Entrenamiento de CNN para Clasificación de Imágenes COVID-19

Marta Cuevas Rodríguez

Herramientas y Algoritmos en Bioinformática
Universidad de Málaga

Diciembre 2024

Índice

1. Introducción

La pandemia de COVID-19, causada por el SARS-CoV-2, desbordó los sistemas de salud a nivel mundial. En la detección y monitoreo de la enfermedad, las radiografías de tórax han jugado un papel fundamental debido a su accesibilidad y rapidez frente a otras pruebas diagnósticas como la RT-PCR. Estas imágenes permiten identificar signos de infecciones respiratorias, como opacidades pulmonares o infiltrados, comunes en COVID-19, neumonía viral y bacteriana, entre otras patologías.

Este trabajo se enfoca en desarrollar un modelo de red neuronal convolucional (CNN) para clasificar imágenes de radiografías en cuatro categorías: *COVID-19*, *NORMAL*, *PNEUMONIA* y *Lung Opacity*. Para ello, se utiliza el conjunto de datos *COVID-19 Radiography Dataset*, el cual contiene imágenes etiquetadas y balanceadas. El objetivo principal es analizar el rendimiento de las CNN para esta tarea, implementando técnicas de preprocesamiento, aumento de datos y optimización de hiperparámetros.

2. Objetivos

El objetivo principal de este proyecto es crear y evaluar un modelo basado en redes neuronales convolucionales (CNN) para la clasificación de imágenes de radiografías de tórax en las categorías *COVID-19*, *NORMAL*, *PNEUMONIA* y *Lung Opacity*.

Los objetivos específicos son:

- Preprocesar y organizar el conjunto de datos *COVID-19 Radiography Dataset* para su uso en el entrenamiento del modelo.
- Diseñar y entrenar una red neuronal convolucional (CNN) que clasifique las imágenes en las categorías mencionadas.
- Implementar técnicas de aumento de datos para mejorar la capacidad de generalización del modelo.
- Ajustar los hiperparámetros y aplicar estrategias como *Early Stopping* para prevenir el sobreajuste.

3. Configuración Inicial

El entorno de trabajo fue configurado con las siguientes características:

- **Lenguaje:** Python con TensorFlow [1] y Keras [2].
- **Hardware:** CUDA para aceleración con GPU (aunque con errores de inicialización).
- **Dataset:** *COVID-19 Radiography Dataset* [4]., dividido en cuatro clases: COVID, NORMAL, PNEUMONIA y Lung_Opacity.
- **Preprocesamiento:** Redimensionamiento a 150×150 y normalización de imágenes en el rango $[0, 1]$.
- **División:** 80 % para entrenamiento y 20 % para validación.
- **Modelo:** Red neuronal convolucional simple.

4. Modelo CNN Simple con Cuatro Clases

4.1. Descripción del Modelo

En este primer enfoque, se implementó una red neuronal convolucional (CNN) simple para clasificar las imágenes en cuatro categorías: *COVID*, *NORMAL*, *PNEUMONIA* y *Lung Opacity*. El modelo consta de:

- Tres bloques de capas `Conv2D` con activación ReLU y `MaxPooling2D` para reducir la dimensionalidad.
- Una capa densa con 128 neuronas y activación ReLU, seguida de un Dropout del 50 % para evitar el sobreajuste.
- Una capa de salida con activación *softmax* para las cuatro clases.

Las imágenes fueron redimensionadas a 150×150 com se indicó en la configuración inicial y los valores de los píxeles se normalizaron en el rango $[0, 1]$. El modelo fue entrenado con un tamaño de lote inicial de 32, utilizando *categorical crossentropy* como función de pérdida y el optimizador Adam durante **10 épocas**.

4.2. Resultados y Análisis

El modelo alcanzó una precisión global del **35 %** en el conjunto de validación, con los siguientes resultados por clase:

- **COVID:** Precisión del 17 %.
- **NORMAL:** Precisión del 29 %.
- **PNEUMONIA:** Precisión del 49 %, la mejor entre las clases.
- **Lung Opacity:** Precisión baja del 6 %.

Estos resultados indican que el modelo tiene dificultades para distinguir entre las clases, especialmente en la categoría *Lung Opacity*. Las posibles causas de este rendimiento subóptimo incluyen la arquitectura relativamente simple del modelo, que podría no ser suficiente para capturar las características complejas presentes en las imágenes.

Además, es posible que exista un desbalance en el conjunto de datos, dado que la clase *Lung Opacity* tiene menos muestras que las otras clases, lo que podría dificultar el aprendizaje de patrones representativos para esta categoría. Finalmente, la falta de técnicas de preprocesamiento o aumento de datos más avanzadas podría estar limitando la capacidad del modelo para generalizar mejor en los casos de *Lung Opacity*.

5. Modelo CNN con Dropout y Early Stopping

5.1. Descripción del Modelo

Para abordar el sobreajuste observado en el modelo anterior, se implementaron dos técnicas de regularización:

- **Dropout:** Se añadieron capas `Dropout` del 20 % después de cada bloque convolucional y del 30 % antes de la capa de salida.
- **Early Stopping:** Se configuró un callback para detener el entrenamiento si la pérdida de validación no mejoraba durante 3 épocas consecutivas.

Además, se simplificó el modelo reduciendo la cantidad de neuronas en la capa densa final a 64. El resto de la arquitectura permaneció igual.

Por otro lado, la distribución de los datos en el conjunto de entrenamiento y validación es la siguiente:

Clase	Entrenamiento	Validación
COVID	5786 imágenes	1446 imágenes
Lung_Opacity	9620 imágenes	2404 imágenes
Normal	16308 imágenes	4076 imágenes
Viral Pneumonia	2152 imágenes	538 imágenes

Cuadro 1: Distribución de las clases en los conjuntos de entrenamiento y validación.

5.2. Resultados y Análisis

El modelo con Dropout y Early Stopping obtuvo una precisión global del **36 %**, con los siguientes resultados por clase:

- **COVID:** 19 % de precisión.
- **NORMAL:** 28 % de precisión.
- **PNEUMONIA:** 48 % de precisión, la mejor entre las clases.
- **Lung Opacity:** 6 % de precisión.

Los resultados se resumen en la siguiente tabla:

Cuadro 2: Reporte de Clasificación

Clase	Precisión	Exhaustividad (Recall)	F1-Score	Soporte
COVID	0.19	0.19	0.19	1446
NORMAL	0.28	0.25	0.26	2404
PNEUMONIA	0.48	0.52	0.50	4076
Lung_Opacity	0.06	0.05	0.05	538
Precisión			0.36	
Promedio macro	0.25	0.25	0.25	8464
Promedio ponderado	0.35	0.36	0.35	8464

Durante el entrenamiento, la exactitud de validación superó significativamente la exactitud del entrenamiento (ver Figura 1). Este comportamiento podría ser causado por sobreajuste, un uso excesivo de la regularización como el Dropout, que pudo haber limitado la capacidad del modelo para aprender patrones complejos, o por la distribución desequilibrada del dataset, que favorece a las clases con más muestras.

Además, en la curva ROC (Figura 2) se pueden observar que las cuatro clases muestran líneas diagonales, lo que indica que el modelo está realizando predicciones aleatorias. Esto suele suceder debido a un desequilibrio en las clases o un entrenamiento inadecuado, posiblemente por un exceso de regularización. Esto sugiere que el modelo no está logrando diferenciar correctamente entre las clases.

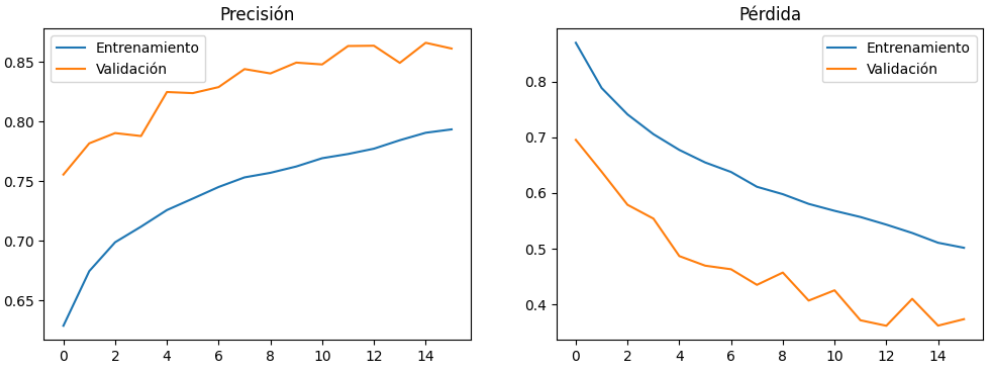

Figura 1: Curvas de exactitud del entrenamiento y validación

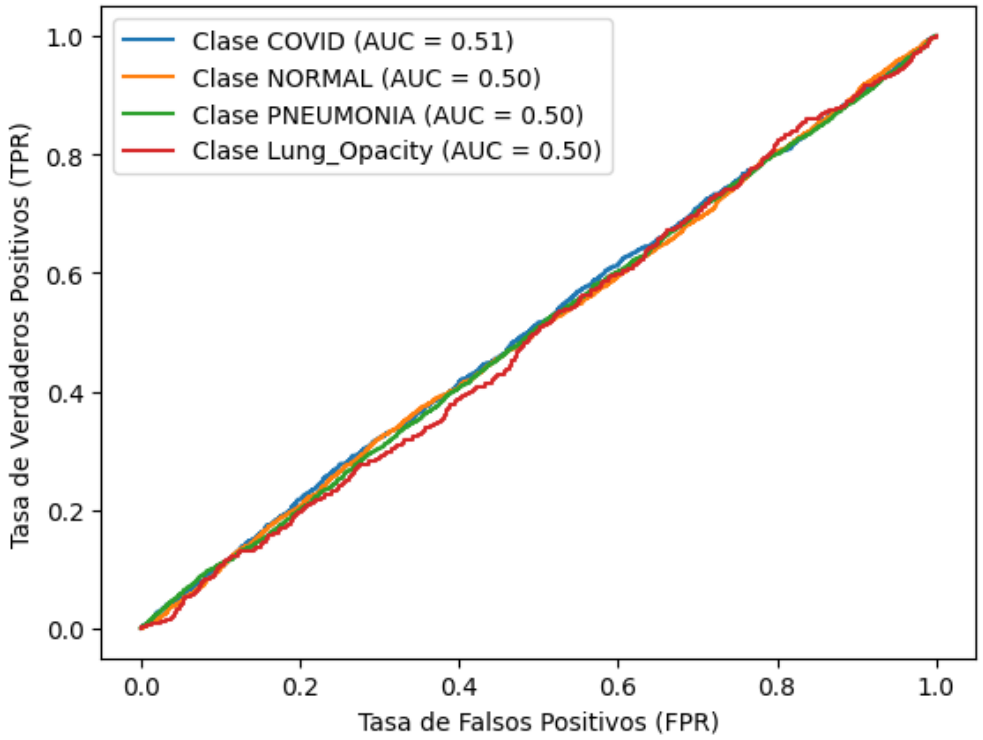

Figura 2: Curva ROC para el segundo modelo

6. Modificación del modelo y evaluación de desempeño

En esta etapa, se implementó un modelo basado en una estructura más típica de clasificación de imágenes utilizando una arquitectura de red neuronal convolucional (CNN) con tres capas convolucionales, seguidas de capas de max pooling. Se añadió además una capa densa al final del modelo, junto con una capa de *Dropout* para intentar mitigar el sobreajuste.

6.1. Cambios implementados

- Se mantuvieron las tres capas convolucionales con un aumento en el número de filtros de 32, 64 y 128 respectivamente, para extraer características más complejas de las imágenes.
- Se introdujo una capa de *Dropout* del 50 % después de la capa densa para prevenir el sobreajuste, con la esperanza de que el modelo pudiera generalizar mejor.
- Se utilizó la función de activación *relu* para las capas convolucionales y la capa densa, y *softmax* para la salida, adecuada para clasificación multiclase.
- La compilación y el entrenamiento del modelo se realizaron utilizando el optimizador *Adam* y la función de pérdida *categorical crossentropy*, ya que es un problema de clasificación multiclase.

6.2. Resultados y Análisis

Durante las primeras épocas del entrenamiento, los resultados fueron los siguientes: en la primera época, la exactitud de entrenamiento fue de 51.72 %, con una exactitud de validación de 65.90 %. La pérdida en entrenamiento fue de 1.1066, mientras que en validación fue de 0.8270. En la segunda época, la exactitud de entrenamiento aumentó a 63.12 %, y la pérdida en entrenamiento fue de 0.9021.

A pesar de la mejora en la exactitud de validación, la diferencia entre entrenamiento y validación no es suficientemente significativa para considerar que el modelo haya logrado un buen rendimiento generalizado. La exactitud en entrenamiento sigue siendo baja, lo que sugiere que el modelo no ha aprendido de manera efectiva de los datos, indicando un posible subajuste o una arquitectura demasiado simple. Además, la pérdida en entrenamiento es considerablemente mayor que en validación, lo que podría señalar que el modelo no está capturando adecuadamente los patrones en los datos de entrenamiento. Aunque se aplicó una capa de *Dropout* para evitar el sobreajuste, esta no resolvió el problema del subajuste, y el modelo sigue teniendo dificultades para aprender de los datos.

En resumen, a pesar de la implementación de una estructura típica de CNN con *Dropout*, los resultados no muestran mejoras sustanciales. Esto sugiere que el modelo sigue enfrentando problemas tanto en el aprendizaje de las características como en la generalización.

7. Análisis del Modelo con MobileNetV2 Preentrenado

En esta parte, se probó el uso de un modelo preentrenado, específicamente *MobileNetV2*, para mejorar el rendimiento en la tarea de clasificación de radiografías de COVID-19.

En esta sección, se utilizó el modelo preentrenado ***MobileNetV2*** [5], una arquitectura ligera y eficiente diseñada para dispositivos móviles. Este modelo fue preentrenado en el conjunto de datos ***ImageNet*** [3], que contiene millones de imágenes clasificadas en diversas categorías, lo que facilita su adaptación a tareas específicas como la clasificación de radiografías médicas.

A continuación, se detallan las modificaciones realizadas, los resultados obtenidos y un análisis de su desempeño.

7.1. Modificaciones Realizadas

Se realizaron las siguientes modificaciones clave:

- **Uso de un modelo preentrenado MobileNetV2**: Se cargó MobileNetV2 preentrenado con pesos de ImageNet, excluyendo las capas superiores, para usar solo las capas convolucionales para la extracción de características.

- **Congelación de capas**: Se congelaron las capas de MobileNetV2, excepto por las últimas 50 capas, para que el modelo pudiera seguir aprendiendo sobre características específicas de nuestro conjunto de datos, mientras se aprovechaban las representaciones preentrenadas.
- **Construcción de un nuevo modelo**: Se añadió una capa de `GlobalAveragePooling2D` después de la base de MobileNetV2, seguida de capas densas con `ReLU` y `Dropout(0.5)`, finalizando con una capa `softmax` para la clasificación multiclase.

7.2. Resultados Obtenidos

El modelo mostró un rendimiento decente desde el principio, con una precisión de validación del **74.13 %** en la primera época. Sin embargo, el rendimiento no mejoró significativamente después de esta época. La precisión alcanzada en la validación se estancó y no mostró una mejora sustancial a lo largo de las siguientes épocas.

A pesar de utilizar un modelo preentrenado, el rendimiento no mejoró significativamente en comparación con los intentos anteriores. Esto puede indicar que el **modelo preentrenado MobileNetV2 no logró ajustarse bien al conjunto de datos**, ya que las características aprendidas de ImageNet pueden no ser lo suficientemente representativas para las radiografías de COVID-19. Además, la **congelación de capas no fue adecuada**, ya que al congelar muchas de las capas, el modelo no tuvo suficiente flexibilidad para aprender patrones específicos del nuevo conjunto de datos. También, a pesar de contar con un conjunto de datos relativamente grande, la **cantidad de datos de entrenamiento** podría haber sido insuficiente para que el modelo generalizara correctamente debido a la complejidad del problema. En resumen, aunque el uso de MobileNetV2 preentrenado proporcionó una mejora en el desempeño inicial, no logró un avance significativo con respecto a los modelos anteriores, lo que sugiere que se deben realizar más ajustes y pruebas para mejorar el rendimiento.

8. Análisis con Modelo Reducido a Dos Clases

Finalmente, se probó un modelo donde se redujo el número de clases a solo **COVID** y **NORMAL**.

8.1. Modificaciones Realizadas

La modificación clave en este experimento fue:

- **Reducción del número de clases**: Se eliminaron las clases **PNEUMONIA** y **Lung Opacity**, quedando solo las clases **COVID** y **NORMAL** para simplificar el problema y evaluar el rendimiento en un escenario de clasificación binaria.
- **Generación de datos**: Los generadores de datos para entrenamiento y validación fueron configurados para dividir el dataset de manera adecuada, con un 20 % de los datos destinados a validación.
- **Modelo de red neuronal**: Se utilizó una red neuronal convolucional estándar con tres capas convolucionales, capas de max pooling, y una capa densa final con activación softmax.
- **Early stopping**: Se mantuvo la técnica de **early stopping** para evitar el sobreajuste, monitoreando la precisión de validación y deteniendo el entrenamiento cuando no se observaban mejoras.
- **Prescindir del dropout**: Se decidió prescindir del uso de **dropout** para permitir que el modelo aprenda mejor las representaciones de los datos sin restricciones adicionales.

8.2. Resultados Obtenidos

El modelo mostró una precisión de **0.63** en el conjunto de validación. A pesar de que la clase **NORMAL** presentó un desempeño significativamente mejor, el modelo mostró dificultades para clasificar correctamente las imágenes de la clase **COVID**.

A continuación se presenta el informe de clasificación obtenido después de entrenar el modelo.

Cuadro 3: Informe de Clasificación

Clase	**Precisión**	**Recall**	**F1-Score**	**Soporte**
COVID	0.27	0.23	0.25	1446
NORMAL	0.74	0.77	0.76	4076
Precisión			0.63	
Promedio macro	0.50	0.50	0.50	5522
Promedio ponderado	0.62	0.63	0.62	5522

Aunque los resultados obtenidos aún no son ideales, las curvas de **accuracy** y **loss** (Figura 3) muestran un comportamiento más esperado durante el entrenamiento. En la gráfica se puede observar que, a medida que avanzan las épocas, la **exactitud** en el conjunto de entrenamiento y validación mejora de manera progresiva, lo que indica que el modelo está aprendiendo correctamente de los datos. La **pérdida**, por su parte, muestra una disminución constante tanto en entrenamiento como en validación, lo cual es una señal positiva de que el modelo está optimizando su capacidad para predecir las clases correctamente. A pesar de que no se alcanzaron los niveles de precisión deseados, las curvas reflejan un rendimiento más estable y coherente en comparación con intentos anteriores, lo que sugiere que el modelo está avanzando hacia una mejor generalización.

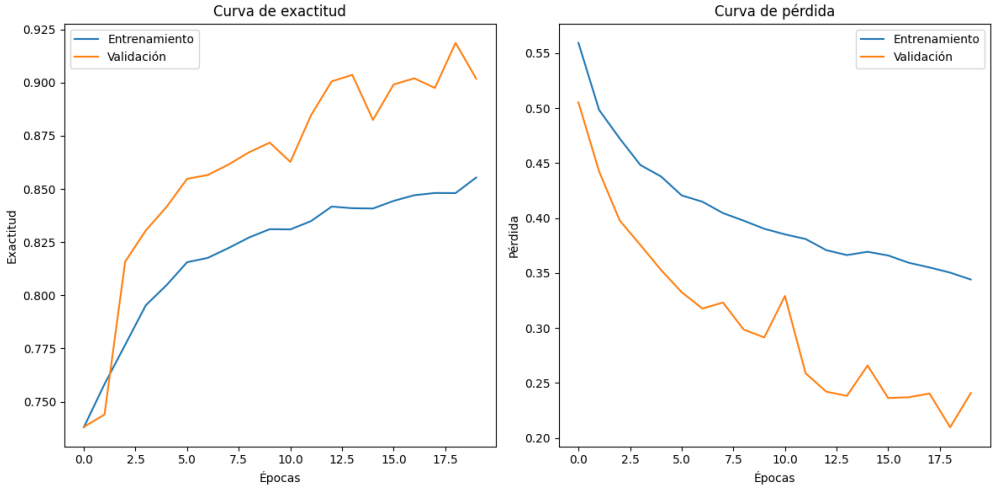

Figura 3: Curvas de exactitud del entrenamiento y validación para los conjuntos de datos COVID y NORMAL

La curva ROC por su lado (Figura 4), sigue sin mostrar unos resultados coherentes, ya que continúa apareciendo casi diagonal. Esto indica que el modelo no está siendo capaz de distinguir adecuadamente entre las clases, lo que sugiere que el desempeño en la clasificación binaria sigue siendo subóptimo. Una curva ROC diagonal, que representa un modelo aleatorio, indica que el modelo no está aprendiendo patrones útiles de los datos y que la capacidad de discriminación entre las clases sigue siendo insuficiente.

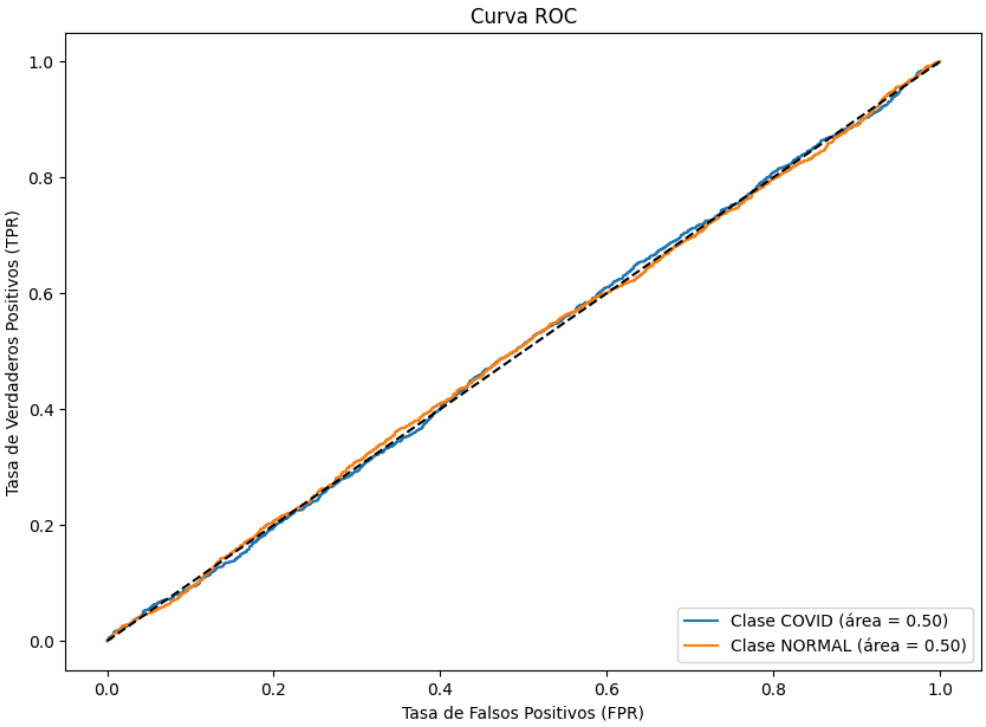

Figura 4: Curvas ROC para los conjuntos de datos COVID y NORMAL

En cuanto a la matriz de confusión, los resultados obtenidos son los siguientes:

Esto indica que, de los 1446 ejemplos de la clase *COVID*, solo 338 fueron correctamente clasificados, mientras que 1108 fueron clasificados erróneamente como *NORMAL*. Por otro lado, de los 4076 ejemplos de la clase *NORMAL*, 3147 fueron correctamente clasificados, mientras que 929 fueron mal clasificados como *COVID*. Esto refleja un claro desajuste en la clasificación de la clase minoritaria *COVID*, lo que está afectando negativamente el rendimiento global del modelo.

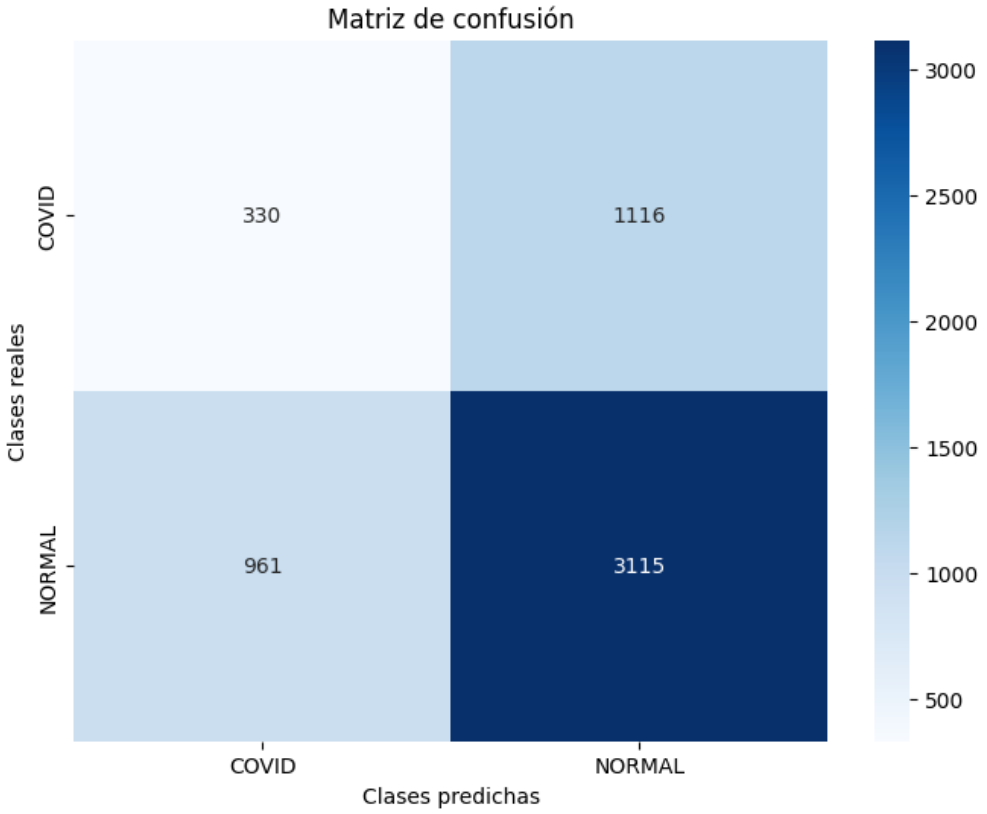

Figura 5: Matriz de confusión para los conjuntos de datos COVID y NORMAL

8.3. Análisis de Resultados

Durante la experimentación con el modelo reducido a dos clases, se observaron los siguientes puntos clave:

- **Rendimiento desigual entre clases**: La clase *NORMAL* tiene un desempeño significativamente mejor que la clase *COVID*, lo que indica que el modelo podría estar sesgado hacia la clase mayoritaria, *NORMAL*.
- **Desbalance de clases**: Es posible que el conjunto de datos esté desbalanceado entre las clases, lo que podría afectar la capacidad del modelo para aprender características representativas de la clase *COVID*. Se podría explorar el uso de técnicas como el *oversampling* o el *undersampling* para equilibrar las clases.
- **Precisión de validación**: La precisión en el conjunto de validación alcanzó un **91 %**, lo que sugiere que el modelo tiene un buen desempeño en general, pero podría mejorarse en términos de clasificación de la clase *COVID*.

La reducción a dos clases ha mejorado el rendimiento global del modelo en términos de precisión en comparación con el modelo de múltiples clases, pero todavía presenta un desempeño subóptimo para la clasificación de *COVID*.

9. Discusión y Conclusiones

A pesar de los cambios implementados, el modelo aún presenta un rendimiento subóptimo. Aunque la reducción del problema a dos clases (*COVID* y *NORMAL*) simplificó la clasificación, las dificultades para detectar la clase *COVID* persisten. Las técnicas de aumento de datos y regularización (como el Dropout) contribuyeron a mitigar el sobreajuste, pero no fueron suficientes para mejorar de manera significativa el desempeño del modelo.

El uso de un modelo preentrenado como MobileNetV2 no proporcionó mejoras sustanciales, lo que sugiere que la naturaleza del conjunto de imágenes, posiblemente con características complejas o difíciles de generalizar, dificulta el entrenamiento. Además, el desbalance de clases entre *COVID* y *NORMAL* probablemente sigue afectando la capacidad del modelo para aprender adecuadamente de la clase minoritaria.

En resumen, aunque se implementaron varios enfoques para mejorar el rendimiento, el modelo sigue enfrentando desafíos significativos debido a la dificultad del conjunto de datos y el desequilibrio entre clases. Futuras mejoras podrían incluir técnicas más avanzadas de balanceo de clases, métricas adicionales como AUC-ROC para evaluar mejor la clase minoritaria y la implementación de redes más profundas.

Referencias

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. `https://www.tensorflow.org/`.

[2] François Chollet et al. Keras, 2015. `https://keras.io`.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[4] Tawsifur Rahman, Md. Mahmudur Rahman, A. S. M. Ashraful Alam, Sakib Mahmud, Sharif A. Chowdhury, Abul Kashem Sarker, Qi Jing, Nabil Ibtehaz, Abidur Rahman Khan, Md. Nazmus Saqib, Khandakar Tanveer Ahmed, Kaushik Sarker, Rushdi Shams, Muhammad E. H. Chowdhury, and U. Rajendra Acharya. Covid-19 radiography database, 2020. https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database/data.

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.