

Prediagnóstico de enfermedades respiratorias a partir de imágenes de tomografías computarizadas torácicas mediante aprendizaje profundo

Francisco Javier Aragón-González¹, Abril Valeria Uriarte-Arcia²,
Luz María Sánchez-García¹

¹ Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
México

² Instituto Politécnico Nacional,
Centro de Innovación y Desarrollo Tecnológico en Cómputo,
México

faragong1300@alumno.ipn.mx, {auriartea,lmsanchez}@ipn.mx

Resumen. En esta investigación, se evaluaron tres arquitecturas de redes neuronales convolucionales: VGG16, ResNet-50 e InceptionV3, en la tarea de clasificación sobre un dataset de imágenes de tomografías computarizadas torácicas (TC). El dataset proviene de un hospital en México, cuyo nombre se omite debido a razones de privacidad y confidencialidad. Estas imágenes representaban dos clases de pacientes: pacientes que egresaron por mejora y pacientes que fallecieron debido a neumonía por COVID-19. Esta clasificación es particularmente desafiante, ya que las imágenes entre estas dos clases son similares, dado que todos los individuos están enfermos, a diferencia de estudios donde las categorías pueden ser visiblemente diferentes, por ejemplo, pacientes sanos vs. pacientes enfermos. Se realizó una búsqueda por cuadrícula (*grid search*) para determinar los mejores valores de hiperparámetros para cada modelo. También se aplicó preprocesamiento a las imágenes como segmentación del pulmón y mejora del brillo. Los resultados obtenidos indicaron que la mejora de brillo tuvo el impacto más positivo en el rendimiento general. Los mejores resultados obtenidos fueron presentados por la VGG16, alcanzando F1-score de 90% y *accuracy* de 89%. Este trabajo demuestra que las redes neuronales convolucionales tienen el potencial de beneficiar significativamente al campo de la salud. A futuro, se propone explorar imágenes con mayor profundidad tonal, pasando de 8 bits a 16 bits por canal, para capturar variaciones más sutiles, lo que podría ser esencial en contextos médicos.

Palabras clave: Prognosis de pacientes, COVID-19, Redes neuronales convolucionales.

Prediagnosis of Respiratory Diseases from Computed Thoracic Tomogram Images Using Deep Learning

Abstract. In this research, three convolutional neural network architectures were evaluated: VGG16, ResNet-50 and InceptionV3, in the classification task on a dataset of thoracic computed tomography (CT) images. The dataset comes from

a hospital in Mexico, whose name is omitted due to privacy and confidentiality reasons. These images represented two classes of patients: patients who were discharged due to improvement and patients who died due to COVID-19 pneumonia. This classification is particularly challenging, since the images between these two classes are similar, given that all individuals are sick, unlike studies where the categories may be visibly different, for example, healthy patients vs. sick patients. A grid search was performed to determine the best hyperparameter values for each model. Preprocessing such as lung segmentation and brightness enhancement was also applied to the images. The results obtained indicated that the brightness improvement had the most positive impact on overall performance. The best results obtained were presented by the VGG16, reaching F1-score of 90% and accuracy of 89%. This work demonstrates that convolutional neural networks have the potential to significantly benefit the healthcare field. In the future, it is proposed to explore images with greater tonal depth, going from 8 bits to 16 bits per channel, to capture more subtle variations, which could be essential in medical contexts.

Keywords: Patient prognosis, COVID-19, convolutional neural.

1. Introducción

La neumonía es una infección respiratoria aguda que afecta a los pulmones. En México es la novena causa de mortalidad en todos los grupos etarios, ocasionada generalmente por bacterias y en menor proporción por virus y hongos; daña en particular a niños y a adultos mayores. Cuando se detecta a tiempo, es controlable; ante las primeras manifestaciones, que llegan a confundirse con una gripe [1].

Para complementar un diagnóstico, los doctores se pueden apoyar de diferentes recursos para analizar el estado de los pulmones, por ejemplo, radiografía de tórax y tomografía computarizada. La tomografía computarizada torácica (TC) tiene una mayor sensibilidad que la radiografía de tórax y permite valorar tanto la afectación pulmonar como posibles complicaciones, además de proporcionar diagnósticos alternativos [2].

Dado que la gravedad de la enfermedad implica una amenaza directa para la vida del paciente, es necesario contar con herramientas que auxilien el diagnóstico médico del paciente con alguna enfermedad respiratoria.

La utilidad de esta investigación es aportar una herramienta para ayudar al personal de la salud a complementar su diagnóstico médico, basándose en métodos y técnicas de Deep learning para la clasificación de imágenes de tomografías torácicas. Uno de los principales problemas que enfrenta el sector salud es la falta de personal para atender la gran cantidad de pacientes, ocasionando que los diagnósticos se entreguen de forma tardía y en algunos casos erróneos. Se decidió utilizar imágenes de TC en lugar de imágenes de radiografías torácicas, debido a que en las imágenes de radiografías torácicas las estructuras se superponen, por ejemplo, las costillas se superponen a los pulmones [3].

De esta forma, y resaltando que el tiempo que dedica un doctor a un paciente en ocasiones es reducido debido a la carga de pacientes, es necesario contar con una herramienta de apoyo para complementar su diagnóstico.

Para esta investigación se utilizará un dataset de imágenes TC de pacientes que ingresaron a un hospital de México con un cuadro de neumonía con sospecha de ser ocasionada por COVID-19. El dataset se empleará para la tarea de clasificación, diferenciando entre pacientes que fueron dados de alta por mejoría y pacientes que recibieron alta por fallecimiento, de esta forma se abordará un problema de predicción utilizando algoritmos de clasificación. Así, se busca detectar a los pacientes con mayor riesgo de fallecimiento para intensificar el cuidado hacia ellos. El nombre del hospital no se puede revelar por cuestiones de confidencialidad al momento de ser redactado este documento.

2. Trabajo relacionado

En [4], los autores discuten el uso exitoso de algoritmos de machine learning para el diagnóstico asistido por computadora usando imágenes médicas. Resaltan las redes neuronales convolucionales por su eficacia en clasificar lesiones y anomalías cerebrales. También introducen un método que emplea transfer learning para distinguir radiografías torácicas de personas con y sin neumonía, utilizando la red preentrenada Xception con ImageNet.

En [5], los autores presentan un sistema para identificar casos de COVID-19 usando tomografías del tórax, aplicando modelos de Ensemble Learning. Usando transfer learning, transfieren conocimientos de una red previamente entrenada a una nueva tarea. Por otro lado, emplean Stacking y Weighted Average Ensemble (WAE) combinando tres modelos: VGG19, ResNet-50 y DenseNet201. Tras entrenar con imágenes de tomografías y evaluar con varias métricas, el método WAE demostró ser el más confiable, destacando en sensibilidad, vital para la detección precisa en contextos médicos.

En [6], los autores analizan imágenes médicas usando técnicas de machine learning y redes neuronales convolucionales. Discuten métodos de detección de enfermedades aplicados a imágenes de rayos X, tomografías y resonancias, resaltando sus pros y contras. Exploran arquitecturas de Deep learning detallando su funcionamiento. Entre los modelos de Deep learning utilizados en esta investigación destacan: LeNet, AlexNet, ZfNet, VGG16, GoogleNet, ResNet-50, ResNet-101 y DenseNet. Además, enfatizan la importancia de la segmentación de imágenes médicas en la detección de tumores y su relevancia en medicina.

En [7], los autores presentan un método de detección automatizada de infecciones pulmonares usando tomografías. Para mejorar la calidad de las imágenes, aplican una técnica de ecualización y eliminan áreas no relevantes segmentando el área de los pulmones.

Utilizan la arquitectura neuronal U-Net para segmentar las imágenes y una red convolucional de tres capas para la clasificación. Implementan validación cruzada dividiendo los datos en un 70% para entrenamiento y 30% para validación. Además, con el objetivo de obtener una estimación más precisa de la capacidad del modelo en datos no vistos, se ha empleado una cuádruple validación cruzada como procedimiento de remuestreo.

En este proceso, los datos se dividen en cuatro conjuntos, utilizándose tres de ellos para el entrenamiento y el restante para la validación en cada iteración. El sistema

Tabla 1. Distribución de imágenes del dataset por clase.

Clase (Egreso)	Cantidad
Defunción	660
Mejoría	544

exhibe una notable exactitud en diversas funciones críticas, alcanzando casi la perfección en la segmentación pulmonar y la clasificación general. Además, muestra una competencia muy alta en la detección de infecciones.

Con base en la investigación de trabajo relacionado se ha definido utilizar, en esta investigación, las redes neuronales convolucionales preentrenadas **ResNet50**, **VGG16** e **InceptionV3**, pues son las más usadas para resolver problemas similares al planteado en este trabajo. Las arquitecturas se utilizarán preentrenadas para reducir el tiempo de entrenamiento.

3. Materiales y método

3.1. Banco de datos

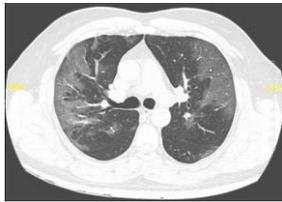
En esta investigación se utilizará un dataset de imágenes de tomografías torácicas (TC) anónimas de pacientes que ingresaron a un hospital del Estado de México, cuyo nombre no se puede revelar por cuestiones de confidencialidad. Las imágenes corresponden a tomografías torácicas de pacientes que fueron hospitalizados por un cuadro grave de neumonía bajo sospecha de COVID-19. El dataset inicial consiste en un total de **1,731 imágenes TC** nombradas con el ID del paciente, acompañadas de un archivo CSV que contiene 108 columnas con datos clínicos de cada paciente. Este archivo contiene una columna con el nombre **“Egreso”** que indica la forma en que el paciente salió del hospital. Dicha columna cuenta con 4 valores diferentes clases: **“Mejoría”, “Defunción”, “Voluntario” y “No se registró el tipo de egreso”**.

Se efectuó un preprocesamiento sobre el dataset para conservar información de interés, conservando solo los pacientes cuyo valor en esa columna fuera **“Mejoría” o “Defunción”**, siendo estas las clases que se usarán para plantear el problema de clasificación. Como resultado de este procesamiento el dataset cuenta finalmente con **1,204 imágenes**. La Tabla 1 muestra la distribución de las imágenes por cada clase.

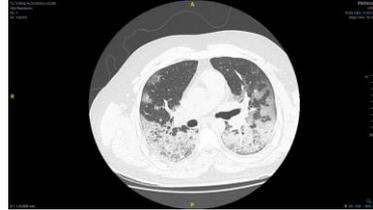
El problema planteado en este trabajo es complejo, dado que al examinar las imágenes del dataset, se encontró que son notablemente similares entre sí, independientemente del paciente del cual provengan, puesto que todos padecen algún grado de afectación pulmonar debido a la neumonía.

Esta semejanza es tal que las variaciones entre las imágenes son mínimas. Esto contrasta con otros estudios o investigaciones, donde la tarea es clasificar entre pacientes sanos y enfermos.

En esos casos, las imágenes TC tienden a presentar diferencias más evidentes, ya que, por ejemplo, en un paciente enfermo podríamos observar anomalías como opacidades pulmonares, las cuales no estarían presentes en un individuo sano. Esta clara distinción facilita el proceso de clasificación.



(a) Imagen correspondiente a clase Defunción.



(b) Imagen correspondiente a clase Mejoría.

Fig. 1. Ejemplos de imágenes TC correspondientes a ambas clases.

Sin embargo, en nuestro caso, la homogeneidad visual de las imágenes eleva la complejidad de categorizar adecuadamente. Ver ejemplo en la Fig. 1 donde se observa que el área de los pulmones en ambas imágenes presenta opacidades que indican la presencia de afectación pulmonar.

3.2. Procesamiento digital de imágenes

Las redes convolucionales pueden verse beneficiadas del preprocesamiento de las imágenes [7], por lo que se plantea el realizar experimentos donde se apliquen técnicas de procesamiento digital de imágenes.

La primera técnica de procesamiento digital que se utilizó sobre las imágenes fue un algoritmo de segmentación automática con el objetivo de obtener la parte central de la imagen TC. Esta zona central contiene el área de interés, que es el pulmón. El algoritmo utilizado para esta tarea se encargaba de detectar el objeto más grande dentro de la imagen, calculando los contornos de los objetos y quedándose con el más grande.

La segunda técnica aplicada fue para mejorar el brillo de la imagen con el uso de la función `cv2.convertScaleAbs()`. Dicha función realiza una operación lineal en cada píxel de la imagen, definida por la relación:

$$output = \alpha \times input + \beta \quad (1)$$

donde:

- **input**: es el valor de intensidad del píxel original.
- **output**: es el valor de intensidad del píxel ajustado.
- α : es un factor de ganancia que controla el contraste.
- β : es un valor de desplazamiento que controla el brillo.

Factor de Ganancia α (Contraste):

- Si $\alpha > 1$: aumenta el contraste de la imagen. Los valores de los píxeles se alejarán más de la intensidad media de la imagen.
- Si $\alpha = 1$: no se realiza ningún cambio en el contraste.
- Si $\alpha < 1$: reduce el contraste de la imagen. Los valores de los píxeles se acercarán más a la intensidad media de la imagen.
- El factor de ganancia α utilizado en este trabajo corresponde al valor de: **1.5**.

Tabla 2. *One-hot encoding* para la variable **Egreso**.

Valor Categórico	Codificación
Mejoría	[1 0]
Defunción	[0 1]

Valor de Desplazamiento β (Brillo):

- Si $\beta > 0$: aumenta el brillo de la imagen al añadir un valor constante a cada píxel.
- Si $\beta = 0$: no se realiza ningún cambio en el brillo.
- Si $\beta < 0$: reduce el brillo de la imagen al restar un valor constante a cada píxel.
- El valor de desplazamiento β utilizado en este trabajo corresponde al valor de: **50**.

El uso de la función **convertScaleAbs** garantiza que los valores resultantes estén en el rango válido [0, 255] para imágenes de 8 bits. Si el cálculo produce un valor por debajo de 0, se fija en 0, y si produce un valor por encima de 255, se fija en 255.

3.3. One-Hot Encoding

El método One-hot encoding permite convertir variables categóricas en valores numéricos usando vectores binarios. Esta técnica se emplea en machine learning cuando un algoritmo no puede usar valores categóricos o para mejorar resultados.

En este método, cada categoría única es representada por un vector cuyo tamaño corresponde al número de categorías, este vector tendrá el valor 1 en la posición que le corresponde a la categoría y el resto de posiciones estarán en cero.

Por ejemplo, para la variable Egreso de nuestro dataset, que tiene valores Mejoría y Defunción, se genera un vector de dos posiciones usando one-hot encoding, en donde la primera posición del vector corresponde a la categoría Mejoría y la segunda posición corresponde a la categoría Defunción. Si la categoría que se quiere codificar es Mejoría, la primera posición del vector tendrá un valor de 1 y la segunda posición un valor de cero y será lo contrario para la categoría Defunción. La Tabla 2 muestra esta codificación.

3.4. Método de validación

Como método de validación se usó **k-fold cross validation estratificado**. Este tipo de validación cruzada elimina el sesgo en la selección de datos que son usados para entrenamiento y prueba, así como garantizar que la proporcionalidad de los datos con respecto a la clase sea mantenida. Con este método los datos se dividen en k particiones, y el modelo se entrena k veces. En cada entrenamiento, se utilizan $k-1$ particiones como conjunto de entrenamiento y 1 partición como conjunto de prueba.

Esto se repite variando qué partes se emplean en el conjunto de entrenamiento y cuál se utiliza para el conjunto de prueba en cada ejecución. Esta técnica permite calcular métricas de rendimiento más precisas al minimizar el riesgo de resultados sesgados por

Tabla 3. Matriz de confusión: clase positiva = Defunción.

Real	Predicho	
	Mejoría	Defunción
Mejoría	VN	FP
Defunción	FN	VP

divisiones no representativas. Después de ejecutar el modelo utilizando las k particiones (folds), se calcula el promedio de los resultados.

3.5. Matriz de confusión y métricas

Una matriz de confusión es una herramienta utilizada en machine learning para evaluar el rendimiento de un algoritmo de clasificación. Muestra cómo se distribuyen las predicciones del modelo en comparación con los valores verdaderos.

La matriz se presenta en un formato tabular y, para una clasificación binaria, tiene cuatro componentes principales (ver Tabla 3):

- **Verdaderos Positivos (VP):** casos en los que el modelo predijo positivo y la verdadera clase también es positiva.
- **Verdaderos Negativos (VN):** casos en los que el modelo predijo negativo y la verdadera clase también es negativa.
- **Falsos Positivos (FP):** casos en los que el modelo predijo positivo pero la verdadera clase es negativa.
- **Falsos Negativos (FN):** casos en los que el modelo predijo negativo pero la verdadera clase es positiva.

Las métricas de evaluación sirven para cuantificar el rendimiento y la calidad de los modelos predictivos, especialmente en tareas de clasificación y regresión. Existen varias métricas de evaluación. A continuación, se presentan las métricas de evaluación para tareas de clasificación que serán usadas en este trabajo.

Accuracy (Exactitud): mide la proporción de predicciones correctas en el conjunto total de observaciones:

$$\text{Accuracy} = \frac{VP+VN}{VP+VN+FP+FN}$$

Recall (Sensibilidad): indica la capacidad del modelo para identificar correctamente todas las muestras positivas reales:

$$\text{Recall} = \frac{VP}{VP + FN}$$

Precision (Precisión): muestra la capacidad del modelo para no etiquetar como positiva una muestra que es negativa:

$$\text{Precision} = \frac{VP}{VP+FP}$$

F1-Score: combina Precision y Recall en una única métrica, buscando un balance entre ambas:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3.6. Modelos de redes neuronales convolucionales

VGG16: es una red neuronal convolucional profunda compuesta por 16 capas. Está estructurada en bloques: utiliza múltiples capas convolucionales con filtros 3x3 seguidas de una capa de max pooling. Estas capas convolucionales extraen características jerárquicas de las imágenes.

Después de procesar a través de todos los bloques convolucionales, la información se aplana y se pasa a través de tres capas totalmente conectadas para la clasificación final. La VGG16 es conocida por su simplicidad, utilizando consistentemente filtros 3x3 y su capacidad para capturar características complejas en imágenes [8].

ResNet-50: es una variante de las redes residuales o “ResNets”. La característica clave de las ResNets es el uso de conexiones residuales o “skip connections”, que permiten que las activaciones se salten una o más capas. Estas conexiones se añaden a la salida de una capa y se suman a la entrada de una capa posterior, facilitando el entrenamiento de redes muy profundas al abordar el problema de la desaparición del gradiente.

La ResNet-50 tiene 50 capas, incluyendo tanto capas convolucionales como totalmente conectadas. Se compone de bloques residuales donde cada bloque tiene tres capas convolucionales. Estas capas tienen filtros de diferentes tamaños (1x1, 3x3, 1x1) diseñados para reducir la dimensionalidad, capturar características espaciales y luego aumentar la dimensionalidad [9].

InceptionV3: es una variante de la familia de redes “Inception”, también conocidas como GoogleNet. El módulo de Inception contiene múltiples operaciones paralelas con diferentes tamaños de filtro convolucional (por ejemplo, 1x1, 3x3, 5x5) y también incluye pooling. Al combinar todas estas operaciones, la red puede capturar una variedad de características espaciales en diferentes escalas simultáneamente. InceptionV3, en particular, mejora las versiones anteriores con optimizaciones para mejorar la eficiencia y el rendimiento.

Una de las características distintivas de InceptionV3 es el uso de factorizaciones convolucionales, donde una convolución 2D se descompone en dos convoluciones 1D consecutivas, lo que reduce el número total de parámetros y acelera la red [10].

4. Diseño experimental

4.1. Flujo experimental

En este trabajo se realizaron varios experimentos con el objetivo de obtener los mejores resultados de los modelos. En la primera etapa de experimentos se ejecutaron los 3 modelos de redes convolucionales con el banco de datos original.

Con este banco de datos se ejecutó el proceso de grid search que se detalla en la sección 4.2. Una vez determinado los mejores hiperparámetros para cada modelo, se procedió a realizar la aumentación de datos, que se presenta en la sección 4.3, en cada modelo. Finalmente se selecciona el mejor modelo con base en la métrica F1-score. La

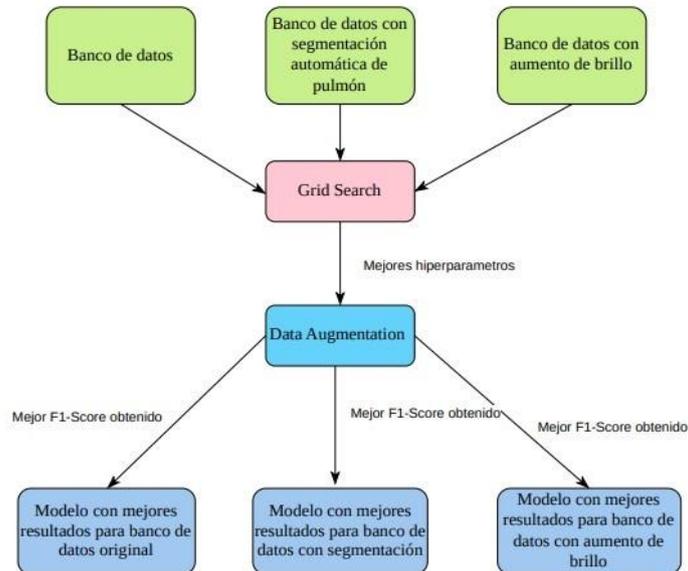


Fig. 2. Diagrama de flujo del diseño experimental.

segunda etapa de experimentación sigue el mismo flujo, pero se utiliza el banco de datos modificado mediante la técnica de segmentación que se presenta en la sección 3.2. El mismo flujo se sigue en la tercera etapa de experimentos, solo que en esta ocasión se usa el banco de datos modificado con la técnica de ajuste de brillo que también se detalla en la sección 4.3. En la Fig. 2 se muestra el flujo experimental.

4.2. Grid Search

El Grid Search o búsqueda en cuadrícula, es un método que realiza diferentes combinaciones de hiperparámetros y permite elegir la mejor combinación, es decir aquella con la que un determinado modelo produce un menor error. En este trabajo se realizó Grid Search para los modelos ResNet-50, VGG16 e InceptionV3 tomando en cuenta los hiperparámetros de learning rate y optimizador.

Se evaluaron **9 combinaciones** que se derivan de las combinaciones posibles de los valores de learning rate (0.00001, 0.0001 y 0.001) y los valores del optimizador (SGD, Adam y RMSprop). Por otro lado, la función de pérdida utilizada en todas las combinaciones y experimentos fue **nn.CrossEntropyLoss**, para evaluar la discrepancia entre las predicciones y las etiquetas reales.

Cada combinación fue probada para cada modelo mediante entrenamientos de **100 épocas**, y en cada iteración se guardaba el modelo cuando se alcanzaba el valor de pérdida más bajo en el conjunto de validación. Para concluir, se compararon los resultados de cada combinación para determinar cuál de todos tenía el valor más bajo de error para determinar la mejor combinación de hiperparámetros (ver Tabla 4).

4.3. Aumentación de datos

Una vez que identificada la mejor arquitectura con base en sus hiperparámetros mediante Grid Search, se llevaron a cabo experimentos adicionales para determinar si la incorporación de técnicas de aumentación de datos (**Data Augmentation**) podría mejorar aún más el rendimiento. Estos experimentos se realizaron usando diferentes combinaciones de los siguientes métodos de Data Augmentation: “**Random Horizontal Flip**”, “**Random Equalization**” y “**Normalization**”.

4.4. Selección del modelo

La selección del mejor modelo en cada experimento realizado se hizo con base en los resultados obtenidos por la métrica F1-Score, dado que dicha métrica toma en cuenta el promedio armónico entre la precisión (Precision) y la sensibilidad (Recall). Esta métrica es importante dado que, en aplicaciones médicas, tanto los falsos positivos como los falsos negativos pueden tener consecuencias significativas. Un falso positivo puede llevar a pruebas médicas adicionales innecesarias, por otro lado, un falso negativo puede resultar en una enfermedad o condición no diagnosticada a tiempo.

El F1-Score, al considerar tanto Precision como Recall, proporciona una métrica que penaliza ambos tipos de errores. Al buscar el mejor ajuste en esta métrica, estamos buscando un balance en el desempeño del clasificador entre Precision y Recall.

En este estudio, se designó como clase positiva a la categoría “Defunción”. Esto se debe al interés particular en identificar a los pacientes con riesgo de fallecimiento.

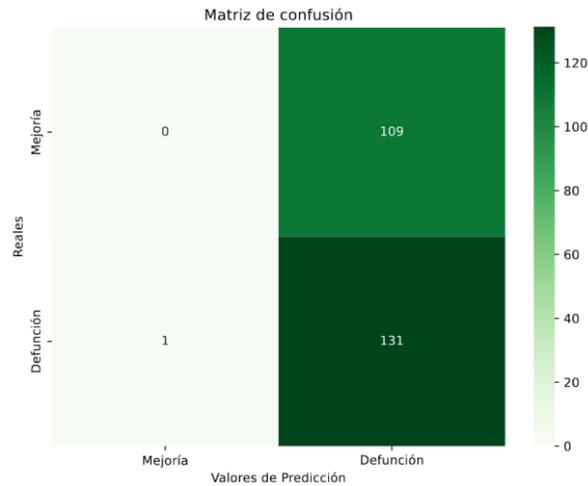
5. Resultados y discusión

5.1. Modificaciones a modelos

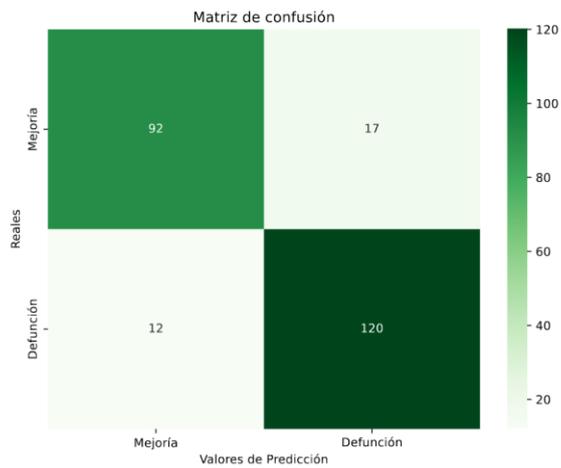
En este estudio, se emplearon tres modelos preentrenados de redes neuronales convolucionales: **InceptionV3**, **ResNet-50** y **VGG16**. Durante los primeros experimentos realizados se notaron algunos problemas en el entrenamiento de 2 de los modelos, con el objetivo de mejorar los resultados se realizaron algunas modificaciones a las arquitecturas respectivas.

Para el modelo InceptionV3, los resultados de la clasificación eran muy bajos, como se puede observar en la Fig. 3a. Para mejorar estos resultados se modificó la arquitectura añadiendo tres capas completamente conectadas. Esta modificación fue necesaria debido a que la configuración original del modelo tenía dificultades para clasificar adecuadamente la categoría “Mejoría”, donde las imágenes de esa categoría se estaban clasificando en la clase “Defunción” como se puede observar en los 109 Falsos Positivos que muestra la matriz de confusión. En la Fig. 3b se observan los resultados obtenidos después de realizar la modificación a la arquitectura.

Por otro lado, el modelo VGG16 mostraba problemas en el ajuste del error durante el entrenamiento, como se puede observar en la Fig. 4a, lo que hacía que el entrenamiento de la red, además de consumir más tiempo, no alcanzaría un mejor desempeño. Para mejorar este comportamiento se incorporó la técnica de Batch Normalization en la arquitectura del modelo. La decisión se basó en la capacidad de



(a) Matriz de confusión de InceptionV3 con arquitectura original.



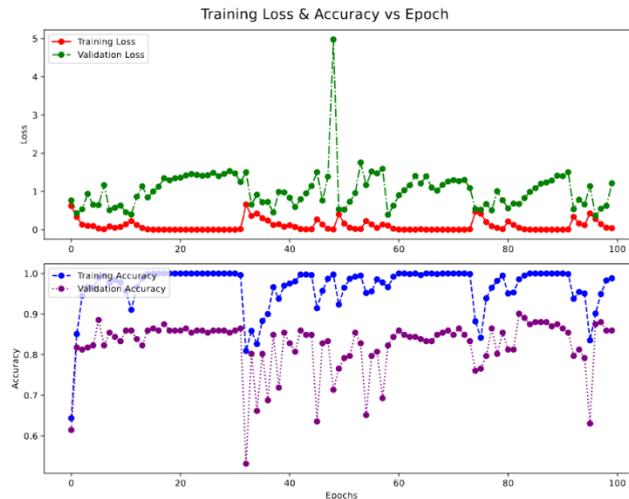
(b) Matriz de confusión de InceptionV3 con arquitectura modificada.

Fig. 3. Matrices de confusión de InceptionV3.

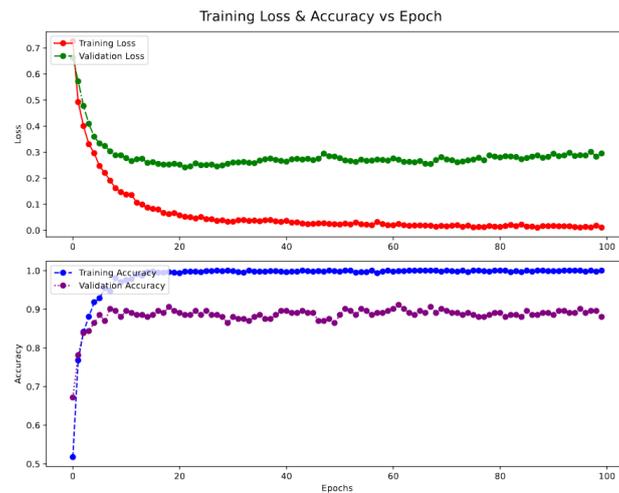
esta técnica para normalizar las activaciones de las capas, lo que puede promover un entrenamiento más estable y acelerar la convergencia. El comportamiento de la VGG16 se mejoró como podemos ver en la Fig. 4b.

2. Experimentos realizados con el Dataset original

Al realizar el Grid Search en los 3 modelos de redes neuronales convolucionales usando el banco de datos sin modificar, se determinaron los mejores hiperparámetros para cada arquitectura.



(a) VGG16 sin Batch Normalization.



(b) VGG16 con Batch Normalization.

Fig. 4. Gráficas de entrenamiento VGG16. – InceptionV3: los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son learning rate = 0.0001 y optimizador = Adam.

Hiperparámetros obtenidos con Grid Search

- **VGG16:** los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son **learning rate = 0.00001** y **optimizador = Adam**.

Tabla 5. Resultados VGG16 (dataset original).

VGG16				
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.8672	0.8672	0.8863	0.8796
H ¹ y Normalización	0.8838	0.8838	0.8863	0.8931
E ¹ y Normalización	0.8713	0.8713	0.8836	0.8803
H, E ¹ y Normalización	0.8672	0.8672	0.8484	0.8750
H, E ¹ sin Normalización	0.8796	0.8796	0.9015	0.8913

¹ H: Inversión Horizontal, E: Ecuilizado.

Tabla 6. Resultados ResNet-50 (dataset original).

ResNet-50				
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.8423	0.8423	0.8257	0.8515
H ¹ y Normalización	0.8755	0.8755	0.8863	0.8863
E ¹ y Normalización	0.8174	0.8174	0.8787	0.8405
H, E ¹ y Normalización	0.8340	0.8340	0.7954	0.8400
H, E ¹ sin Normalización	0.8340	0.8340	0.8636	0.8507

¹ H: Inversión Horizontal, E: Ecuilizado.

Tabla 7. Resultados InceptionV3 (dataset original).

InceptionV3				
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.8796	0.8796	0.9090	0.8921
H ¹ y Normalización	0.8838	0.8838	0.9015	0.8947
E ¹ y Normalización	0.8298	0.8298	0.9166	0.8551
H, E ¹ y Normalización	0.8506	0.8506	0.9015	0.8686
H, E ¹ sin Normalización	0.9004	0.9004	0.9242	0.9104

¹ H: Inversión Horizontal, E: Ecuilizado.

- **ResNet-50:** los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son **learning rate = 0.0001** y **optimizador = Adam**.

Resultados obtenidos con Data Augmentation

Tras determinar los hiperparámetros mediante Grid Search, se sometieron los modelos a diversas estrategias de Data Augmentation previamente descritas en la sección 4.3, con el objetivo de discernir si alguna de estas técnicas podía potenciar los resultados obtenidos. Las tablas 5, 6 y 7 muestran los desempeños al aplicar Data Augmentation en los tres modelos empleados para abordar este problema.

En la Tabla 5 se muestra que la VGG16 alcanzó sus mejores resultados cuando se implementó Data Augmentation de Inversión Horizontal. En cuanto al modelo ResNet-50, mostrado en la Tabla 6, también este alcanzó su mejor desempeño al aplicar Data Augmentation mediante Inversión Horizontal.

Para el modelo InceptionV3, cuyos resultados se muestran en la Tabla 7, se observa que el modelo alcanzó su mejor rendimiento cuando se aplicaron técnicas de Inversión Horizontal, Ecuilización y omitiendo la Normalización de la imagen.

5.3. Experimentos realizados con el dataset con segmentación automática de pulmón

Al realizar el Grid Search en los 3 modelos de redes neuronales convolucionales usando el banco de datos con imágenes segmentadas, se determinaron los mejores hiperparámetros para cada arquitectura.

Hiperparámetros obtenidos con Grid Search

- **VGG16:** los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son **learning rate = 0.00001** y **optimizador = Adam**.
- **ResNet-50:** los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son **learning rate = 0.0001** y **optimizador = Adam**.
- **InceptionV3:** los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son **learning rate = 0.0001** y **optimizador Adam**.

Resultados obtenidos con Data Augmentation sobre el dataset con segmentación

Tras determinar los hiperparámetros mediante Grid Search, se sometieron los modelos a diversas estrategias de Data Augmentation previamente descritas en la sección 4.3, con el objetivo de discernir si alguna de estas técnicas podía potenciar los resultados obtenidos.

Las tablas 8, 9 y 10 muestran los desempeños al aplicar Data Augmentation en los tres modelos utilizando el dataset de imágenes con segmentación.

En la Tabla 8 se observa que VGG16 obtuvo los mejores resultados sin utilizar ninguna técnica de data augmentation. El modelo ResNet-50 mostrado en la Tabla 9 alcanzó su mejor desempeño al aplicar Data Augmentation mediante Ecuilización y Normalización.

Por otro lado, en la Tabla 10, se observa que el modelo InceptionV3 alcanzó mejores resultados utilizando Data Augmentation con Inversión Horizontal y Normalización.

Tabla 8. Resultados VGG16 (dataset con segmentación).

	VGG16 con Segmentación			
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.8962	0.8962	0.8863	0.9034
H ¹ y Normalización	0.8755	0.8755	0.8484	0.8818
E ¹ y Normalización	0.8423	0.8423	0.8560	0.8560
H, E ¹ y Normalización	0.8464	0.8464	0.8181	0.8537
H, E ¹ sin Normalización	0.8464	0.8464	0.8257	0.8549

¹ H: Inversión Horizontal, E: Ecuilizado.

Tabla 9. Resultados ResNet-50 (dataset con segmentación).

	ResNet-50 con Segmentación			
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.7634	0.7634	0.7196	0.7692
H ¹ y Normalización	0.8174	0.8174	0.7500	0.8181
E ¹ y Normalización	0.8464	0.8464	0.8787	0.8624
H, E ¹ y Normalización	0.6473	0.6473	0.4090	0.5595
H, E ¹ sin Normalización	0.8132	0.8132	0.9015	0.8409

¹ H: Inversión Horizontal, E: Ecuilizado.

Tabla 10. Resultados InceptionV3 (dataset con segmentación).

	InceptionV3 con Segmentación			
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.8091	0.8091	0.7651	0.8145
H ¹ y Normalización	0.8796	0.8796	0.8863	0.8897
E ¹ y Normalización	0.8257	0.8257	0.8787	0.8467
H, E ¹ y Normalización	0.8091	0.8091	0.8787	0.8345
H, E ¹ sin Normalización	0.8381	0.8381	0.8560	0.8528

¹ H: Inversión Horizontal, E: Ecuilizado.

5.4. Experimentos con dataset con aumento de brillo

Al realizar el Grid Search en los 3 modelos de redes neuronales convolucionales usando el banco de datos con ajuste de brillo en las imágenes, se determinaron los hiperparámetros para cada arquitectura.

Hiperparámetros obtenidos con Grid Search

- **VGG16:** los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son **learning rate = 0.00001** y **optimizador = Adam**.

Tabla 11. Resultados VGG16 (dataset con ajuste de brillo).

	VGG16 con ajuste de brillo			
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.8796	0.8796	0.8257	0.8825
H ¹ y Normalización	0.8879	0.8879	0.8636	0.8941
H ¹ sin Normalización	0.8630	0.8630	0.8484	0.8715

¹ H: Inversión Horizontal.**Tabla 12.** Resultados ResNet-50 (dataset con ajuste de brillo).

	ResNet-50 con ajuste de brillo			
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.7842	0.7842	0.7803	0.7984
H ¹ y Normalización	0.7551	0.7551	0.9318	0.8065
H ¹ sin Normalización	0.8796	0.8796	0.9015	0.8913

¹ H: Inversión Horizontal**Tabla 13.** Resultados InceptionV3 (dataset con ajuste de brillo).

	InceptionV3 con ajuste de brillo			
	Accuracy	Precision	Recall	F1-Score
Sin Data Augmentation	0.7676	0.7676	0.8636	0.8028
H ¹ y Normalización	0.8174	0.8174	0.8030	0.8281
H ¹ sin Normalización	0.8796	0.8796	0.9015	0.8913

¹ H: Inversión Horizontal

- **ResNet-50:** los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son **learning rate = 0.0001** y **optimizador = Adam**.
- **InceptionV3:** los mejores hiperparámetros para esta arquitectura dentro del espacio de búsqueda descrito en la sección 4.2 son **learning rate = 0.001** y **optimizador = SGD**.

Resultados obtenidos con data augmentation sobre el conjunto con aumento de brillo

Tras determinar los hiperparámetros mediante Grid Search, se sometieron los modelos a diversas estrategias de Data Augmentation previamente descritas en la sección 4.3, con el objetivo de discernir si alguna de estas técnicas podía potenciar los resultados obtenidos.

Las tablas 11, 12 y 13 muestran los desempeños al aplicar Data Augmentation de los tres modelos utilizando el dataset de imágenes con ajuste de brillo. En la Tabla 11 se observa que VGG16 obtuvo los mejores resultados utilizando Data Augmentation con inversión horizontal. El modelo ResNet-50 mostrado en la Tabla 12 alcanzó su mejor desempeño al aplicar Data Augmentation mediante la Inversión Horizontal, Ecuilización y omitiendo la Normalización de la imagen.

Tabla 14. Comparación de resultados para VGG16 y sus variantes.

Modelo	Métrica	Resultado promedio
VGG16	Accuracy	0.8970
	Precision	0.9141
	Recall	0.8969
	F1-Score	0.9046
VGG16 (segmentación)	Accuracy	0.8753
	Precision	0.8976
	Recall	0.8727
	F1-Score	0.8844
VGG16 (ajuste de brillo)	Accuracy	0.8920
	Precision	0.9085
	Recall	0.8939
	F1-Score	0.9011

En la Tabla 13, se observa que el modelo InceptionV3 alcanzó mejores resultados utilizando Data Augmentation mediante la Inversión Horizontal, Ecuilización y omitiendo la Normalización de la imagen.

5.5. Resultados finales de los modelos

Las tablas 14, 15 y 16 nos presentan los mejores desempeños de cada modelo para cada dataset (original, con segmentación y ajuste de brillo) con la mejor técnica de data augmentation. Estos resultados se obtuvieron al promediar las métricas de los 5 ejecuciones sobre diferentes particiones de cada banco de datos usando el método de 5-folds cross-validation.

5.6. Discusión de resultados

La Tabla 14 presenta una comparativa de los resultados obtenidos utilizando el modelo VGG16 y sus variantes: segmentación automática del pulmón y aumento de brillo. La VGG16 con el dataset original sirve como modelo de referencia y obtuvo un F1-score de 0.9046, mientras que el Accuracy alcanzado fue de 0.8970.

La segmentación del pulmón tiene como objetivo centrarse únicamente en las regiones de interés, eliminando posibles ruidos que podrían afectar el rendimiento del modelo. A pesar de esto, el F1-score de la VGG16 con segmentación fue 0.8844, ligeramente inferior al modelo de referencia.

El Accuracy también disminuyó a 0.8753. Esto podría sugerir que la segmentación automática, aunque útil en teoría, en este caso particular no aportó una mejora significativa. Es posible que el modelo VGG16 ya era lo suficientemente robusto para manejar el ruido en las imágenes sin segmentar. El aumento de brillo puede ayudar a resaltar características sutiles en las imágenes, lo que podría ser útil especialmente en imágenes médicas donde los detalles son cruciales.

Tabla 15. Comparación de resultados para ResNet-50 y sus variantes.

Modelo	Métrica	Resultado promedio
ResNet-50	Accuracy	0.8729
	Precision	0.8844
	Recall	0.8848
	F1-Score	0.8844
ResNet-50 (segmentación)	Accuracy	0.8089
	Precision	0.8278
	Recall	0.8394
	F1-Score	0.8244
ResNet-50 (ajuste de brillo)	Accuracy	0.8629
	Precision	0.8878
	Recall	0.8606
	F1-Score	0.8721

Esta variante, VGG16 con ajuste de brillo, mostró una mejora en comparación con la segmentación del pulmón, obteniendo un F1-Score de 0.9011 y un Accuracy de 0.8920, por lo que el aumento de brillo podría ser una técnica de preprocesamiento más efectiva para este dataset en particular que la segmentación del pulmón.

La aplicación de diferentes técnicas de preprocesamiento y la variación de la arquitectura VGG16 con batch normalization mostró diferencias en el rendimiento. Mientras que el aumento de brillo mejoró el rendimiento en comparación con la segmentación automática del pulmón, ninguna de las técnicas superó significativamente al modelo VGG16 de referencia en términos de F1-Score.

Esto destaca la robustez del modelo VGG16 y sugiere que futuras investigaciones podrían centrarse en otras técnicas de preprocesamiento o ajustes de arquitectura para mejorar aún más el rendimiento. La Tabla 15 presenta una comparativa de los resultados obtenidos utilizando el modelo ResNet-50 y sus variantes: segmentación automática del pulmón y aumento de brillo.

La ResNet-50 con el banco de datos original obtuvo un F1-Score de 0.8844 y Accuracy de 0.8729. La variante de ResNet-50 con segmentación experimentó una disminución en todas las métricas comparadas con el modelo base, el F1-Score se redujo a 0.8244 y el Accuracy a 0.8089.

Estos resultados podrían indicar que, para el modelo ResNet-50, la segmentación automática no necesariamente aporta mejoras. Podría ser que la segmentación estuviera eliminando información útil o que ResNet-50 ya tiene la capacidad de manejar el ruido o información adicional presente en las imágenes sin segmentar.

ResNet-50 con aumento de brillo alcanzó F1-Score de 0.8721 y Accuracy de 0.8629. Aunque el aumento de brillo mostró una mejora con respecto a la segmentación del pulmón, todavía son inferiores al modelo ResNet-50 base. Estos resultados sugieren

Tabla 16. Comparación de resultados para InceptionV3 y sus variantes.

Modelo	Métrica	Resultado promedio
InceptionV3	Accuracy	0.8488
	Precision	0.8616
	Recall	0.8666
	F1-Score	0.8377
InceptionV3 (segmentación)	Accuracy	0.8555
	Precision	0.8701
	Recall	0.8651
	F1-Score	0.8673
InceptionV3 (ajuste de brillo)	Accuracy	0.8787
	Precision	0.9010
	Recall	0.8757
	F1-Score	0.8877

que, aunque el aumento de brillo puede aportar ciertas ventajas, no es suficiente para superar el rendimiento del modelo base.

La comparativa entre las diferentes variantes del modelo ResNet-50 revela que el modelo estándar, sin modificaciones adicionales, ofrece el mejor rendimiento en este dataset específico. Tanto la segmentación automática del pulmón como el aumento de brillo no lograron superar las métricas del modelo con el banco de datos original.

La Tabla 16 presenta una comparativa de los resultados obtenidos utilizando el modelo InceptionV3 y sus variantes: segmentación automática del pulmón y aumento de brillo. InceptionV3 con el banco de datos original presentó un F1Score de 0.8377 y Accuracy de 0.8488. Esta misma arquitectura con segmentación obtuvo un ligero aumento en el F1-Score, alcanzando 0.8673.

Además, se observa una mejora en todas las métricas en comparación con el modelo base, lo que sugiere que la segmentación automática beneficia al modelo InceptionV3 al trabajar con este dataset. La variante con ajuste de brillo mostró mejores resultados, con F1-Score de 0.8877 y un Accuracy de 0.8787. La Precision alcanzó 0.9010, lo que indica una alta proporción de verdaderos positivos en relación con los falsos positivos.

Mientras que en los modelos previamente discutidos (VGG16 y ResNet-50) no siempre se observó una mejora con las técnicas de preprocesamiento, InceptionV3 mostró un incremento en el rendimiento con ambas variantes. De hecho, el aumento de brillo demostró ser la técnica más beneficiosa para este modelo, llevando al F1-Score más alto.

Estos hallazgos subrayan cómo diferentes arquitecturas pueden responder de manera distinta a técnicas de preprocesamiento similares, y cómo es crucial experimentar y evaluar múltiples combinaciones para encontrar la mejor solución para un dataset específico.

5.7. Discusión integrada de resultados globales

Las tablas y las discusiones individuales presentadas anteriormente ofrecen una perspectiva detallada sobre el rendimiento de tres arquitecturas prominentes de redes neuronales convolucionales: VGG16, ResNet-50 e InceptionV3. A lo largo de este análisis, nos enfocaremos en el F1-Score, ya que es una métrica que combina Precision y Recall, ofreciendo una visión más completa del rendimiento del modelo.

VGG16:

- F1-Score dataset original: 0.9046
- F1-Score dataset con segmentación: 0.8844
- F1-Score dataset con aumento de brillo: 0.9011

VGG16 en el dataset original mostró el F1-Score más alto entre sus variantes. Curiosamente, la segmentación, que suele ser beneficiosa para resaltar áreas de interés, redujo ligeramente el rendimiento. Sin embargo, el aumento de brillo mostró resultados casi idénticos al modelo estándar.

ResNet-50:

- F1-Score dataset original: 0.8844
- F1-Score dataset con segmentación: 0.8244
- F1-Score dataset con aumento de brillo: 0.8721

ResNet-50 en el dataset original no solo superó a sus variantes, sino que la segmentación automática redujo significativamente el F1-Score. Esto podría sugerir que ResNet-50, podría ya estar extrayendo características esenciales sin la necesidad de un preprocesamiento tan enfocado.

InceptionV3:

- F1-Score dataset original: 0.8377
- F1-Score dataset con segmentación: 0.8673
- F1-Score dataset con aumento de brillo: 0.8877

InceptionV3 mostró una tendencia interesante. Aunque su rendimiento en el dataset original fue el más bajo de los tres, las técnicas de procesamiento mostraron ser beneficiosas, con el aumento de brillo se obtuvo el mayor incremento en el F1-Score.

Al comparar los tres modelos, VGG16 demostró tener el F1-Score más alto en el dataset original. Sin embargo, es crucial notar cómo diferentes modelos responden a técnicas de procesamiento sobre el dataset. Mientras que VGG16 y ResNet-50 mostraron un rendimiento reducido con la segmentación, InceptionV3 se benefició de ella. Esto subraya la importancia de considerar el dataset, la arquitectura y las técnicas de preprocesamiento como un sistema integrado.

6. Conclusiones

Las arquitecturas de redes neuronales convolucionales, específicamente VGG16, ResNet-50 e InceptionV3, han demostrado ser herramientas potentes para la clasificación de imágenes de tomografía computarizada. Cada modelo tiene sus propias

fortalezas y debilidades, y su eficacia puede variar dependiendo del tipo de procesamiento aplicado sobre el dataset.

VGG16 presentó el F1-Score más alto en el dataset original, lo que sugiere que esta arquitectura es robusta y capaz de capturar características esenciales sin la necesidad de procesamiento especializado de las imágenes.

ResNet-50 aunque tuvo un buen rendimiento, mostró una mayor sensibilidad a la segmentación en comparación con las otras arquitecturas, lo que indica que las conexiones residuales podrían estar optimizadas para trabajar con la información original sin modificaciones sustanciales.

InceptionV3 a pesar de tener el rendimiento más bajo en el dataset original, mostró una notable mejora con técnicas de procesamiento, subrayando su flexibilidad y adaptabilidad.

Tras un análisis exhaustivo de los resultados, se puede inferir que las diferentes arquitecturas de redes neuronales convolucionales tienen peculiaridades que pueden influir en su rendimiento en función de la naturaleza del dataset. La VGG16, por ejemplo, es una arquitectura más compacta comparada con InceptionV3. Esta característica podría hacerla más apta cuando se requiere un costo computacional más bajo.

Por otro lado, arquitecturas como ResNet-50 e InceptionV3, al ser más profundas y complejas, podrían ser más adecuadas para conjuntos de datos con variaciones más amplias y características más intrincadas. Sin embargo, en nuestro dataset, que consta de imágenes TC de pacientes que egresaron y aquellos que fallecieron por neumonía de COVID-19, la similitud entre las imágenes podría haber desafiado la capacidad discriminativa de estas redes más complejas.

Estas observaciones indican que no existe una solución única para la clasificación de imágenes TC. La elección del modelo y las técnicas de procesamiento deben adaptarse a los datos y el contexto clínico específico. A partir de los hallazgos de nuestra investigación, queda en evidencia el potencial de las redes neuronales convolucionales como herramientas de apoyo a los sistemas de salud.

Los resultados conseguidos no alcanzan una cifra superior al 90% en términos de Accuracy y F1-Score, estos valores subrayan la necesidad de continuar con experimentaciones variadas con el objetivo de reducir el margen de error. Es esencial ampliar los métodos y técnicas implementados para lograr una mayor precisión en los diagnósticos y potenciar aún más la contribución de la inteligencia artificial en el ámbito médico.

7. Trabajo a futuro

Mencionamos las siguientes direcciones de trabajo futuro.

- **Técnicas de procesamiento digital:** aunque se exploraron la segmentación y el ajuste de brillo, existen muchas otras técnicas de procesamiento, como la ampliación, la rotación y morfología matemática, que podrían mejorar el rendimiento de la clasificación.

- **Modelos más recientes:** explorar arquitecturas otras arquitectura, como EfficientNet o Vision Transformers, que podrían ofrecer mejoras en términos de eficacia y rendimiento.
- **Redes neuronales generativas (GANs):** se puede considerar un método para **ampliar** el conjunto de datos utilizando redes neuronales generativas, como las GANs (Generative Adversarial Networks). Estas redes son capaces de crear imágenes sintéticas que, aunque no son reales, comparten características similares con las imágenes originales del conjunto de datos. Implementando GANs, podríamos generar un mayor volumen de imágenes, enriqueciendo así el dataset y posiblemente mejorando el desempeño de los modelos en entrenamientos futuros.

Además, en consideración de aspectos futuros, también se está considerando el uso de imágenes de 16 bits por las razones que se mencionan a continuación:

- **Mejor resolución de detalles:** dada la alta precisión de las imágenes a 16 bits, sería posible identificar y clasificar estructuras y patologías en la imagen con un nivel de detalle anteriormente inaccesible.
- **Preprocesamiento avanzado:** las imágenes a 16 bits permiten técnicas de preprocesamiento más sofisticadas, aprovechando el mayor rango dinámico para, por ejemplo, realzar características sutiles o correcciones de imagen.
- **Redes neuronales profundas especializadas:** diseñar y entrenar redes neuronales específicamente optimizadas para trabajar con la profundidad de 16 bits, teniendo en cuenta la naturaleza de los datos.

Considerando que las imágenes convencionales utilizan una profundidad de 8 bits por canal, lo que ofrece de 0 a 255 niveles de tonalidad, se propone explorar el uso de imágenes a 16 bits por canal, dado que una imagen a 16 bits tiene 65,536 niveles de tonalidad (desde 0 hasta 65,535). La utilización de imágenes con mayor profundidad podría potenciar la capacidad de identificación y clasificación de modelos de Deep learning en contextos donde la precisión es esencial.

Referencias

1. Romero, L.: Neumonía, novena causa de mortalidad en México (2019) <https://www.gaceta.unam.mx/neumonia-novenacausa-de-mortalidad-en-mexico/>
2. Chamorro, E. M., Tascón, A. D., Sanz, L. I., Vélez, S. O., Nacenta, S. B. Diagnóstico radiológico del paciente con COVID-19. Radiología, vol. 63, no. 1, pp. 56–73 (2021). doi: 10.1016/j.rx.2020.11.001
3. ¿Cuáles son los beneficios de las exploraciones por TC? https://www.radiologyinfo.org/es/info/safety-hiw_04 (2022)
4. Luján-García, J. E., Yáñez-Márquez, C., Villuendas-Rey, Y., Camacho-Nieto, O.: A transfer learning method for pneumonia classification and visualization. Applied Sciences, vol. 10, no. 8, pp. 2908 (2020) doi: 10.3390/app10082908

5. Mouhafid, M., Salah, M., Yue, C., Xia, K.: Deep ensemble learning-based models for diagnosis of Covid-19 from chest CT images. *Healthcare*, vol. 10, no. 1, p. 166 (2022) doi: 10.3390/healthcare10010166
6. Iqbal, S., Qureshi, A. N., Li, J., Mahmood, T.: On the analyses of medical images using traditional machine learning techniques and convolutional neural networks. *Archives of Computational Methods in Engineering*, vol. 30, no. 5, pp. 3173–3233 (2023) doi: 10.1007/s11831-023-09899-9
7. Mahmoudi, R., Benameur, N., Mabrouk, R., Mohammed, M. A., Garcia-Zapirain, B., Bedoui, M. H.: A deep learning-based diagnosis system for COVID-19 detection and pneumonia screening using CT imaging. *Applied Sciences*, vol. 12, no. 10, pp. 4825 (2022) doi: 10.3390/app12104825
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pp. 770–778 (2016)
10. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pp. 2818–2826 (2016)