



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM ATLACOMULCO

**“Análisis comparativo de modelos de redes neuronales
convolucionales para la detección de cáncer de pulmón”**

T E S I S

Que para obtener el Título de:

Ingeniero en Computación

Presenta:

Alexander Garduño Morales

Director de Tesis:

Dr. Everardo Efrén Granda Gutiérrez

Atlacomulco, México; Agosto del 2025.

RESUMEN

En este trabajo se realiza un análisis comparativo de tres modelos de redes neuronales convolucionales (CNN) para la detección de cáncer de pulmón, con base en imágenes de tomografías computarizadas, con el fin de identificar la arquitectura que ofrece el mejor desempeño bajo condiciones controladas. Los modelos de aprendizaje profundo considerados para este estudio son: *DenseNet201*, *InceptionV3* y *ResNet50*, que fueron validados mediante un esquema de subconjunto estratificado.

El estudio incluye un preprocesamiento de las imágenes médicas y la aplicación de técnicas de aumento de datos, destacando particularmente *Grid Distortion*, *CLAHE* y *Gauss Noise*, para simular variaciones que imiten posibles situaciones clínicas y técnicas.

Como parte principal del estudio, se realiza una comparación entre dos escenarios: un conjunto de imágenes balanceado mediante técnicas propias (Caso A) y un conjunto previamente aumentado por un tercero (Caso B), lo que permite evaluar el impacto del preprocesamiento en el rendimiento final de los tres modelos CNN.

Los resultados muestran que la arquitectura *ResNet50* alcanza la mejor combinación de precisión (99.82%), sensibilidad (99.81%) y eficiencia computacional (15.91 segundos de clasificación), superando a *InceptionV3* y *DenseNet201*. Además, el modelo demuestra una correcta capacidad de generalización, con una puntuación kappa de Cohen del 99.72%, lo que refleja una buena coincidencia entre las predicciones y las etiquetas reales del conjunto de datos de prueba.

La validez del estudio se asegura mediante validación cruzada estratificada, conjuntos de prueba independientes y comparación con el estado del arte. La aportación se manifiesta en la generación de un marco reutilizable para el desarrollo de sistemas de apoyo al diagnóstico médico. Además, el uso de transformaciones complejas va más allá del aumento básico, mejorando la robustez de los modelos evaluados.

Palabras clave: Redes neuronales convolucionales, cáncer de pulmón, inteligencia artificial, diagnóstico por imágenes.

ABSTRACT

In this work, a comparative analysis of three convolutional neural network (CNN) models for lung cancer detection is conducted using computed tomography (CT) images, aiming to identify the architecture that delivers the best performance under controlled conditions. The deep learning models evaluated in this study are *DenseNet201*, *InceptionV3*, and *ResNet50*, which were validated using a stratified subsampling scheme.

The study includes medical image preprocessing and the application of data augmentation techniques, specifically *Grid Distortion*, *CLAHE*, and *Gaussian Noise*, designed to simulate variations mimicking potential clinical and technical scenarios.

As a central part of the study, a comparison is carried out between two scenarios: a dataset balanced using custom techniques (Case A) and another previously augmented by a third party (Case B), allowing assessment of the impact of preprocessing on the final performance of the three CNN models.

Results show that the *ResNet50* architecture achieves the best combination of accuracy (99.82%), sensitivity (99.81%), and computational efficiency (15.91 seconds for classification), outperforming both *InceptionV3* and *DenseNet201*. Furthermore, the model demonstrates strong generalization capability, with a Cohen's kappa score of 99.72%, indicating excellent agreement between predictions and true labels in the test dataset.

The validity of the study is ensured through stratified cross-validation, independent test sets, and comparison with the state of the art. The contribution lies in the development of a reusable framework for building medical diagnostic support systems. Additionally, the use of complex transformations goes beyond basic augmentation, enhancing the robustness of the evaluated models.

Keywords: Convolutional neural networks, lung cancer, artificial intelligence, medical imaging.

ÍNDICE

RESUMEN.....	i
ABSTRACT	ii
ÍNDICE	iii
ÍNDICE DE TABLAS	vi
ÍNDICE DE FIGURAS.....	vii
1 INTRODUCCIÓN	1
2 PLANTEAMIENTO DEL PROBLEMA	3
2.1 Definición del problema	3
2.2 Objetivos de investigación	4
2.3 Preguntas de investigación	5
2.4 Hipótesis	5
2.5 Justificación	6
2.6 Impactos	6
2.6.1 Aportación científica.....	6
2.6.2 Aportación tecnológica	7
3 ESTADO DEL ARTE	8
3.1 Cáncer de Pulmón	8
3.2 Inteligencia computacional.....	10
3.3 Inteligencia Artificial	10
3.3.1 Flujo de trabajo de la Inteligencia Artificial	11
3.3.2 Aprendizaje automático	14
3.3.3 Aprendizaje profundo.....	17
3.4 Redes Neuronales Convolucionales	18

3.4.1	Métricas de rendimiento.....	21
3.5	Procesamiento de imágenes.....	22
3.6	Trabajos previos en salud: análisis de imágenes médicas.....	25
3.7	Conclusiones sobre el estudio del estado del arte	29
4	METODOLOGÍA.....	32
4.1	Requerimientos del proyecto.....	32
4.1.1	Hardware	33
4.1.2	Software	33
4.1.3	Datos y recursos	34
4.1.4	Flujo de trabajo	38
4.2	Balanceo de datos del caso A.....	42
4.2.1	Procesamiento de datos	43
4.3	Experimentación.....	44
4.3.1	Entrenamiento de <i>InceptionV3</i>	45
4.3.2	Entrenamiento de <i>ResNet50</i>	48
4.3.3	Entrenamiento de <i>DenseNet201</i>	49
5	RESULTADOS Y DISCUSIÓN	51
5.1	Técnicas de aumento de datos	51
5.2	Dataset final creado	56
5.3	Resultados en los modelos entrenados	57
5.3.1	<i>Inception V3</i>	57
5.3.2	<i>ResNet 50</i>	59
5.3.3	<i>DenseNet 201</i>	61
5.3.4	Google Net	62

5.4	Pruebas de validación	63
5.4.1	Caso A. Dataset creado	64
5.4.2	Caso B: dataset existente.....	68
5.5	Métricas finales por modelo	70
5.6	Comparación con otros trabajos	73
CONCLUSIONES		76
REFERENCIAS		78

ÍNDICE DE TABLAS

Tabla 3.1 Diferencias entre Aprendizaje Automático y Profundo (Microsoft, 2024)	18
Tabla 3.2 Ejemplos de Redes Neuronales Convolucionales (Artola Moreno, 2019)	20
Tabla 3.3 Métricas de rendimiento para la evaluación de un modelo de machine learning (Artola Moreno, 2019)	21
Tabla 3.4 Matriz de confusión, basada en (Artola Moreno, 2019)	22
Tabla 3.5. Resultados obtenidos por Erick García Espinosa y colaboradores (2024)	26
Tabla 5.1 Métricas finales por modelo.....	71
Tabla 5.2 Comparación de los resultados obtenidos con respecto a otros trabajos	74

ÍNDICE DE FIGURAS

Figura 3.1. Esquema general de un modelo de aprendizaje supervisado (Gonzalo, 2018)	15
Figura 3.2. Esquema de un modelo de clustering o agrupamiento (Gonzalo, 2018).	16
Figura 4.1. Distribución de las imágenes del dataset.	34
Figura 4.2. Estructura anatómica básica de los pulmones.	36
Figura 4.3. Imagen perteneciente a la clase 0 o benigno.	36
Figura 4.4. Imagen perteneciente a la clase 1 o Maligno.	37
Figura 4.5. Imagen perteneciente a la clase 2 o Normal.	38
Figura 4.6. Diagrama de flujo sobre el proceso seguido durante la elaboración del proyecto (Parte 1)	39
Figura 4.7. Diagrama de flujo sobre el proceso seguido durante la elaboración del proyecto (Parte 2)	40
Figura 4.8 Código para aplicar las técnicas de aumento de datos.	44
Figura 5.1 Resultado de aplicar Grid Distortion.	52
Figura 5.2 Resultado de aplicar Optical Distortion.	53
Figura 5.3 Resultado de aplicar Affine Transformations.	54
Figura 5.4 Resultado de aplicar Random Brightness.	54
Figura 5.5 Resultado de aplicar Gauss Noise.	55
Figura 5.6 Resultado de aplicar CLAHE.	55
Figura 5.7 Grafica sobre la distribución de datos del dataset creado.	56
Figura 5.8 Grafica sobre la precisión del entrenamiento por subconjunto con el modelo Inception V3.	58
Figura 5.9 Grafica sobre los valores de pérdida del mejor Subconjunto del modelo Inception V3.	58

Figura 5.10 Grafica sobre la precisión del entrenamiento por subconjunto del modelo ResNet 50	60
Figura 5.11 Grafica sobre los valores de pérdida del mejor subconjunto del modelo ResNet 50.....	60
Figura 5.12 Gráfica de la precisión del entrenamiento por subconjunto, modelo DenseNet201.....	61
Figura 5.13. Gráfica de pérdida del mejor subconjunto del modelo DenseNet201.	62
Figura 5.14 Gráfica de la precisión del entrenamiento por subconjunto del modelo GoogleNet.	63
Figura 5.15 Matriz de confusión de Inception V3 caso A	64
Figura 5.16 Gráfica sobre la complejidad temporal del modelo Inception V3	65
Figura 5.17 Matriz de confusión de ResNet 50 caso A.....	66
Figura 5.18 Gráfica sobre la complejidad temporal del modelo ResNet 50.	66
Figura 5.19 Matriz de confusión de DenseNet 201 caso A.....	67
Figura 5.20 Gráfica sobre la complejidad temporal del modelo DenseNet 201	68
Figura 5.21 Matriz de confusión de Inception V3 caso B.....	69
Figura 5.22 Matriz de confusión de ResNet 50 caso B.....	69
Figura 5.23 Matriz de confusión de DenseNet 201 caso B.....	70

1 INTRODUCCIÓN

En los últimos años, el desarrollo de tecnologías que automatizan procesos ha sido clave; un ejemplo de esto son las tecnologías de reconocimiento de patrones, las cuales se usan para solucionar problemas o facilitar procesos en distintas áreas (MathWorks, 2023), como en el caso particular de este trabajo, que es en la medicina.

Existen muchas enfermedades que son tratables en la medida de la etapa en la que se detectan. Un ejemplo claro es el cáncer de pulmón, que no solamente es una de las principales causas de muerte por cáncer a nivel mundial, sino que el tratamiento efectivo depende de cuan temprano se detecte (Cabo García, del Campo Mulet, Rubio González, Nápoles Smith, & Columbie Reguifero, 2018). Este y otros problemas similares han impulsado el desarrollo de herramientas computacionales que apoyen a diagnosticar de manera rápida y eficiente a pacientes, mediante el uso de algoritmos de inteligencia artificial.

Como primeros pasos del desarrollo del trabajo se realizó una investigación sobre las distintas técnicas para clasificar imágenes dentro de la que se destacó el uso de las redes neuronales convolucionales (CNN, por las siglas en inglés de *Convolutional Neural Networks*). Esta tecnología, en conjunto con las redes neuronales recurrentes (RNN, por las siglas en inglés de *Recurrent Neural Networks*) es usada en mayor manera para tareas de reconocimiento y procesamiento de imágenes digitales (Striker, 2024).

De la misma forma en que se encontró que esta técnica es de gran ayuda a la hora de clasificar imágenes, pero se encontraron problemas al usar este tipo de tecnología; una de las principales problemáticas es la brecha de conocimiento respecto a que modelo resulta más adecuado para cada conjunto de datos.

A partir de esta investigación inicial se concluyó en que existe una problemática concreta: la necesidad de comparar distintos modelos de CNN bajo condiciones igualadas y controladas para evaluar su capacidad diagnóstica en imágenes, particularmente aplicadas a la detección de cáncer.

El cáncer de pulmón se destaca por ser la principal causa de muerte por cáncer a nivel mundial, y especialmente presenta una elevada tasa de mortalidad asociada a diagnósticos tardíos. En este sentido, el uso de técnicas de inteligencia artificial, particularmente las CNN, presenta un potencial significativo para el análisis automatizado de imágenes médicas, como las tomografías computarizadas (TC), lo que potencialmente puede servir para mejorar la precisión y oportunidad en el diagnóstico.

Para esto, en este documento se presentan los resultados de un análisis comparativo de diversos modelos de CNN, entre ellos, *InceptionV3*, *ResNet50* y *DenseNet201*, evaluando su desempeño en la clasificación de imágenes entre tres clases: benigno, maligno y normal (sano), con el fin de identificar cuál de las tres arquitecturas es más eficaz para su aplicación en esta tarea.

2 PLANTEAMIENTO DEL PROBLEMA

El rápido avance de las tecnologías basadas en inteligencia artificial ha abierto nuevas posibilidades en el campo de la medicina, y particularmente es de interés para este estudio el análisis y diagnóstico por imágenes. Dentro de las herramientas disponibles, las redes neuronales convolucionales son de las más destacadas en el aprendizaje profundo, ya que han demostrado ser altamente efectivas en la detección y clasificación de patrones visuales complejos, lo que puede ser útil en la identificación de imágenes que contengan lesiones pulmonares. Sin embargo, a pesar del creciente número de aplicaciones exitosas, persisten desafíos importantes relacionados con la selección adecuada del modelo a utilizar, así como con las estrategias de preprocesamiento y aumento de datos que maximizan su rendimiento.

2.1 Definición del problema

El cáncer de pulmón es el tipo de cáncer con mayor frecuencia en ambos sexos en México (Cabo García, del Campo Mulet, Rubio González, Nápoles Smith, & Columbie Reguifero, 2018). Es por ello por lo que la detección oportuna de esta enfermedad es esencial, pues la tasa de supervivencia de los pacientes muchas veces es definida por esta detección temprana.

Para la detección de esta enfermedad, se hace uso de herramientas visuales como las tomografías (Remon, 2023); es por ello por lo que, gracias a los avances tecnológicos de los últimos años, se puede hacer uso de la visión artificial, pues esta ha demostrado ser una herramienta prometedora en el análisis de imágenes médicas.

En este orden de ideas, la inteligencia artificial actualmente es uno de los temas más impulsados; gracias a esto cada vez se van desarrollando subcampos de esta. Uno de estos es el Aprendizaje Automático (*Machine Learning*, por su origen en inglés), cuyo propósito es el de desarrollar sistemas con capacidad de aprender, es decir, que su rendimiento mejora a partir de los datos que reciben (Oracle, 2024).

De la misma forma, existen subconjuntos del aprendizaje automático que se utilizan para reconocer patrones en imágenes; uno de estos es las Redes Neuronales Convolucionales (abreviado en adelante como CNN). En consecuencia, a lo anterior, se han desarrollado múltiples modelos capaces de extraer y aprender características clave en una imagen; sin embargo, no existe un consenso claro sobre cuál de estos ofrece el mejor rendimiento, en especial cuando se usan grandes conjuntos de datos. Aunque diversas arquitecturas de redes neuronales convolucionales han demostrado un buen desempeño en tareas de clasificación de imágenes médicas, su eficacia puede variar dependiendo del tipo de lesión, la calidad de las imágenes y las condiciones bajo las que se entrenan.

En particular, existen múltiples factores que pueden influir en el rendimiento de estos sistemas, entre los cuales destacan las técnicas utilizadas para el preprocesamiento de las imágenes, como la normalización, el redimensionamiento, el realce del contraste y el balanceo de clases (Ruiz Castilla, Rangel Cortes, García Lamont, & Cervantes Canales, 2019). Estas operaciones, que típicamente se utilizan en estos sistemas, no tienen un estándar definido en la literatura y su impacto en el desempeño final de los modelos no siempre es claro, y depende de la aplicación específica en la que se vean utilizados.

Por otro lado, también es relevante identificar si ciertas arquitecturas son más robustas o adaptables a distintas condiciones de entrenamiento, lo cual podría orientar futuros desarrollos potencialmente aplicables en el ámbito médico. Por ello, este trabajo se centra específicamente en evaluar tres modelos de CNN ampliamente utilizados: *DenseNet201*, *InceptionV3* y *ResNet50*, y analizar cómo diferentes estrategias de preprocesamiento afectan sus métricas de rendimiento en la detección de cáncer de pulmón a partir de imágenes de tomografías.

2.2 Objetivos de investigación

Objetivo General: Evaluar el desempeño de distintos modelos de redes neuronales convolucionales, para la detección de cáncer de pulmón en tomografías, comparando su precisión, sensibilidad y especificidad en diferentes condiciones de entrenamiento y preprocesamiento de datos.

Objetivos Específicos

- Investigar sobre modelos de CNN que puedan ser utilizados para la clasificación de tomografías pulmonares.
- Implementar los modelos seleccionados utilizando un dataset público previamente preprocesado.
- Evaluar el impacto del aumento de datos en el rendimiento de los modelos aplicando técnicas alternativas de aumento de datos.
- Comparar los modelos en función de métricas de rendimiento como precisión, sensibilidad, especificidad y área bajo la curva de característica operativa del receptor (ROC).
- Determinar qué modelo y condición de preprocesamiento ofrecen la mejor capacidad de clasificación en la detección de cáncer de pulmón.

2.3 Preguntas de investigación

- ¿Cuál de los modelos de CNN proporciona el mejor desempeño en la detección de cáncer de pulmón en tomografías?
- ¿Cómo varía el rendimiento de los modelos según el preprocesamiento de los datos?
- ¿Cuál es el modelo más adecuado para la detección de cáncer de pulmón considerando distintos escenarios de entrenamiento?

2.4 Hipótesis

Los modelos de redes neuronales convolucionales pre entrenados, específicamente *DenseNet201*, *InceptionV3* y *ResNet50*, en conjunto con técnicas de aumento de datos, pueden obtener valores en las métricas de rendimiento como lo son precisión, sensibilidad, F1-Score y puntuación Kappa de Cohen, asegurando una precisión mayor al 90%.

2.5 Justificación

Los métodos convencionales de diagnóstico dependen en gran medida de la experiencia del radiólogo, lo que puede llevar a variabilidad en la interpretación de las imágenes. Este trabajo aporta al conocimiento sobre diferentes arquitecturas de CNN en la clasificación de imágenes médicas y en el análisis de si es que influye llevar a cabo o no un preprocesamiento de los datos. Tecnológicamente, el estudio puede servir de base para futuras implementaciones de sistemas de diagnóstico asistido por computadora.

La inteligencia artificial ha tenido un gran desarrollo en épocas recientes, y por ende se han creado diversos modelos que buscan reconocer e identificar patrones a través de las CNN. El llevar a cabo este estudio contribuye al análisis comparativo de los distintos modelos existentes, a partir de métricas de evaluación. Además, en este estudio se examina el efecto de diferentes técnicas de aumento de datos sobre la precisión del modelo, lo que puede servir como evidencia sobre la importancia del preprocesamiento en imágenes, con posibles aplicaciones médicas.

2.6 Impactos

2.6.1 Aportación científica

Este trabajo aporta al campo de la inteligencia artificial, con posibles aplicaciones en la medicina, particularmente en el diagnóstico asistido por imágenes médicas. A través de una comparación bajo condiciones controladas, se evalúan tres arquitecturas profundas de redes neuronales convolucionales: *DenseNet201*, *InceptionV3* y *ResNet50*. Estas arquitecturas, seleccionadas por tener buenos resultados en tareas de clasificación de imágenes, son adaptadas y entrenadas para la detección de cáncer de pulmón a partir de tomografías computarizadas.

Como un aspecto importante, esta investigación realiza una evaluación basada en múltiples métricas de desempeño: precisión, sensibilidad (*recall*), especificidad, puntaje F1 y área bajo la curva ROC (*Area Under the Receiver Operator Curve*, también identificada como *AUC*). Esto permite identificar el modelo con mayor exactitud global,

y con ello, analizar su capacidad para minimizar falsos negativos, que es considerado un aspecto importante en contextos médicos, donde omitir un caso positivo puede tener consecuencias graves, pues implicaría que no se dé una atención oportuna al paciente. Además, el estudio considera aspectos metodológicos, como la estratificación de los conjuntos de entrenamiento, validación y prueba, así como el uso de técnicas de validación cruzada para garantizar la reproducibilidad.

2.6.2 Aportación tecnológica

Desde una perspectiva tecnológica, esta investigación profundiza en el análisis de estrategias de preprocesamiento de imágenes y aumento de datos (conocido en inglés como *data augmentation*). Estas son técnicas para mejorar la calidad del entrenamiento de modelos de aprendizaje profundo en escenarios con limitaciones de datos. En el ámbito médico, los conjuntos de datos suelen ser escasos, desbalanceados y difíciles de obtener debido a restricciones éticas o limitaciones de privacidad de datos de los pacientes. Esta situación incrementa el riesgo de sobreajuste (*overfitting*) y limita la capacidad de generalización de los modelos.

En este trabajo, se implementan técnicas de preprocesamiento, como la normalización de intensidades, ajuste de contraste, reescalado de imágenes y corrección de sesgos de iluminación, con el fin de optimizar la calidad de entrada para las redes neuronales. También se exploran estrategias de aumento de datos, incluyendo rotaciones, traslaciones, volteos horizontales, acercamientos y distorsiones, diseñadas para aumentar artificialmente la diversidad del conjunto de entrenamiento. El impacto de estas técnicas en el rendimiento de cada modelo es analizado de forma cuantitativa, permitiendo identificar combinaciones entre arquitectura y pretratamiento de datos.

3 ESTADO DEL ARTE

La aplicación de las CNN en el ámbito médico ha crecido en los últimos años, especialmente en el análisis y diagnóstico por imágenes. Por ello, se presentan diversos estudios que han demostrado su eficacia en la detección de enfermedades como el cáncer de piel, melanomas y enfermedades respiratorias, destacando su capacidad para identificar patrones complejos a partir de datos visuales.

Sin embargo, a pesar del avance tecnológico, persisten desafíos en cuanto a la estandarización de modelos, condiciones en el preprocesamiento de datos y técnicas de aumento de datos que mejoren su desempeño. Este estado del arte presenta una revisión de trabajos recientes que abordan el uso de CNN en diagnóstico médico, con énfasis en su aplicación para la detección de cáncer pulmonar y el impacto del procesamiento previo de imágenes en los resultados obtenidos.

3.1 Cáncer de Pulmón

En los últimos años, tanto en México como a nivel mundial, el cáncer de pulmón ha tomado una relevancia importante, como se menciona en el artículo “*Aspectos clínicos y epidemiológicos en pacientes con cáncer de pulmón en un servicio de neumología*” (Cabo García, del Campo Mulet, Rubio González, Nápoles Smith, & Columbie Reguifero, 2018). Este tipo de cáncer es una de las enfermedades de origen respiratorio con mayor mortalidad; se estima que a nivel mundial son diagnosticados 1,040,000 nuevos casos al año. Además, se ha observado que esta afección predomina en hombres de entre 55 y 65 años, mientras que para las mujeres se ha presentado un aumento considerable relacionado con el hábito de fumar.

Según datos reportados por el Instituto Nacional de Salud Pública (INSP), se registraron 7 mil 811 casos nuevos y 6 mil 733 muertes por cáncer de pulmón en México. Se hace énfasis en que las personas que desarrollan esta enfermedad no acuden a los servicios médicos sino hasta que se encuentra ya en un estado muy avanzado del padecimiento. En estadios tempranos, este tipo de tumor solo se diagnostica en el 0.6 %

de los casos, mientras que el 24 % se detecta en una etapa avanzada y cerca del 65 % en un periodo crítico (Instituto Nacional de Salud Pública, 2021).

Existen múltiples formas de detectar este tipo de cáncer; de acuerdo con el informe “*Cáncer de pulmón*” redactado por el Dr. Jordi Remon (Remon, 2023), algunas de las formas de diagnóstico principales son:

- Una historia clínica con la exploración física.
- TAC (tomografía axial computarizada) o escáner torácico y abdominal.
- TAC o resonancia magnética cerebral.
- Tomografía por emisión de positrones.
- Biopsia del tumor o de un ganglio.
- Broncoscopia.
- Mediastinoscopia.
- Punción transtorácica.
- Toracentesis.
- Gammagrafía ósea.
- Espirometría.
- Radiografía de tórax.

De todas estas formas mencionadas, este estudio se centra particularmente en las tomografías axiales computarizadas (TAC), ya que estas ofrecen una visión mucho más detallada de las zonas afectadas, y esto permite identificar con mayor facilidad posibles deformaciones en los pulmones.

Debido a que este tipo de cáncer es del tipo “silencioso” pues al comienzo no presenta síntomas que adviertan a la persona que lo padece la detección temprana de esta enfermedad es crucial, es por ello por lo que haciendo uso de las TAC y de tecnologías nuevas como lo son la Inteligencia Artificial o subramas importantes de esta como es en este caso la Visión Artificial se plantea la idea de comparar algoritmos clasificadores para la detección de cáncer de pulmón.

3.2 Inteligencia computacional

La inteligencia computacional (IC) es un campo interdisciplinario que agrupa un conjunto de técnicas inspiradas en mecanismos de la naturaleza y procesos cognitivos humanos, con el objetivo de resolver problemas complejos, no lineales y de difícil modelado mediante métodos convencionales. Entre las principales técnicas de IC se identifican las redes neuronales artificiales, algoritmos evolutivos, sistemas de enjambre, lógica difusa y modelos híbridos, que combinan dos o más de estos enfoques para potenciar sus capacidades (Grupo Integrado de Ingeniería, 2025).

Una característica distintiva de la IC es su capacidad de aprender de la experiencia, adaptarse a cambios en el entorno y operar, aun ante datos con incertidumbre, ruido o incompletos. Estas propiedades la diferencian de los algoritmos deterministas tradicionales y la hacen especialmente útil en aplicaciones donde la precisión absoluta no es posible, pero donde se puede mejorar el comportamiento del sistema con base en patrones (Engelbrecht, 2007).

Las redes neuronales artificiales, y en particular las CNN, son herramientas con un gran potencial en el procesamiento de datos visuales y en la resolución de problemas de clasificación y reconocimiento de patrones. Inspiradas en la estructura del cerebro humano, las CNN son capaces de identificar automáticamente características espaciales complejas en datos de imagen sin necesidad de extracción manual de atributos, lo que las hace ideales para tareas como el diagnóstico médico por imágenes, la visión artificial y la vigilancia inteligente (LeCun, Bengio, & Hinton, 2015).

3.3 Inteligencia Artificial

El conjunto de algoritmos que pertenecen a la inteligencia artificial (IA) considera diversas técnicas que imitan los procedimientos inductivos y deductivos del cerebro humano. Surgió a partir de la investigación de las neuronas humanas y su finalidad es la de representar electrónicamente el funcionamiento del cerebro (Gómez Herrera, 2025).

Actualmente el crecimiento de la inteligencia artificial es uno de los factores más importantes, pues su implementación en áreas que dependen en gran medida de la toma

rápida de decisiones es crucial, pero para esto es necesario seguir una serie de pasos en específico estos pasos van desde la extracción de datos hasta la implementación de un modelo en un entorno productivo.

Existe un gran número de subramas que derivan de la inteligencia artificial, cada una de estas se enfoca en alguno de los tres siguientes aspectos: la toma de decisiones, el aprendizaje, la percepción. A su vez, estas subramas pueden derivar aún más subramas con el fin de atender problemáticas más específicas. Dentro de las subramas más destacables se encuentran el aprendizaje automático (*Machine Learning*), aprendizaje profundo (*Deep Learning*) y la visión por computadora (*Computer Vision*). Con estas subramas se pueden desarrollar algoritmos que identifiquen patrones y con base en estos patrones sean capaces de clasificar un grupo de imágenes (Codificando Bits, 2024).

3.3.1 Flujo de trabajo de la Inteligencia Artificial

El desarrollo exitoso de sistemas basados en IA, especialmente en aplicaciones como el diagnóstico médico, requiere un enfoque estructurado y sistemático. De acuerdo con (Martin, 2022), el éxito de un proyecto de IA depende de la correcta ejecución de cuatro etapas fundamentales: 1) preparación de los datos, 2) modelización, 3) simulación y pruebas, y 4) despliegue. Este flujo de trabajo garantiza la integridad técnica del modelo, y permite su integración efectiva en entornos reales, donde la precisión, confiabilidad y seguridad son prioritarias.

A continuación, se describe cada una de estas fases, contextualizadas en el desarrollo de modelos de CNN para la detección de cáncer de pulmón a partir de imágenes de tomografía computarizada (abreviada como TC o TAC).

1. Preparación de los datos

Esta etapa es considerada de las más importantes del proceso, ya que la calidad del modelo final está directamente ligada a la calidad, cantidad y representatividad de los datos utilizados en su entrenamiento. En las imágenes médicas, los datos suelen provenir de bases de datos clínicas o repositorios públicos, los cuales contienen estudios de TC etiquetados por radiólogos expertos, por lo que este etiquetado también es trascendental.

La preparación incluye tareas como la limpieza de datos (eliminación de imágenes corruptas o duplicadas), normalización de intensidades (para homogenizar el contraste), reescalado espacial (ajuste de resolución) y etiquetado preciso, principalmente en clases, tales como benignos, malignos o sanos, como los que se utilizan en este proyecto. Además, se debe abordar el desequilibrio de clases, común en *datasets* médicos, donde los casos positivos (que padecen algún cáncer, por ejemplo) suelen ser minoritarios. Técnicas como el *oversampling*, *undersampling*, ayudan a mitigar este sesgo y mejorar el desempeño del modelo.

Como se menciona en (Martin, 2022), sin una base de datos robusta, representativa y bien anotada, incluso los modelos más sofisticados pueden fallar al generalizar a nuevos casos. Por ello, esta fase no debe subestimarse, ya que representa hasta el 70% del esfuerzo total en un proyecto de IA.

2. Modelización

Una vez que los datos han sido adecuadamente preprocesados, se procede a la construcción y al entrenamiento del modelo. Esta etapa consiste en seleccionar una arquitectura de aprendizaje automático o aprendizaje profundo capaz de extraer características relevantes de las imágenes y clasificarlas con precisión.

Particularmente en el presente trabajo se exploran arquitecturas de redes neuronales convolucionales, *DenseNet201*, *InceptionV3* y *ResNet50*, todas preentrenadas en grandes conjuntos de datos como ImageNet (basada en *transfer learning*). Esta estrategia permite aprovechar conocimientos previos sobre características visuales generales, acelerando el entrenamiento y mejorando el rendimiento en conjuntos de datos más pequeños.

La modelización también incluye la definición de la función de pérdida, el optimizador (por ejemplo, Adam o SGD), la tasa de aprendizaje y la arquitectura de la capa de clasificación. Además, se implementan técnicas de regularización como *dropout* y *batch normalization* para prevenir el sobreajuste. Como sugiere Martin (2022), es recomendable evaluar múltiples modelos y comparar sus métricas de desempeño para seleccionar el más adecuado, en lugar de depender de una única arquitectura.

3. Simulación y pruebas

Antes de cualquier despliegue en un entorno o escenario real, es necesario validar el modelo en entornos controlados. Esta fase implica evaluar el desempeño utilizando un conjunto de pruebas independiente, nunca visto durante el entrenamiento, y aplicar métricas como precisión, sensibilidad, especificidad, F1-score y área bajo la curva ROC (AUC), de tal modo que se cuantifica la exactitud de los modelos entrenados previamente.

Además, se realizan pruebas de robustez ante variaciones en la calidad de las imágenes, presencia de ruido, o diferentes protocolos de adquisición. También se pueden simular escenarios reales mediante estudios de caso o integración con sistemas de información hospitalaria. Esto último no forma parte del alcance de este trabajo.

Según Martin (2022), la confianza en un sistema de IA solo se alcanza tras una validación exhaustiva, que incluya tanto evaluaciones cuantitativas como cualitativas (por ejemplo, revisión por expertos médicos). Este paso es clave para garantizar que el modelo no solo funcione en el laboratorio, sino que también sea seguro y eficaz en la práctica clínica. Dado que este estudio es exploratorio, tampoco se contempla esta retroalimentación, aunque debe identificarse como un posible trabajo a futuro.

4. Despliegue

La última etapa consiste en integrar el modelo entrenado en un entorno operativo, ya sea como parte de un sistema de apoyo a la decisión clínica, una aplicación web, un plugin para estaciones de trabajo de radiología o en dispositivos de borde (*edge computing*) para procesamiento en tiempo real. Esta fase requiere la selección del hardware adecuado, la optimización del modelo para reducir su tamaño y latencia, y su implementación en lenguajes o *frameworks* compatibles, como Python con TensorFlow o PyTorch, o mediante contenedores (*Docker*) para facilitar la escalabilidad.

El despliegue también debe considerar aspectos de seguridad de datos, cumplimiento de normativas, trazabilidad del modelo y actualización continua basada en retroalimentación clínica. Como señala Martin (2022), el desarrollo no termina con la implementación, sino que inicia un ciclo de monitoreo y mejora continua, esencial para mantener la confianza y utilidad del sistema a largo plazo.

3.3.2 Aprendizaje automático

Aunque están relacionados, aprendizaje automático e inteligencia artificial no son el mismo concepto. Algunas veces se confunden ambas en cuanto a que son usadas para automatizar tareas simples. Sin embargo, el aprendizaje automático es una rama específica de la IA enfocada en que las computadoras aprendan de los humanos y de la forma en que las máquinas interactúan con ellos, de una manera similar a la que los humanos lo hacen entre sí (AWS Amazon, 2025).

La IA funciona gracias a los algoritmos diseñados por desarrolladores e ingenieros informáticos. Una gran cantidad de datos se construye y reconstruye mediante estos algoritmos hasta que están listos para funcionar, de tal forma que las máquinas son capaces de ayudar en diferentes aspectos del comportamiento y actividades humanas (Mayorga, 2019).

Diseñar un algoritmo de IA requiere esencialmente la combinación de conceptos que le permitan al modelo ser autosuficiente. Ejemplo de esto son la evaluación, representación y organización. Al combinarlos, se asegura de que el algoritmo de IA sea capaz de responder al comportamiento de manera efectiva.

El aprendizaje automático emplea tres tipos de técnicas: el aprendizaje supervisado (impulsado por las tareas), el aprendizaje no supervisado (impulsado por los datos) y aprendizaje por refuerzo (que puede aprender de los errores).

A. Aprendizaje supervisado

Un modelo de aprendizaje supervisado utiliza un conjunto de datos de entrenamiento etiquetado (estructurado) para establecer que un atributo específico pertenece a una etiqueta; este conjunto de datos es conocido comúnmente como *dataset* (Google Cloud, 2025). A partir de estos datos de entrenamiento, se pretende deducir una función que genere un mapeo entre cierto número de entradas y una salida, de la mejor manera posible. El conjunto de datos de entrenamiento se conforma de tuplas (X, Y) , donde X representa a las variables que predicen una determinada salida Y (Gonzalo, 2018).

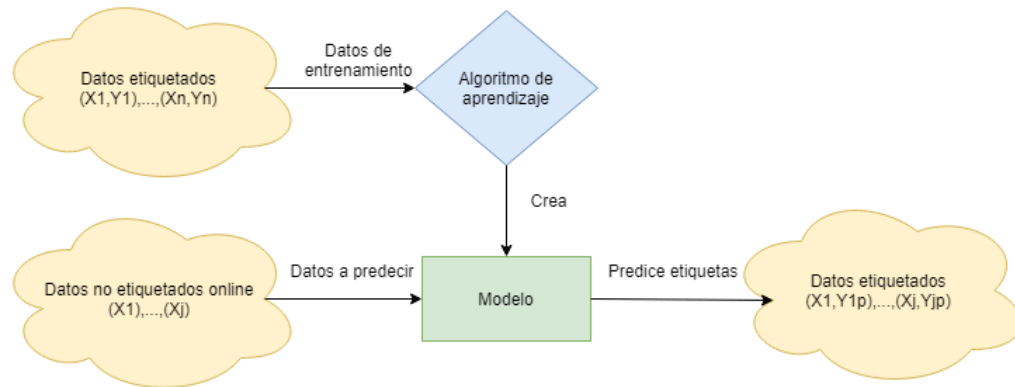


Figura 3.1. Esquema general de un modelo de aprendizaje supervisado (Gonzalo, 2018)

Los algoritmos de aprendizaje supervisado más comunes que se usan en la actualidad incluyen los siguientes (Google Cloud, 2025):

- Regresión lineal, que es un modelo que relaciona variables mediante una línea recta, asumiendo una relación lineal entre entrada y salida.
- Regresión polinómica, que se considera una extensión de la regresión lineal que ajusta una curva polinómica para capturar relaciones no lineales.
- k-vecinos más cercanos (kNN, por sus siglas en inglés), que es un algoritmo que predice basándose en la mayoría o promedio de los k datos más similares.
- Naive Bayes, que es un clasificador probabilístico que aplica el teorema de Bayes asumiendo independencia entre características.
- Árboles de decisión, modelo que divide los datos mediante reglas del tipo antecedente-consecuente (*if-else*) para predecir resultados de forma jerárquica.

B. Aprendizaje no supervisado

En el otro extremo del aprendizaje supervisado, se identifica al aprendizaje no supervisado. Esta es una variante de los algoritmos de aprendizaje automático que usa datos sin etiquetar (datos no estructurados) para reconocer patrones en estos datos. A diferencia del aprendizaje supervisado, la “precisión” de la variable de salida no se conoce de antemano, sino que se evalúa conforme se ejecuta el algoritmo (Google Cloud, 2025).

La característica principal de estas técnicas es que son capaces de aprender de los datos sin intervención humana y los categorizan en grupos, a partir de sus atributos. Aquí ya no se tienen tuplas (X, Y) , simplemente se tienen X , que sirve tanto para entender como para resumir un conjunto de datos (Gonzalo, 2018).

Los algoritmos de aprendizaje no supervisado más comunes que se usan en la actualidad incluyen los siguientes (Google Cloud, 2025):

- Agrupación en clústeres con el modelo k-medias (*k-means*), que es un algoritmo que divide los datos en k grupos mediante minimización de la distancia al centroide (ver Figura 3.2).
- Hierarchical clustering, que es un método de agrupamiento que construye una jerarquía de clústeres, ya sea de forma aglomerativa o divisiva.
- Mínimos cuadrados parciales, que es una técnica de regresión que encuentra componentes latentes maximizando la covarianza entre variables predictoras y respuesta.

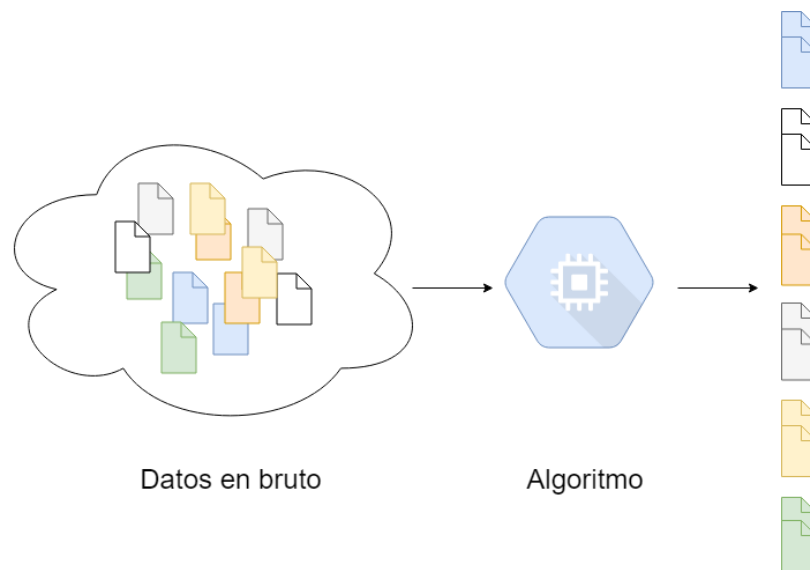


Figura 3.2. Esquema de un modelo de clustering o agrupamiento (Gonzalo, 2018).

C. Aprendizaje por refuerzo

El aprendizaje por refuerzo es un modelo de aprendizaje automático basado en el enfoque "aprende haciendo", donde un agente adquiere habilidades mediante

experimentos de prueba y error. A través de un ciclo continuo de acciones y retroalimentación, el agente mejora su desempeño: recibe refuerzos positivos al actuar correctamente y negativos ante errores, hasta alcanzar un nivel de rendimiento satisfactorio (Google Cloud, 2025).

3.3.3 Aprendizaje profundo

El aprendizaje profundo (*Deep Learning*, por su origen del inglés) es un subconjunto del aprendizaje automático, que se caracteriza por el uso de múltiples capas de redes neuronales, que son capaces de aprender a partir de grandes volúmenes de datos. A su vez, una red neuronal es un modelo funcional del cerebro humano, expresado mediante operaciones matemáticas. En cada una de las capas de la red neural profunda, los algoritmos de *Deep Learning* realizan múltiples cálculos y hacen predicciones, mejorando gradualmente la precisión del resultado durante un proceso de entrenamiento automatizado y progresivo. El *Deep Learning* se alimenta de información de múltiples fuentes de datos y la analiza en tiempo real (González Barrio, Calleja Ochoa, Gómez-Escudero, Rodríguez Ezquerro, & López de Lacalle Marcaide, 2021).

Al igual que con el aprendizaje automático, el aprendizaje profundo tiene varios tipos entre los que destacan principalmente: las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN).

Las redes neuronales profundas se construyen a partir de múltiples capas de neuronas (nodos) interconectadas, cada una de las cuales utiliza un algoritmo de aprendizaje profundo para extraer e identificar características y patrones en los datos. A partir de ello, se calcula la probabilidad de que la muestra pueda clasificarse dentro de una o más categorías o clases. Las capas de entrada y salida en una red neuronal profunda se conocen como capas visibles, siendo la capa de entrada la que recibe los datos iniciales para su procesamiento, mientras que la capa de salida genera la predicción, clasificación o resultado final del modelo (González Barrio, Calleja Ochoa, Gómez-Escudero, Rodríguez Ezquerro, & López de Lacalle Marcaide, 2021).

A manera de resumen, la Tabla 3.1 muestra algunas de las diferencias más significativas entre el aprendizaje automático y el aprendizaje profundo.

Tabla 3.1 Diferencias entre Aprendizaje Automático y Profundo (Microsoft, 2024)

	Aprendizaje Automático	Aprendizaje Profundo
Cantidad de datos	Es capaz de trabajar con cantidades de datos relativamente pequeñas para realizar predicciones sobre ellos.	Requiere de grandes cantidades de datos para realizar predicciones eficientes. Por lo tanto, la fase de entrenamiento es extensiva.
Requerimientos en hardware	Por lo general, no es indispensable contar con altos requerimientos o especificaciones demandantes en procesador o memoria.	Los algoritmos implican realizar una gran cantidad de operaciones, principalmente multiplicaciones matriciales, sobre los datos. Se hace necesario el uso de unidades de procesamiento gráfico o tensorial para ejecutar los algoritmos en un tiempo razonable.
Procesamiento de las características	Los usuarios deben crear e identificar con precisión las características en sus datos.	El aprendizaje de las características se realiza de manera abstracta, en un alto nivel de datos. Implica también la creación automática de nuevas características.
Enfoque del aprendizaje	El flujo del aprendizaje se realiza en pasos pequeños, mediante división de procesos. Los resultados de cada paso se combinan en una sola salida.	El proceso de aprendizaje se realiza de extremo a extremo, pasando progresivamente por estructuras complejas.
Tiempo de ejecución	Típicamente puede tardar desde unos segundos hasta unas pocas horas.	El proceso de aprendizaje suele requerir de mucho tiempo, dada la complejidad de las estructuras, la cantidad de operaciones y el tamaño del conjunto de datos. Dependiendo del hardware pueden ser horas o días.
Salida	Generalmente es un valor numérico, como puede ser una puntuación relativa o una clasificación de pertenencia.	Se presenta en diversos formatos, tales como texto, una puntuación numérica relativa, o incluso señales, como el sonido.

3.4 Redes Neuronales Convolucionales

Una Red Neuronal Convolutiva (abreviada por su acrónimo en inglés CNN) consiste en mapas de campos receptivos locales que responden a los estímulos únicamente en una región del campo visual, y disminuyen a medida que la corteza se mueve hacia

adelante. Estos campos receptivos se superponen de modo que cubren todo el campo visual (Sarmiento-Ramos, 2020). Su nombre indica que la red usa el operador matemático conocido como convolución; este se refiere a una operación matemática que combina dos funciones para describir la superposición entre ambas (MathWorks, 2023).

En una red de este tipo, los datos de entrada son transformados mediante diferentes filtros convolucionales, mediante el desplazamiento a través de las neuronas de diferentes capas neuronales. Cada filtro comparte los mismos parámetros en cada capa de la red neuronal, reduciendo la cantidad de parámetros del algoritmo. Su principal aplicación se encuentra en visión artificial, para resolver problemas como la clasificación o la segmentación de imágenes, entre otras (Sarmiento-Ramos, 2020).

Una CNN se compone esencialmente de una capa de entrada, varias capas ocultas intermedias y una capa de salida. Esto se puede visualizar mediante la Figura 4.4, donde cada una de las capas realiza operaciones sobre los datos, modificándolos con el propósito de extraer sus características particulares. Los tres tipos de capas más comunes en las CNN son: convolución, activación o ReLU (*Rectified Linear Unit*, o unidad lineal rectificada), y agrupación. Estas operaciones se repiten en decenas o cientos de capas; cada capa aprende a identificar diferentes características (MathWorks, 2023).

Las capas de las redes neuronales convolucionales pueden clasificarse en dos categorías: capas primarias y capas secundarias. Las capas primarias son las principales usadas por la Red Neuronal Convolucional y consiste en las capas de la convolución, capas de activación, capas de agrupación, capas planas y capas densas. Las capas secundarias son capas opcionales que pueden agregarse al modelo para que este sea más robusto contra el ajuste y aumentar la capacidad de la generalización, capas como lo son: abandono, capas de normalización por lotes y capas de regularización (Kandel & Castelli, 2020).

Existen un sinnúmero de modelos CNN desarrollados para cubrir tareas o áreas específicas, es por ello por lo que comúnmente para la resolución de problemáticas con este tipo de tecnología distintos modelos son usados como base para la creación o entrenamiento de nuevos modelos, esto se hace con el fin de solucionar una problemática

más específica cambiando únicamente las capas requeridas por cada usuario conservando así la información guardada por anteriores autores.

La Tabla 3.2 muestra ejemplos de redes neuronales convolucionales comúnmente usadas como base para el desarrollo de nuevos proyectos, así como sus características más significantes.

Tabla 3.2 Ejemplos de Redes Neuronales Convolucionales (Artola Moreno, 2019)

Modelo	No. de Capas	Tamaño de imagen de entrada	Características
<i>VGG-16</i>	16	224x224	13 capas son de convolución. 2 capas están totalmente conectadas. La capa final es una <i>softmax</i> para clasificar.
<i>VGG-19</i>	19	224x224	16 capas son de convolución. 2 capas están totalmente conectadas. La capa final es una <i>softmax</i> para clasificar.
<i>ResNet</i>	152	224x224	Su nombre es la abreviatura de Redes Residuales. Aprenden funciones más complejas, lo que mejora su rendimiento.
<i>Inception-V3</i>	42	299x299	Es una arquitectura convolucional profunda ampliamente utilizada para tareas de clasificación.
<i>AlexNet</i>	8	227x227	Las primeras 5 capas son convolucionales y las restantes están totalmente conectadas. El <i>dropout</i> , que es una técnica para reducir el sobreajuste, se aplica antes de la primera y segunda capa. Tiene 62,3 millones de parámetros.
<i>GoogleNet</i>	22	224x224	Formado por módulos de inicio que permiten cubrir un área más amplia del mapa de características, manteniendo alta resolución y un uso eficiente de la información, incluso con entradas de tamaño reducido.
<i>DenseNet 201</i>	201	224 × 224	Conexiones densas. Reutilización de características. Mejor propagación de gradientes.

3.4.1 Métricas de rendimiento

Dado que cada CNN trabaja de manera diferente, es necesario evaluar qué modelo responde mejor bajo qué parámetros, es por esto por lo que existen las métricas de rendimiento. Las métricas con las que se evalúa un modelo dependen de cada autor, pues lo que necesita cada uno es diferente; sin embargo, existen métricas que son usadas por la mayoría de los autores y las cuales solventan en gran parte la necesidad de medir el rendimiento de un modelo.

La Tabla 3.3 muestra las métricas de rendimiento comúnmente usadas para evaluar el desempeño de un modelo, así como su fórmula y una descripción sobre lo que hace o resuelve.

Tabla 3.3 Métricas de rendimiento para la evaluación de un modelo de machine learning (Artola Moreno, 2019)

Métrica	Formula	Descripción
Área bajo la curva	$\frac{Se + Sp}{2}$	Evalúa la capacidad del modelo para distinguir entre clases. Utiliza la curva ROC, que traza la tasa de verdaderos positivos (<i>recall</i>) contra la tasa de falsos positivos. <i>Se</i> es la sensibilidad mientras que <i>Sp</i> es la especificidad.
Exactitud	$\frac{TP + TN}{TP + TN + FP + FN}$	Mide la proporción de muestras clasificadas correctamente con respecto al número total de muestras.
Especificidad	$\frac{TN}{TN + FP}$	Identificada como <i>Sp</i> . Relación entre los verdaderos negativos clasificados y los verdaderos negativos reales.
Sensibilidad	$\frac{TP}{TP + FN}$	Identificada como <i>Se</i> . Relación entre los verdaderos positivos clasificados y los verdaderos positivos reales.
Precisión	$\frac{TP}{TP + FP}$	Identificada como <i>Pr</i> . Mide la proporción de casos clasificados como positivos que realmente lo son.
Puntuación F1	$\frac{Se + Pr}{2}$	Es la media armónica entre precisión (<i>Pr</i>) y sensibilidad (<i>Se</i>). Es útil cuando hay un desbalance en las clases.

Donde:

- Verdaderos positivos (abreviado del inglés como TP): casos en los que la clase real fue 1 (verdadero) y la predicción también lo es.
- Verdaderos negativos (abreviado del inglés como TN): casos en los que la clase real fue falsa (0) y el pronóstico también lo es.
- Falsos positivos (abreviado del inglés como FP): casos en los que la clase real fue 0 (falso) y el pronóstico es 1 (verdadero). Es falso porque el modelo ha hecho una predicción errónea y positiva porque la clase fue predicha como positiva.
- Falsos negativos (abreviado del inglés como FN): casos en los que la clase real fue 1 (verdadero) y la predicción es 0 (falsa). Es falso porque el modelo ha realizado una predicción incorrecta y negativa porque la clase pronosticada fue negativa.

De la misma forma, existen herramientas que ayudan a evaluar el desempeño del modelo de una manera gráfica, por ejemplo, la matriz de confusión, que no debe confundirse con una medida del rendimiento del modelo; sin embargo, la mayoría de las métricas de rendimiento parten de ella. La Tabla 3.4 es una representación de una matriz de confusión.

Tabla 3.4 Matriz de confusión, basada en (Artola Moreno, 2019)

		Resultado actual	
		Positivos (1)	Negativos (0)
Resultado Predicho	Positivos (1)	Verdadero-Positivo (TP)	Verdadero- Negativo (TN)
	Negativos (0)	Falso-Positivo (FP)	Falso-Negativo (FN)

3.5 Procesamiento de imágenes

Una imagen digital es una representación bidimensional de una imagen a partir de una matriz numérica, frecuentemente binaria. Estas imágenes, generalmente, se obtienen al convertir señales continuas en formato digital, las cuales se pueden visualizar en

diversos medios como impresoras digitales, monitores y dispositivos de proyección digital (Catalán Urzúa, 2019).

El procesamiento de imágenes digitales hace uso de distintos algoritmos para crear, manipular, modificar, comunicar y visualizar imágenes. Dichos algoritmos se pueden emplear para distintas tareas, entre las que se encuentran las siguientes (MathWorks, 2023):

- Convertir señales en imágenes digitales, según la salida de un sensor.
- Mejorar la claridad o eliminar el ruido.
- Extraer el tamaño o la escala de un objeto.
- Identificar la cantidad de objetos en una escena.
- Preparar imágenes para su visualización o impresión.
- Comprimir imágenes para su transmisión, por ejemplo, a través de una red.

Con el fin de resaltar detalles importantes de las tomografías de los pulmones de distintos casos, se identifican técnicas que brinden una mejora significativa a las imágenes, puesto que como se menciona con anterioridad el éxito de un modelo depende en gran medida del *dataset* que se tenga, así como del procesamiento que reciben las imágenes antes de pasarlas al entrenamiento del modelo.

Existen técnicas que ayudan en el aumento de datos cuando se tiene un dataset de imágenes pequeño, estas técnicas crean imágenes nuevas a partir de las que ya se tienen. Dentro de todas las técnicas de procesamiento de imágenes algunas de las más conocidas y utilizadas son las siguientes:

- Grid Distortion

Esta transformación divide la imagen en una cuadrícula y distorsiona aleatoriamente cada celda, creando efectos de deformación localizados. Es particularmente útil para el aumento de datos en tareas como el análisis de imágenes médicas, OCR y otros dominios donde las variaciones geométricas locales son significativas (Albumentations, 2025).

- Optical Distortion

Aplica una distorsión óptica a imágenes, máscaras, cuadros delimitadores y puntos clave. Admite dos modelos de distorsión (Albumentations, 2025):

- Modelo de matriz de cámara: Utiliza el modelo de calibración de cámara de OpenCV.
- Modelo de ojo de pez: Aplica una distorsión radial directa.

- Affine Transformations

Aplica transformaciones afines a las imágenes. Las transformaciones afines implican:

- Traslación ("mover" la imagen en el eje x/y).
- Rotación.
- Escalado ("acercar/alejar").
- Desplazamiento.

Todas estas transformaciones pueden crear nuevos píxeles en la imagen sin un contenido definido, por ejemplo, si la imagen se traslada a la izquierda, se crean píxeles a la derecha. Algunas transformaciones implican interpolaciones entre varios píxeles de la imagen de entrada para generar valores de píxeles de salida (Albumentations, 2025).

- Random Brightness Contrast

Cambia aleatoriamente el brillo y el contraste de la imagen de entrada. Esta transformación ajusta el brillo y el contraste de una imagen simultáneamente, lo que permite una amplia gama de variaciones de iluminación y contraste. Es particularmente útil para la ampliación de datos en tareas de visión artificial, ya que ayuda a que los modelos sean más resistentes a diferentes condiciones de iluminación (Albumentations, 2025).

- Gauss Noise

Añade ruido gaussiano a las imágenes, que puede usarse para simular procesos aleatorios naturales en la ampliación de imágenes (Albumentations, 2025).

- CLAHE

Aplica la ecualización de histograma adaptativa limitada por contraste (CLAHE) a la imagen de entrada. CLAHE es un método avanzado para mejorar el contraste de una imagen. A diferencia de la ecualización de histograma normal, que funciona en toda la imagen, CLAHE funciona en pequeñas regiones (mosaicos) de la imagen. Esto da como resultado una ecualización más equilibrada, lo que evita la sobre amplificación del contraste en áreas con un contraste inicialmente bajo (Albumentations, 2025).

- Unsharp Masking

Se usa realmente para “afilarse” una imagen, esto ayuda a enfatizar la textura y los detalles, y es crítico en el posprocesamiento de las imágenes digitales. No puede crear detalles adicionales, pero puede mejorar en gran medida la apariencia de los detalles al aumentar la pequeña escala agudeza (Cambridge in Colour, 2025).

3.6 Trabajos previos en salud: análisis de imágenes médicas

El informe titulado “*¿Sirve la inteligencia artificial para ver el cáncer de formas nuevas y más eficaces?*” (Equipo del NCI, 2022), presenta una idea de la relevancia que ha tomado la IA en los últimos años en el ámbito médico, específicamente se menciona su aplicación en la detección de cáncer. Se presenta un caso de ejemplo en el que un radiólogo con 15 años de experiencia identifica lo que posiblemente sería una zona afectada con cáncer de próstata. El mismo caso se le pasa a un modelo de IA, donde este presenta resultados bastante similares a los detallados por el especialista. En el trabajo mencionado se aclara que, si bien la IA facilita la tarea de identificar zonas de interés, aún se necesita la intervención de un radiólogo, pues se requiere la confirmación de que efectivamente la IA realizó una evaluación correcta. Por último, los especialistas explican que este campo aún sigue siendo muy nuevo y quedan muchas preguntas por resolver antes de poner en escenarios reales este tipo de tecnologías de manera eficaz.

Erick García Espinosa propone en su trabajo “*Detección de Lesiones en la Piel Usando Visión Artificial*” (Espinosa, 2024), usar redes neuronales para identificar enfermedades como lo son: cáncer de piel, sarna, sarampión, entre otras. Para lograr esto,

se hace uso de modelos como Inception V3 para extraer características de las imágenes de un directorio previamente entrenado, cambiando únicamente la salida de la penúltima capa. La propuesta de García Espinosa reduce el trabajo de crear una red neuronal desde cero a la vez que optimiza los modelos ya existentes adecuándolo para cubrir con sus objetivos. El número de imágenes usadas para el entrenamiento se encuentra entre un rango de 11,900 a 12,500. El modelo se ocupa en el entorno Orange Data Mining, y el autor menciona que en este se obtienen los mejores resultados según las métricas de rendimiento obtenidas tanto en los conjuntos de entrenamiento como en los de validación.

La Tabla 3.5 muestra los resultados obtenidos por (García Espinosa, Sergio Ruiz Castilla, & García Lamont, 2024), en donde se observa que en varios casos se logra un 100% de precisión en la fase de validación, por lo que estos resultados son muy prometedores. Sin embargo, deberá revisarse si estos son o no realistas, pues no es común que la precisión sea superior en la fase de validación que en la de entrenamiento.

Tabla 3.5. Resultados obtenidos por Erick García Espinosa y colaboradores (2024)

Enfermedad	Entrenamiento	Cantidad de imágenes para entrenamiento	Validación	Cantidad de imágenes para validación
Melanoma	99.6%	12738	100.0%	100
Sarampión	99.1%	11901	100.0%	100
MonkeyPox	98.4%	12040	97.0%	100
Sarna	97.7%	11901	99.0%	100
Herpes	97.5%	11918	100.0%	100
Varicela	97.9%	12204	99.0%	100
Lupus	97.5%	12096	100.0%	100

Trabajos desarrollados por la comunidad de expertos en Aprendizaje Automático hacen un análisis comparativo de múltiples algoritmos para resolver una misma problemática. Un ejemplo de ellos es el trabajo titulado “Comparación de algoritmos de visión artificial en la clasificación de enfermedades de la piel utilizando redes neuronales” desarrollado por Erick García Espinosa, José Sergio Ruiz Castilla y Farid García Lamont (2024). Este trabajo evalúa el desempeño de tres herramientas para la detección de

enfermedades dermatológicas: Orange Data Mining, Azure Custom Vision y una Red Neuronal Convolucional. El dataset que se utiliza en ese trabajo tiene un total de 84,794 imágenes divididas en 7 categorías donde cada una representa una enfermedad dermatológica diferente.

El desarrollo de sistemas que pre diagnostiquen diferentes tipos de enfermedades tiene una gran relevancia en la actualidad, pues aparte de que su costo es menor, la facilidad y precisión que presentan son de gran ayuda para identificar padecimientos que representen un peligro para la salud. Esto es planteado por Dulce Charin Venegas Flores en su trabajo “Sistema de Pre-Diagnóstico de Cáncer de Melanomas Usando Redes Neuronales Artificiales” (Venegas Flores, 2019). La autora compara tres tipos de Redes Neuronales diferentes: Redes Neuronales Perceptrón con metadatos, Redes Neuronales con metadatos y Redes Neuronales Convolucionales. Con base en sus resultados obtenidos, se remarca que, si bien el mejor resultado se presenta con las redes neuronales convolucionales, estas presentan una desventaja significativa con respecto a las otras dos, esta es el procesamiento que se requiere para recolectar datos, procesarlos y entrenarlos, a su vez esto se compensa con la precisión que el modelo presenta una vez que esté se encuentra completo.

En otras aproximaciones, existen investigaciones en las que en lugar de probar con modelos de CNN diferentes ocupan técnicas diferentes en el procesamiento de los datos. Un ejemplo de esto se encuentra en el trabajo “Redes Neuronales Convolucionales en la identificación de melanomas benignos y malignos” (Ruiz Castilla, Rangel Cortes, García Lamont, & Cervantes Canales, 2019). Los autores plantean aplicar tres diferentes técnicas en su dataset para comparar e identificar que configuración muestra mejores resultados en la detección de melanomas benignos y malignos. La primera técnica que presentan es añadir metadatos como lo son: edad, género y tamaño del melanoma. La segunda técnica plantea quitar la piel de las imágenes segmentando las áreas con melanomas. La tercera técnica propone un aumento del doble de datos con en base en el dataset original, en esta técnica no se aplicó algún preprocesamiento distinto. De acuerdo con los resultados obtenidos por los autores, se remarca la importancia de llevar a cabo un preprocesamiento

de los datos, por más mínimo que sea, pues esto mejora el entendimiento que tiene la CNN con respecto a los datos, mejorando notablemente la capacidad de predicción.

Mauricio Gómez Macedo presenta en su trabajo “Detección de covid-19 y neumonía en imágenes de rayos X de pulmones utilizando redes neuronales convolucionales” (Gómez Macedo, 2022), propone el desarrollo de un nuevo algoritmo enfocado a diagnosticar enfermedades pulmonares como lo son COVID-19 y neumonía. Su propuesta radica en entrenar dos modelos distintos, posteriormente realizar una comparación y finalmente encontrar cuál es más eficiente. El primer modelo entrena dos redes neuronales convolucionales binarias con el fin de clasificar las imágenes en tres diferentes clases. La primera red neuronal identifica si la imagen pertenece a una de dos clases: “enfermo” o “sano”, donde dentro de la clase “enfermo” se agruparon al mismo tiempo las imágenes de COVID-19 y neumonía. En caso de que la detección indicara que pertenece a “enfermo”, entonces se utiliza la segunda red neuronal, que tiene la función de clasificar si pertenece a COVID-19 o a neumonía. Por otro lado, el segundo modelo utiliza una red neuronal convolucional con tres salidas para identificar todas las clases.

Roger Salinas González muestra en su trabajo titulado “Aprendizaje profundo para la clasificación de imágenes de plantas de albahaca con diferencias de nitrógeno durante su cultivo, utilizando la técnica de entrenamiento curriculum-by-smoothing” (Salinas González, 2021), una forma de obtener soluciones cada vez más innovadoras y que a su vez estas aprovechen las tecnologías actuales. El autor propone el uso de diferentes modelos (*VGG16* y *ResNet50V2*) para analizar el desempeño en la clasificación de sus imágenes, y propone el uso de la técnica de entrenamiento que denominó como curriculum-by-smoothing. Como punto de comparación el autor realiza dos tipos de entrenamiento, el primero de ellos con valores comunes en el proceso de entrenamiento sin aplicar ninguna técnica extra al procesamiento de los datos muestra una deficiencia significativa al no tener suficientes datos para el proceso de entrenamiento, otro problema destacable en este primer entrenamiento es que los modelos solamente muestran buenos resultados con imágenes que contengan un fondo específico. El segundo entrenamiento implementa ahora la técnica antes mencionada, con ella presenta mejores resultados en la precisión en ambos modelos, sino que además al segmentar las imágenes sobre fondos

específicos los resultados no se ven afectados de mejor o peor manera, lo que indica que de esta manera los modelos fueron capaces de aprender las características clave de las imágenes.

Santiago Andrés Martínez Tarazona en su trabajo titulado “Prototipo de herramienta de software con aprendizaje de máquinas para la diferenciación de neumonía bacteriana o viral con el uso de radiografías de tórax” (Martínez Tarazona, 2022), propone la idea de generar un sistema especializado en clasificar pacientes con neumonía viral, bacteriana y sin neumonía, a partir de imágenes de rayos-X, y usando aprendizaje profundo. La propuesta radica en primero analizar el lenguaje de programación que se utiliza para el procedimiento, para ello el autor realizó un primer entrenamiento con el modelo *VGG16* en dos lenguajes distintos: Matlab y Python. Este entrenamiento se realizó con las mismas métricas en ambos lenguajes para garantizar una comparación equitativa; después de las pruebas se concluyó que el mejor lenguaje y por ende el cual se usaría sería Matlab. El segundo entrenamiento realizado fue ahora con tres modelos diferentes: *VGG16*, *ResNet50*, y *XCEPTION*; no obstante, algunos de los parámetros de entrenamiento cambiaron con respecto al primero, algunos de ellos fueron el *batch*, el ritmo de aprendizaje y por último el algoritmo.

3.7 Conclusiones sobre el estudio del estado del arte

El análisis crítico del estado del arte revela un creciente interés en la aplicación de redes neuronales convolucionales para el diagnóstico asistido por imágenes médicas, particularmente en la detección de enfermedades pulmonares como el cáncer. Diversos estudios han demostrado resultados prometedores al utilizar arquitecturas profundas como *InceptionV3*, *ResNet50*, *DenseNet201* y sus variantes, logrando altos niveles de precisión en la clasificación de nódulos pulmonares. No obstante, a pesar de los avances en el aspecto tecnológico y metodológico, persisten limitaciones que pueden afectar la reproducibilidad, comparabilidad y aplicabilidad de estos sistemas.

En primer lugar, se observa que muchos de los trabajos revisados se basan en *datasets* privados, pequeños o no estandarizados, lo que dificulta la validación externa de los modelos y limita su capacidad de generalización. Aunque existen repositorios

públicos, su uso frecuente sin un tratamiento homogéneo del preprocesamiento o el aumento de datos introduce sesgos que afectan la validez de las comparaciones entre modelos. Además, varios estudios reportan métricas de desempeño desproporcionadamente altas, incluso cercanas al 100%, sin proporcionar evidencia suficiente sobre la estratificación de los datos, el riesgo de sobreajuste o la independencia del conjunto de prueba.

Otro hallazgo relevante es la falta de estandarización en las estrategias de preprocesamiento y aumento de datos. Mientras que trabajos como el de Ruiz Castilla et al. (2019) destacan la importancia de técnicas como el segmentado de lesiones o la inclusión de metadatos, otros aplican transformaciones básicas (rotaciones, volteos) sin analizar su impacto específico en el rendimiento del modelo. Esto impide establecer buenas prácticas claras para el tratamiento de imágenes médicas, especialmente si los datos son escasos o desbalanceados, como es el caso del dataset utilizado en este trabajo.

Asimismo, se identifica una brecha crítica en cuanto a la comparación sistemática y controlada de múltiples arquitecturas CNN bajo las mismas condiciones. La mayoría de los estudios se centran en un solo modelo o realizan comparaciones parciales sin considerar variables clave como el tamaño del *dataset*, el tipo de aumento de datos, la normalización de píxeles o el uso de *transfer learning*. Esta limitación dificulta la toma de decisiones sobre qué arquitectura es más adecuada para una tarea específica, como la detección de cáncer de pulmón a partir de tomografías computarizadas.

Por otro lado, aunque se reconoce su potencial para apoyar el diagnóstico, también queda claro que la IA no reemplaza al especialista, sino que debe funcionar como una herramienta de apoyo. Como se menciona en el informe del (Equipo del NCI, 2022), incluso los modelos más precisos requieren la validación por parte de radiólogos.

En este orden de ideas, el presente trabajo se posiciona como una contribución oportuna, y una aplicación de la computación. En este proyecto se implementan modelos de CNN ampliamente utilizados (*DenseNet201*, *InceptionV3* y *ResNet50*), pero adicionalmente se realiza una evaluación comparativa y bajo condiciones controladas, utilizando dos escenarios: un dataset propio balanceado con técnicas de aumento de datos (caso A) y un dataset externo previamente aumentado (caso B). Además, se analiza el

impacto de transformaciones específicas (*Grid Distortion*, *CLAHE* o *Gauss Noise*), con el fin de mejorar los modelos frente a variaciones.

Finalmente, se confirma la necesidad de una metodología que combine preprocesamiento de imágenes, estrategias de aumento de datos, evaluación multivariada basada en métricas, y una comparación transparente entre arquitecturas, para elegir la más viable en la identificación de cáncer de pulmón en tomografías.

4 METODOLOGÍA

La presente investigación se enmarca en una aproximación cuantitativa y experimental, centrada en la evaluación comparativa del desempeño de diferentes arquitecturas de redes neuronales convolucionales para la detección de cáncer de pulmón a partir de imágenes de tomografía computarizada.

El diseño metodológico se estructura en torno a un flujo de trabajo que abarca la adquisición, el preprocesamiento de los datos, el entrenamiento, la validación y un análisis comparativo de tres modelos de aprendizaje profundo. Esto permite evaluar la eficacia de cada arquitectura, y al mismo tiempo, analizar el impacto de distintas estrategias de aumento de datos en la calidad del modelo final, garantizando condiciones controladas y reproducibles.

El proyecto se desarrolla en dos escenarios: uno basado en un dataset balanceado mediante técnicas de aumento de datos (Caso A), y otro utilizando un dataset previamente aumentado por un tercero (Caso B). Esta doble evaluación permite contrastar el comportamiento de los modelos frente a diferentes tratamientos del conjunto de datos, identificando así cuál combinación de arquitectura y preprocesamiento ofrece el mejor equilibrio entre precisión, sensibilidad y eficiencia computacional.

Las métricas de rendimiento se analizan mediante validación cruzada estratificada, matrices de confusión y pruebas de generalización, asegurando una evaluación integral que responde directamente a los objetivos planteados y contribuye al establecimiento de buenas prácticas en el desarrollo de sistemas de IA aplicados al diagnóstico médico.

4.1 Requerimientos del proyecto

El desarrollo de modelos de inteligencia artificial basados en redes neuronales convolucionales para la detección de cáncer de pulmón a partir de imágenes médicas exige un entorno computacional adecuado que garantice tanto el correcto procesamiento de las imágenes como un entrenamiento eficiente de los modelos.

Debido a la naturaleza intensiva en recursos del aprendizaje profundo, especialmente cuando se trabaja con *transfer learning* y grandes volúmenes de datos, es fundamental contar con una configuración de hardware y software que permita manejar adecuadamente las operaciones matriciales, el almacenamiento de modelos y la aceleración del entrenamiento mediante unidades de procesamiento gráfico (GPU). En este contexto, se establecen los requerimientos técnicos necesarios para la implementación del sistema, divididos en especificaciones de hardware, herramientas de software y el flujo general de funcionamiento del modelo.

Estos componentes aseguran la viabilidad técnica del proyecto. A continuación, se detallan las especificaciones empleadas en este trabajo, diseñadas para cumplir con los objetivos de precisión, robustez y escalabilidad en el análisis de tomografías computarizadas.

4.1.1 Hardware

El desarrollo del modelo requiere de un sistema de cómputo con los requerimientos intermedios, esto debido a que la duración del entrenamiento depende en gran medida del sistema de cómputo. El equipo de cómputo que se propone cuenta con las siguientes especificaciones: procesador AMD Ryzen 5 5500, memoria RAM: 16GB, tarjeta de video Radeon RX6600.

4.1.2 Software

Con base en los objetivos del proyecto, el lenguaje de programación seleccionado es Python debido a la gran cantidad de librerías que se pueden usar, esto facilita en gran medida el trabajo con distintos modelos de redes neuronales e imágenes. En complemento a esto, las librerías usadas para el desarrollo del proyecto son: *Numpy*, *Os*, *PyTorch*, *PIL*, y *OpenCV*.

Dado que es necesario manipular arreglos multidimensionales, *Numpy* cubre esta necesidad pues incorpora funciones para realizar las principales operaciones algebraicas con vectores y matrices. La librería *Os* permite leer datos a través de archivos en carpetas.

PyTorch es la librería más importante que se ocupó, gracias a esta fue posible probar y entrenar todas las redes neuronales propuestas, dentro de la misma librería se encuentra la subbiblioteca *TorchVision* esta ayuda específicamente en la visión por computadora.

El preprocesamiento de las imágenes utiliza a la librería *OpenCV*, esta permite implementar técnicas que permiten resaltar y eliminar detalles en las imágenes. De la misma forma permiten crear nuevas imágenes con base en un conjunto, esto con el fin de poder brindarle al modelo diferentes casos para lograr que identifique de la mejor manera.

4.1.3 Datos y recursos

El dataset *The IQ-OTH/NCCD lung cancer dataset* proporcionado por Hamdalla F. Al-Yasriy (Al-Yasriy, 2020), presenta un total de 1097 imágenes sobre el cáncer de pulmón distribuidas en 3 clases principales: Benigno, Maligno y Normal. La distribución de los datos puede ser apreciada en la Figura 4.1.

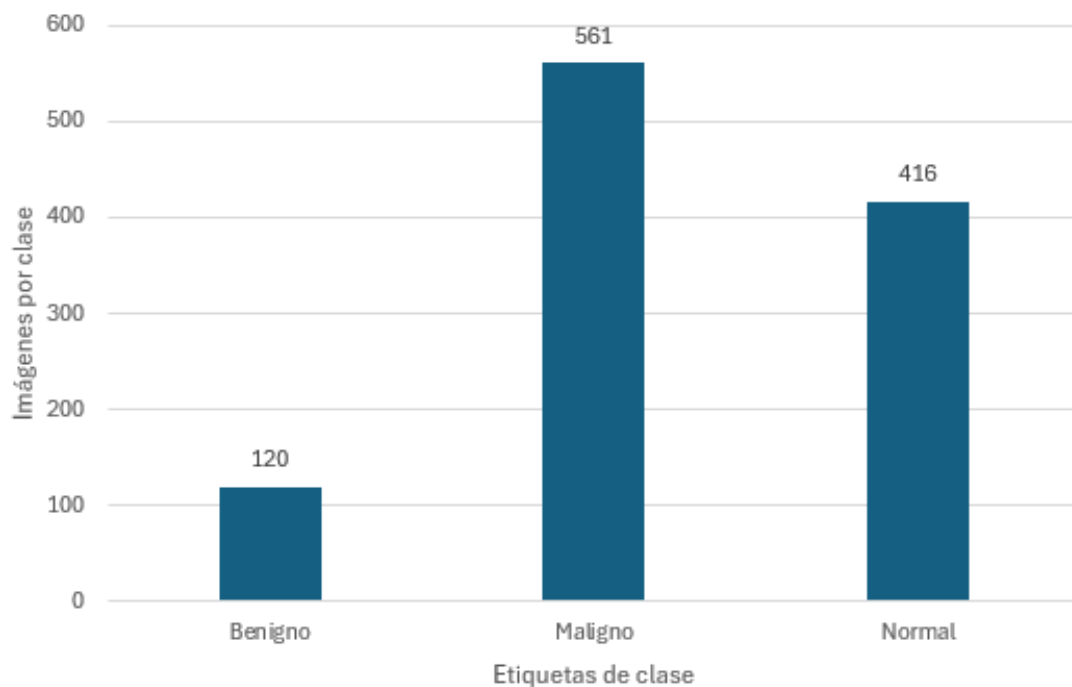


Figura 4.1. Distribución de las imágenes del dataset.

Con base en la gráfica se puede notar que existe un gran desbalanceo de datos sobre todo entre la categoría Benigno y las dos restantes. Como manera de solventar esto se proponen 2 casos diferentes:

- A. Balancear los datos con técnicas de aumento de datos.
- B. Explorar casos de trabajo en los que se usaron este dataset y extraer sus datos.

El caso A toma como base el dataset proporcionado por (Al-Yasriy, 2020), implementando las siguientes técnicas de aumento de datos para balancear los datos:

- Grid Distortion
- Optical Distortion
- Affine Transformations
- Random Brightness Contrast
- Gauss-Noise
- CLAHE
- Unsharp Masking

Antes de proceder con los casos de estudio, se analizan las imágenes del conjunto de datos con el fin de identificar posibles irregularidades que puedan afectar el entrenamiento y evaluación de los modelos. Para ello, se utiliza como referencia la estructura anatómica básica de los pulmones en la Figura 4.2, para reconocer las regiones de interés y las características morfológicas normales en las tomografías. Esta base anatómica contextualiza las alteraciones observadas y para guiar la interpretación visual de las imágenes durante el preprocesamiento y la validación de resultados.

La clase 0 (Benigno) se caracteriza por la presencia de lesiones o nódulos pulmonares no malignos, los cuales, aunque pueden presentar cambios estructurales visibles en la imagen, no exhiben comportamiento invasivo ni potencial de metástasis. Estas alteraciones suelen ser estables en el tiempo, bien delimitadas y asociadas a procesos inflamatorios, infecciosos o granulomatosos previos. Un ejemplo representativo de esta categoría se muestra en la Figura 4.3, donde se observa un nódulo con bordes definidos y sin signos de invasión tisular, consistente con un diagnóstico de naturaleza benigna.

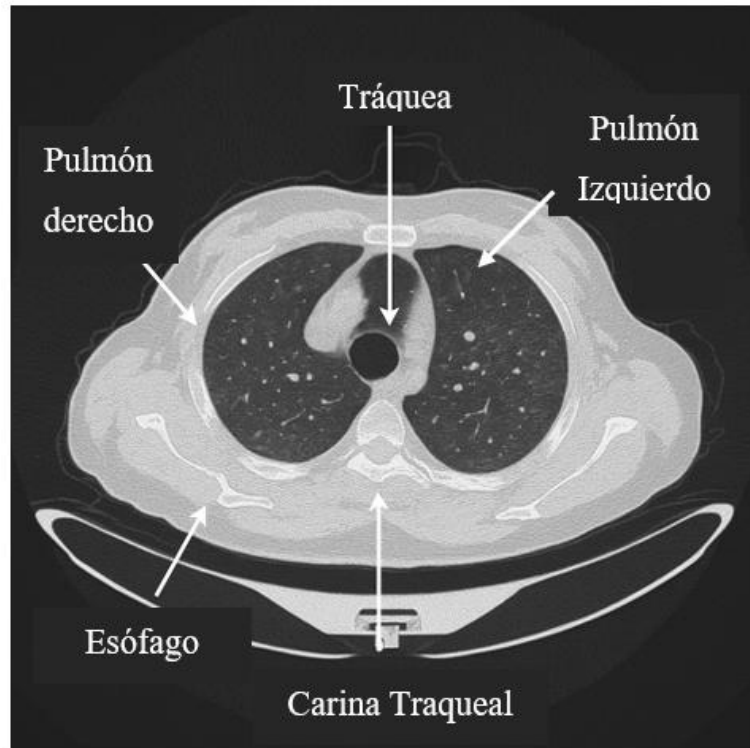


Figura 4.2. Estructura anatómica básica de los pulmones.

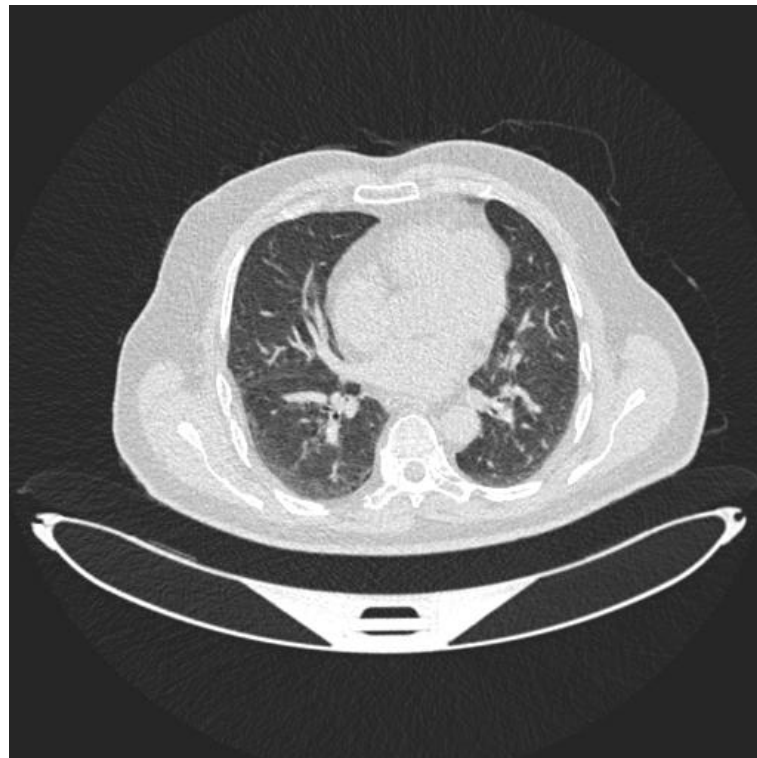


Figura 4.3. Imagen perteneciente a la clase 0 o benigno.

La clase 1 (Maligno) se caracteriza por la presencia de nódulos o masas pulmonares con características de neoplasia maligna. Estas lesiones suelen presentarse con bordes irregulares, forma asimétrica y mayor densidad en comparación con tejidos circundantes, lo que las distingue de los nódulos benignos. Además, pueden observarse signos indirectos de patología avanzada, tales como desplazamiento de estructuras mediastínicas, engrosamiento pleural, adenopatías hiliares o mediastínicas, y en algunos casos, evidencia de invasión local. Estas características visibles en las tomografías computarizadas son indicativas de un comportamiento agresivo, que representa el cáncer de pulmón. Un ejemplo representativo de esta categoría se muestra en la Figura 4.4.

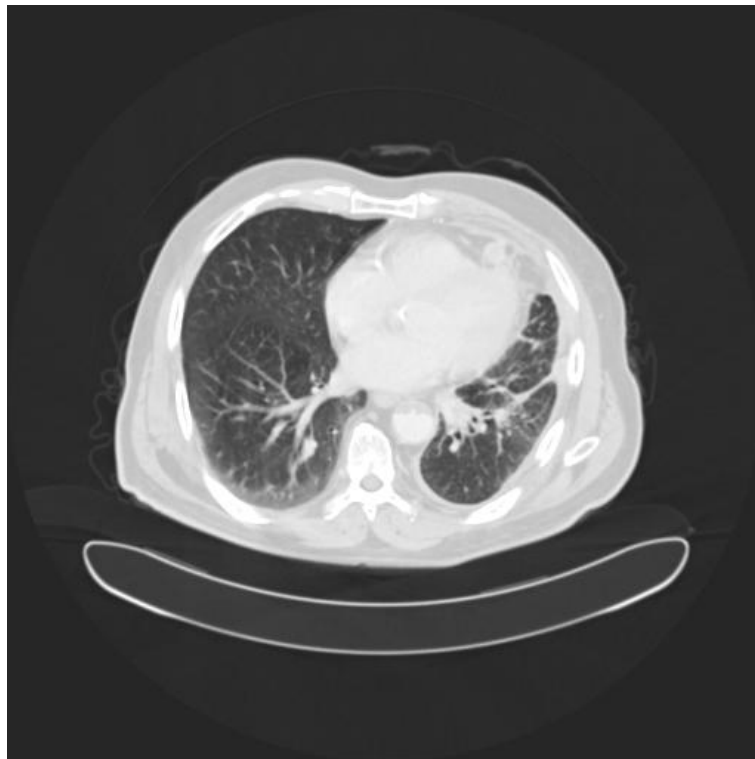


Figura 4.4. Imagen perteneciente a la clase 1 o Maligno

La clase 2 (Normal) comprende imágenes de tomografía computarizada en las que no se observan lesiones, nódulos ni alteraciones estructurales en el parénquima pulmonar, lo que indica la ausencia de patología maligna o benigna significativa. Estas imágenes representan el estado anatómico esperado de los pulmones, con estructuras bien definidas, como los vasos sanguíneos, bronquios y pleura, distribuidos de manera simétrica y sin signos de consolidación, atelectasia, engrosamiento pleural o adenopatías. La

identificación precisa de esta categoría es fundamental para evitar falsos positivos en el sistema de clasificación, ya que un modelo que no distinga adecuadamente entre tejido pulmonar sano y patológico podría generar alertas innecesarias. Un ejemplo representativo de esta clase se muestra en la Figura 4.5, donde se aprecia una imagen con densidad homogénea, bordes pulmonares regulares y sin masas o nódulos detectables, lo cual es consistente con un hallazgo radiológico normal.



Figura 4.5. Imagen perteneciente a la clase 2 o Normal.

4.1.4 Flujo de trabajo

Este proyecto adopta un diseño metodológico mixto con predominio cuantitativo, dado que se trabajará con datos estructurados (imágenes médicas) y se evaluarán modelos mediante métricas cuantificables (precisión, eficiencia, error, etc.). El enfoque es de investigación básica aplicada, ya que se busca generar conocimiento teórico y práctico sobre técnicas de inteligencia computacional, con aplicaciones en problemas reales de clasificación de imágenes.

El flujo de trabajo se presenta en las Figura 4.6 y Figura 4.7, y representa a grandes rasgos el proceso a seguir para la elaboración del proyecto. Comenzando con el estudio de la situación actual del cáncer de pulmón, esto se hace con el fin de entender que es la enfermedad y que se está haciendo actualmente para tratarla, esto hace posible identificar brechas mejorables y aplicarlas al proyecto. Una vez que estas brechas han sido identificadas de forma ambigua es necesario plantearlas de manera que se pueda definir un fin específico, logrando así una delimitación del proyecto.

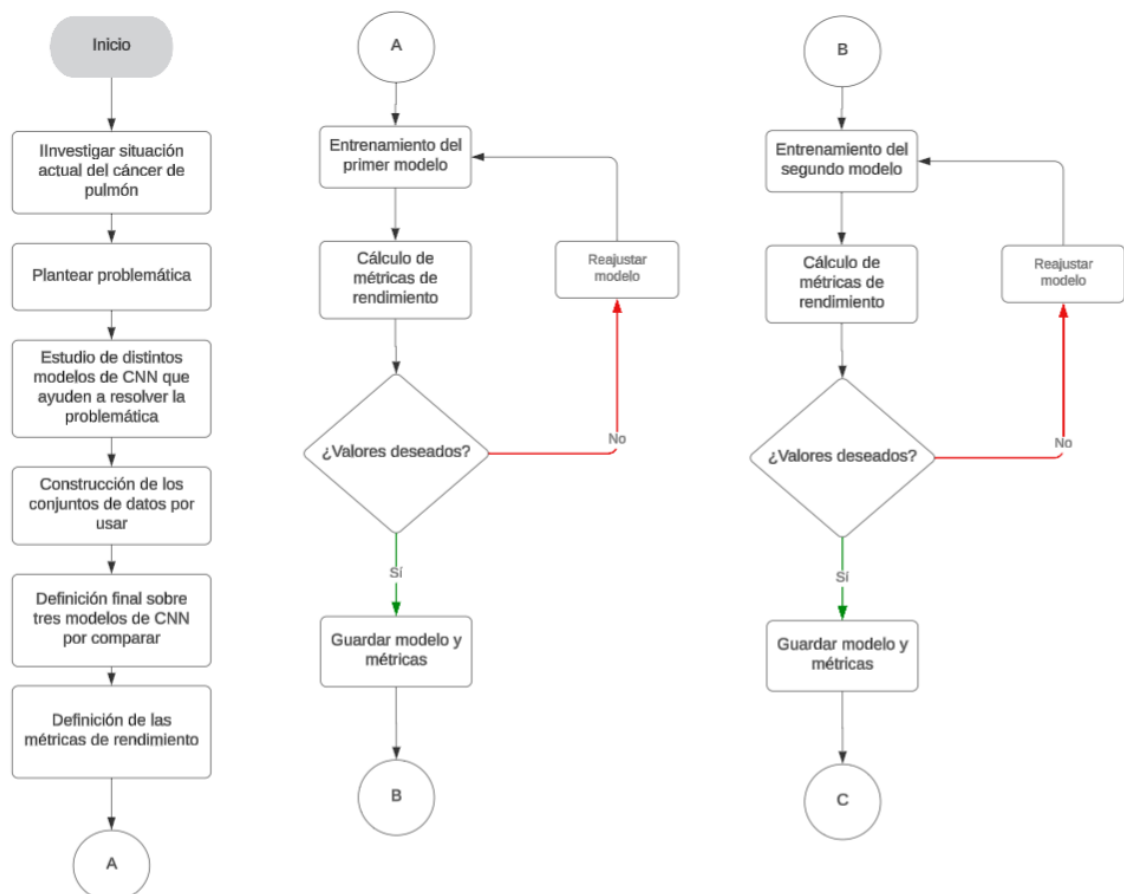


Figura 4.6. Diagrama de flujo sobre el proceso seguido durante la elaboración del proyecto (Parte 1)

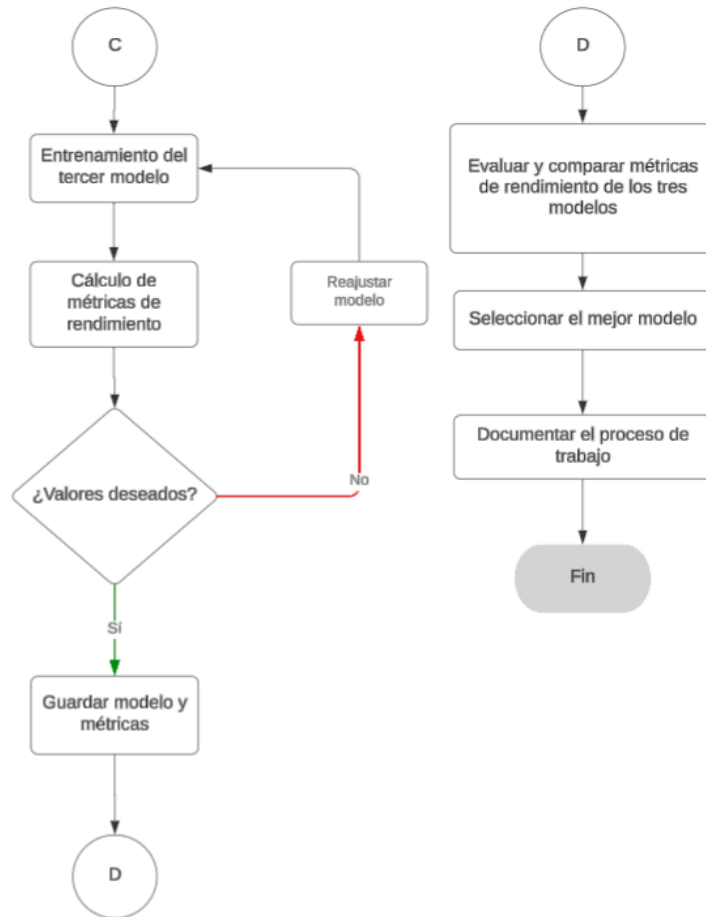


Figura 4.7. Diagrama de flujo sobre el proceso seguido durante la elaboración del proyecto (Parte 2)

Frente a la problemática identificada (la falta de un marco comparativo sistemático y estandarizado para evaluar el desempeño de diferentes arquitecturas de redes neuronales convolucionales en la detección de cáncer de pulmón), se hace necesario proponer una solución que no solo sea técnica y viable, sino también innovadora y con un impacto tangible en el ámbito clínico. Es por ello por lo que, en la tercera fase del proyecto, se lleva a cabo un análisis de las tecnologías a emplear, con el fin de seleccionar aquellas que ofrezcan las mejores condiciones para un entrenamiento adecuado y una generalización efectiva del modelo. Este proceso implica una evaluación de las características de cada arquitectura, sus capacidades de extracción de características, la eficiencia computacional de cada modelo y la experiencia previa de uso en tareas de clasificación de imágenes.

Entre las arquitecturas más prometedoras se encuentran *DenseNet201*, *InceptionV3* y *ResNet50*, que pertenecen a la categoría de aprendizaje profundo y han sido entrenadas previamente con grandes conjuntos de datos como ImageNet, lo que permite aprovechar el *transfer learning* para mejorar el rendimiento en dominios con datos limitados.

La elección de estas tres arquitecturas responde a sus diferencias estructurales que influyen directamente en su capacidad para aprender representaciones jerárquicas de las imágenes. Por ejemplo, *ResNet50* utiliza conexiones residuales que permiten el flujo de gradientes a través de múltiples capas; *InceptionV3* emplea bloques *Inception* con convoluciones de diferentes tamaños para capturar patrones a múltiples escalas; y *DenseNet201* implementa conexiones densas entre capas, fomentando la reutilización de características y mejorando la eficiencia del aprendizaje.

Una vez definidas las tecnologías a utilizar, se procede a la construcción y preparación del conjunto de datos. Dado que los modelos requieren imágenes de tamaños y formatos estandarizados (por ejemplo, 224x224 píxeles para *ResNet50* y *DenseNet201*, y 299x299 para *InceptionV3*), el preprocesamiento de las imágenes debe ser diseñado para cumplir con estos requisitos.

Además, considerando el desbalanceo en el dataset original, donde una clase (benigno) está sobrerrepresentada frente a las otras (maligno y normal), se implementan técnicas de aumento de datos y mejora de imágenes. Estas incluyen transformaciones geométricas como *Grid Distortion* y *Affine Transformations*, ajustes fotométricos como *Random Brightness Contrast* y *CLAHE*, y la adición controlada de ruido (*Gauss Noise*). El objetivo es equilibrar las clases, al tiempo que se aumenta la diversidad del conjunto de entrenamiento, lo que permite al modelo aprender a reconocer patrones bajo variaciones de posición, iluminación y calidad de imagen.

Posteriormente, se define un conjunto de métricas de rendimiento para la evaluación de los modelos. Estas métricas son: precisión, sensibilidad (*recall*), especificidad, F1-Score, puntuación Kappa de Cohen y área bajo la curva ROC (AUC). La selección de estas métricas responde a su capacidad para evaluar la exactitud global del modelo, así como su habilidad para minimizar falsos negativos, que se puede

considerar como un aspecto muy importante en el diagnóstico médico, así como su consistencia en la clasificación de casos de datos desbalanceados.

Una vez establecidas las bases metodológicas, se inicia el proceso de entrenamiento y validación. Este proceso se estructura en las siguientes subetapas: definición del número de épocas, implementación de validación cruzada estratificada con *k-folds* ($k=5$), cálculo de métricas durante el entrenamiento y validación, graficación de las curvas de pérdida y precisión, y, por último, la realización de pruebas de clasificación con un conjunto independiente de prueba.

Finalmente, para la parte de comparación y análisis, se contrastan los resultados obtenidos por los tres modelos en ambos escenarios: dataset balanceado mediante técnicas propias (Caso A) y dataset previamente aumentado por un tercero (Caso B). Esta doble evaluación permite identificar el modelo con mejor desempeño y evidenciar la influencia del preprocesamiento y aumento de datos. El modelo que demuestre el mejor equilibrio entre precisión, sensibilidad, eficiencia computacional y capacidad de generalización será considerado el más adecuado.

4.2 Balanceo de datos del caso A

El caso A toma como base el dataset proporcionado por (Al-Yasriy, 2020), en este se implementarán las siguientes técnicas de aumento de datos para balancear los datos de las 3 categorías:

- Grid Distortion
- Optical Distortion
- Affine Transformations
- Random Brightness Contrast
- Gauss-Noise
- CLAHE
- Unsharp Masking

Usar estas técnicas garantiza que se aumente la cantidad de imágenes, pero que también, al aumentar la diversidad de características, el modelo no solo aprenda a

reconocer patrones muy específicos, si no que más bien aprenda posibles variaciones que pueden surgir en diferentes pacientes, equipos o condiciones. Esto corresponde a un escenario más real y adecuado al problema de diagnóstico asociado.

4.2.1 Procesamiento de datos

La implementación de las técnicas anteriores en el conjunto de datos es crear los suficientes datos para que el modelo identifique las características más notorias, y también para asegurar que el modelo realmente aprenda y que sea capaz de generalizar bien.

De manera específica Grid Distortion y Optical Distortion logran deformar la imagen de manera que se simula variaciones tanto en pacientes como en los equipos que se usan para realizar la tomografía. En complemento con esto, Affine Transformations ayuda a simular diferentes posiciones o ángulos del cuerpo del paciente al momento del escaneo. De la misma forma Random Brightness Contrast y Gauss-Noise, simulan el comportamiento que presentan equipos viejos o deteriorados, pues estas técnicas añaden ruido a las imágenes, no obstante, este no es muy alto para evitar dañar el dataset y por ende al modelo final. Por último, tanto CLAHE como Unsharp Masking están destinados a mejorar detalles de las imágenes, resaltando áreas que tengan bajo contraste, con el fin de identificar mejores zonas que presenten pequeños nódulos o lesiones. Ejemplos de imágenes resultantes luego de aplicar estas técnicas pueden verse en la sección de Resultados.

Python es la herramienta que se utilizó para implementar estas técnicas, esto debido a que existen librerías que facilitan la implementación de estas. Con este enfoque se desarrolla el código de la Figura 4.8, este contiene todas las técnicas mencionadas anteriormente, pero se le asigna un porcentaje de que se le aplique a la imagen esto con el fin de aplicar las técnicas de forma aleatoria.

```

augmentations = A.Compose([
    A.GridDistortion(num_steps=5, distort_limit=0.2, p=0.5),
    A.OpticalDistortion(distort_limit=0.02, p=0.5),
    A.Affine(scale=(0.95, 1.05), translate_percent=(0.02, 0.05), shear=(-3, 3), p=0.5),
    A.RandomBrightnessContrast(brightness_limit=0.05, contrast_limit=0.05, p=0.3),
    A.GaussNoise(std_range=(0.1, 1.0), p=0.2),
    A.CLAHE(clip_limit=1.0, tile_grid_size=(8,8), p=0.2),
])

```

Figura 4.8 Código para aplicar las técnicas de aumento de datos

Esto crea imágenes nuevas con las técnicas; sin embargo, como se observa, únicamente están seis de las siete técnicas mencionadas. La última técnica no entra dentro de este código, pues la finalidad de esta es mejorar las características de todas las imágenes. Para lograr esto se unen dos técnicas: Gaussian Blur y Subtract. Estas dos técnicas permiten restar un suavizado a la imagen original, para posteriormente guardar esto en una imagen nueva y finalmente sumar esta nueva imagen a la imagen original, logrando así resaltar detalles esenciales en la imagen.

4.3 Experimentación

Con el dataset balanceado y preprocesado según las técnicas definidas en el caso A, se inicia la fase de experimentación, cuyo objetivo principal es evaluar y comparar el desempeño de tres arquitecturas de redes neuronales convolucionales ampliamente utilizadas: *InceptionV3*, *ResNet50* y *DenseNet201*.

El entrenamiento de cada modelo se realiza bajo condiciones controladas y estandarizadas, utilizando *transfer learning* con pesos preentrenados, lo que permite aprovechar características visuales generales y acelerar la convergencia del proceso de aprendizaje.

Para garantizar una evaluación robusta y evitar el sobreajuste, se implementa una validación cruzada estratificada con $k=5$ *folds*, asegurando que cada modelo sea entrenado y validado en diferentes particiones del conjunto de datos. Además, se define una proporción de 90% para entrenamiento y validación cruzada, y 10% para pruebas finales, con el fin de evaluar la capacidad de generalización de cada arquitectura. A continuación,

se detalla el proceso de entrenamiento para cada modelo, incluyendo las transformaciones de entrada, la configuración del optimizador Adam con una tasa de aprendizaje de 0.001, la función de pérdida por entropía cruzada categórica y el número de épocas, estableciendo así las bases para un análisis comparativo riguroso de su rendimiento en la detección de cáncer de pulmón.

4.3.1 Entrenamiento de *InceptionV3*

Después de balancear los datos con las técnicas propuestas, el primer modelo con el que se prueba el dataset es *InceptionV3*. Las imágenes que se utilizan para el entrenamiento están almacenadas de manera local, separadas en carpetas con base en las tres clasificaciones antes mencionadas. Usualmente, cuando se trabaja con proyectos de *Machine Learning*, se entrenan múltiples modelos y, a su vez, estos modelos resultantes son puestos a prueba con el fin de determinar cuál de ellos es el mejor.

Para lograr esto es necesario separar el dataset en tres secciones: entrenamiento, validación y prueba. Esto no siempre resulta conveniente, pues esto significaría que el entrenamiento tendría menos datos y en caso de datasets pequeño esto puede ser contradictorio. Para solventar esto se hace uso del método de validación cruzada y *k-folds*.

El algoritmo de validación cruzada y *k-folds* parte de una mezcla aleatoria de los datos. Enseguida, se inicializa el parámetro k , que es un número entero que representa el número de particiones del conjunto de datos. Finalmente se establecen, tanto el número de iteraciones de entrenamiento como el de validación que se usan al construir el modelo. Este modelo consta de los siguientes pasos, que se repiten dependiendo del valor de k :

- Selecciona una k partición, que no será vista por el modelo (“oculta”).
- Con las particiones restantes se entrena el modelo.
- Una vez entrenado el modelo, se almacena su desempeño
- Se repite el proceso, pero cambiando la partición que se mantiene “oculta”.

Como parte del procesamiento de imágenes se considera establecer un porcentaje del total de imágenes para entrenamiento y otro porcentaje para validación del modelo clasificador, en donde antes de pasar el conjunto de datos a un modelo pasa por un

preprocesamiento en el que implica convertir el valor de los píxeles de las imágenes a tensores que son soportados, así como el previo procesamiento que los modelos requieren. En este caso se hizo en una proporción de 90%-10%, en donde el 90% de las imágenes serán utilizadas para el proceso del entrenamiento con la validación cruzada y los *k-folds*, el 10% restante será utilizado para la parte de prueba. Cabe mencionar que esta separación se hizo de manera aleatoria mediante Python.

Comenzando con el entrenamiento, es necesario el uso de librerías que permitan manejar datos (PyTorch, torchvision), dividir datos (*stratified k-fold*) y calcular métricas.

De manera general el modelo *InceptionV3* requiere un tamaño de imagen de 299x299 píxeles además de normalizar los valores de los píxeles usando valores medios y la desviación estándar de ImageNet, por esto es necesario preprocesar las imágenes antes de pasarlas al entrenamiento, esto se hace con el siguiente código:

```
transform = transforms.Compose([
    transforms.Resize((299, 299)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

Después de realizar pruebas de entrenamiento con valores de *k-folds* diferentes, se concluyó que el valor que mejor presenta resultados es 5, por lo que cada partición es de 180 imágenes. Para el entrenamiento se carga un modelo preentrenado de *InceptionV3* al que se reemplaza la capa final, con el fin de adaptarla a 3 clases: Benigno, Maligno y Normal. Cada subconjunto se entrena 10 épocas con el fin de mejorar la precisión del modelo, se guarda y se identifica el mejor modelo según la precisión. Con el modelo final se utiliza entropía cruzada debido a la clasificación multiclase, además del optimizador Adam con una tasa de aprendizaje de 0.001.

En el transcurso del entrenamiento, además, se recolectan las métricas de rendimiento mencionadas en capítulos anteriores. Estas métricas y la sección de pruebas pueden ser apreciadas en la sección de Resultados.

El procedimiento general, que aplica para este modelo puede ser apreciado de mejor forma mediante el siguiente pseudocódigo, pero debe observarse que en esencia puede replicarse para cualquiera de los tres modelos elegidos, con un mínimo de ajustes:

INICIO

Crear carpetas para guardar reportes y modelos

Definir transformaciones para las imágenes:

- Redimensionar a 299x299
- Convertir a tensor
- Normalizar con medias y desviaciones estándar

Cargar dataset de imágenes desde una ruta

Obtener etiquetas y nombres de clases

Imprimir etiquetas e índices

Definir número de *k-folds* para validación cruzada

Preparar validación cruzada estratificada

Inicializar mejor precisión y ruta del mejor modelo

PARA cada *k-fold*:

Mostrar número de *k-fold* actual

Crear subconjuntos de entrenamiento y validación

Crear DataLoaders para entrenamiento y validación

Cargar modelo *InceptionV3* preentrenado

Desactivar salida auxiliar

Cambiar la última capa para que tenga 3 clases

Definir función de pérdida y optimizador

PARA cada época de entrenamiento:

Poner modelo en modo entrenamiento

Inicializar pérdida acumulada

PARA cada lote de datos en el entrenamiento:

Mover datos al dispositivo

Hacer forward-pass

Calcular pérdida

Propagar gradientes

Actualizar pesos

Sumar la pérdida

Mostrar la pérdida de la época actual

Evaluar el modelo

Inicializar listas de etiquetas verdaderas y predichas

PARA cada lote del conjunto de validación:

Mover imágenes al dispositivo

Hacer predicciones

Obtener etiquetas predichas

Guardar etiquetas verdaderas y predichas

Calcular métricas

Mostrar métricas

FIN

4.3.2 Entrenamiento de *ResNet50*

Si bien el flujo general de entrenamiento sigue la misma estructura metodológica definida para *InceptionV3*, el modelo *ResNet50* introduce una arquitectura basada en bloques residuales (*skip connections*), diseñados para mitigar el problema del desvanecimiento de gradientes en redes profundas. Esta característica le permite entrenar de manera más estable y eficiente, incluso con un gran número de capas.

En este trabajo, *ResNet50* se adapta a imágenes de entrada de 224x224 píxeles, tamaño estándar para esta arquitectura, y se utiliza *transfer learning* con pesos preentrenados, reemplazando únicamente la capa final para ajustarla a las tres clases del problema (Benigno, Maligno, Normal).

El entrenamiento se realiza bajo las mismas condiciones de validación cruzada estratificada ($k=5$), optimizador Adam y tasa de aprendizaje de 0.001, permitiendo una comparación justa con los otros modelos.

4.3.3 Entrenamiento de *DenseNet201*

El modelo *DenseNet201* representa un enfoque distinto en la conexión entre capas, caracterizado por su estructura densamente conectada, donde cada capa recibe como entrada las características de todas las capas anteriores. Esta arquitectura promueve una fuerte reutilización de características y mejora el flujo de información y gradientes a través de la red, lo que puede traducirse en un aprendizaje más eficiente y una menor tendencia al sobreajuste.

Al igual que *ResNet50*, *DenseNet201* opera con imágenes de entrada de 224x224 píxeles y se implementa mediante *transfer learning*, conservando los pesos preentrenados y modificando solo la capa de clasificación para adaptarla al problema de tres clases. El proceso de entrenamiento sigue el mismo esquema de validación cruzada, optimización y métricas de evaluación, garantizando la consistencia metodológica con los otros dos modelos evaluados.

Con la finalización del proceso de entrenamiento y validación de los modelos *InceptionV3*, *ResNet50* y *DenseNet201*, se completa la fase metodológica del presente trabajo. El diseño experimental, basado en validación cruzada estratificada con $k=5$ folds, el uso de técnicas de aumento de datos y un preprocesamiento estandarizado, garantiza una evaluación justa y reproducible del desempeño de cada arquitectura.

Además, la comparación se realiza bajo los escenarios definidos previamente, que son: el dataset balanceado mediante técnicas propias (Caso A) y un dataset previamente

aumentado por un tercero (Caso B), lo que permite evaluar la eficacia de los modelos y analizar el impacto de las estrategias de aumento de datos.

Esta metodología sienta las bases para un análisis de los resultados, los cuales se presentan en el siguiente capítulo. A continuación, se detallan los valores de precisión, sensibilidad, F1-Score, puntuación Kappa y tiempo de clasificación para cada modelo, acompañados de matrices de confusión y gráficas de desempeño que permiten una interpretación clara y comparativa. Asimismo, se contrastan estos resultados con otros trabajos del estado del arte, con el fin de contextualizar el aporte del presente estudio dentro del campo de la inteligencia artificial aplicada al diagnóstico médico.

5 RESULTADOS Y DISCUSIÓN

Una vez completado el proceso metodológico, que incluye la selección y preprocesamiento de los datos, el balanceo del dataset mediante técnicas avanzadas de aumento de datos, y el entrenamiento y validación cruzada de tres arquitecturas de redes neuronales convolucionales (*DenseNet201*, *InceptionV3* y *ResNet50*), se procede a presentar y analizar los resultados obtenidos en este estudio.

El objetivo principal de este capítulo es exponer de manera sistemática y objetiva el desempeño de cada modelo en dos escenarios diferentes: uno basado en un dataset balanceado mediante técnicas de aumento de datos desarrolladas en este trabajo (Caso A), y otro utilizando un dataset previamente aumentado por un tercero (Caso B).

Los resultados se presentan en orden lógico, comenzando con la descripción de las técnicas de aumento de datos aplicadas y su impacto visual en las imágenes, seguido de la evaluación del entrenamiento de cada modelo mediante métricas de rendimiento como precisión, sensibilidad, F1-Score, puntuación Kappa de Cohen y tiempo de clasificación. Posteriormente, se muestran las matrices de confusión y gráficas de complejidad temporal para analizar la capacidad de generalización y eficiencia computacional de los modelos. Finalmente, se realiza la comparación cuantitativa de los resultados.

5.1 Técnicas de aumento de datos

Como se ha insistido, la calidad y diversidad del conjunto de datos de entrenamiento son determinantes en el desempeño de los modelos de aprendizaje profundo, como en este caso, en donde los datos etiquetados suelen ser escasos y desbalanceados. Por esto, las técnicas de aumento de datos permiten la generación artificial de nuevas imágenes a partir de las existentes, introduciendo variaciones controladas que mejoran la capacidad del modelo para generalizar.

En este trabajo, se implementaron seis técnicas: *Grid Distortion*, *Optical Distortion*, *Affine Transformations*, *Random Brightness Contrast*, *Gauss Noise* y *CLAHE*, seleccionadas para simular condiciones reales de variabilidad en las tomografías, como

posibles errores de posicionamiento del paciente, calibración del equipo, ruido en la adquisición de imágenes o diferencias en el contraste. A diferencia de aumentos básicos como rotaciones o volteos, estas transformaciones más complejas permiten al modelo aprender a reconocer patrones bajo una amplia gama de condiciones, lo que contribuye para reducir el sobreajuste y para mejorar el sistema de clasificación.

A continuación, se muestran ejemplos visuales de cada técnica aplicada a algunas de las imágenes del dataset, para ilustrar su comportamiento y justificando su inclusión en el proceso de preprocesamiento de imágenes.

La Figura 5.1 muestra una imagen del dataset original a la que se ha aplicado la técnica de *Grid Distortion*. Esta transformación divide la imagen en una cuadrícula y desplaza aleatoriamente los puntos de control de cada celda, introduciendo deformaciones locales que simulan variaciones durante la adquisición de la tomografía. Su uso en el aumento de datos permite al modelo aprender a reconocer patrones incluso bajo algunas distorsiones espaciales moderadas.

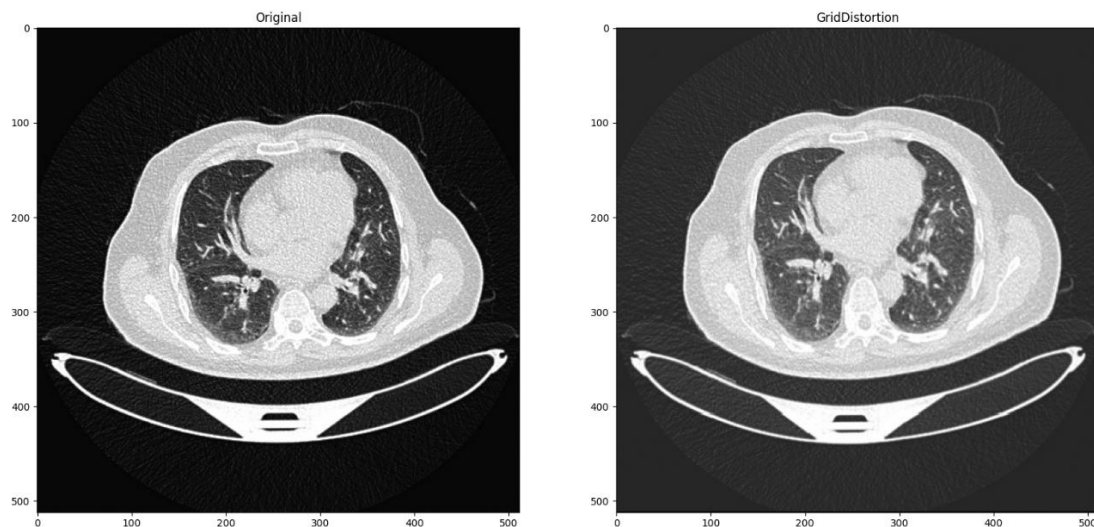


Figura 5.1 Resultado de aplicar Grid Distortion.

En otro ejemplo, se presenta la Figura 5.2 que corresponde con una imagen del dataset modificada con la técnica de *Optical Distortion* aplicada. Esta transformación simula efectos ópticos no lineales, como los causados por errores de calibración en el

equipo de imagenología, introduciendo una distorsión radial que afecta principalmente los bordes de la imagen.

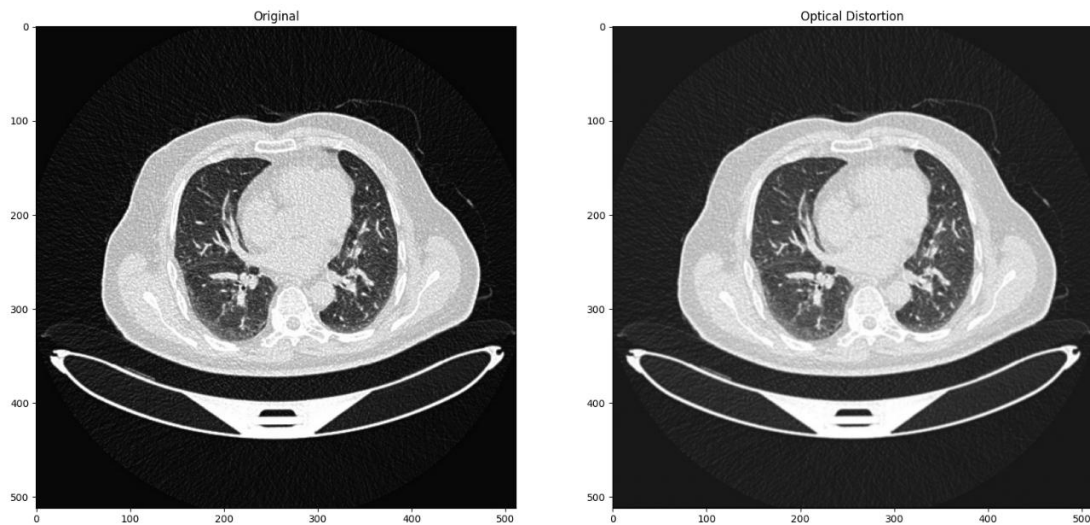


Figura 5.2 Resultado de aplicar Optical Distortion.

En la Figura 5.3 se pretende que se pueda ver el efecto de transformaciones sobre una imagen del dataset. Para esto, esta técnica combina operaciones como rotación, traslación, escalado y cizallamiento, simulando diferentes ángulos o posiciones del paciente durante el escaneo. Con este tipo de variaciones, el modelo se vuelve potencialmente más inmune a cambios en la orientación del pulmón, por ejemplo, lo cual permitiría tener una clasificación, de manera independiente a la postura del sujeto o del equipo.

De manera complementaria, en la Figura 5.4 se muestra el resultado de modificar aleatoriamente el brillo y el contraste mediante la técnica *Random Brightness Contrast*. Aquí se simulan condiciones variables de iluminación o posibles diferencias en la intensidad del escáner, ayudando al modelo a enfocarse en las características estructurales de la imagen en lugar de depender de aprender valores de píxeles específicos. Esto es especialmente útil para mejorar la adaptabilidad del modelo a imágenes provenientes de distintos equipos.

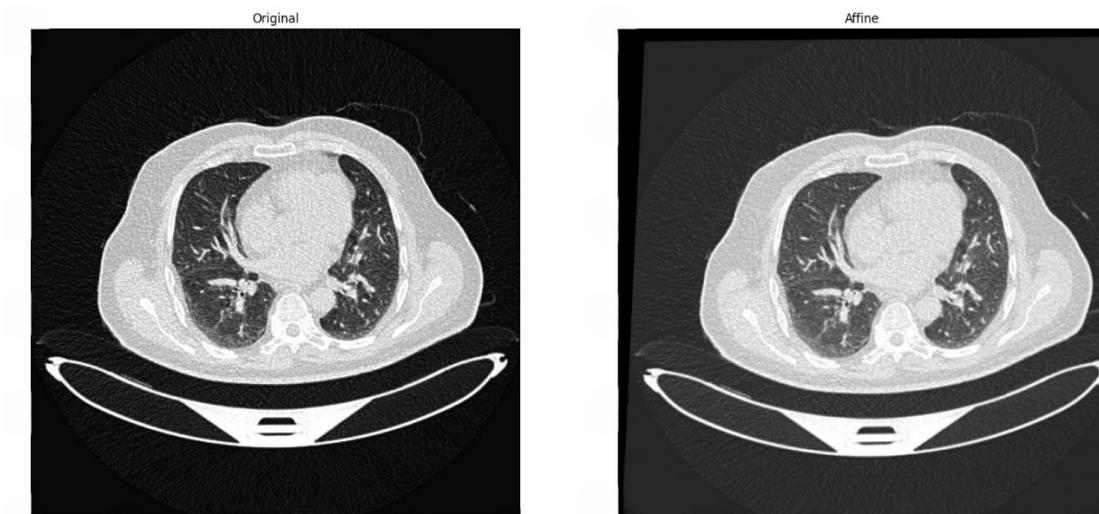


Figura 5.3 Resultado de aplicar Affine Transformations.

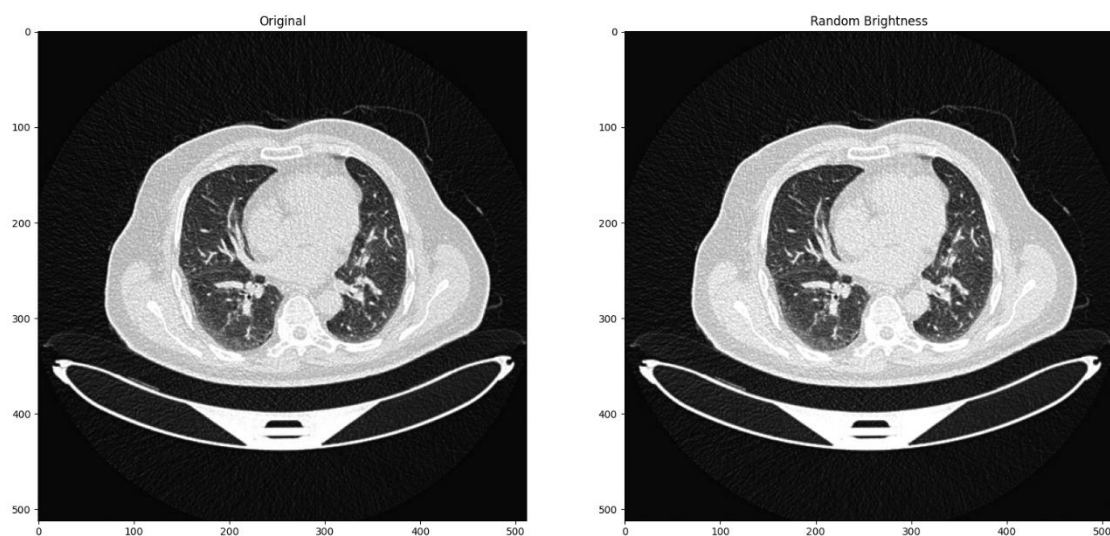


Figura 5.4 Resultado de aplicar Random Brightness.

En cuestiones de alteraciones más sutiles, la Figura 5.5 presenta una imagen del dataset con ruido gaussiano añadido. Este ruido simula perturbaciones aleatorias que pueden ocurrir durante la adquisición de la imagen, como interferencias electrónicas o limitaciones del hardware. Si se entrena con imágenes ruidosas, el modelo desarrolla mayor tolerancia a imperfecciones comunes en estudios clínicos, reduciendo el riesgo de sobreajuste a datos limpios o muy idealizados.

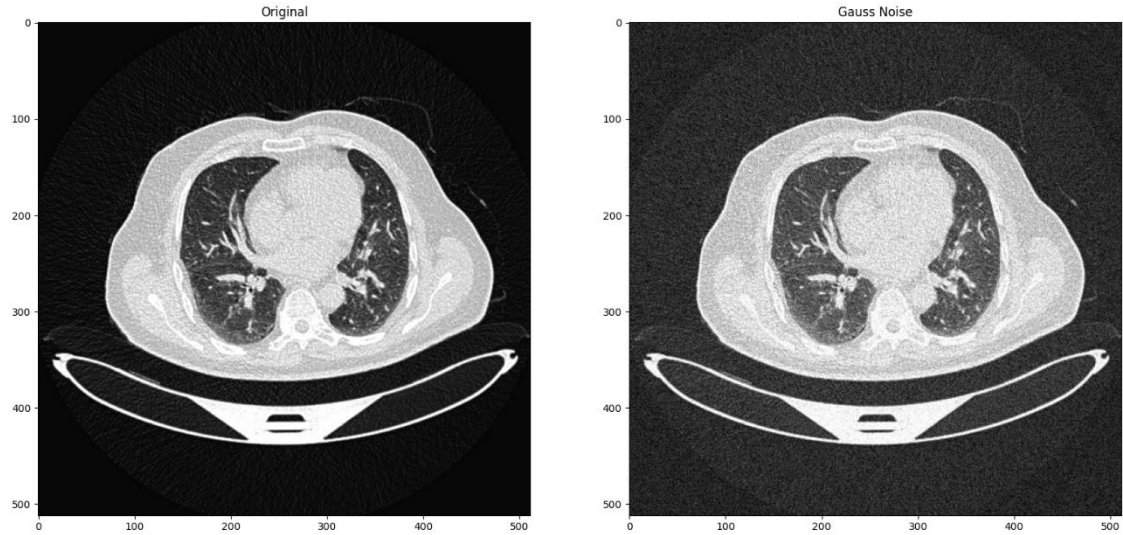


Figura 5.5 Resultado de aplicar Gauss Noise

Finalmente, la Figura 5.6 exhibe un ejemplo de imagen luego de ser procesada con *CLAHE*. Esta técnica mejora localmente el contraste de regiones específicas, resaltando detalles sutiles en áreas de baja intensidad, como pequeños defectos o bordes difusos. Es decir, que CLAHE facilita que el modelo detecte, por ejemplo, lesiones incipientes que podrían pasar desapercibidas en imágenes con contraste homogéneo.

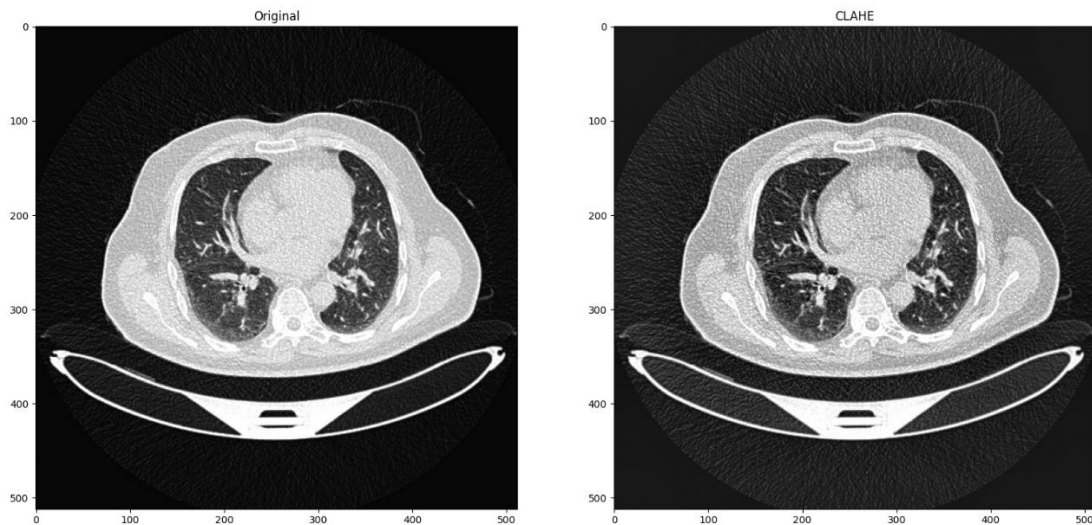


Figura 5.6 Resultado de aplicar CLAHE.

5.2 Dataset final creado

Después de aplicar las técnicas de aumento de datos, se pasó de tener 1097 imágenes distribuidas en tres categorías de manera diferente entre ellas, a tener 3000 imágenes distribuidas de manera equitativa. La Figura 5.7 muestra la nueva distribución con respecto a los datos del dataset original.

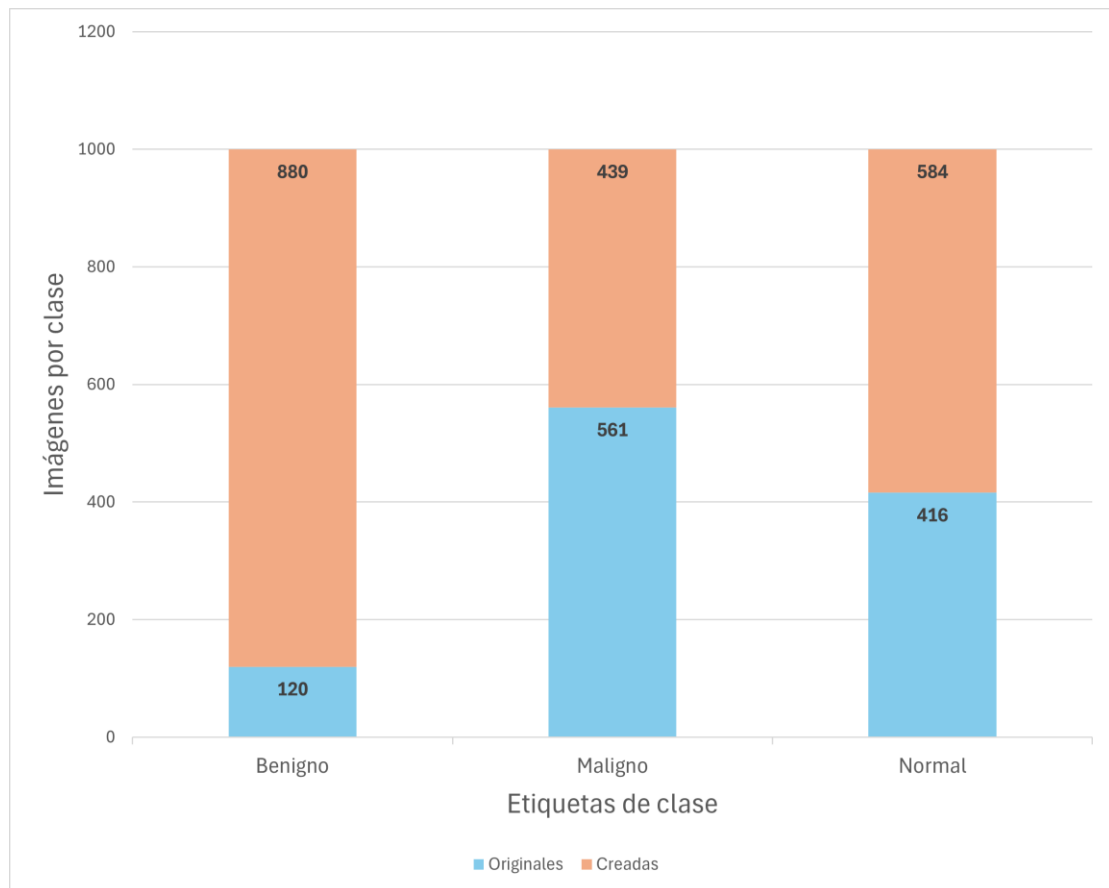


Figura 5.7 Grafica sobre la distribución de datos del dataset creado

Esta figura anterior muestra la nueva distribución del dataset tras la aplicación de técnicas avanzadas de aumento de datos, resultando en un conjunto de imágenes divididas equitativamente en 1000 imágenes por clase (Benigno, Maligno y Normal). A diferencia del dataset original, que presentaba un marcado desbalanceo con una clara sobrerrepresentación de la clase *Benigno*, esta nueva distribución garantiza que cada categoría contribuya de manera proporcional al entrenamiento del modelo.

El balanceo de clases es un paso muy importante en el desarrollo de sistemas de inteligencia artificial para diagnóstico médico, ya que evita que el modelo desarrolle un sesgo hacia la clase mayoritaria, lo cual podría llevar a una alta precisión general, pero con una baja sensibilidad para detectar casos de cáncer (principalmente en la clase *Maligno*). Al equilibrar las clases, se mejora significativamente la capacidad del modelo para aprender características distintivas de cada categoría, especialmente de aquellas menos representadas, lo que se traduce en una detección más confiable.

5.3 Resultados en los modelos entrenados

Tras la preparación y balanceo del conjunto de datos, se procede al entrenamiento de las tres arquitecturas CNN profundas: *InceptionV3*, *ResNet50* y *DenseNet201*. Cada modelo se entrena bajo un esquema de validación cruzada estratificada con $k=5$ folds. El proceso se realiza con un conjunto de hiperparámetros comunes, como el optimizador Adam, una tasa de aprendizaje de 0.001 y una función de pérdida por entropía cruzada categórica (esta última como parámetro por defecto en las bibliotecas usadas), garantizando así una comparación controlada entre las arquitecturas. A continuación, se presenta el desempeño de cada modelo durante el entrenamiento, analizando la evolución de la precisión y la pérdida a lo largo de las épocas, con el fin de identificar el mejor subconjunto para su posterior evaluación en el conjunto de prueba.

5.3.1 *Inception V3*

La Figura 5.8 muestra la métrica de precisión del entrenamiento de cada subconjunto, con base en esta es que se eligió el mejor modelo con el cual se realizan las pruebas de clasificación. Los resultados son buenos; de manera general significa que por cada categoría el modelo en el proceso de validación al entrenar clasificó de 1 a 3 imágenes mal (del total del conjunto de datos). Complementariamente, la Figura 5.9 muestra la función de pérdida durante el entrenamiento; esta figura solo muestra la función de pérdida del mejor subconjunto obtenido y del cual se hizo uso durante las pruebas de clasificación finales. Se observa una adecuada convergencia de la razón de aprendizaje.

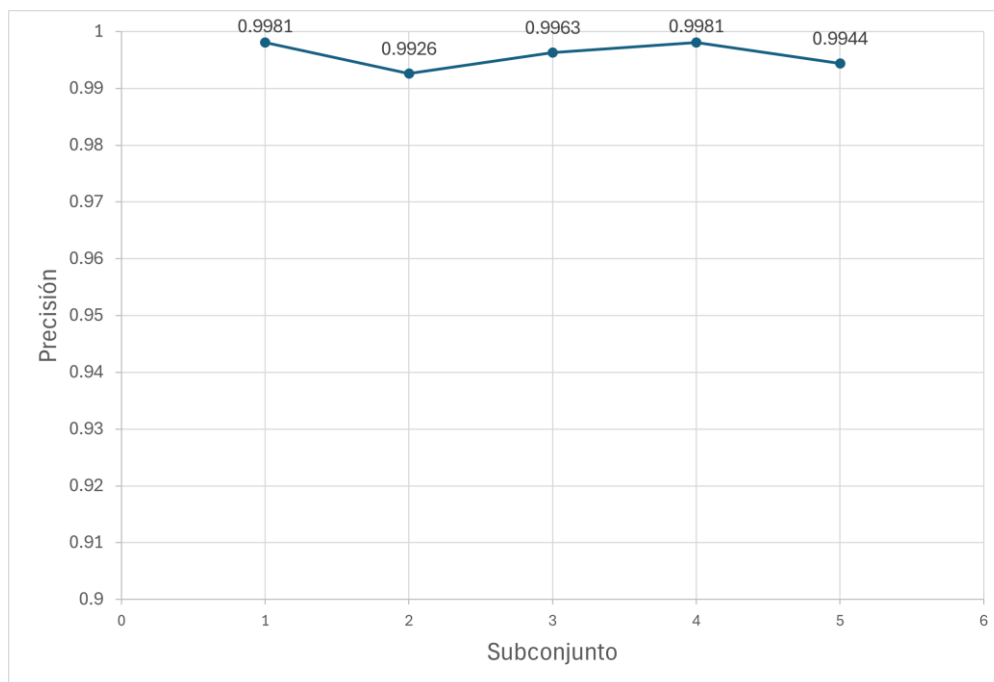


Figura 5.8 Grafica sobre la precisión del entrenamiento por subconjunto con el modelo Inception V3.

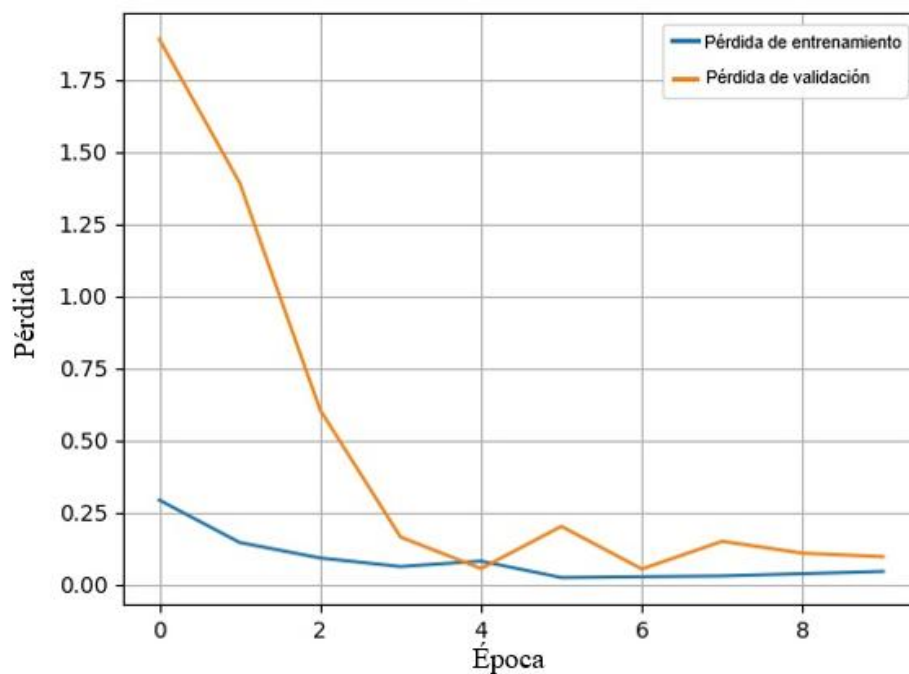


Figura 5.9 Grafica sobre los valores de pérdida del mejor Subconjunto del modelo Inception V3

Las métricas representadas en la Figura 5.9, se usan para analizar cómo evoluciona el aprendizaje del modelo. Mientras que la pérdida de entrenamiento es ocupada para medir el error que el modelo comete sobre los datos de entrenamiento, la pérdida de validación es ocupada para medir el error que el modelo presenta sobre datos que no se han visto antes. Ambas métricas son usadas para corroborar que no exista un sobreajuste.

De manera general, el modelo *InceptionV3* mostró un desempeño altamente consistente durante el entrenamiento, alcanzando una precisión superior al 99% en todos los subconjuntos de validación cruzada. Su arquitectura, basada en bloques *Inception* que procesan características a múltiples escalas, le permite capturar patrones finos en las tomografías, lo que se refleja en su rápida convergencia y baja pérdida de validación. El mejor subconjunto seleccionado presentó una evolución estable de las métricas, sin signos de sobreajuste, lo que demuestra su capacidad para generalizar a partir del conjunto de datos balanceado.

5.3.2 *ResNet 50*

El modelo *ResNet50* presentó una buena estabilidad en cada subconjunto. La Figura 5.10 muestra la precisión de cada subconjunto de entrenamiento del modelo *ResNet 50*. Con base en estos datos, se determina que existen dos subconjuntos que presentan la misma precisión, sin embargo, en algunas otras métricas como lo es la sensibilidad salió mejor el subconjunto dos, por lo que se establece como modelo final este. Por otra parte, la Figura 5.11 muestra los valores de pérdida que tuvo el mejor subconjunto del modelo durante el proceso de entrenamiento.

ResNet50 destacó por su estabilidad y alto rendimiento, con valores de precisión que se mantuvieron por encima del 99.5% en todos los subconjuntos. La presencia de conexiones residuales permitió un flujo eficiente de gradientes, facilitando un entrenamiento robusto incluso con un número reducido de épocas. El subconjunto seleccionado como modelo final mostró una de las menores pérdidas de validación, lo que, sumado a su eficiencia computacional, lo posiciona como uno de los modelos más confiables del estudio.

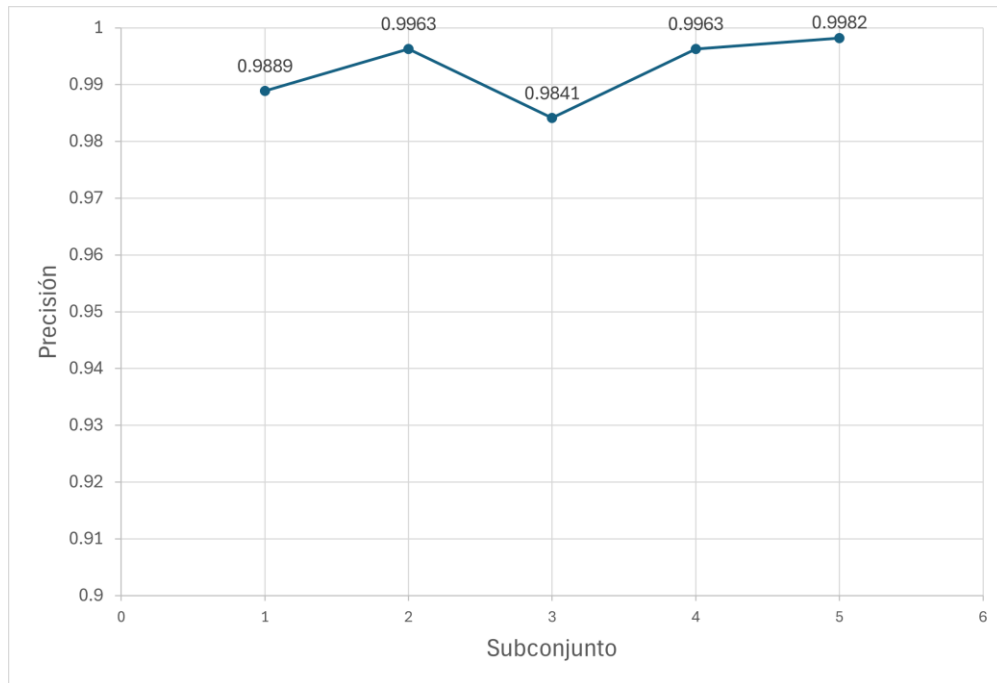


Figura 5.10 Grafica sobre la precisión del entrenamiento por subconjunto del modelo ResNet 50

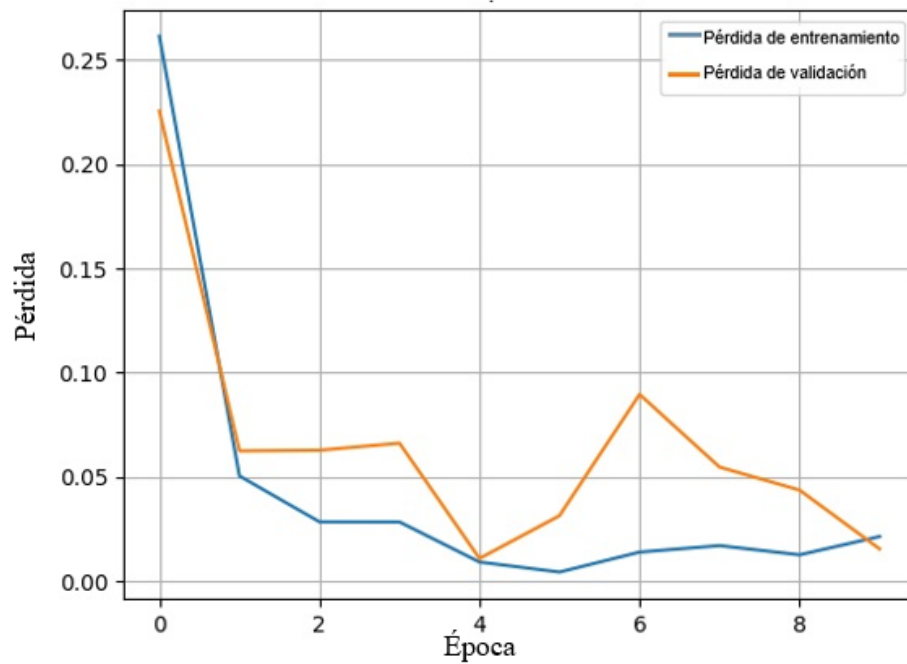


Figura 5.11 Grafica sobre los valores de pérdida del mejor subconjunto del modelo ResNet 50.

5.3.3 DenseNet 201

La Figura 5.12 muestra los datos de precisión de cada subconjunto de entrenamiento del modelo *DenseNet 201*. Se observa que este modelo presenta de igual forma buen comportamiento, pues los valores de precisión no bajan del 90% lo que significa que en ocasiones puede clasificar imágenes mal, destacando mayormente fallos en la categoría Normal.

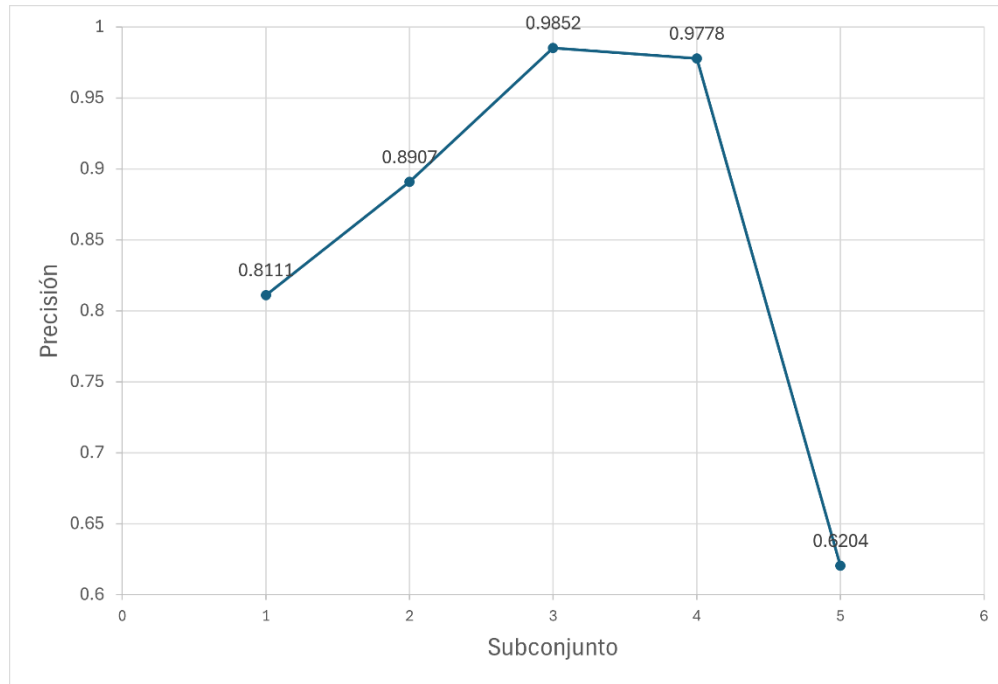


Figura 5.12 Gráfica de la precisión del entrenamiento por subconjunto, modelo DenseNet201.

Por otra parte, la Figura 5.13 muestra los valores de pérdida que tuvo el mejor subconjunto del modelo *DenseNet 201*. Este modelo exhibió un buen desempeño, aunque con una convergencia más lenta en comparación con los otros modelos. Su arquitectura densamente conectada promueve una fuerte reutilización de características, lo que se tradujo en una precisión cercana al 98.5%. Aunque no alcanzó los valores máximos de los otros modelos, mostró una buena capacidad de generalización, especialmente en la detección de nódulos benignos, gracias a la propagación efectiva de información a través de sus capas.

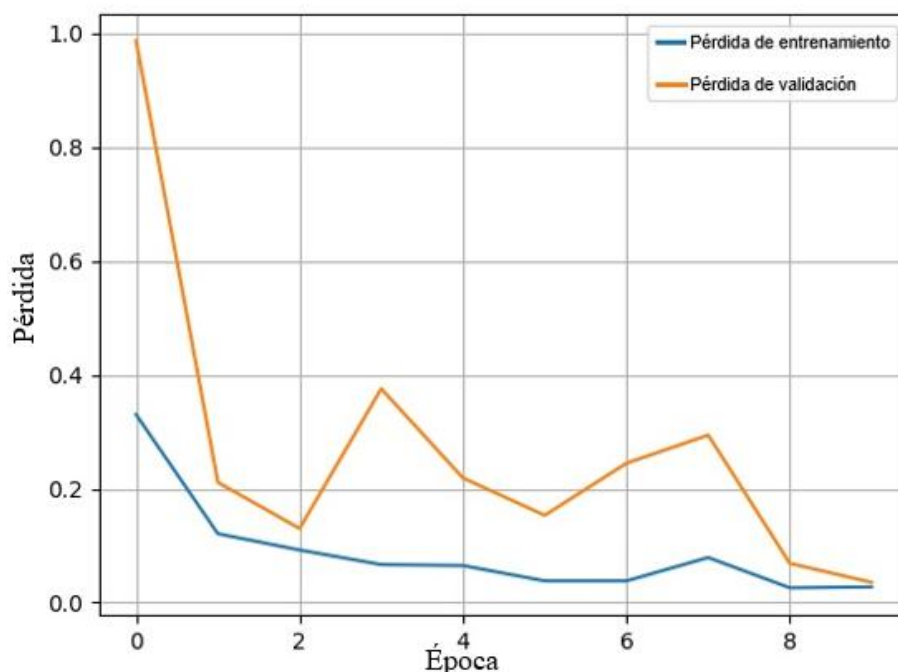


Figura 5.13. Gráfica de pérdida del mejor subconjunto del modelo DenseNet201.

5.3.4 Google Net

Google Net es una versión anterior de *InceptionV3*. Se incluyó en este estudio con el fin de comparar el resultado que entregan ambos modelos e identificar si las imágenes trabajan mejor con versiones anteriores o por el contrario las versiones más actualizadas entregan mejores resultados. La Figura 5.14 muestra la precisión de los subconjuntos durante el proceso de entrenamiento.

Este modelo presenta de igual forma buen comportamiento, pues los valores de precisión no bajan del 90% lo que significa que en ocasiones puede clasificar imágenes mal, destacando mayormente fallos en la categoría Normal.

GoogleNet mostró un comportamiento aceptable, con una precisión que superó el 90% en todos los subconjuntos, aunque inferior a la de su versión mejorada. Este resultado sugiere que las mejoras introducidas en *InceptionV3*, como una estructura más profunda y optimizaciones en los bloques *Inception*, tienen un impacto positivo significativo en el rendimiento. Aunque este no fue el mejor modelo, su desempeño confirma las bondades de la familia de arquitecturas *Inception* para tareas de clasificación de imágenes.

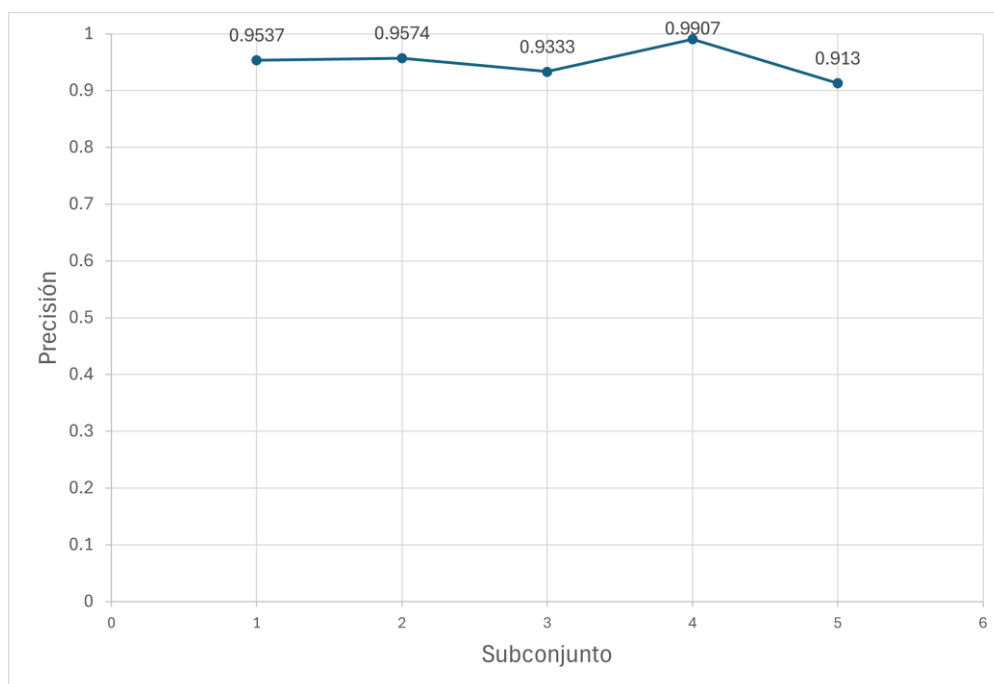


Figura 5.14 Gráfica de la precisión del entrenamiento por subconjunto del modelo GoogleNet.

5.4 Pruebas de validación

Una vez seleccionado el mejor modelo de cada arquitectura luego del entrenamiento con la validación cruzada, se procede a evaluar su desempeño en un conjunto de prueba independiente, que nunca fue visto durante el entrenamiento. Esta fase de validación tiene como objetivo medir la capacidad de generalización de los modelos y analizar su comportamiento en escenarios reales.

Las pruebas en esta fase se realizan en los dos casos ya explicados: primero, con el dataset balanceado desarrollado en este trabajo (Caso A), y segundo, con un dataset previamente aumentado por un tercero (Caso B). Los resultados se presentan mediante matrices de confusión y análisis de rendimiento temporal, proporcionando una visión completa de la precisión, así como de la eficiencia computacional de cada modelo.

Las siguientes matrices de confusión muestran los resultados al clasificar imágenes nunca vistas en los modelos finales. Hay que mencionar que como se indicó con anterioridad, el dataset se dividió en una proporción de 90% para el entrenamiento y 10%

restante para el proceso de pruebas de validación. Sabiendo esto, el número de imágenes para el dataset propio es de 100, mientras que para el dataset consultado en internet es de 120 imágenes.

5.4.1 Caso A. Dataset creado

A. Inception V3

En la Figura 5.15 se visualizan los resultados de utilizar el mejor subconjunto del modelo *Inception V3* para realizar predicciones. De acuerdo con las pruebas realizadas, los resultados no varían demasiado, identificando principalmente que surgen predicciones erróneas comúnmente de 1 a 2 en todas las categorías.

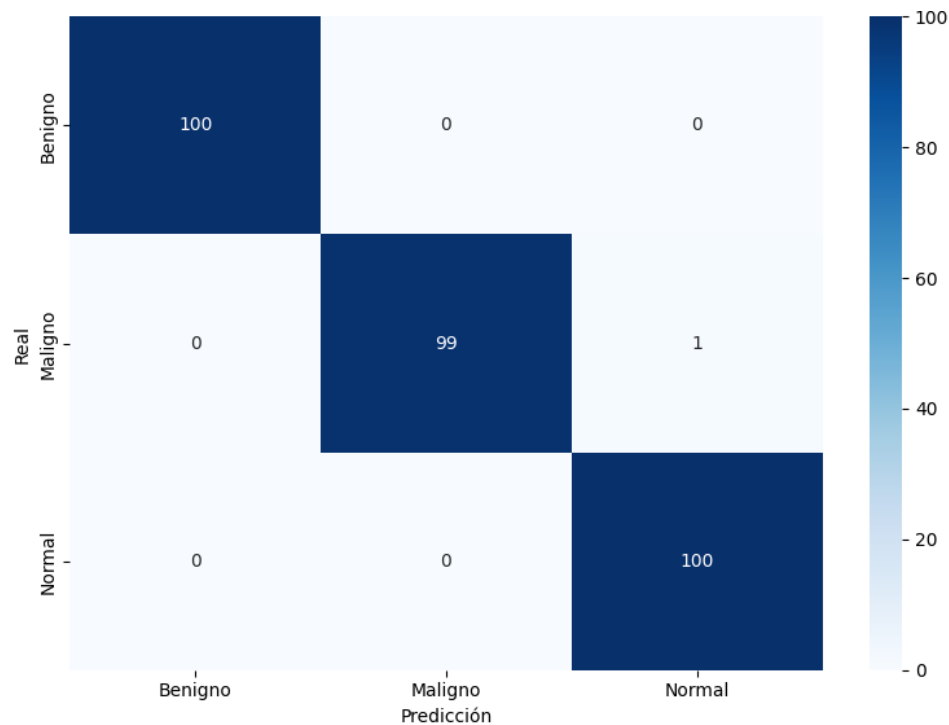


Figura 5.15 Matriz de confusión de Inception V3 caso A

A su vez, se realizó el análisis del tiempo que le toma al modelo final reconocer las imágenes, con el fin de determinar si el modelo será eficiente con un conjunto mayor de datos. Para resolver esto el dataset de validación se dividió en segmentos de 50 datos por prueba aumentando cada vez hasta integrar todos los datos de las 3 categorías. Esta

división se llevó a cabo de manera segmentada, si bien la selección de las imágenes no fue de manera aleatoria, se tomaron segmentos de cada categoría para cumplir la métrica de 50 imágenes por prueba. El resultado de esto puede ser apreciado en la Figura 5.16.

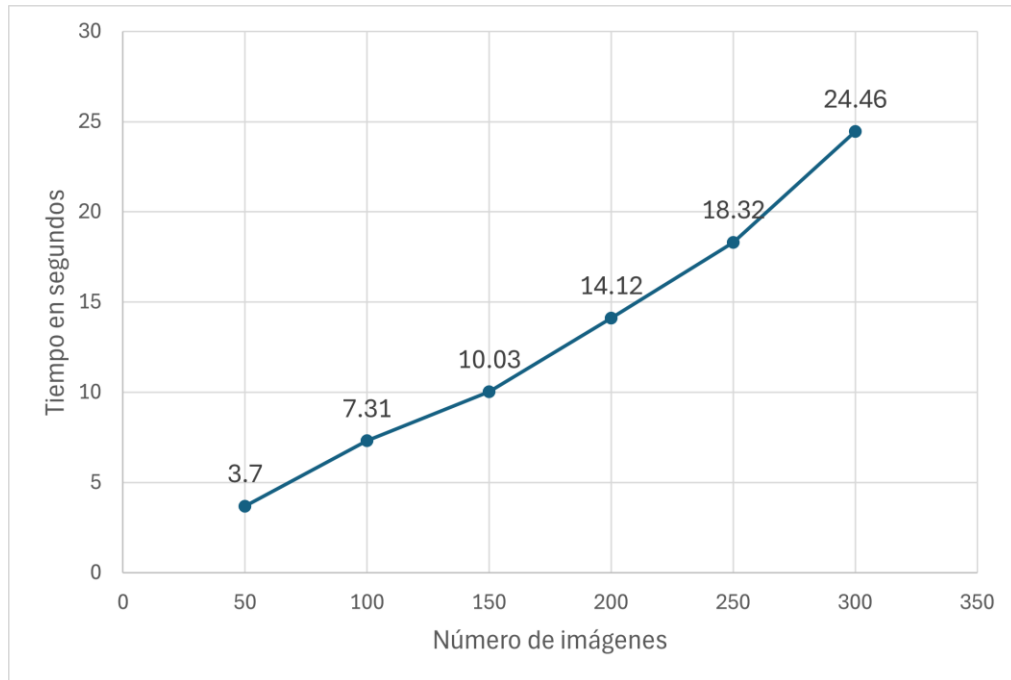


Figura 5.16 Gráfica sobre la complejidad temporal del modelo Inception V3

B. ResNet 50

La Figura 5.17 presenta los resultados de utilizar el mejor subconjunto del modelo *ResNet 50* para realizar predicciones. A diferencia del anterior modelo, los resultados de las predicciones suelen variar, pero en este caso solo se presentan en la clase “Normal”.

De la misma forma que con el modelo anterior, se evaluó la complejidad temporal del modelo *ResNet 50* siguiendo el mismo flujo de trabajo de segmentar bloques de imágenes de 50 en 50. Los datos graficados pueden ser apreciados en la Figura 5.18.

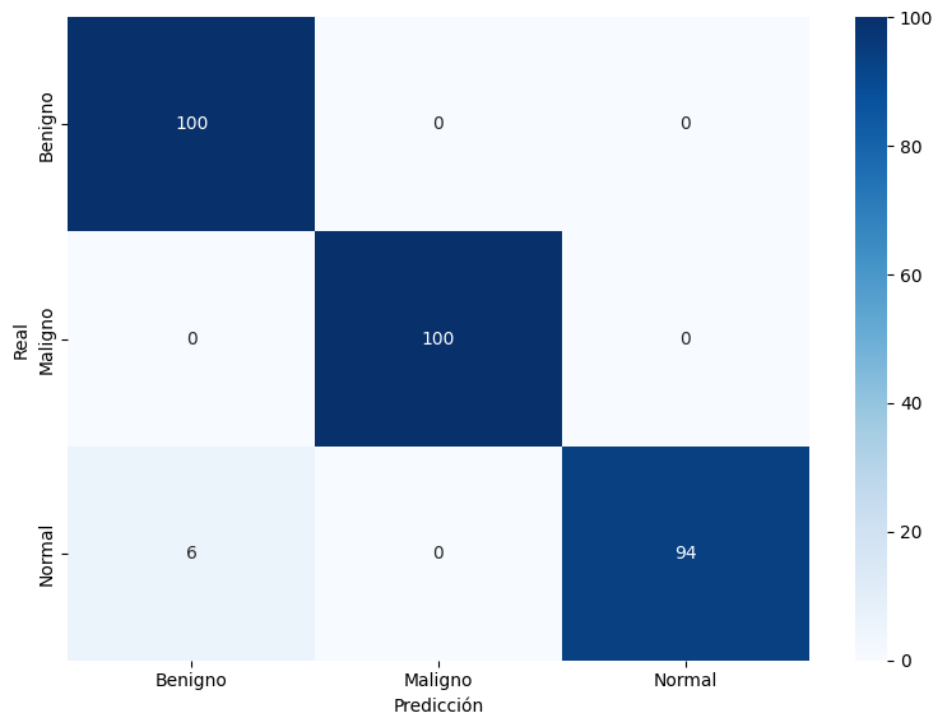


Figura 5.17 Matriz de confusión de ResNet 50 caso A

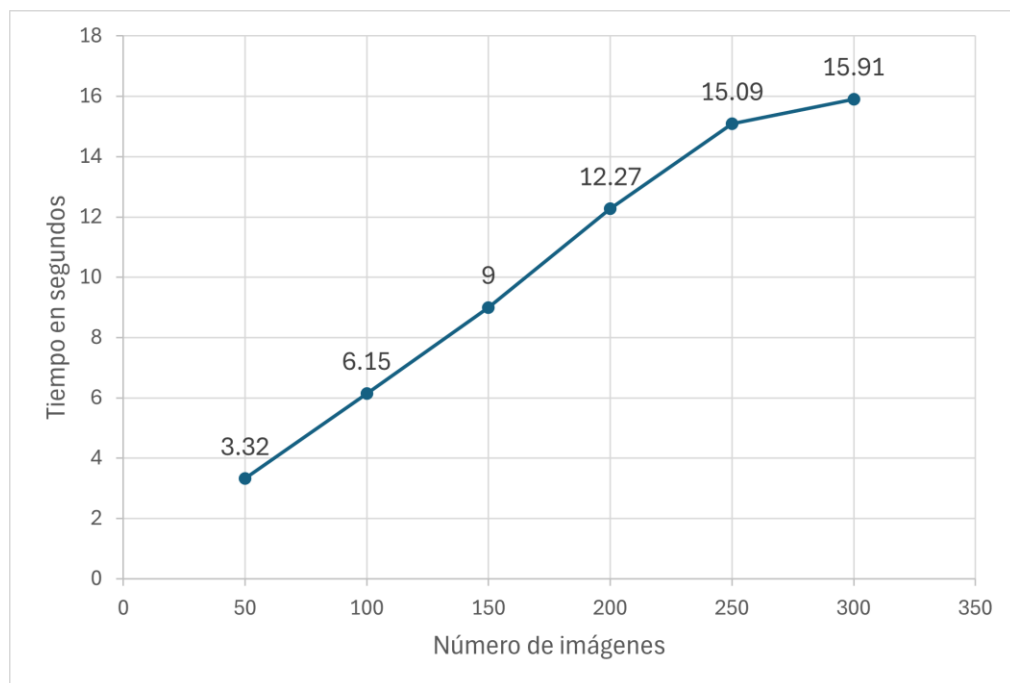


Figura 5.18 Gráfica sobre la complejidad temporal del modelo ResNet 50.

C. DenseNet 201

En la Figura 5.19 se visualizan los resultados de utilizar el mejor subconjunto del modelo *DenseNet 201* para realizar predicciones. Este modelo suele presentar variaciones en la clase “Benigno”, clasificando de 1 a 3 imágenes como normales.

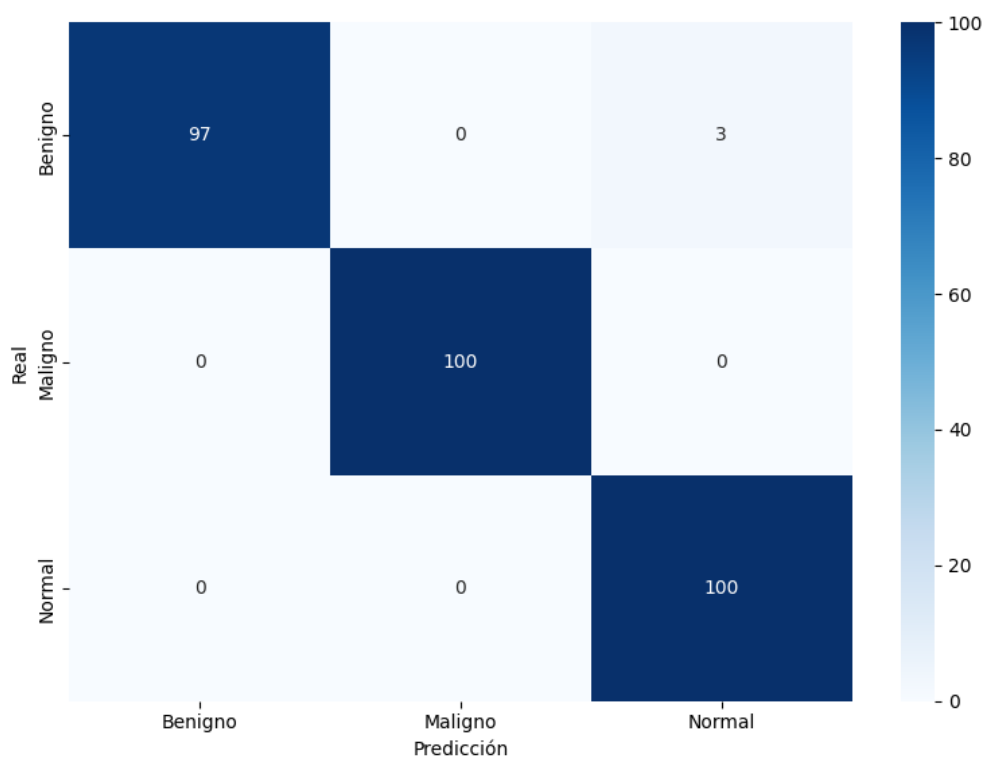


Figura 5.19 Matriz de confusión de DenseNet 201 caso A

De la misma forma que con los modelos anteriores, se evaluó la complejidad temporal del modelo *DenseNet 201* siguiendo el mismo flujo de trabajo de segmentar bloques de imágenes de 50 en 50. Los datos graficados pueden ser apreciados en la Figura 5.20.

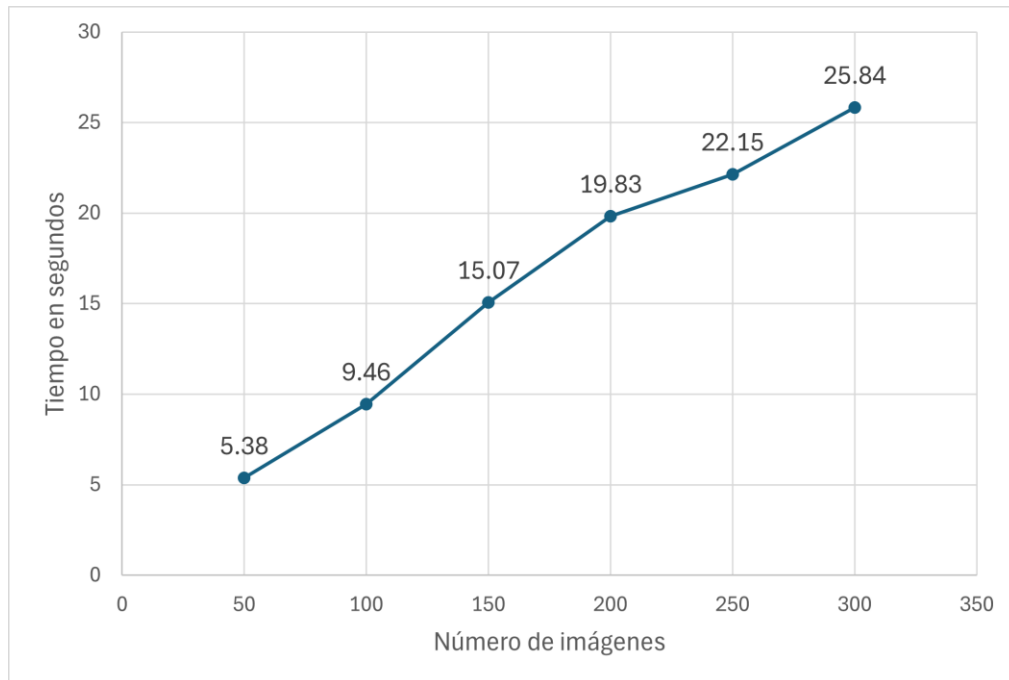


Figura 5.20 Gráfica sobre la complejidad temporal del modelo DenseNet 201

5.4.2 Caso B: dataset existente

A. Inception V3

En la Figura 5.21 se visualizan los resultados de utilizar el mejor subconjunto del modelo *Inception V3* para realizar predicciones, pero ahora utilizando los datos del dataset dos. En contraparte del primer caso en donde normalmente no existen grandes variaciones al clasificar, para este caso las predicciones tienden a variar al clasificar en las clases “Benigno” y “Normal”. El número de muestras en las que las predicciones suelen fallar ronda entre las 20-50, surgiendo más fallos en la clase “Benigno”.

B. ResNet 50

En la Figura 5.22 se visualizan los resultados de utilizar el mejor subconjunto del modelo *ResNet 50* para realizar predicciones, pero ahora utilizando los datos del dataset dos. Los resultados muestran que las variaciones surgen en las clases “Benigno” y “Maligno”, con tendencia a fallar más en la clase “Benigno”.

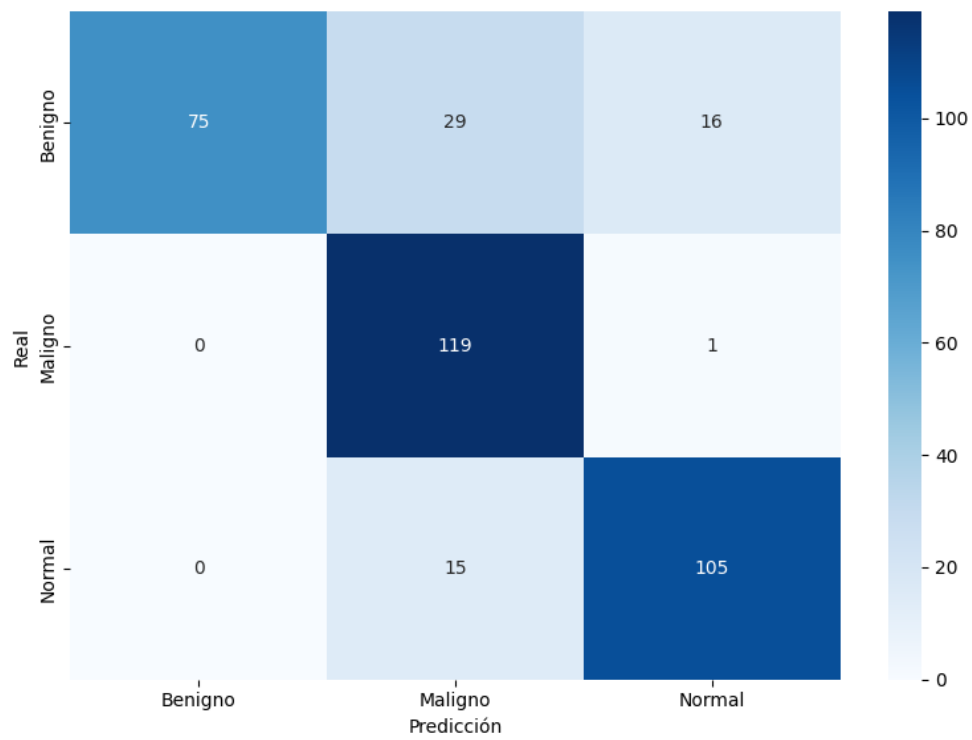


Figura 5.21 Matriz de confusión de Inception V3 caso B

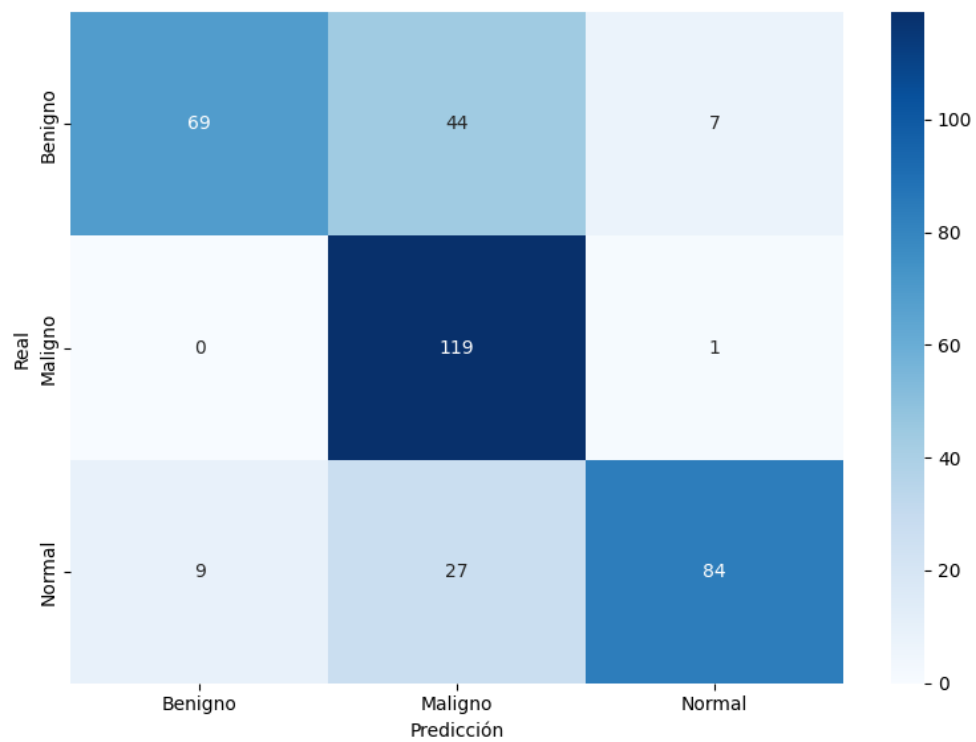


Figura 5.22 Matriz de confusión de ResNet 50 caso B

C. DenseNet 201

En la Figura 5.23 se visualizan los resultados de utilizar el mejor subconjunto del modelo *DenseNet 201* para realizar predicciones, pero ahora utilizando los datos del dataset dos. Los resultados muestran que las predicciones suelen fallar en las clases “Benigno” y “Normal”, siguiendo la tendencia de fallar en la clase “Benigno”.

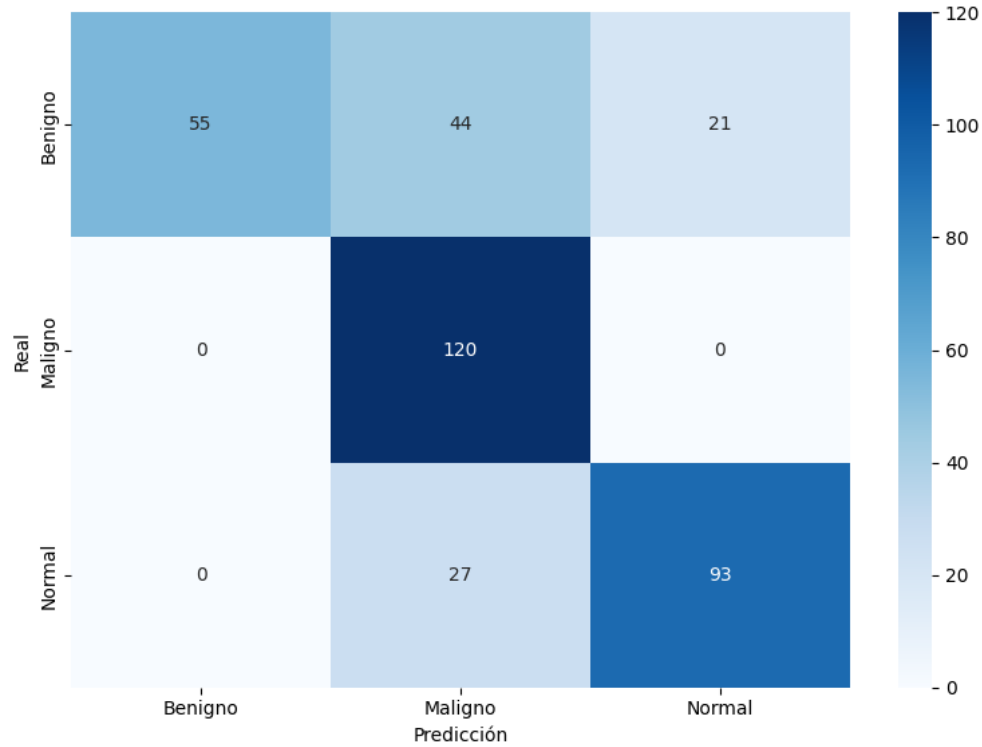


Figura 5.23 Matriz de confusión de DenseNet 201 caso B

5.5 Métricas finales por modelo

Una vez finalizado el proceso de entrenamiento y validación cruzada para cada uno de los modelos evaluados, se procede a la comparación de su desempeño mediante métricas de clasificación estandarizadas.

En la Tabla 5.1 se presentan los resultados finales obtenidos por las arquitecturas *DenseNet201*, *InceptionV3* y *ResNet50* en el Caso A (dataset balanceado con técnicas propias), incluyendo valores de precisión, sensibilidad, F1-Score, puntuación Kappa de Cohen y tiempo de clasificación. Estas métricas permiten evaluar la exactitud del modelo,

así como su capacidad para generalizar, su equilibrio entre verdaderos positivos y falsos negativos, y su eficiencia computacional, aspectos fundamentales para su posible aplicación en entornos clínicos.

Tabla 5.1 Métricas finales por modelo

Modelo	Precisión	Sensibilidad	F1-Score	Puntuación Kappa de Cohen	Tiempo de clasificación (segundos)
<i>DenseNet 201</i>	98.52%	98.66%	98.66%	97.78%	25.84
<i>Inception V3</i>	99.81%	99.66%	100%	99.72%	24.46
<i>ResNet 50</i>	99.82%	99.81%	99.81%	99.72%	15.91

La comparativa de los modelos CNN usados se basa en métricas de rendimiento que permiten cuantificar la precisión general del modelo. Adicionalmente, esta comparativa es útil para también analizar su capacidad para detectar correctamente casos positivos (sensibilidad), minimizar falsos positivos (especificidad), y equilibrar ambos aspectos (F1-Score y puntuación Kappa). Los resultados presentados en la Tabla 5.1 muestran un desempeño sobresaliente de los tres modelos evaluados, todos entrenados bajo las mismas condiciones de preprocesamiento, aumento de datos y validación cruzada estratificada, lo que garantiza una comparación justa y reproducible.

En primer lugar, *ResNet50* destaca con una precisión del 99.82%, seguido muy de cerca por *InceptionV3* con 99.81%, mientras que *DenseNet201* alcanza un 98.52%. Estos valores indican que ambos modelos clasifican correctamente más del 99% de las imágenes del conjunto de prueba, lo cual representa un nivel de confiabilidad extremadamente alto para una aplicación en entornos clínicos. Esta relativamente alta precisión sugiere que los modelos son capaces de generalizar adecuadamente a partir del conjunto de entrenamiento, incluso cuando se enfrentan a variaciones introducidas por técnicas de aumento de datos.

Al analizar la sensibilidad (*recall*), métrica crítica en diagnóstico médico donde el objetivo es minimizar los falsos negativos, se observa que *ResNet50* nuevamente obtiene

el mejor resultado con 99.81%, seguido por *InceptionV3* con 99.66% y *DenseNet201* con 98.66%. Este alto nivel de sensibilidad implica que los modelos son altamente efectivos para identificar casos de cáncer de pulmón (clase Maligno) y lesiones pulmonares, lo cual es fundamental para garantizar una detección temprana. En un contexto clínico, esto se traduce en una mayor probabilidad de que un nódulo maligno no pase desapercibido, reduciendo así el riesgo de diagnósticos tardíos.

El F1-Score, que combina precisión y sensibilidad en un solo indicador, refuerza lo anterior. *InceptionV3* alcanza un F1-Score del 100%, lo cual es un resultado excepcional, aunque debe interpretarse con cautela: este valor perfecto puede deberse a una combinación óptima entre ambas métricas en el conjunto de prueba, pero también podría indicar una ligera sobreestimación si el dataset de prueba no es lo suficientemente amplio o diverso. *ResNet50* obtiene un F1-Score del 99.81%, muy cercano al máximo, mientras que *DenseNet201* registra un 98.66%, lo que indica un rendimiento bueno, aunque ligeramente inferior en el equilibrio entre precisión y sensibilidad.

Por otra parte, se cuantificó la puntuación Kappa de Cohen, que mide el acuerdo entre las predicciones del modelo y las etiquetas reales, ajustando por el azar. Ambos modelos, *ResNet50* e *InceptionV3*, obtienen una puntuación Kappa de 99.72%, lo que refleja un acuerdo casi perfecto entre las predicciones y las etiquetas verdaderas. Esto es que el modelo acierta con frecuencia, pero además lo hace de forma consistente y significativa desde el punto de vista estadístico. Por su parte, *DenseNet201* alcanza un 97.78%, lo cual sigue siendo un valor muy alto, pero ligeramente menor, lo que podría indicar una mayor variabilidad en la clasificación de ciertas clases, como se observa en correspondencia con las matrices de confusión.

Además de las métricas de rendimiento, se consideró el tiempo de clasificación. En este aspecto, *ResNet50* sobresale con un tiempo promedio de 15.91 segundos para el conjunto de prueba, lo que lo convierte en el modelo más eficiente computacionalmente. *InceptionV3* requiere 24.46 segundos, y *DenseNet201* 25.84 segundos. Esta diferencia es significativa en un entorno donde el tiempo de respuesta puede influir en la toma de decisiones. La mayor eficiencia de *ResNet50* puede atribuirse a su arquitectura más ligera en comparación con las otras dos, a pesar de que todas operan con tamaños de imagen

similares (224x224 píxeles para *ResNet50* y *DenseNet201*, y 299x299 para *InceptionV3*), lo cual también influiría en la carga computacional.

Es importante destacar que estos resultados se obtuvieron en el Caso A, es decir, con el dataset balanceado mediante las técnicas propuestas en este trabajo, lo que permite ver el impacto positivo de estas estrategias. En comparación con los resultados del Caso B (dataset aumentado por un tercero), se observa una disminución en el rendimiento, puesto que no todas las técnicas de aumento de datos son igualmente efectivas.

En conjunto, los resultados indican que *ResNet50* es el modelo que ofrece el mejor equilibrio entre precisión, sensibilidad, eficiencia computacional y consistencia estadística, lo que lo posiciona como la mejor opción para su integración en sistemas de apoyo al diagnóstico clínico. Si bien *InceptionV3* muestra un F1-Score perfecto, su mayor tiempo de inferencia y un desempeño ligeramente inferior en sensibilidad lo hacen menos favorable en escenarios donde la rapidez y la detección de casos críticos son prioritarias.

Este análisis permite seleccionar el modelo más adecuado, y al mismo tiempo valida la metodología propuesta: el uso de técnicas de aumento de datos, la validación cruzada estratificada y la evaluación multivariada de métricas clínicamente relevantes.

5.6 Comparación con otros trabajos

Con el fin de evaluar el desempeño del modelo seleccionado dentro del estado del arte y validar su eficacia frente a implementaciones previas, se realiza una comparación con trabajos recientes que han utilizado arquitecturas similares en tareas de clasificación de imágenes médicas. En la Tabla 5.2 se presentan los resultados de investigaciones independientes que emplearon el modelo *ResNet50*, que fue el que reportó el mejor desempeño en este estudio, aplicado a diferentes conjuntos de datos y problemáticas relacionadas con el diagnóstico por imágenes.

Esta comparación permite evaluar la competitividad de los resultados obtenidos en este trabajo, considerando métricas como precisión y sensibilidad. A diferencia de otros estudios, el enfoque propuesto en esta investigación incorpora un conjunto de técnicas de aumento de datos y validación cruzada, lo que podría explicar el superior desempeño.

Tabla 5.2 Comparación de los resultados obtenidos con respecto a otros trabajos

Autor	Modelo	Precisión	Sensibilidad
Este trabajo	<i>ResNet 50</i>	98.52%	98.66%
(Martínez Tarazona, 2022)	<i>ResNet 50</i>	97.83%	96.80%
(Salinas González, 2021)	<i>ResNet 50</i>	87.85%	No reportado

La comparación entre el desempeño del modelo *ResNet50* obtenido en este trabajo y los resultados reportados por otros investigadores que han utilizado la misma arquitectura en contextos similares de clasificación de imágenes médicas se discute enseguida.

Como se observa en la tabla, el modelo *ResNet50* entrenado en este trabajo, utilizando el *Caso A* (dataset balanceado con técnicas avanzadas de aumento de datos), alcanza una precisión del 99.82% y una sensibilidad del 99.81%, valores que superan significativamente los reportados por otros autores. Por ejemplo, (Martínez Tarazona, 2022) obtiene una precisión del 97.83% y una sensibilidad del 96.8%, mientras que (Salinas González, 2021) reporta una precisión aún menor, del 87.85%, sin especificar la sensibilidad.

La superioridad de los resultados obtenidos en este trabajo puede atribuirse a varios factores clave del diseño metodológico. En primer lugar, el uso de técnicas diversificadas de aumento de datos permite al modelo aprender a generalizar mejor frente a variaciones reales en la calidad de las imágenes, el ruido, la iluminación y la posición del paciente, lo cual mejora notablemente al modelo. A diferencia de muchos estudios que aplican aumentos básicos (rotaciones, volteos), este trabajo incorpora transformaciones más complejas, lo que impacta directamente en la capacidad del modelo para distinguir entre clases con alta precisión.

En segundo lugar, el balanceo equitativo del dataset mediante aumento de datos asegura que ninguna clase domine el entrenamiento, lo que previene sesgos y mejora el rendimiento en clases minoritarias, como lo es el caso de la clase *Maligno*. En contraste con otros estudios que trabajan con datos desbalanceados o que no especifican cómo abordan este problema, lo que puede afectar negativamente la sensibilidad del modelo.

Además, la implementación de validación cruzada estratificada con $k=5$ folds promueve que se realice una evaluación más confiable y reproducible del modelo, reduciendo el riesgo de sobreajuste y favoreciendo que el rendimiento no dependa de una partición aleatoria específica.

Otro aspecto para considerar es la calidad del preprocesamiento. En este estudio, se realizó una normalización de los valores de píxeles, ajuste de tamaño a 224x224, y se utilizó *transfer learning* con pesos preentrenados, optimizando con esto la extracción de características. Estas decisiones técnicas, sumadas al uso de métricas como sensibilidad y F1-Score, robustecen al modelo.

Es importante destacar también que, aunque algunos trabajos reportan métricas altas, en ocasiones se carece de transparencia sobre el tamaño del dataset, el grado de desbalanceo, el tipo de aumento de datos o la independencia del conjunto de prueba. Esto puede llevar a resultados sobre estimados, o al menos, que no se pueden reproducir ante otras condiciones, aun con cambios ligeros.

Finalmente, es importante considerar que la comparación presentada en la Tabla 5.2 demuestra que el modelo de *ResNet50* desarrollado en este trabajo es competitivo, al superar significativamente el desempeño de implementaciones previas en escenarios similares. Esto valida la hipótesis de que un preprocesamiento adecuado, combinado con técnicas de aumento de datos, puede mejorar sustancialmente el rendimiento de los modelos de inteligencia artificial.

CONCLUSIONES

El avance en la tecnología es cada vez mayor, por ello la propuesta de soluciones que automaticen procesos son cada vez más relevantes. En ámbitos como lo son la medicina, el uso de herramientas como la inteligencia artificial es cada vez más relevante, al proporcionar soluciones rápidas y efectivas, hacen posible que los tratamientos en enfermedades en los que la detección temprana son de suma importancia eleven en gran medida la efectividad y por ende la calidad de vida de las personas mejore con el paso del tiempo.

En este trabajo se utilizaron técnicas de IA, pero se tiene además la complementación de esta tecnología con otras ramas como lo son el procesamiento de imágenes digitales hace posible la mejora en la obtención de resultados, al tener la disponibilidad de mejorar los datos con los que los modelos de IA trabajan esto se traduce en mejores resultados al momento de comenzar con pruebas.

Al implementar diferentes disciplinas en el desarrollo de la solución a la problemática se asegura el cumplimiento de los objetivos propuestos al inicio del trabajo. El primero de ellos, “investigación de diferentes modelos de CNN para clasificar tomografías pulmonares” fue cumplido al desarrollar el estado del arte pues con la investigación de distintos trabajos se identificaron modelos clave en la clasificación de imágenes.

El segundo de ellos “implementar los modelos seleccionados utilizando un dataset publico previamente procesado” conllevó a diferentes procesos para su cumplimiento, primeramente, se investigaron diferentes repositorios populares para extraer un primer dataset lo suficientemente extenso para poder cubrir con las necesidades del proyecto, a este mismo posteriormente se le realizó un primer filtro para corroborar el estado en el que se encontraba, y con base en este primer filtro se tomaron acciones para trabajarlo y dejarlo en condiciones adecuadas para su posterior utilización.

El tercero y cuarto objetivos fueron desarrollados en sincronía, pues para evaluar el impacto que tenía aumentar los datos del dataset con técnicas de aumento de datos era necesario comparar las métricas de rendimiento que se obtenían en cada modelo después

de entrenarlos. Estas métricas son clave para el cumplimiento de estos dos objetivos pues con base en ellas se determinaría si es que en verdad existe una gran diferencia y por ende importancia al preprocesar los datos en comparación con solo entrenar el modelo sin realizar algún cambio extra a los ajustes del modelo en sí.

El ultimo objetivo que comprende “Determinar qué modelo y condición de preprocesamiento ofrecen la mejor capacidad de clasificación en la detección de cáncer de pulmón” se cumplió de manera similar a los anteriores, pues con la ayuda de las métricas de rendimiento fue posible detectar el mejor comportamiento tanto de los datos como de los modelos y destacar sobre qué condiciones era mejor trabajar la propuesta de la investigación realizada.

Estas acciones en conjunto conllevaron a la comprobación del objetivo general “Evaluar el desempeño de distintos modelos de redes neuronales convolucionales para la detección de cáncer de pulmón en tomografías, comparando su precisión, sensibilidad y especificidad en diferentes condiciones de entrenamiento y preprocesamiento de datos”, por lo que se determinó un flujo de trabajo adecuado para esta evaluación, permitiendo además alcanzar valores muy satisfactorios en las métricas propuestas.

REFERENCIAS

- Albumentations. (2025). *Albumentations Documentation*. Recuperado el 05 de Marzo de 2025, de Choosing Augmentations for Model Generalization: https://albumentations.ai/docs/api_reference/augmentations/geometric/transforms
- Al-Yasriy, H. F. (24 de Mayo de 2020). *The IQ-OTH/NCCD lung cancer dataset*. Obtenido de Kaggle: <https://www.kaggle.com/datasets/hamdallak/the-iqothnccd-lung-cancer-dataset>
- Artola Moreno, Á. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python*. Sevilla, España: Trabajo de fin de grado en Ingeniería de las Tecnologías de Telecomunicación, Universidad de Sevilla. Recuperado el 06 de Marzo de 2025, de <https://idus.us.es/server/api/core/bitstreams/c22e9088-218a-4554-9b27-6c0ebf89d0e8/content>
- AWS Amazon. (2025). *Machine Learning*. Recuperado el 05 de Marzo de 2025, de ¿Cuál es la diferencia entre el aprendizaje profundo y las redes neuronales?: <https://aws.amazon.com/es/compare/the-difference-between-deep-learning-and-neural-networks/>
- Cabo García, A., del Campo Mulet, E., Rubio González, T., Nápoles Smith, N., & Columbie Reguifero, J. C. (2018). Aspectos clínicos y epidemiológicos en pacientes con cáncer de pulmón en un servicio de neumología. *MEDISAN*, 22(4), 12.
- Cambridge in Colour. (2025). *Cambridge in Colour, A Learning Community for Photographers*. Recuperado el 05 de Marzo de 2025, de Sharpening: unsharp mask: <https://www.cambridgeincolour.com/tutorials/unsharp-mask.htm>
- Catalán Urzúa, A. (2019). Técnicas de procesamiento digital de imágenes. *Revista de Marina*, 136(969). Recuperado el 04 de Marzo de 2025, de Técnicas de procesamiento digital de imágenes: <https://revistamarina.cl/es/articulo/tecnicas-de-procesamiento-digital-de-imagenes>

- Codificando Bits. (2024). *Codificando Bits*. Recuperado el 06 de Marzo de 2025, de Introducción a las Redes Neuronales Recurrentes: <https://codificandobits.com/blog/introduccion-redes-neuronales-recurrentes/>
- Engelbrecht, A. P. (2007). *Computational Intelligence: An Introduction* (2nd ed.). John Wiley & Sons.
- Equipo del NCI. (22 de Abril de 2022). ¿Sirve la inteligencia artificial para ver el cáncer de formas nuevas y más eficaces? *NIH - Instituto Nacional del Cancer*. Obtenido de <https://www.cancer.gov/espanol/noticias/temas-y-relatos-blog/2022/inteligencia-artificial-imagenes-cancer>
- Espinosa, E. G. (Octubre de 2024). *Detección de Lesiones en la Piel Usando Visión Artificial*. Toluca, México: Tesis de la Universidad Autónoma del Estado de México. Obtenido de <https://ri.uaemex.mx/handle/20.500.11799/141833>
- García Espinosa, E., Sergio Ruiz Castilla, J., & García Lamont, F. (18 de Marzo de 2024). Comparación de algoritmos de visión artificial en la clasificación de enfermedades de la piel utilizando redes neuronales. *Repositorio Institucional UAEM*, pág. 12.
- Gómez Herrera, R. (2 de Marzo de 2025). ¿cómoves? *Divulgación de la Ciencia - UNAM*. Recuperado el 25 de Febrero de 2025, de La inteligencia artificial ¿Hacia dónde nos lleva?: <https://www.comoves.unam.mx/numeros/articulo/2/la-inteligencia-artificial-hacia-donde-nos-lleva>
- Gómez Macedo, M. (2022). Detección de covid-19 y neumonía en imágenes rayos x de pulmones utilizando redes neuronales convolucionales. *Repositorio de Tesis DGBSDI*, pág. 71.
- González Barrio, H., Calleja Ochoa, A., Gómez-Escudero, G., Rodríguez Ezquerro, A., & López de Lacalle Marcaide, L. (6 de Abril de 2021). *Interempresas Metalmecánica*. Recuperado el 05 de Marzo de 2025, de Los conceptos de Machine Learning y Deep Learning en la industria: <https://www.interempresas.net/MetalMecanica/Articulos/347471-Los-conceptos-de-Machine-Learning-y-Deep-Learning-en-la-industria.html>

- Gonzalo, Á. (26 de Mayo de 2018). *Machine Learning para todos*. Recuperado el 05 de Marzo de 2025, de Tipos de aprendizaje automático:
<https://machinelearningparatodos.com/tipos-de-aprendizaje-automatiko/>
- Google Cloud. (2025). *Google Cloud*. Recuperado el 05 de Marzo de 2025, de ¿Qué es el aprendizaje automático (AA)? : <https://cloud.google.com/learn/what-is-machine-learning?hl=es-419>
- Grupo Integrado de Ingeniería. (30 de 04 de 2025). *Inteligencia Computacional*.
 Obtenido de <https://gii.udc.es/investigacion/detalle/3>
- Instituto Nacional de Salud Publica. (5 de Abril de 2021). *Página oficial del Instituto Nacional de Salud Publica, Gobierno de México*. Recuperado el 04 de Marzo de 2025, de México frente al cáncer de pulmón:
<https://www.insp.mx/avisos/mexico-frente-al-cancer-de-pulmon#sup2>
- Kandel, I., & Castelli, M. (2020, March 16). Transfer Learning with Convolutional Neural Networks for Diabetic Retinopathy Image Classification. A Review. *Applied Sciences*, 2021(10). doi:<https://doi.org/10.3390/app10062021>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.
- Martin, J. (3 de Enero de 2022). *digitalbiz magazine, Mathworks*. Recuperado el 05 de Marzo de 2025, de Flujo de trabajo de la inteligencia artificial:
<https://www.digitalbizmagazine.com/workflow-de-la-inteligencia-artificial/>
- Martinez Tarazona, S. A. (2022). *Prototipo de herramienta de software con aprendizaje de máquinas para la diferenciación de neumonía bacteriana o viral con el uso de radiografías de tórax*. Obtenido de Repositorio Institucional UNAB:
<http://hdl.handle.net/20.500.12749/16509>
- MathWorks. (2023). *MathWorks Discovery*. Recuperado el 06 de Marzo de 2025, de Introducción a la convolución:
<https://la.mathworks.com/discovery/convolution.html>

- MathWorks. (2023). *Procesado de imágenes digitales*. Recuperado el 04 de Marzo de 2025, de Introducción al procesamiento digital de señales:
<https://la.mathworks.com/discovery/digital-image-processing.html>
- Mayorga, L. (24 de Julio de 2019). *UNAM Global - Revista*. Recuperado el 05 de Marzo de 2025, de ¿Qué es el aprendizaje automático y cómo funciona?:
https://unamglobal.unam.mx/global_revista/que-es-el-aprendizaje-automatico-y-como-funciona/
- Microsoft. (2024). *Microsoft Learn*. Recuperado el 05 de Marzo de 2025, de
<https://learn.microsoft.com/es-es/azure/machine-learning/concept-deep-learning-vs-machine-learning?view=azureml-api-2>
- Oracle. (25 de Noviembre de 2024). *Oracle Cloud Infrastructure (OCI)*. Recuperado el 19 de Febrero de 2025, de ¿Qué es el machine learning?:
<https://www.oracle.com/mx/artificial-intelligence/machine-learning/what-is-machine-learning/>
- Remon, J. (2 de Enero de 2023). *Cáncer de pulmón*. Recuperado el 04 de Marzo de 2025, de Sociedad Española de Oncología Médica: <https://seom.org/info-sobre-el-cancer/cancer-de-pulmon?showall=1&start=0>
- Ruiz Castilla, J. S., Rangel Cortes, J. J., García Lamont, F., & Cervantes Canales, J. (13 de Junio de 2019). Redes Neuronales Convolucionales en la identificación de melanomas benignos y malignos. *Risti*, págs. 15-27.
- Salinas González, R. (2021). *Aprendizaje profundo para la clasificación de imágenes de plantas de albahaca con diferencias de nitrógeno durante su cultivo, utilizando la técnica de entrenamiento curriculum-by-smoothing*. Obtenido de Repositorio Institucional de Acceso Abierto de la Universidad Autónoma del Estado de Morelos: <http://riaa.uaem.mx/handle/20.500.12055/2138>
- Sarmiento-Ramos, J. L. (2020). Aplicaciones de las redes neuronales y el deep learning a la ingeniería biomédica. *Revista UIS Ingenierías*, 19(4), 1-8.
doi:<https://doi.org/10.18273/revuin.v19n4-2020001>

Striker, C. (4 de Octubre de 2024). *IBM*. Recuperado el 06 de Marzo de 2025, de ¿Qué es una RNN (red neuronal recurrente)?: <https://www.ibm.com/mx-es/think/topics/recurrent-neural-networks>

Venegas Flores, D. C. (Marzo de 2019). Sistema de Pre-Diagnóstico de Cáncer de Melanomas Usando Redes Neuronales Artificiales. *Repositorio Institucional UAEM*, pág. 88.