrava. Por tanto, el uso de técnicas como *data augmentation* se vuelve especialmente relevante para simular cobertura del espacio sin necesidad de recolectar datos reales adicionales.

1.2. Objetivos

Este Trabajo de Fin de Grado tiene como objetivo principal analizar el impacto de la generación de datos artificiales mediante técnicas de *data augmentation* en modelos de clasificación de imágenes basados en aprendizaje profundo.

Para ello se definen los siguientes objetivos específicos:

- Aplicar y comparar diferentes técnicas de *data augmentation* en tareas de clasificación de imágenes
- Evaluar el comportamiento de dos arquitecturas convolucionales preentrenadas (VGG16 y ResNet50) usando *transfer learning* sobre conjuntos de datos con características diferentes: uno binario (COVID vs No-COVID) y otro multiclase (CUB-200-2011)
- Medir el rendimiento con y sin *data augmentation* mediante métricas como precisión, pérdida, matriz de confusión y curva ROC dependiendo del conjunto.
- Identificar las mejoras obtenidas, si las hubiera, y las limitaciones del enfoque, proponiendo futuras líneas de investigación

Estado del arte

En esta sección se presenta un breve repaso de los principales conceptos, técnicas y enfoques relacionados con el trabajo realizado. Se introducen las bases teóricas del aprendizaje profundo aplicado a imágenes, las estrategias empleadas para afrontar la escasez de datos y las contribuciones previas más relevantes.

2.1. Redes Neuronales Convolucionales (CNN)

El aprendizaje profundo ha supuesto un cambio de paradigma en el procesamiento de imágenes, logrando resultados superiores a los métodos tradicionales en tareas como clasificación, segmentación o detección. Dentro de este ámbito, las redes neuronales convolucionales (en adelante, CNN) han demostrado ser especialmente eficaces debido a su capacidad para extraer representaciones jerárquicas y espaciales de los datos visuales (Simonyan & Zisserman, 2014; He et al., 2016).

Su arquitectura está diseñada para aprovechar la estructura bidimensional de las imágenes mediante capas convolucionales, de *pooling* y de activación, lo que las hace especialmente adecuadas para reconocer patrones locales como bordes, texturas o formas. Las CNN modernas, como VGG16 o ResNet50, han sido preentrenadas en conjuntos masivos como ImageNet y constituyen la base de numerosos sistemas actuales de clasificación de imágenes.

2.2. Data augmentation

Una de las principales limitaciones de las CNN es su alta dependencia de grandes cantidades de datos bien etiquetados. Para mitigar este problema, en los últimos años ha surgido la técnica de *data augmentation*, que consiste en generar variantes artificiales de las imágenes originales mediante transformaciones geométricas (rotación, traslación, volteo), de color (brillo, contraste) o de composición (recortes, mezcla de imágenes).

Estas técnicas, descritas ampliamente por Shorten y Khoshgoftaar (2019) y evaluadas empíricamente por Perez y Wang (2017), permiten ampliar la diversidad del conjunto de entrenamiento sin necesidad de nuevos datos reales, reduciendo el riesgo de *overfitting* y mejorando la capacidad de generalización de los modelos.

2.3. Transfer learning

Otra estrategia clave en entornos con pocos datos es el *transfer learning*, que consiste en reutilizar un modelo previamente entrenado en una tarea diferente; ajustándolo a la nueva tarea mediante el entrenamiento de sus capas finales (Pan & Yang, 2010). Este enfoque permite aprovechar representaciones ya aprendidas sobre conjuntos de datos extensos (como ImageNet) y adaptarlas eficientemente a nuevos dominios con mucho menos esfuerzo computacional.

Modelos como VGG16 o ResNet50 han sido ampliamente utilizados bajo este enfoque, ya que encapsulan representaciones visuales generales útiles para multitud de tareas. En este trabajo, se ha optado por congelar inicialmente las capas convolucionales y entrenar solo las densas, lo que reduce el riesgo de sobreajuste y disminuye el tiempo de entrenamiento

2.4. Maldición de la dimensionalidad y escasez de datos

En el contexto del aprendizaje profundo, la necesidad de grandes volúmenes de datos se justifica también desde una perspectiva teórica. El concepto de maldición de la dimensionalidad, formulado por Bellman (1961), establece que cuando el número de dimensiones del espacio de entrada crece, el volumen necesario de muestras para representar adecuadamente ese espacio aumenta exponencialmente.

En imágenes, una entrada típica puede tener más de 150.000 dimensiones (por ejemplo, una imagen RGB de 224×224 píxeles), lo que dificulta que los modelos generalicen correctamente si no se dispone de un número suficiente de muestras (Verleysen & François, 2005). Esta situación es habitual en el mundo real donde recopilar y etiquetar imágenes no siempre es una opción viable por diversas razones.

Como consecuencia de este fenómeno, si no se dispone de suficientes datos, los modelos pueden aprender patrones espurios o ruido en lugar de representaciones útiles (Hughes, 1968; Keogh & Mueen, 2017), generando predicciones no fiables en datos no vistos. Esta problemática refuerza el valor de técnicas como el *data augmentation* o el *transfer learning*, que permiten combatir la escasez de datos desde un punto de vista práctico y computacionalmente eficiente.

Diseño del sistema

Esta sección presenta en detalle el problema al que se enfrenta este trabajo. En la definición del problema, en primer lugar, se describe el marco teórico de la maldición de la dimensionalidad, que explica por qué los modelos de aprendizaje profundo son especialmente sensibles a la escasez de datos. A continuación, se analizan las dificultades prácticas asociadas a la adquisición y etiquetado de conjuntos de datos reales, así como su impacto en el proceso de entrenamiento.

Posteriormente, se revisan las principales técnicas de *data augmentation* como solución para ampliar artificialmente los conjuntos de datos, y se expone el papel de aplicar *transfer learning* como estrategia para reutilizar conocimiento previamente adquirido y reducir la necesidad de entrenamiento desde cero.

3.1. Planteamiento del problema

3.1.1. Maldición de la dimensionalidad

3.1.1.1. Definición

La maldición de la dimensionalidad es un fenómeno que describe cómo la complejidad de los datos crece exponencialmente al aumentar el número de dimensiones o características. Introducido por Bellman (1961), este concepto señala que, en espacios de alta dimensión, las distancias entre puntos se vuelven menos significativas, los datos tienden a dispersarse y muchas técnicas estadísticas y de *machine learning* pierden eficacia.

En términos simples, cuantos más atributos tenga un conjunto de datos, más difícil es extraer patrones significativos con un número limitado de muestras. Esto se debe a que el espacio de búsqueda crece de manera exponencial, lo que exige una cantidad también exponencial de datos para representarlo adecuadamente (Verleysen & François, 2005).

3.1.1.2. Impacto en el procesamiento de imágenes

En el procesamiento de imágenes, cada píxel se considera una dimensión, por lo que una imagen en color de 224x224 píxeles tiene más de 150.000 características ($224 \times 224 \times 3$). Esta alta dimensionalidad genera varios problemas:

- Los modelos utilizados requieren una gran cantidad de datos bien categorizados para poder generalizar correctamente el problema.
- Los datos escasos o mal distribuidos provocan que el modelo aprenda ruido (fluctuaciones aleatorias o impredecibles en los datos que dificultan la habilidad del modelo para identificar patrones).
- El riesgo de sobreajuste aumenta si el conjunto de entrenamiento es pequeño en relación con la dimensión del espacio de entrada que, como ya se ha expuesto, es alto en conjuntos de imágenes

3.1.1.3. Ejemplos prácticos

Un ejemplo claro ocurre al entrenar CNN con pocos datos, como sucede en problemas médicos (e.g., clasificación de radiografías) o en conjuntos de datos de nicho como CUB-200-2011, donde algunas clases tienen pocas muestras. Sin una estrategia para mitigar el problema de la dimensionalidad, los modelos tienden a sobreajustar (aprender de memoria las muestras) y fracasan al generalizar sobre datos nuevos.

Por ello, técnicas como *data augmentation* y *transfer learning* se vuelven imprescindibles para reducir el impacto negativo de este fenómeno. Ambas permiten explorar más eficazmente el espacio de entrada sin necesidad de tener que recopilar nuevas imágenes para el conjunto de datos.

3.1.2. Limitaciones en la obtención de datos reales

Uno de los principales desafíos en el entrenamiento de estos modelos de aprendizaje profundo es la necesidad de disponer de grandes volúmenes de datos bien etiquetados. Algo que resulta difícil de cumplir en la práctica.

En muchos dominios especializados los datos son escasos, heterogéneos y costosos de obtener. Además, el proceso de etiquetado manual suele requerir intervención experta, lo que eleva aún más los recursos necesarios y por lo tanto aumenta los costes de producir estos conjuntos de datos.

Esta escasez de datos afecta directamente a la capacidad del modelo para generalizar como ya se ha explorado anteriormente. Cuando el conjunto de entrenamiento no es representativo del dominio completo, los modelos tienden a aprender características específicas del conjunto visto, presentando un alto riesgo de sobreajuste (*overfitting*).

Este problema se intensifica en tareas multiclase o con alta variabilidad visual, donde las clases pueden estar desequilibradas o pobremente representadas. Por eso en este trabajo se quiere estudiar el impacto en un problema binario y otro multiclase.

Otro aspecto para considerar es la baja disponibilidad de *datasets* públicos en ciertas áreas de aplicación, lo que obliga a reutilizar conjuntos limitados o poco variados. Incluso cuando los

datos están disponibles, pueden estar sujetos a restricciones éticas, legales o técnicas que dificultan su uso. Todo ello hace imprescindible buscar estrategias que permitan mitigar estas limitaciones sin comprometer la calidad del entrenamiento.

Sumando todos estos problemas, se justifica el uso de técnicas como *data augmentation*, que permiten generar instancias artificiales a partir de imágenes reales, aumentando así la diversidad del conjunto sin necesidad de recolectar nuevos datos y los costes que ello implica.

Asimismo, el uso de *transfer learning* se presenta como una solución eficaz para aprovechar modelos previamente entrenados sobre grandes bases de datos, adaptándolos a tareas nuevas con un esfuerzo computacional y de datos mucho menor.

3.2. Estrategia de solución adoptada

3.2.1. Aplicación de data augmentation

3.2.1.1. Explicación teórica

La técnica de *data augmentation* es una estrategia ampliamente utilizada en el mundo del aprendizaje profundo para aumentar, de forma artificial, el conjunto de datos de entrenamiento sin necesidad de recolectar nuevas muestras para el conjunto inicial del problema que se está trabajando.

En lugar de agregar imágenes completamente nuevas, se generan variaciones de las existentes mediante transformaciones que no alteran la clase a la que pertenecen. Algunas de estas alteraciones son: girarlas, reflejarlas o cambiar su brillo. Esto permite al modelo exponerse a una mayor diversidad de datos, lo que a su vez mejora su capacidad de generalización.

Desde el punto de vista estadístico, la técnica de *data augmentation* actúa como una forma de regularización. Al incrementar la variedad de instancias vistas por el modelo durante el entrenamiento, se reduce la posibilidad de que memorice patrones específicos del conjunto de entrenamiento (*overfitting*).

Además, ayuda al modelo a ser invariante a ciertas transformaciones que pueden darse en escenarios reales: un objeto puede aparecer rotado, parcialmente oculto o con condiciones de iluminación distintas, pero seguir siendo el mismo.

3.2.1.2. Técnicas comunes de *data augmentation*

Las técnicas de *data augmentation* pueden clasificarse en varias categorías según el tipo de transformación que aplican:

Transformaciones geométricas

Las transformaciones geométricas son una de las técnicas más clásicas y efectivas, a la vez que simples, dentro del *data augmentation*. Se basan en modificar la geometría de la imagen. Es decir, su forma, posición o escala sin alterar el significado semántico del objeto representado.

Estas transformaciones permiten simular situaciones reales en las que un objeto puede aparecer rotado, desplazado o encuadrado de manera distinta, favoreciendo así la generalización del modelo ante variaciones espaciales. A continuación, se describen las transformaciones geométricas más habituales:

- **Rotación aleatoria:** consiste en girar la imagen completa en torno al eje central. Esto es útil para que el modelo no dependa de la orientación exacta del objeto. En contextos como la clasificación de aves, donde estas pueden aparecer inclinadas de forma natural es especialmente relevante.
- **Traslación o desplazamiento (*shift*):** esta transformación se refiere a mover la imagen horizontal o verticalmente, habitualmente en un porcentaje relativo del tamaño de la imagen. Esto simula pequeños errores de encuadre.
- **Zoom aleatorio:** permite acercar o alejar el objeto recortando una parte de la imagen o ampliándola ligeramente. Ayuda a aprender características locales más robustas.
- **Volteo horizontal:** invierte la imagen a lo largo del eje vertical, como si se viera reflejada en un espejo. Esta transformación es muy útil en conjuntos de datos donde la orientación izquierda-derecha no altera la clase como es este caso.
- **Recortes aleatorios:** simulan distintos encuadres del mismo objeto, eliminando bordes o añadiendo márgenes.

La ventaja de estas transformaciones es que se mantiene la clase original de la imagen lo que permite ampliar la distribución espacial del conjunto de datos de forma artificial y sin apenas costes. Estas transformaciones son especialmente eficaces en problemas de visión por computador donde el objetivo a identificar puede encontrarse en diferentes posiciones y ángulos de visión que contribuyen a mejorar la invarianza de las CNN.

Transformaciones en el espacio de color

Las transformaciones en el espacio de color constituyen otro conjunto fundamental de técnicas de *data augmentation*. A diferencia de las transformaciones geométricas, que modifican

la estructura espacial de la imagen, estas técnicas alteran las propiedades cromáticas y de iluminación.

Su principal objetivo es hacer que el modelo sea robusto frente a variaciones de luz, contraste, saturación y composición de color que pueden darse de forma natural en diferentes condiciones de captura de las imágenes. A continuación, se describen las transformaciones más comunes en este ámbito:

- **Ajuste de brillo:** consiste en incrementar o reducir la intensidad de la luz en la imagen de forma aleatoria. Esto simula condiciones de iluminación diversas.
- **Variación de contraste:** modifica la diferencia entre las zonas claras y oscuras de la imagen. Aumentar el contraste realza los bordes y formas, mientras que reducirlo suaviza las transiciones. Esta variación entrena al modelo para extraer características relevantes.
- **Ajuste de saturación y tono:** consiste en cambiar la intensidad o riqueza del color y la tonalidad general de la imagen. Obliga al modelo a centrarse en aprender la forma y textura en lugar de depender únicamente del color.
- **Perturbación de los canales RGB:** consiste en modificar los valores individuales de los canales rojo, verde y azul. Esta técnica genera combinaciones cromáticas atípicas que aumentan la variabilidad del conjunto, aunque debe aplicarse con cuidado.
- **Conversión entre espacios de color:** en algunos casos se puede trabajar directamente en espacios de color alternativos (como HSV o LAB).

Estas transformaciones son especialmente relevantes en dominios donde las condiciones de adquisición de imágenes son muy variables, como ocurre, sobre todo, en fotografía natural o imágenes capturadas con distintos dispositivos (que son los conjuntos que se van a utilizar en este estudio).

Técnicas avanzadas de aumento de datos

Además de las transformaciones geométricas y cromáticas tradicionales, en los últimos años han surgido una serie de técnicas de *data augmentation* más sofisticadas; diseñadas para alterar parcialmente la imagen o incluso combinar múltiples ejemplos entre sí.

A continuación, se mencionan las más relevantes, pero ninguna ha sido utilizada en este trabajo por el cuidado que hay que tener en su aplicación para no afectar al rendimiento del modelo: *random erasing*, *cutout*, *mixup*, *CutMix*, *GridMask*.

3.2.1.3. Ejemplos de cómo estas técnicas mitigan la maldición de la dimensionalidad

Volviendo a uno de los problemas expuesto anteriormente, la maldición de la dimensionalidad implica que, para cubrir adecuadamente un espacio de entrada de alta dimensión, muchas veces se necesita una cantidad descomunal de muestras.

Aplicar *data augmentation* no elimina este problema, pero lo reduce considerablemente al poblar de forma más eficiente ese espacio con ejemplos sintéticos que simulan condiciones variadas que se podrían dar en el conjunto de datos. Por ejemplo:

- Una imagen de un ave en el *dataset* CUB puede verse girada, aumentada o recortada, generando múltiples representaciones en el espacio de entrada, todas válidas para la misma clase.
- En un entorno médico, como el *dataset* COVID, aplicar pequeños desplazamientos y ajustes de brillo puede simular condiciones reales de toma de radiografías sin alterar el diagnóstico.

De esta manera, el modelo aprende a detectar características invariables frente a transformaciones, desarrollando una representación interna más robusta y menos sensible a la dispersión de los datos.

3.2.1.4. Conclusión

En resumen, *data augmentation* es una técnica esencial para cualquier sistema de visión artificial basado en *deep learning*, especialmente cuando se trabaja con conjuntos de datos limitados o en contextos donde obtener datos nuevos es costoso.

Al actuar como una forma de generación artificial de datos, mejora la capacidad de generalización del modelo, reduce el riesgo de sobreajuste y mitiga los efectos de la maldición de la dimensionalidad. Su implementación es sencilla con tecnologías como TensorFlow y Keras, y sus beneficios, tanto teóricos como empíricos, están ampliamente demostrados en la literatura.

3.2.2. Uso de modelos preentrenados con *transfer learning*

El *transfer learning*, o aprendizaje por transferencia, es una técnica fundamental en el campo del aprendizaje profundo que permite reutilizar modelos previamente entrenados en una tarea para resolver una nueva, normalmente diferente pero relacionada. Su principal ventaja radica en que evita entrenar un modelo desde cero, lo cual es especialmente útil cuando se dispone de un conjunto de datos limitado y un nivel de computación bajo.

En el contexto de visión por computador, el *transfer learning* se aplica comúnmente utilizando arquitecturas convolucionales profundas entrenadas sobre grandes conjuntos de datos como ImageNet, que contiene más de 14 millones de imágenes y 1000 clases distintas. Estas redes, al haber aprendido a detectar bordes, texturas, formas y patrones comunes, desarrollan representaciones jerárquicas del contenido visual que pueden reutilizarse para tareas de clasificación de imágenes.

3.2.2.1. Funcionamiento y fases

El proceso típico de *transfer learning* en imágenes se compone de dos fases:

- **Extracción de características:** en esta fase, se utiliza el modelo preentrenado como un extractor de características. Las capas convolucionales iniciales del modelo original se mantienen fijas (no se entrenan) y solo se añaden nuevas capas densas para adaptarse a la nueva tarea. Esto permite aprovechar el conocimiento general aprendido previamente, reduciendo el tiempo de entrenamiento y la necesidad de grandes volúmenes de datos.
- **Fine-tuning:** en una segunda fase opcional, se pueden descongelar algunas capas superiores del modelo base para reajustarlas al nuevo dominio.

3.2.2.2. Ventajas y limitaciones

Las principales ventajas del *transfer learning* son:

- **Reducción del coste computacional:** evita entrenamientos prolongados desde cero.
- **Mejor rendimiento en *datasets* pequeños:** permite alcanzar buenos resultados incluso con datos limitados.
- **Aprovechamiento de conocimiento general:** facilita que el modelo aprenda representaciones útiles más rápidamente.

Sin embargo, también presenta algunas limitaciones:

- Si el dominio original y el nuevo son muy distintos (por ejemplo, imágenes naturales vs imágenes médicas), el modelo puede no transferir eficazmente.
- Requiere precaución en la fase de *fine-tuning* para evitar degradar lo aprendido en el preentrenamiento.

3.2.2.3. Aplicación en este trabajo

En este Trabajo de Fin de Grado se han utilizado dos modelos preentrenados ampliamente reconocidos: VGG16 y ResNet50, ambos con pesos entrenados en ImageNet. En la primera fase, sus capas convolucionales fueron congeladas y se añadieron capas densas personalizadas para adaptarlas a los *datasets* binario (COVID vs No-COVID) y multiclase (CUB-200-2011). En fases posteriores, se exploró la posibilidad de aplicar *fine-tuning* descongelando las últimas capas para mejorar la adaptación a la tarea específica.

Este enfoque ha permitido obtener modelos robustos y eficientes, incluso en condiciones donde los datos disponibles eran escasos o estaban desbalanceados, demostrando la eficacia del *transfer learning* como estrategia complementaria a *data augmentation*.

Implementación

Esta sección describe detalladamente el proceso seguido para implementar, entrenar y evaluar los modelos de aprendizaje profundo en los diferentes escenarios definidos para este trabajo. Se presentan las decisiones técnicas tomadas, los recursos utilizados, y la estructura del experimento diseñada para luego poder discutir los beneficios empíricamente.

El diseño experimental se ha estructurado en torno a dos conjuntos de datos contrastados: uno binario (COVID-19 Radiography Database) y otro multiclase (CUB-200-2011), lo que permite evaluar la robustez de las técnicas propuestas en contextos con diferentes grados de complejidad. Ambos *datasets* presentan limitaciones reales en cuanto al volumen de datos por clase, lo que los convierte en candidatos ideales para aplicar técnicas de *data augmentation* y *transfer learning*.

A nivel de implementación, se ha optado por utilizar Google Colab como entorno de ejecución, debido a su accesibilidad, facilidad de configuración y disponibilidad de aceleración por GPU. La implementación se ha desarrollado en Python, utilizando la biblioteca TensorFlow y su API de alto nivel Keras. El trabajo se ha diseñado de forma modular para permitir la reutilización de funciones y facilitar la comparación de resultados.

El flujo de trabajo adoptado que se puede ver reflejado en el *notebook* puede resumirse en los siguientes pasos principales:

1. **Selección del conjunto de datos:** Se eligen dos *datasets* públicos con características complementarias: uno binario (COVID vs No-COVID) y uno multiclase (CUB-200-2011).
2. **Configuración del entorno de trabajo:** Se utiliza Google Colab como plataforma de ejecución, con aceleración por GPU, y se emplean bibliotecas como TensorFlow, Keras, scikit-learn y Matplotlib.
3. **Preparación de los datos:** Descarga, preprocesamiento, reorganización y partición automática de los conjuntos de entrenamiento y validación, manteniendo una resolución fija (224x224) y normalizando los valores de los píxeles para la entrada.
4. **Configuración de los generadores de datos:** Creación de un `ImageDataGenerator` para cada experimento, con y sin técnicas de *data augmentation*, incluyendo transformaciones geométricas, cromáticas y avanzadas.

5. **Construcción del modelo:** Para cada configuración experimental se construye un modelo basado en VGG16 o ResNet50, utilizando pesos preentrenados y adaptando su arquitectura para ajustarse al número de clases del *dataset* correspondiente.
6. **Entrenamiento de los modelos:** Se entrenan las combinaciones de ambos modelos y ambos *datasets* en diferentes configuraciones: con y sin *data augmentation*.
7. **Evaluación y análisis de resultados:** Se recopilan métricas clave (precisión, pérdida, matriz de confusión, curva ROC) y se visualizan los resultados obtenidos para su posterior comparación y discusión.
8. **Almacenamiento y documentación:** Todos los modelos, históricos y gráficas se almacenan para su análisis posterior y para reproducibilidad del trabajo.

4.1. Selección del conjunto de datos

Para llevar a cabo este trabajo, se han seleccionado dos conjuntos de datos con características contrastadas, que permiten evaluar el rendimiento de las técnicas aplicadas en escenarios con distinta complejidad.

4.1.1. COVID-19 Radiography Database

Este *dataset*, publicado en Kaggle por Tawsifur Rahman et al. (2021), contiene imágenes de radiografías torácicas clasificadas en cuatro clases: *COVID-19*, *Normal*, *Viral Pneumonia* y *Lung Opacity*. Para los fines de este trabajo, se ha reducido el problema a un escenario binario (*COVID* vs *NORMAL*), seleccionando una muestra equilibrada de cada clase.

Este conjunto de datos representa un caso real en el que el volumen de datos es limitado, la obtención de imágenes es costosa, y la etiqueta asociada requiere validación clínica, lo que justifica plenamente el uso de *data augmentation* y *transfer learning*.

4.1.2. Caltech-UCSD Birds 200 (CUB-200-2011)

El segundo *dataset* utilizado es el conocido conjunto de datos CUB-200-2011, publicado por Wah et al. (2011), que contiene 11.788 imágenes clasificadas en 200 especies distintas de aves. Este conjunto se ha utilizado ampliamente en tareas de clasificación donde las clases son visualmente muy similares entre sí y presentan alta variabilidad dentro de la misma categoría.

Con una media de 30 imágenes por clase, este *dataset* representa un reto importante para modelos de aprendizaje profundo, sobre todo sin *data augmentation*. Su inclusión permite analizar el comportamiento de las técnicas propuestas en un entorno multiclase con muchas clases y pocas instancias por clase.

Ambos *datasets* comparten una característica clave: la disponibilidad limitada de datos y su alto valor semántico, lo que los convierte en casos representativos para estudiar el impacto de técnicas de *data augmentation* y *transfer learning*.

4.2. Configuración del entorno de trabajo

El desarrollo experimental de este trabajo se ha llevado a cabo en el entorno de Google Colab, una plataforma en la nube basada en Jupyter Notebooks que permite ejecutar código en Python con acceso gratuito a aceleración por GPU.

Esta elección ha facilitado la ejecución de entrenamientos más intensivos sin necesidad de hardware local especializado, adaptándose así a las restricciones típicas de un entorno académico.

Las principales herramientas y tecnologías utilizadas han sido:

- **Python 3.10:** como lenguaje de programación principal, por su versatilidad y su amplia disponibilidad de librerías para ciencia de datos.
- **TensorFlow 2.x y Keras:** como *framework* principal para construir y entrenar CNN. Keras, como interfaz de alto nivel, ha permitido una implementación más clara y modular.
- **Google Drive:** como sistema de almacenamiento auxiliar para guardar modelos entrenados, resultados intermedios y gráficas.
- **Matplotlib y Seaborn:** para la visualización de métricas, evolución del entrenamiento y análisis de resultados.
- **scikit-learn:** para el cálculo de métricas complementarias como matriz de confusión, precisión, *recall* y curva ROC.
- **Kaggle API:** para la descarga directa del *dataset* COVID desde el entorno de Colab.

Este entorno ha permitido desarrollar un flujo de trabajo reproducible y flexible, facilitando la gestión de múltiples experimentos con distintas configuraciones de modelo, *augmentation* y *dataset*.

4.3. Preparación de los datos

La preparación adecuada de los datos es un paso fundamental para garantizar que los modelos de aprendizaje profundo reciban entradas consistentes, equilibradas y correctamente etiquetadas. Dado que este trabajo se centra en evaluar el impacto de aplicar *data augmentation*, se ha diseñado un proceso de preprocesamiento coherente y reproducible para ambos conjuntos de datos, asegurando condiciones comparables.

4.3.1. Dataset binario: COVID-19 Radiography Database

El *dataset* fue descargado directamente desde la plataforma Kaggle utilizando el API de Kaggle, e importado en el entorno de Google Colab. A partir del conjunto completo, se seleccionaron únicamente las clases *COVID* y *Normal*, descartando las otras dos categorías para simplificar el problema a una clasificación binaria clara.

Se tomaron 1000 imágenes de cada clase, para un total de 2000 imágenes. A efectos de mantener un flujo similar al del *dataset* multiclase, no se reorganizaron las imágenes en carpetas separadas por *train* y *val*, sino que se aplicó una partición automática mediante la funcionalidad `validation_split` del generador `ImageDataGenerator`.

Todas las imágenes fueron redimensionadas a una resolución fija de 224x224 píxeles y reescaladas dividiendo los valores de cada píxel por 255 para normalizar la entrada.

4.3.2. Dataset multiclase: CUB-200-2011

El *dataset* CUB-200-2011 se descargó desde su repositorio oficial (Caltech Vision), en formato comprimido .tgz. Una vez descomprimido, las imágenes se organizaron en carpetas por clase. Al igual que con el *dataset* binario, no se creó una separación manual entre entrenamiento y validación, sino que se aprovechó la funcionalidad de partición interna de `ImageDataGenerator`.

Dado que este *dataset* contiene imágenes de diferentes tamaños, todas fueron redimensionadas a 224x224 píxeles, y se aplicó el mismo preprocesamiento de normalización.

Debido a la elevada cantidad de clases (200) y al número reducido de imágenes por clase (~60), el uso de *data augmentation* resulta especialmente crítico para evitar el sobreajuste y dotar al modelo de más variabilidad.

4.3.3. Consideraciones comunes

Para ambos conjuntos, la carga de imágenes se realizó utilizando `flow_from_directory`, permitiendo la generación dinámica de *batches* tanto para entrenamiento como validación. Esta estrategia permite trabajar con grandes volúmenes de imágenes sin necesidad de cargarlas todas en memoria, y facilita la integración de técnicas de *data augmentation* aplicadas en tiempo real.

Se utilizó un tamaño de *batch* de 32 imágenes, y un porcentaje fijo del 20% de los datos se reservó automáticamente para validación (`validation_split=0.2`), de modo que el modelo no tuviera acceso a las imágenes de validación durante el entrenamiento.

4.4. Configuración de los generadores de datos

Una vez preparado el conjunto de imágenes, se definieron los generadores de datos encargados de alimentar los modelos durante el entrenamiento. Para ello se utilizó la clase `ImageDataGenerator` de Keras, que permite generar lotes de imágenes a partir de un directorio, aplicar preprocesamiento y, si se desea, aplicar técnicas de *data augmentation* de forma dinámica.

Para cada experimento, se configuraron dos versiones de los generadores:

- Un generador sin *data augmentation*, utilizado como referencia base.
- Un generador con *data augmentation*, donde se aplicaron transformaciones aleatorias como rotaciones, desplazamientos horizontales y verticales, zoom, volteo horizontal, *shear* y ajustes de brillo.

Estas transformaciones se eligieron por su capacidad para simular variaciones realistas en las imágenes sin alterar su contenido semántico. El objetivo era evaluar si estas perturbaciones controladas mejoraban la capacidad de generalización de los modelos, especialmente en situaciones con datos escasos.

Todos los generadores se configuraron con las siguientes opciones comunes:

- `target_size=(224, 224)`: para adaptar las imágenes al tamaño de entrada esperado por los modelos VGG16 y ResNet50.
- `batch_size=32`: para garantizar un equilibrio entre uso de memoria y estabilidad del gradiente.
- `class_mode='categorical' o 'binary'`: según el número de clases del *dataset*.
- `validation_split=0.2`: para reservar el 20% de las imágenes para validación del modelo tras el entrenamiento.

4.5. Construcción del modelo

Para cada conjunto de datos y cada configuración experimental (con y sin *augmentation*), se construyó un modelo basado en una de las siguientes arquitecturas preentrenadas:

- VGG16: una red convolucional profunda con capas secuenciales, conocida por su simplicidad y robustez.
- ResNet50: una arquitectura más moderna y profunda, que incorpora bloques residuales para facilitar el entrenamiento de redes más profundas sin degradación.

Ambos modelos se cargaron con pesos preentrenados en ImageNet (`weights='imagenet'`) y se excluyó su parte superior (`include_top=False`) para permitir la personalización del clasificador final. Para cada problema se utilizó una estructura distinta.

En el caso del *dataset* binario se utilizó una arquitectura sencilla:

- `GlobalAveragePooling2D`: para aplanar la salida convolucional.
- `Dense`: con 1024 neuronas y activación ReLU para aprender representaciones específicas de la tarea.
- Una capa de salida con dos clases y activación *sigmoid*.

Para el caso del conjunto no binario, se añadieron las siguientes capas:

- `GlobalAveragePooling2D`: como primera capa, para aplanar la salida convolucional.
- `BatchNormalization`: para estabilizar el entrenamiento y acelerar la convergencia del modelo.
- `Dense`: dos capas, de 256 y 128 neuronas con activación ReLU.
- `Dropout`: dos capas de 0.5 y 0.3 después de cada capa `Dense` para reducir el riesgo de sobreajuste
- Por último, una capa de salida con activación *softmax* y tantas neuronas como clases.

Además, se ajustó la tasa de aprendizaje para cada entrenamiento: $1e-3$ para el entrenamiento inicial y $1e-5$ para la fase de *fine-tuning*.

4.6. Entrenamiento de los modelos

Para el caso binario, todos los modelos se entrenaron durante 10 épocas, utilizando el optimizador Adam con una tasa de aprendizaje fija de $1e-5$. Se utilizó `binary_crossentropy` como función de pérdida y tanto para el *dataset* multiclase como para el binario se registraron las métricas de precisión y pérdida.

En el caso del conjunto multiclase se utilizó un enfoque más avanzado con dos fases:

- Fase 1 – Entrenamiento del clasificador: Se congelaron todas las capas de los modelos y se entrenó únicamente la parte superior personalizada que se ha discutido anteriormente.

Se utilizaron 80 épocas y una tasa fija de aprendizaje de $1e-3$ con las mismas métricas que el conjunto binario y con `categorical_crossentropy` como función de pérdida.

Además, debido a la complejidad añadida y para evitar el sobreajuste y los posibles problemas de recursos se añadieron tres *callbacks*: *EarlyStopping* para detener el entrenamiento cuando deje de disminuir la pérdida del conjunto de validación, *ReduceLROnPlateau* para reducir la tasa de aprendizaje cuando el modelo se estanque durante el entrenamiento y *ModelCheckpoint* para guardar mejor el modelo.

- *Fase 2 – Fine-tuning*: Para esta fase se descongelaron las capas superiores de los modelos base. En VGG16 desde la capa `block5_conv1` y en ResNet50 las últimas treinta capas. También se redujo la tasa de aprendizaje a $1e-5$ para evitar grandes oscilaciones al modificar pesos preentrenados.

Además, se implementó una lógica para reutilizar el mejor modelo entrenado en la fase anterior previo a comenzar el *fine-tuning* para que el punto de partida siempre fuese el óptimo detectado por el *EarlyStopping*.

El entrenamiento se realizó con `steps_per_epoch` y `validation_steps` calculados en función del tamaño de los generadores, asegurando una cobertura completa de los datos por época.

4.7. Evaluación y análisis de los resultados

Una vez completado el entrenamiento de cada configuración, se realizó una evaluación detallada sobre el conjunto de validación. Para ello se utilizaron las siguientes métricas:

- *Accuracy*: como métrica principal de clasificación.
- *Loss*: para observar la convergencia del modelo.
- Matriz de confusión: para analizar los aciertos y errores por clase.
- Curva ROC y AUC: en el caso binario, para evaluar la capacidad discriminativa del modelo.

Las predicciones se obtuvieron mediante `model.predict`, y se compararon con las etiquetas verdaderas empleando la librería `sklearn.metrics`. Todas las métricas se almacenaron, y sus visualizaciones fueron generadas y guardadas automáticamente para facilitar la comparación entre modelos y configuraciones.

4.8. Almacenamiento y documentación

Con el fin de garantizar la trazabilidad, reproducibilidad y posterior análisis de los experimentos realizados, se implementó un sistema de almacenamiento automático tanto de los modelos entrenados como de los resultados obtenidos. Para cada configuración se guardaron los siguientes elementos:

- Modelo entrenado en formato .h5 y .keras.
- Archivo .txt con el histórico de métricas por época.
- Gráficas de precisión, pérdida, matriz de confusión y curva ROC (si procede).
- Archivos .png con Identificación única del experimento en la nomenclatura (p.e. CUB_VGG16_Augmented.png).

Todos los resultados se almacenaron en carpetas organizadas por experimento y modelo dentro del entorno de Google Drive. Esta estructura ha permitido comparar de forma clara las distintas configuraciones y facilita una posible reutilización o extensión futura del trabajo.

Desarrollo, Experimentación y Análisis de Resultados

Una vez definidas las arquitecturas, parámetros y configuraciones descritas en el capítulo anterior, se procedió al desarrollo experimental del trabajo. Esta fase tenía como objetivo comprobar el impacto del uso de *data augmentation* en tareas de clasificación de imágenes mediante modelos convolucionales preentrenados, tanto en un contexto binario como multiclase.

Para ello, se diseñó una batería de experimentos sistemáticos en la que se entrenaron y evaluaron los modelos VGG16 y ResNet50 sobre ambos conjuntos de datos (COVID-19 y CUB-200-2011), comparando su rendimiento en dos escenarios diferenciados:

1. Sin *data augmentation*: donde el conjunto de datos se mantiene sin modificaciones, salvo la normalización y redimensionado estándar.
2. Con *data augmentation*: donde se aplican transformaciones geométricas, cromáticas y otras técnicas descritas previamente durante la fase de entrenamiento.

El diseño de los experimentos llevados a cabo consistía en pruebas para ambos datasets con ambos modelos aplicando *data augmentation* y sin aplicarlo.

Al finalizar cada entrenamiento, se calcularon y almacenaron las métricas clave (*accuracy*, *loss*, matriz de confusión y curva ROC, cuando aplicaba), junto con las gráficas de evolución durante el entrenamiento.

En las siguientes sub-secciones se presentan los resultados obtenidos para cada combinación, comenzando con el *dataset* binario y continuando con el multiclase.

5.1. Experimento sobre el *dataset* binario: COVID vs. No-COVID

El primer conjunto de experimentos se llevó a cabo utilizando el COVID-19 Radiography Database, transformado en un problema de clasificación binaria entre imágenes etiquetadas como *COVID* y *NORMAL*. Este escenario representa un caso real donde los datos disponibles son limitados y sensibles, y donde el coste de etiquetado y adquisición es elevado, tal como se discutió en la sección de diseño.

5.1.1. Resultados sin *data augmentation*

Tanto VGG16 como ResNet50 fueron entrenados con el conjunto de imágenes tal cual, sin aplicar ninguna transformación aleatoria.

- VGG16 alcanzó una precisión de validación alta (~91%), mostrando cierta estabilidad, pero también señales de posible sobreajuste a partir de la época 6.
- ResNet50 obtuvo un rendimiento moderado en validación (~75%), aunque con una ligera mayor oscilación entre épocas.

Ambos modelos presentaron una pérdida baja en entrenamiento, lo que indica una buena capacidad para aprender los patrones presentes, pero mostraron una menor generalización al conjunto de validación. La matriz de confusión evidencia que la clase *NORMAL* era ligeramente más fácil de clasificar, con más falsos negativos en la clase *COVID*.

5.1.2. Resultados con *data augmentation*

- VGG16 no mejoró su precisión del conjunto validación (~93%), con una curva de pérdida más suave y una menor diferencia entre entrenamiento y validación.
- ResNet50 alcanzó una precisión del 77%, con una curva de entrenamiento más estable y una pérdida de validación significativamente más baja.

La aplicación de *data augmentation* no solo ayudó a reducir el sobreajuste, sino que también incrementó el área bajo la curva ROC (AUC). Esto sugiere que el modelo fue capaz de aprender representaciones más robustas y menos dependientes de patrones específicos de las imágenes originales.

5.1.3. Análisis comparativo

Modelo	<i>Augmentation</i>	<i>Accuracy</i>	AUC ROC	Observaciones
VGG16	No	91%	0.94	Posible sobreajuste
VGG16	Sí	93%	0.97	Mayor estabilidad y generalización
ResNet50	No	75%	0.77	Mejor que VGG sin <i>augmentation</i>

ResNet50	Sí	77%	0.84	Mejor combinación global observada
----------	----	-----	------	------------------------------------

Tabla 5.1: Resultados Dataset Binario

Se muestran los resultados de la precisión y ROC AUC para el *dataset* binario obtenidos tras el entrenamiento. Así como un comentario de lo observado en cada caso.

Estos resultados confirman que el uso de *data augmentation* en un contexto binario con pocos datos tiene un impacto directo y positivo en el rendimiento del modelo, tanto en términos de precisión como de robustez ante datos no vistos. Además, evidencian que los modelos preentrenados como VGG16 y ResNet50 pueden beneficiarse especialmente de esta técnica al disponer de mayor capacidad de aprendizaje.

5.2. Experimentos sobre el *dataset* multiclase: CUB-200-2011

El segundo bloque de experimentos se realizó sobre el conjunto de datos CUB-200-2011, compuesto por 200 clases de aves diferentes y un total de 11.788 imágenes. Este *dataset* es un estándar en tareas de clasificación de grano fino (*fine-grained classification*), donde las clases presentan diferencias sutiles entre sí y una alta variabilidad interna.

Este escenario presenta un reto más complejo que el caso binario, ya que el número medio de imágenes por clase es bajo (~60), lo que lo convierte en un caso ideal para estudiar el impacto de aplicar *data augmentation* y evaluar la capacidad de generalización de los modelos preentrenados.

Para esta sección solo se han tomado en cuenta los resultados obtenidos con el modelo VGG16 ya que los resultados de ResNet50 demostraban que con este modelo el experimento todavía se encontraba en una fase incipiente del aprendizaje. Probablemente debido a una combinación de la arquitectura del modelo que no se adecuaba a las características del conjunto, la falta de datos para el entrenamiento y el número de elevado de clases.

5.2.1. Resultados sin *data augmentation*

Dado el gran número de clases y la escasez relativa de ejemplos por clase los resultados no son tan buenos como en el conjunto binario, pero aun así es interesante analizar los resultados obtenidos.

Durante el entrenamiento se obtuvo una precisión del 68% en el conjunto de entrenamiento mientras que en el de validación se obtuvo un 38% con una pérdida que indica que todavía el modelo tiene margen de mejora (1.1).

La matriz de confusión refleja que el modelo tiene cierto conocimiento de las clases, pero también muestra problemas a la hora de diferenciarlas. Por lo que se puede considerar un rendimiento aceptable para un problema tan complejo.

5.2.2. Resultados con *data augmentation*

Al aplicar *data augmentation* durante el entrenamiento, aunque las métricas absolutas en son similares a las obtenidos en el conjunto sin *data augmentation* se observa que la pérdida es mayor (>1.6) por lo que el margen de mejora es amplio todavía.

Otro detalle que se observa es que la precisión de entrenamiento (52%) es más cercana a la de validación (39%) por lo que podemos deducir que en el caso anterior el modelo estaba comenzando a sobreajustarse a los datos que recibía en el entrenamiento mientras que al aplicar *data augmentation* se consigue proveer de robustez al modelo y una mejor capacidad de generalización.

Observando los resultados de las gráficas y la tendencia de estas, es posible que incrementando el número de épocas y haciendo una búsqueda óptima de los hiperparámetros con un *grid search* se podrían obtener unos resultados bastante mejores.

Quedaría pendiente para trabajos futuros ver cómo se puede solventar el problema de limitación de la capacidad de cómputo para poder mejorar todavía más estos resultados, alcanzando unos valores absolutos cercanos a los que se han podido observar en el conjunto binario.

5.2.3. Análisis comparativo

Modelo	<i>Augmentation</i>	<i>Accuracy</i>	<i>Loss</i>	Observaciones
VGG16	No	38%	1.1	Posible sobreajuste
VGG16	Sí	39%	1.66	Mayor robustez y capacidad de generalización

Tabla 5.2: Resultados *Dataset* Multiclase

Se muestran los resultados de la precisión y pérdida para el *dataset* multiclase obtenidos tras el entrenamiento. Así como una breve observación de cada caso.

El *dataset* CUB-200-2011 es un entorno altamente desafiante, incluso con modelos preentrenados. Las mejoras obtenidas con *data augmentation* muestran que estas técnicas ayudan a los modelos a generar representaciones más invariantes a posibles transformaciones, y a evitar que se centren en características visuales irrelevantes o específicas de la imagen.

El modelo VGG16, a pesar de ser más simple que ResNet50, obtuvo mejores resultados en validación cuando se aplicó *data augmentation*. Esto sugiere que, en tareas con muy pocos

datos por clase, una arquitectura más simple puede generalizar mejor si se complementa con técnicas adecuadas de enriquecimiento de datos.

5.3. Análisis de métricas y visualizaciones

Con el fin de evaluar el comportamiento de los modelos en los distintos escenarios, se han generado y analizado varias métricas cuantitativas y visualizaciones para cada experimento. Estas herramientas permiten identificar no solo qué modelo o configuración obtiene una mayor precisión, sino también cómo evoluciona el aprendizaje y qué errores comete el sistema.

5.3.1. Precisión y pérdida

Para cada entrenamiento se registraron las métricas de *accuracy* y *loss* tanto para el conjunto de entrenamiento como de validación. Estas métricas se representan gráficamente en función del número de épocas, permitiendo observar el comportamiento del modelo durante el proceso de aprendizaje.

5.3.1.1. Dataset binario: COVID vs. No COVID

En el caso binario (COVID), los modelos entrenados con *data augmentation* mostraron curvas más suaves, con una brecha menor entre entrenamiento y validación, lo que indica una mejor generalización.

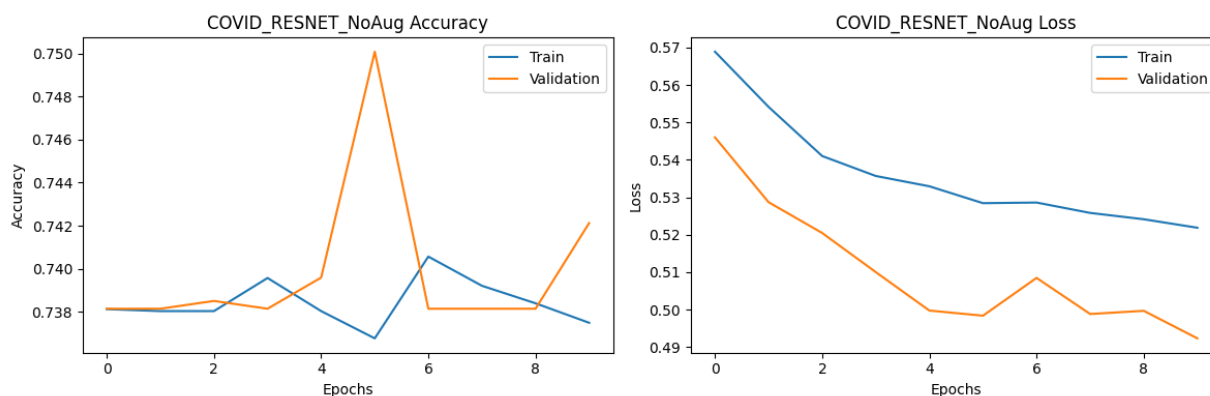


Figura 5.1: Resultados Dataset Binario - Modelo ResNet50 sin *data augmentation*

En la figura se muestran los resultados de precisión y pérdida del dataset binario con el modelo ResNet50 y sin aplicar *data augmentation* en el entrenamiento.

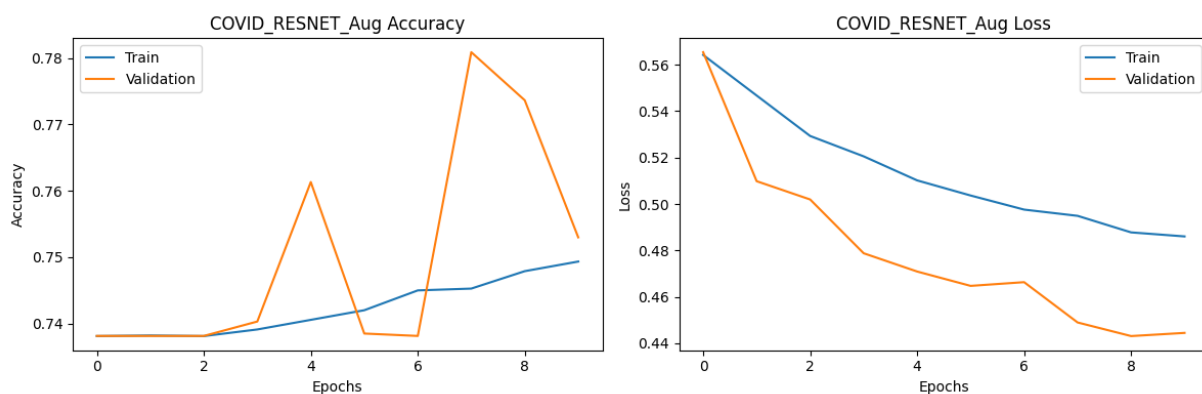


Figura 5.2: Resultados *Dataset* Binario - Modelo ResNet50 & *data augmentation*

En la figura se muestran los resultados de precisión y pérdida del *dataset* binario con el modelo ResNet50 y habiendo aplicado *data augmentation* en el entrenamiento.

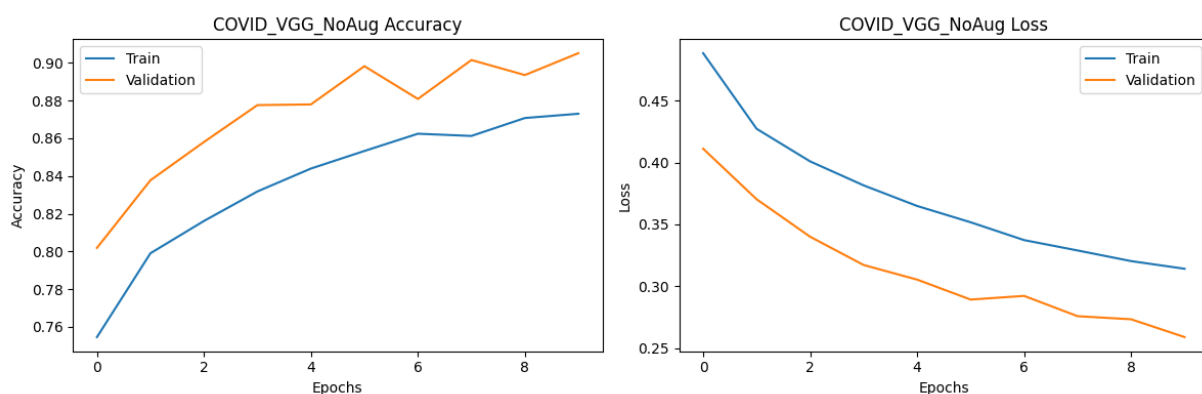


Figura 5.3: Resultados *Dataset* Binario - Modelo VGG16 sin *data augmentation*

En la figura se muestran los resultados de precisión y pérdida del *dataset* binario con el modelo VGG16 sin aplicar *data augmentation* en el entrenamiento.

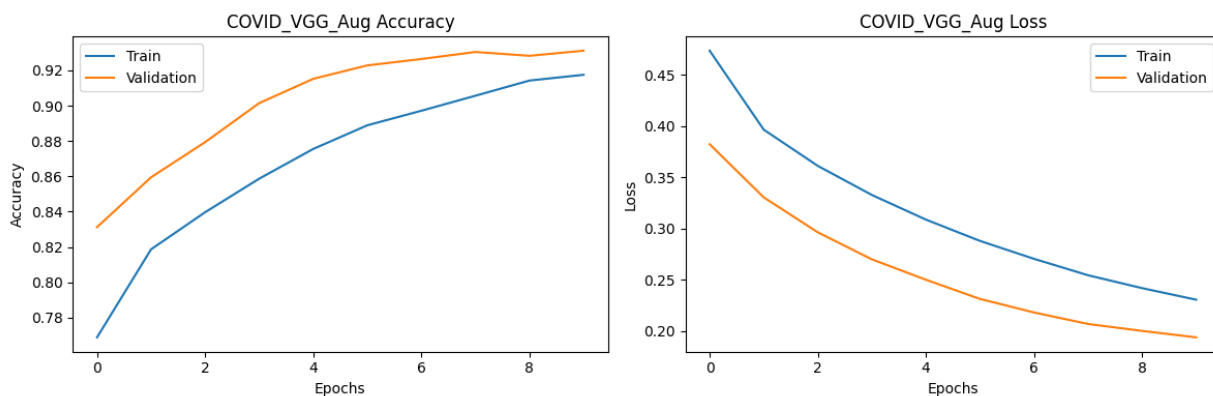


Figura 5.4: Resultados *Dataset* binario - Modelo VGG16 & *data augmentation*

En la figura se muestran los resultados de precisión y pérdida del *dataset* binario con el modelo VGG16 habiendo aplicado *data augmentation* en el entrenamiento.

5.3.1.2. *Dataset* multiclase: CUB-200-2011

En el caso multiclase (CUB-200-2011), aunque las precisiones absolutas fueron más bajas pero el uso de *data augmentation* redujo significativamente la pérdida y estabilizó el entrenamiento.

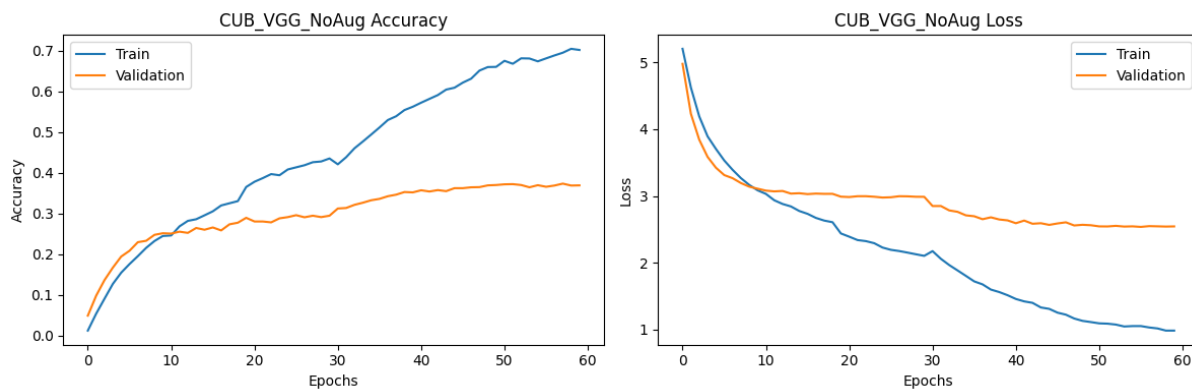


Figura 5.5: Resultados *Dataset* - Modelo VGG16 sin *data augmentation*

En la figura se muestran los resultados de precisión y pérdida del dataset multiclase con el modelo VGG16 y sin aplicar *data augmentation* en el entrenamiento.

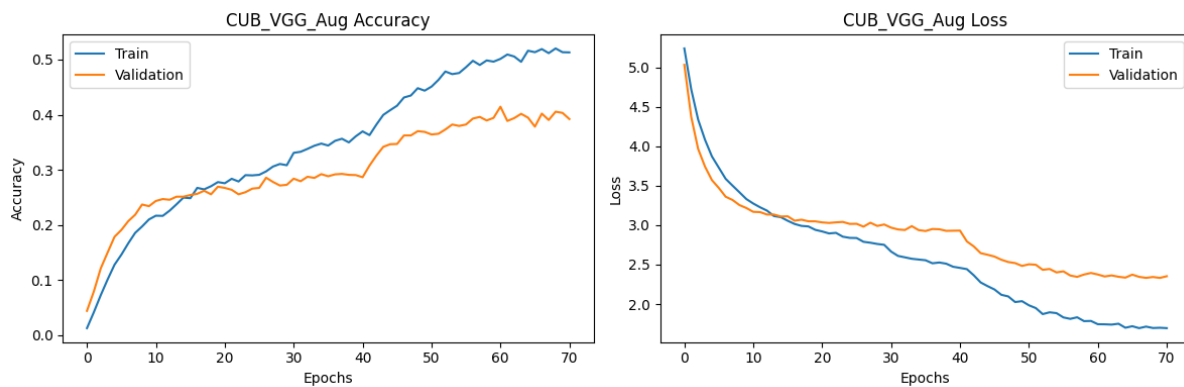


Figura 5.6: Resultados *Dataset* - Modelo VGG16 & *data augmentation*

En la figura se muestran los resultados de precisión y pérdida del dataset multiclase con el modelo VGG16 y habiendo aplicado *data augmentation* en el entrenamiento.

5.3.2. Matriz de confusión

Las matrices de confusión proporcionan una visión detallada de las predicciones del modelo. Se generaron para cada modelo tras el entrenamiento, utilizando el conjunto de validación:

5.3.2.1. *Dataset* binario (COVID vs. No COVID)

En el caso COVID, las matrices evidencian mejoras claras en la detección de la clase COVID cuando se aplica *data augmentation*, con una reducción de falsos negativos.

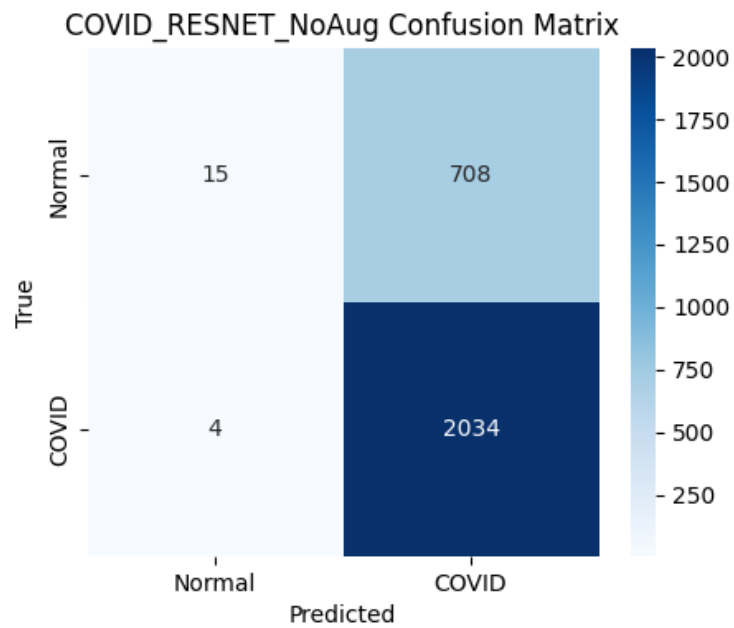


Figura 5.7: Matriz de confusión *Dataset* binario - RestNet50 sin *data augmentation*

En la figura se muestra la matriz de confusión resultante del entrenamiento del *dataset* binario con ResNet50 y sin aplicar *data augmentation*.

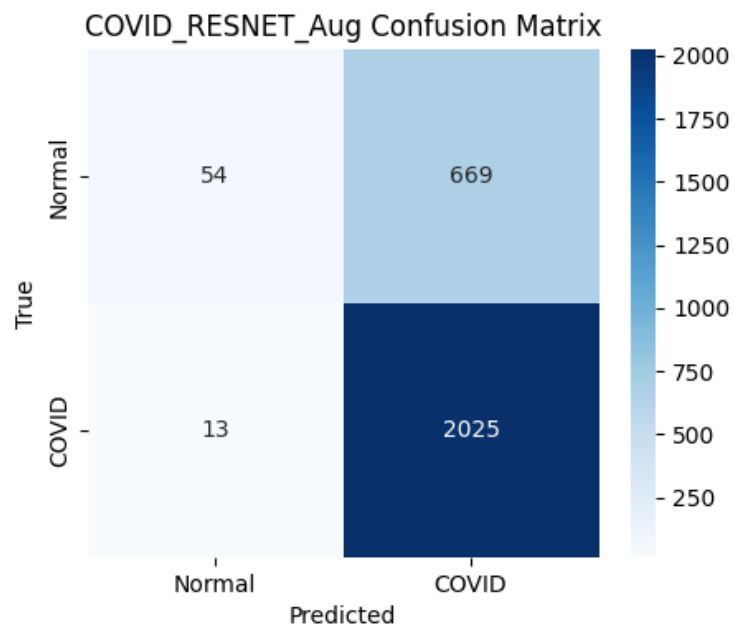


Figura 5.8: Matriz de confusión *Dataset* binario - ResNet50 & *data augmentation*

En la figura se muestra la matriz de confusión resultante del entrenamiento del *dataset* binario con ResNet50 y aplicando *data augmentation*.

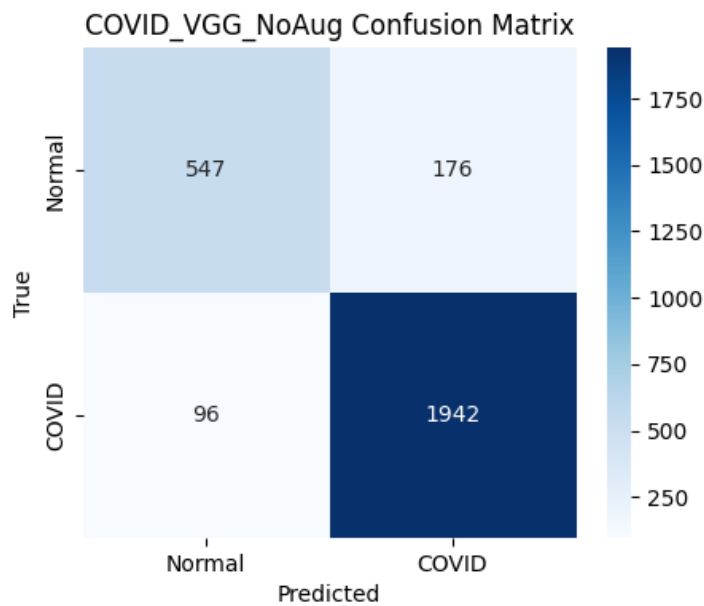


Figura 5.9: Matriz de confusión *Dataset* binario - VGG16 sin *data augmentation*

En la figura se muestra la matriz de confusión resultante del entrenamiento del *dataset* binario con VGG16 y sin aplicar *data augmentation*.

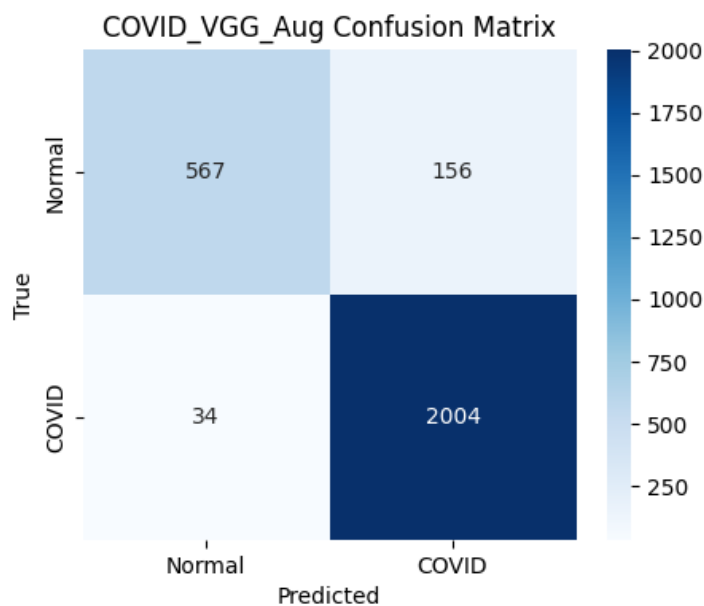


Figura 5.10: Matriz de confusión *Dataset* binario - VGG16 & *data augmentation*

En la figura se muestra la matriz de confusión resultante del entrenamiento del *dataset* binario con VGG16 y aplicando *data augmentation*.

5.3.2.2. *Dataset* multiclase (CUB-200-2011)

En el caso CUB, debido al elevado número de clases (200), se optó por mostrar solo un subconjunto de las clases más representativas o bien un *heatmap* sin etiquetas para visualizar los patrones de error más comunes.

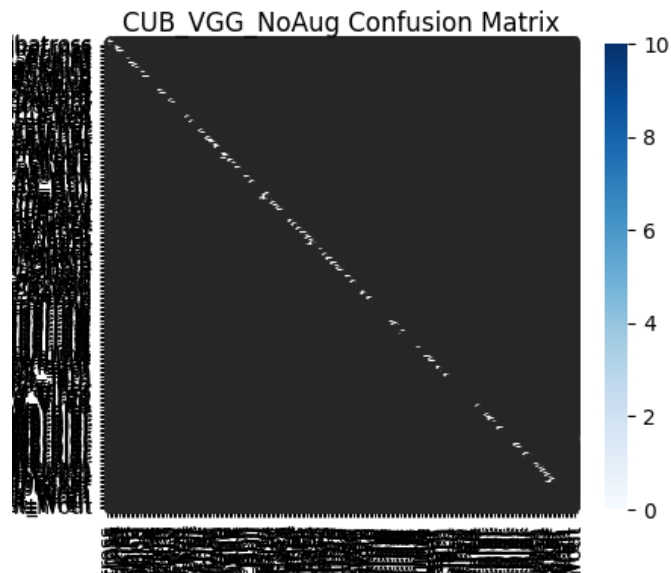


Figura 5.11: Matriz de confusión *Dataset* multiclase - VGG16 sin *data augmentation*

En la figura se muestra la matriz de confusión resultante del entrenamiento del *dataset* multiclase con VGG16 y sin aplicar *data augmentation*.

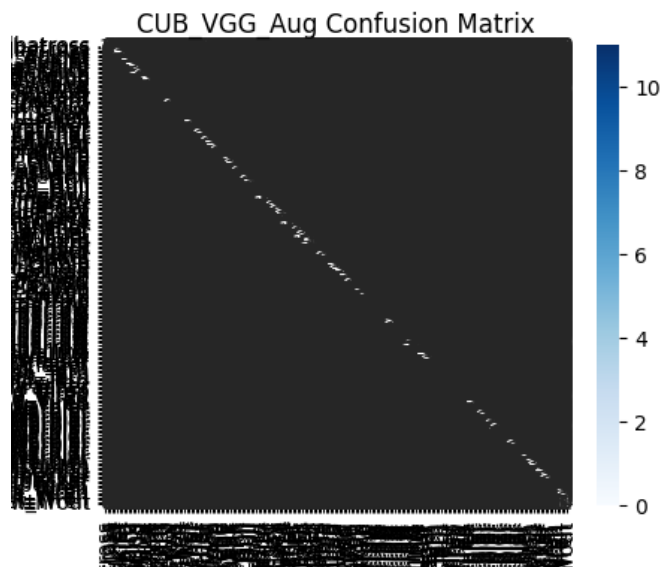


Figura 5.12: Matriz de confusión *Dataset* multiclase - VGG16 & *data augmentation*

En la figura se muestra la matriz de confusión resultante del entrenamiento del *dataset* multiclase con VGG16 y aplicando *data augmentation*.

5.3.3. Curva ROC y AUC (solo *dataset* binario)

Para los experimentos binarios, se calculó la curva ROC y el AUC (Área bajo la curva). Estas gráficas muestran la relación entre la tasa de verdaderos positivos y la de falsos positivos a diferentes umbrales de decisión.

En todos los casos, el uso de *data augmentation* elevó el AUC de forma consistente. La arquitectura VGG16 con *augmentation* obtuvo el mayor AUC (0.97), indicando una alta capacidad de discriminación entre clases. Las curvas ROC se combinaron y etiquetaron según el modelo, lo que permite realizar comparaciones visuales directas entre configuraciones.

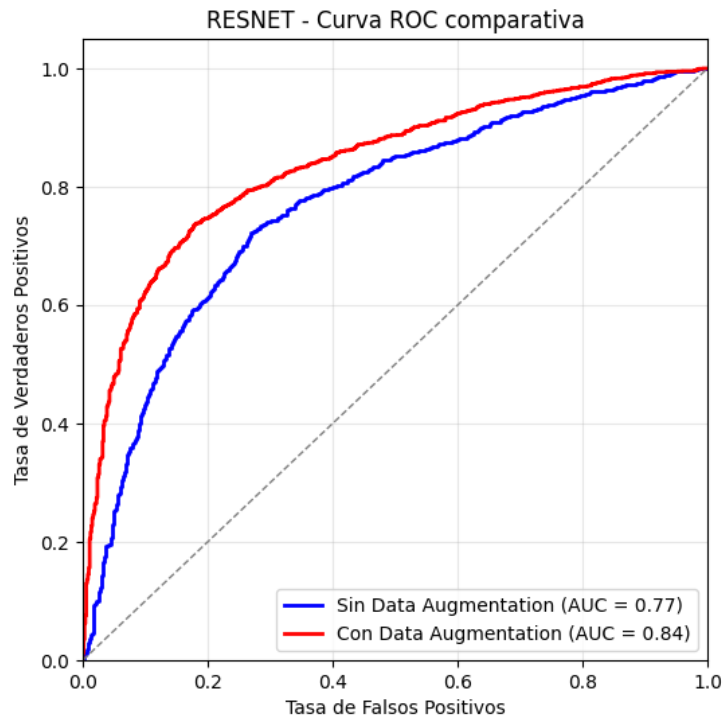


Figura 5.13: Curva ROC - ResNet50

En la figura se muestra la curva ROC resultante del modelo ResNet50 con y sin *data augmentation* y el área bajo la curva (AUC) para mejor visualización.

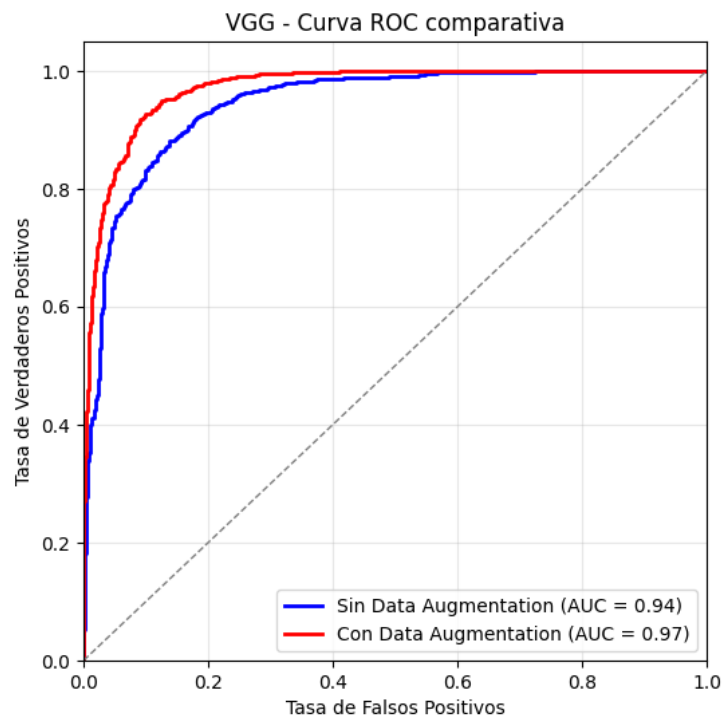


Figura 5.14: Curva ROC - VGG16

En la figura se muestra la curva ROC resultante del modelo VGG16 con y sin *data augmentation* y el área bajo la curva (AUC) para mejor visualización.

Análisis Comparativo, Discusión y Limitaciones del Estudio

Este capítulo recopila e interpreta los resultados obtenidos a lo largo del proceso experimental, estableciendo comparaciones entre modelos, configuraciones y conjuntos de datos. Si bien el capítulo anterior presentó métricas específicas y gráficas detalladas, aquí se destacan las conclusiones más relevantes desde una perspectiva comparativa y crítica, evaluando la utilidad real de aplicar *data augmentation* y *transfer learning* en contextos de datos limitados.

6.1. Comparación entre modelos

Al comparar las arquitecturas VGG16 y ResNet50, se observan diferencias notables según el tipo de problema:

- En el *dataset* binario (COVID), VGG16 ofreció un rendimiento superior en términos de precisión y AUC, especialmente cuando se aplicó *data augmentation*. Su arquitectura más simple parece haber favorecido a la generalización especialmente con clases bien definidas. ResNet50 por su parte parece mostrar mayor inestabilidad en las curvas de validación por lo que su complejidad no ha ayudado al problema.
- En el *dataset* multiclase (CUB-200-2011), los resultados fueron mejores también con VGG16. Aunque ambas arquitecturas mostraron problemas para generalizar debido al bajo número de clases y pocas épocas de entrenamiento, VGG16 mostró rasgos de estar aprendiendo el problema mientras que ResNet50 no.

En resumen, VGG16 parece que se adapta mejor a contextos con pocos datos por clase, probablemente debido a su menor complejidad. Mientras que ResNet50 es posible que pueda requerir de una configuración más precisa para funcionar mejor o más datos para alcanzar su máximo potencial.

6.2. Efecto de aplicar *data augmentation*

La aplicación de técnicas de *data augmentation* tuvo un impacto positivo claro en ambos contextos:

- En el caso binario, mejoró significativamente la precisión y la capacidad de discriminación del modelo (ROC AUC), reduciendo además los falsos negativos en la clase COVID.
- En el caso multiclase, aunque las métricas absolutas siguieron siendo moderadas, el *data augmentation* logró estabilizar las curvas de aprendizaje y reducir la pérdida de validación, indicando que el modelo aprendía patrones más generales y útiles.

Además, se observó que sin aplicar *data augmentation* los modelos presentaban signos de un posible sobreajuste, mientras que con *augmentation* se mantenía una diferencia más razonable entre entrenamiento y validación.

6.3. Discusión de los resultados

Los resultados demuestran que el uso conjunto de *transfer learning* y *data augmentation* son herramientas fundamentales para trabajar con conjuntos de datos reducidos. Aunque los modelos preentrenados ya ofrecen un rendimiento inicial decente, es el *data augmentation* el que permite aprovecharlos de forma más eficaz, especialmente cuando se quiere evitar el sobreajuste o se trabaja con tareas de alta dimensionalidad.

No obstante, se identificaron también algunas limitaciones:

- En el caso del *dataset* CUB, con 200 clases, la precisión sigue siendo baja en términos absolutos. Esto se debe tanto a la complejidad del problema como al bajo número de imágenes por clase.
- El tamaño de los *datasets*, especialmente en el caso multiclase, supuso una limitación ya que disponer de más imágenes o aplicar técnicas más agresivas de *augmentation* podría mejorar los resultados.
- Un mayor número de épocas de entrenamiento podría permitir al modelo converger mejor, aunque incrementa el riesgo de sobreajuste y sería más costoso computacionalmente.

Conclusiones y futuros trabajos

Este Trabajo de Fin de Grado ha tenido como objetivo analizar el impacto del uso de técnicas de *data augmentation* y *transfer learning* en tareas de clasificación de imágenes mediante aprendizaje profundo, especialmente en contextos donde los datos disponibles son limitados. Para ello, se han utilizado dos conjuntos de datos contrastados: uno binario (COVID vs No-COVID) y otro multiclase (CUB-200-2011) y dos modelos preentrenados ampliamente utilizados: VGG16 y ResNet50.

7.1. Conclusiones

A partir de los experimentos realizados y de las métricas obtenidas, se pueden extraer las siguientes conclusiones principales:

- El uso de *data augmentation* mejora de forma consistente la generalización de los modelos, tanto en clasificación binaria como multiclase, reduciendo el sobreajuste y aumentando la precisión en validación.
- VGG16 mostró un mejor rendimiento global en el escenario binario y en el multiclase cuando se utilizó *data augmentation*. ResNet50 tuvo comportamientos atípicos y por eso se descartó para este trabajo, pero sería interesante ver si con más datos es capaz de mejorar a VGG16.
- Las técnicas de aumento de datos más sencillas (rotación, desplazamiento, volteo, etc.) son suficientes para obtener mejoras notables, aunque su combinación cuidadosa es clave para evitar distorsiones semánticas.
- El uso de *transfer learning* permite aplicar modelos complejos preentrenados incluso en *datasets* pequeños, con buenos resultados y un coste computacional razonable.
- La combinación de *data augmentation* sumado al *transfer learning* es una estrategia eficaz, reproducible y flexible para enfrentar el problema de la escasez de datos en visión por computador.

7.2. Líneas de mejora y futuros trabajos

A partir de las observaciones realizadas, se proponen las siguientes líneas de trabajo para futuras investigaciones o desarrollos:

- Aplicar *grid search* para explorar las combinaciones de hiperparámetros y optimizar los que se utilizan en el entrenamiento.
- Explorar técnicas más avanzadas de *augmentation*, como *MixUp*, *CutMix* o *AutoAugment*, que han demostrado en otros artículos de investigación mejoras significativas en tareas complejas.
- Utilizar métodos de explicabilidad de modelos (como Grad-CAM o LIME) para analizar qué regiones de la imagen utilizan los modelos para tomar decisiones, lo que permitiría detectar sesgos o dependencias indeseadas.
- Ampliar el análisis a otros tipos de modelos más recientes, como EfficientNet o Vision Transformers, que podrían ofrecer mejoras de rendimiento adicionales con menos parámetros.

Para finalizar, este trabajo ha demostrado que es posible mejorar el rendimiento de modelos convolucionales en contextos con pocos datos mediante técnicas accesibles y bien fundamentadas.

Los resultados obtenidos refuerzan el valor de aplicar estrategias simples pero eficaces, que permitan avanzar en tareas reales de clasificación de imágenes sin necesidad de disponer de grandes recursos computacionales ni conjuntos de datos masivos.

Bibliografía

- [1] R. Bellman, *Adaptive Control Processes: A Guided Tour*, NJ: Princeton University Press, 1961.
- [2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, Las Vegas, NV, 2016, pp. 770–778.
- [4] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
- [5] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv preprint arXiv:1712.04621*, 2017.
- [6] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [7] M. Verleysen and D. François, “The curse of dimensionality in data mining and time series prediction,” in *Proc. Int. Work-Conf. Artif. Neural Netw.*, 2005, pp. 758–770.
- [8] G. Hughes, “On the mean accuracy of statistical pattern recognizers,” *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, 1968.
- [9] E. Keogh and A. Mueen, “Curse of dimensionality,” in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer, 2017, pp. 314–315.
- [10] T. Rahman, M. Chowdhury, A. Khandakar, M. Mahbub, A. Islam, and M. Ayari, “COVID-19 Radiography Database,” *Kaggle Datasets*, 2021. [Online]. Available: <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>
- [11] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” California Institute of Technology, Pasadena, CA, Tech. Rep. CNS-TR-2011-001, 2011.

Apéndices

Apéndice A: Recursos

El código completo del trabajo, incluyendo notebooks, scripts de entrenamiento, visualizaciones y documentación técnica, está disponible en [github](#). Este repositorio contiene:

- Scripts de carga de datos y preprocesamiento.
- Implementación de modelos VGG16 y ResNet50 con y sin data augmentation.
- Visualización de resultados (matrices de confusión, curvas ROC, métricas).
- Código para reproducibilidad en Google Colab.

Los *datasets* utilizados se pueden obtener de los siguientes enlaces:

- [COVID-19 Radiography Dataset \(Kaggle\)](#)
- [CUB-200-2011 Dataset](#)