
Detección de Neumonía en Radiografías de Tórax Utilizando una Red Neuronal Convolutiva

Johan Riascos Hurtado¹, Victor Alexis Largo Guarumo¹ y Juan Camilo Giraldo Calle¹

¹ *Machine Learning, Ingeniería Física, Facultad de Ingenierías,
Universidad Tecnológica de Pereira, Risaralda, Colombia*

E-mail: johan.riascos@utp.edu.co, victor.largo1@utp.edu.co, juancamilo.giraldo@utp.edu.co

Resumen

En este estudio, desarrollamos y evaluamos un modelo de red neuronal convolutiva (CNN) para la detección de neumonía en radiografías de tórax. Utilizando un conjunto de datos compuesto por imágenes normales y con neumonía, preprocesamos las imágenes para normalizar y aumentar los datos de entrenamiento. Nuestra CNN incluye cinco bloques convolucionales con capas de normalización y dropout para prevenir el sobreajuste. La precisión del modelo fue evaluada utilizando una matriz de confusión, obteniendo una precisión del 90 % en el conjunto de prueba. Además, se analizaron las métricas de precisión, recall y la curva ROC para evaluar la efectividad del modelo. Los resultados demuestran la capacidad de la CNN para identificar correctamente las imágenes de neumonía con alta precisión, proporcionando una herramienta útil para la asistencia médica en el diagnóstico de esta enfermedad.

Key words. Red Neuronal Convolutiva (CNN), Neumonía, Radiografías de Tórax, Clasificación de Imágenes Médicas, Procesamiento de Imágenes, Deep Learning, Normalización de Datos, Dropout, Curva ROC, Precisión-Recall

1. Introducción

La neumonía es una infección respiratoria aguda que afecta a los pulmones y puede ser potencialmente mortal, especialmente en niños pequeños, ancianos y personas con sistemas inmunitarios debilitados. La detección temprana y precisa de la neumonía es crucial para el tratamiento efectivo y la mejora de las tasas de recuperación. Las radiografías de tórax son una herramienta diagnóstica comúnmente utilizada para identificar la presencia de neumonía. Sin embargo, la interpretación de estas imágenes puede ser desafiante y subjetiva, dependiendo en gran medida de la experiencia del radiólogo.

En los últimos años, los avances en el aprendizaje profundo, particularmente en las redes neuronales convolucionales (CNN), han demostrado un potencial significativo para mejorar la precisión y la eficiencia del diagnóstico de imágenes médicas. Las CNN son especialmente adecuadas para tareas

de clasificación de imágenes debido a su capacidad para aprender características jerárquicas y complejas directamente a partir de los datos de imagen sin la necesidad de ingeniería manual de características.

Este estudio se centra en el desarrollo y la evaluación de un modelo de CNN para la detección automática de neumonía en radiografías de tórax. Utilizando un conjunto de datos que incluye imágenes de tórax normales y con neumonía, implementamos un pipeline de preprocesamiento que incluye normalización y aumento de datos para mejorar la robustez del modelo. Nuestra CNN consta de varios bloques convolucionales, seguidos de capas de normalización y dropout para prevenir el sobreajuste y mejorar la generalización del modelo.

El propósito de este trabajo es explorar la efectividad de las CNN en la detección de neumonía y proporcionar una herramienta automatizada que

pueda asistir a los profesionales de la salud en el diagnóstico preciso y rápido de esta enfermedad. A través de la evaluación de diversas métricas de rendimiento, como la matriz de confusión, la precisión, el recall y las curvas ROC, demostramos el potencial de nuestro enfoque basado en CNN para ser implementado en entornos clínicos y contribuir a la mejora del cuidado de la salud.

2. Marco Teórico

2.1. Neumonía

La neumonía es una infección respiratoria aguda que afecta los sacos de aire en uno o ambos pulmones, que pueden llenarse de líquido o pus, causando síntomas como tos con flema o pus, fiebre, escalofríos y dificultad para respirar. La neumonía puede ser causada por diversos agentes infecciosos, incluyendo bacterias, virus y hongos. Dependiendo del tipo de patógeno, la gravedad de la enfermedad y el tratamiento requerido pueden variar significativamente.

- Neumonía bacteriana: Generalmente causada por la bacteria *Streptococcus pneumoniae*, aunque otras bacterias también pueden ser responsables.
- Neumonía viral: Comúnmente causada por virus respiratorios como el virus de la gripe (influenza), el virus respiratorio sincitial (VRS) y, más recientemente, el SARS-CoV-2.
- Neumonía por hongos: Menos común, puede ocurrir en personas con sistemas inmunitarios debilitados y es causada por hongos como *Pneumocystis jirovecii*.

La detección temprana de la neumonía es crucial para iniciar el tratamiento adecuado y mejorar las tasas de recuperación. En los niños, ancianos y personas con enfermedades crónicas o sistemas inmunitarios debilitados, el diagnóstico y tratamiento oportunos pueden prevenir complicaciones graves e incluso la muerte.

Dentro de los métodos tradicionales para la detección y la técnica con la que se obtuvieron las imágenes usadas en el estudio son las **Radiografías de tórax**: Las radiografías son la herramienta diagnóstica principal para confirmar la presencia de neumonía. Permiten a los médicos observar la inflamación en los pulmones y otras características típicas de la infección.

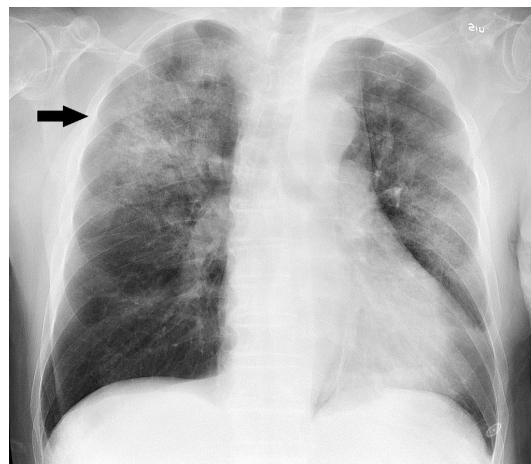


Figura 1: Radiografía de tórax de una neumonía causada por influenza y *Haemophilus influenzae*, con consolidaciones parcheadas, principalmente en el lóbulo superior derecho.

2.2. Redes Neuronales Convolucionales (CNN)

Son una clase de redes neuronales profundas especialmente diseñadas para procesar datos con una estructura de cuadrícula, como las imágenes. A diferencia de las redes neuronales tradicionales, las CNN pueden capturar patrones espaciales y temporales en los datos, lo que las hace extremadamente eficaces para tareas de visión por computadora. Una CNN está compuesta por múltiples capas que transforman una entrada de imagen en una salida de predicción a través de una serie de operaciones convolucionales y otras transformaciones. Los componentes básicos de una CNN incluyen capas convolucionales, capas de pooling, capas de normalización, y capas completamente conectadas. Los componentes principales de una CNN son los siguientes.

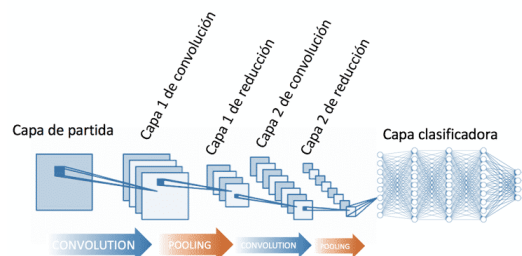


Figura 2: Arquitectura de una CNN.

2.2.1. Capas Convolucionales

- **Filtros (Kernels):** Los filtros son matrices pequeñas que se aplican a la imagen de entrada para detectar características como bordes, texturas y patrones. Cada filtro genera un mapa de características (feature map).
- **Operaciones de Convolución:** La operación de convolución consiste en mover el filtro sobre la imagen de entrada y calcular un producto punto entre el filtro y una porción de la imagen, produciendo un mapa de características.
- **Activaciones:** Después de la convolución, se aplica una función de activación (como ReLU) para introducir no linealidades en el modelo.

2.2.2. Capas de Pooling

- **MaxPooling:** Selecciona el valor máximo en cada ventana de la imagen, reduciendo la dimensionalidad y la cantidad de parámetros, lo que ayuda a controlar el sobre ajuste.
- **AveragePooling:** Calcula el promedio de los valores en cada ventana.

2.2.3. Capas de Normalización

- **Batch Normalization:** Normaliza las activaciones de la capa anterior para mejorar la estabilidad y la velocidad del entrenamiento.

2.2.4. Capas de Regularización

- **Dropout:** Apaga aleatoriamente una fracción de las neuronas durante el entrenamiento para prevenir el sobreajuste.

2.2.5. Capas de Regularización

Conectan todas las neuronas de una capa a todas las neuronas de la siguiente capa, transformando linealmente el input en el output. Suelen estar presentes en las últimas capas de la red para combinar las características extraídas y hacer la predicción final.

En cuanto al **Entrenamiento y Optimización**. El entrenamiento de una CNN implica ajustar los pesos de los filtros y las capas completamente conectadas para minimizar una función de pérdida (loss function), típicamente usando un algoritmo de optimización como Adam o SGD (Stochastic Gradient Descent).

- **Función de Pérdida:** En problemas de clasificación binaria, la función de pérdida comúnmente

utilizada es la entropía cruzada binaria (binary cross-entropy).

- **Algoritmos de Optimización:** Los optimizadores como Adam ajustan los pesos de la red en función del gradiente de la función de pérdida.
- **Técnicas para Prevenir el Sobreajuste:**
 - **Dropout:** Apaga aleatoriamente neuronas durante el entrenamiento.
 - **Batch Normalization:** Ayuda a estabilizar y acelerar el entrenamiento.
 - **Aumentación de Datos:** Genera nuevas muestras de datos a partir de las existentes mediante transformaciones como rotaciones, traslaciones y flips.

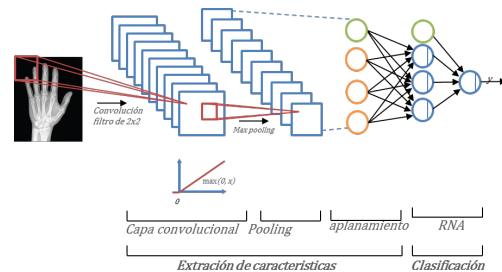


Figura 3: Vistazo general del funcionamiento de una CNN.

2.3. Aumentación y Preprocesamiento de Imágenes

Son técnicas clave que se utilizan para mejorar la eficiencia y la robustez del modelo.

2.3.1. Preprocesamiento de Imágenes

El preprocesamiento de imágenes implica una serie de pasos que transforman las imágenes en su formato bruto a un formato adecuado para la entrada a una CNN. Estos pasos son esenciales para:

- **Normalización:** Asegura que los valores de los píxeles de las imágenes estén dentro de un rango específico, generalmente $[0, 1]$, lo que facilita el proceso de entrenamiento.
- **Reducción de Variabilidad:** Minimiza la variabilidad innecesaria en las imágenes, permitiendo que el modelo se concentre en las características más relevantes.
- **Mejora de la Convergencia:** Mejora la velocidad y la estabilidad del entrenamiento del modelo.

Para llevar a cabo un preprocesamiento adecuado es muy importante tener en cuenta los siguientes pasos.

- **Redimensionamiento:** Las imágenes deben ser redimensionadas a un tamaño consistente que coincida con el tamaño de entrada esperado por la CNN. En nuestro caso, las imágenes se redimensionan a 150x150 píxeles.
- **Escalado:** Las imágenes se escalan para que los valores de los píxeles estén en el rango de $[0, 1]$. Esto se realiza dividiendo cada valor de píxel por 255.
- **Conversión a Tres Canales:** Si las imágenes son en escala de grises, se pueden convertir a un formato RGB de tres canales para que sean compatibles con modelos que esperan tres canales de entrada.

2.3.2. Aumentación de Imágenes

La aumentación de imágenes es una técnica que se utiliza para aumentar la cantidad y la diversidad de los datos de entrenamiento sin necesidad de recopilar nuevas imágenes. Esto se logra aplicando una serie de transformaciones aleatorias a las imágenes de entrenamiento existentes. La aumentación ayuda a prevenir el sobreajuste y mejora la capacidad del modelo para generalizar a datos no vistos. Dentro de las técnicas comunes de aumentación de imágenes tenemos las siguientes.

- **Reescalado (Rescaling):** Normaliza los valores de los píxeles. Por ejemplo, el reescalado por $1./255$.
- **Shear Range:** Aplica un cambio angular a la imagen, como una distorsión en el plano, manteniendo los bordes paralelos.
- **Zoom Range:** Aplica un zoom aleatorio a la imagen, permitiendo que el modelo sea robusto a diferentes escalas de objetos.
- **Flip Horizontal:** Invierte las imágenes horizontalmente, lo cual es especialmente útil en tareas donde la orientación horizontal de los objetos no es relevante.
- **Rotaciones Aleatorias:** Aplica rotaciones aleatorias dentro de un rango especificado.
- **Desplazamientos Aleatorios (Shifts):** Desplaza las imágenes horizontal y verticalmente.

2.4. Balance de los Hiperparámetros

El balance de los hiperparámetros en un modelo de aprendizaje profundo, como una Red Neuronal Convolutiva (CNN), es crucial y delicado debido a varios factores que influyen directamente en el rendimiento del modelo.

2.4.1. Precisión y Generalización

- **Overfitting y Underfitting:** Un balance incorrecto puede llevar al modelo a sobreajustarse (overfitting) o subajustarse (underfitting). El overfitting ocurre cuando el modelo aprende demasiado bien los detalles y el ruido del conjunto de entrenamiento, lo que disminuye su capacidad de generalización a nuevos datos. El underfitting sucede cuando el modelo es demasiado simple para capturar la complejidad de los datos, resultando en un bajo rendimiento tanto en los datos de entrenamiento como en los de prueba.
- **Regularización:** Hiperparámetros como las tasas de Dropout ayudan a prevenir el overfitting al evitar que el modelo se vuelva demasiado complejo.

2.4.2. Eficiencia Computacional

- **Tamaño del Lote (Batch Size):** Afecta tanto el tiempo de entrenamiento como la estabilidad del proceso de optimización. Un tamaño de lote demasiado grande puede requerir más memoria y hacer que el entrenamiento sea más lento, mientras que un tamaño demasiado pequeño puede resultar en una convergencia inestable.
- **Número de Épocas (Epochs):** Determina cuántas veces el algoritmo verá el conjunto completo de datos. Un número insuficiente de épocas puede llevar a un modelo no entrenado adecuadamente, mientras que demasiadas épocas pueden causar overfitting.

2.4.3. Tasa de Aprendizaje (Learning Rate)

- Una tasa de aprendizaje demasiado alta puede hacer que el modelo no converja, saltando sobre el mínimo óptimo, mientras que una tasa demasiado baja puede hacer que el proceso de entrenamiento sea extremadamente lento y se estanque en un mínimo local.
- Los callbacks como *ReduceLROnPlateau* ajustan la tasa de aprendizaje durante el entrenamiento para ayudar a encontrar un balance adecuado.

2.4.4. Estructura del Modelo

- **Número de Capas y Filtros:** El número de capas convolucionales, filtros y neuronas en las capas densas afecta la capacidad del modelo para aprender características complejas de los datos. Un modelo con demasiadas capas puede ser propenso al overfitting, mientras que un modelo con muy pocas capas puede no ser lo suficientemente potente.
- **Función de Activación:** Diferentes funciones de activación como ReLU, sigmoide o tanh tienen distintos comportamientos y afectarán la capacidad del modelo para aprender.

2.4.5. Técnicas de Regularización

- **Dropout:** Apaga aleatoriamente neuronas durante el entrenamiento para evitar la dependencia excesiva de ciertas neuronas, ayudando a generalizar mejor. La tasa de Dropout debe ser cuidadosamente ajustada; demasiada regularización puede llevar al underfitting.

2.4.6. Generación de Datos y Augmentación

La augmentación de datos, como rotaciones, zooms y flips, ayuda a aumentar la variedad de datos y mejora la capacidad del modelo para generalizar. Sin embargo, un exceso de augmentación puede introducir ruido y dificultar el aprendizaje de patrones relevantes.

2.5. Evaluación de Modelos de Clasificación

Este proceso permite determinar la efectividad del modelo para realizar predicciones precisas y confiables. En el caso de la clasificación de imágenes de rayos X para detectar neumonía, varias métricas y métodos se utilizan para evaluar el desempeño del modelo.

2.5.1. Métricas de Evaluación

- **Exactitud (Accuracy):** Es la proporción de predicciones correctas entre el total de predicciones realizadas.

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

donde TP son verdaderos positivos, TN son verdaderos negativos, FP son falsos positivos, y FN son falsos negativos.

- **Matriz de Confusión:** Una herramienta que permite visualizar el desempeño del modelo de clasificación. La matriz de confusión muestra la can-

tidad de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figura 4: Matriz de Confusión.

- **Precisión:** La proporción de verdaderos positivos sobre el total de predicciones positivas.

$$Presicion = \frac{TP}{TP + FP}$$

- **Recall (Sensibilidad):** La proporción de verdaderos positivos sobre el total de verdaderos casos positivos.

$$Recall = \frac{TP}{TP + FN}$$

- **F1-Score:** Es la media armónica de la precisión y el recall, proporcionando una métrica balanceada cuando existe una distribución desigual entre clases.

$$F_1 - Score = 2 * \frac{Presicion * Recall}{Presicion + Recall}$$

- **Curva ROC y AUC:** La Curva ROC (Receiver Operating Characteristic) es una representación gráfica que ilustra la capacidad del modelo para discriminar entre clases. El AUC (Área Bajo la Curva) proporciona una única medida de rendimiento, donde un valor más cercano a 1 indica mejor desempeño.

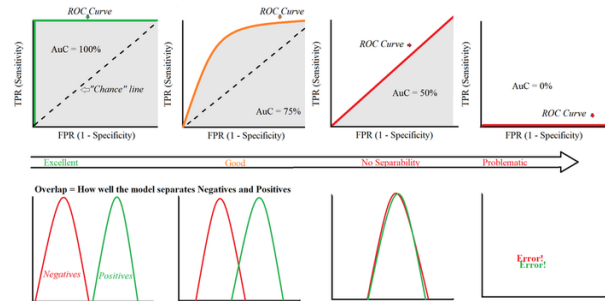


Figura 5: Curva ROC y AUC.

- Curva de Precisión-Recall: Esta curva es especialmente útil en situaciones de clases desbalanceadas, mostrando la relación entre la precisión y el recall para diferentes umbrales.

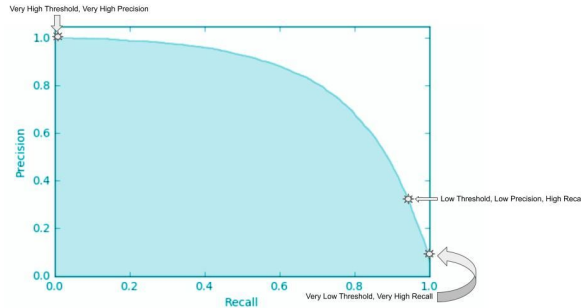


Figura 6: Curva ROC y AUC.

2.5.2. Interpretación de Resultados

- Precisión del Modelo: Una alta precisión indica que el modelo realiza predicciones correctas en la mayoría de los casos. Sin embargo, en conjuntos de datos desbalanceados, esta métrica puede ser engañosa.
- Matriz de Confusión: Proporciona una visión detallada de cómo el modelo está cometiendo errores, diferenciando entre falsos positivos y falsos negativos.
- Curva ROC y AUC: Una curva ROC que se acerca a la esquina superior izquierda y un valor de AUC alto indican un buen rendimiento del modelo en términos de discriminación entre clases.
- Curva de Precisión-Recall: Una alta área bajo esta curva sugiere que el modelo mantiene un buen balance entre precisión y recall, lo cual es crucial en problemas donde los falsos negativos o falsos positivos tienen costos significativos.

3. Procedimiento

Este algoritmo implementa una Red Neuronal Convolutiva (CNN) para la detección de neumonía en imágenes de rayos X del pecho. Primero, se importan todas las librerías necesarias, incluyendo Keras y TensorFlow para construir y entrenar la CNN, OpenCV para la manipulación de imágenes, y otras librerías para la visualización y el análisis de datos. Luego, se monta Google Drive para acceder a los datos y se fijan semillas aleatorias para garantizar la reproducibilidad. Se cargan imágenes de ejemplo

y se muestran para ilustrar el tipo de datos con los que trabajará el modelo.

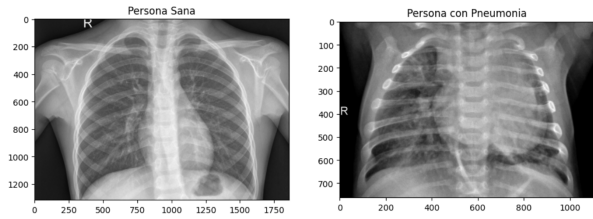


Figura 7: Persona Sana vs Persona con Neumonía.

La función *process-data* se encarga del preprocesamiento de datos, configurando los generadores de datos de entrenamiento, validación y prueba con transformaciones como normalización, distorsión y flip horizontal. Los iteradores de directorio recorren las carpetas correspondientes y cargan las imágenes en lotes ajustando su tamaño. La arquitectura de la CNN se define en varios bloques convolucionales utilizando capas *Conv2D* y *SeparableConv2D*, seguidas de normalización (*BatchNormalization*) y agrupación máxima (*MaxPool2D*). Estas capas extraen características relevantes de las imágenes. Las capas de Dropout se incluyen para reducir el overfitting. Las capas convolucionales están conectadas a capas densas (fully connected) que transforman los datos en una representación plana, pasando por varias capas densas con activación ReLU y Dropout. La capa final es una capa densa con una única unidad y activación sigmoide para realizar la clasificación binaria. El modelo se compila utilizando el optimizador Adam y la función de pérdida binaria (*binary-crossentropy*). Se definen callbacks para reducir la tasa de aprendizaje (*ReduceLROnPlateau*) y detener el entrenamiento temprano (*EarlyStopping*). El modelo se entrena utilizando los generadores de datos de entrenamiento y validación. Finalmente, se guarda el modelo entrenado y se evalúa su rendimiento mediante métricas como precisión, matriz de confusión, precisión, recall y F1-score. Se visualizan los resultados con gráficas de precisión y pérdida, imágenes de prueba con probabilidades predichas y curvas ROC y de Precisión-Recall. Este algoritmo combina técnicas de preprocesamiento de imágenes, generación de datos aumentados y una arquitectura de CNN para construir un modelo capaz de diferenciar entre imágenes de rayos X normales y aquellas con neumonía, proporcionando métricas y visualizaciones detalladas para evaluar su rendimiento.

Mostrando el resumen del modelo implementa-

do.

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 150, 150, 3)	0
conv2d (Conv2D)	(None, 150, 150, 16)	448
conv2d_1 (Conv2D)	(None, 150, 150, 16)	2,320
max_pooling2d (MaxPooling2D)	(None, 75, 75, 16)	0
separable_conv2d (SeparableConv2D)	(None, 75, 75, 32)	688
separable_conv2d_1 (SeparableConv2D)	(None, 75, 75, 32)	1,344
batch_normalization (BatchNormalization)	(None, 75, 75, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 32)	0
separable_conv2d_2 (SeparableConv2D)	(None, 37, 37, 64)	2,400
separable_conv2d_3 (SeparableConv2D)	(None, 37, 37, 64)	4,736
batch_normalization_1 (BatchNormalization)	(None, 37, 37, 64)	256
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 64)	0
separable_conv2d_4 (SeparableConv2D)	(None, 18, 18, 128)	8,896
separable_conv2d_5 (SeparableConv2D)	(None, 18, 18, 128)	17,664
batch_normalization_2 (BatchNormalization)	(None, 18, 18, 128)	512
max_pooling2d_3 (MaxPooling2D)	(None, 9, 9, 128)	0
dropout (Dropout)	(None, 9, 9, 128)	0
separable_conv2d_6 (SeparableConv2D)	(None, 9, 9, 256)	34,176
separable_conv2d_7 (SeparableConv2D)	(None, 9, 9, 256)	68,096
batch_normalization_3 (BatchNormalization)	(None, 9, 9, 256)	1,024
max_pooling2d_4 (MaxPooling2D)	(None, 4, 4, 256)	0
dropout_1 (Dropout)	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 512)	2,097,664
dropout_2 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 128)	65,664
dropout_3 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dropout_4 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

4. Resultados

Después de guardar el modelo lo primero que se hace es probarlo a partir de las métricas de evaluación.

METRICAS DE LA PRUEBA -----

Accuracy: 90.7051282051282%

Precision: 88.24884792626729%

Recall: 98.2051282051282%

F1-score: 92.96116504854369

METRICA DE ENTRENAMIENTO -----

Train acc: 96.55

Se puede observar que el modelo tiene una precisión de 90 %.

Ya que el modelo presenta una buena precisión para identificar entre pacientes sanos y enfermos, se hacen pruebas con un lote de imágenes para comprobar obteniendo lo siguiente

■ Prueba en pacientes sanos

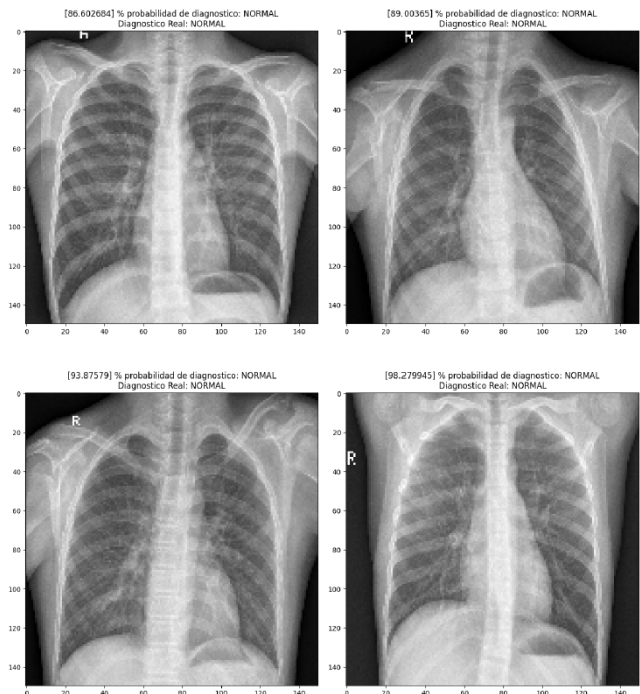


Figura 8: Prueba del modelo para identificar a pacientes sanos.

■ Prueba en pacientes con neumonía

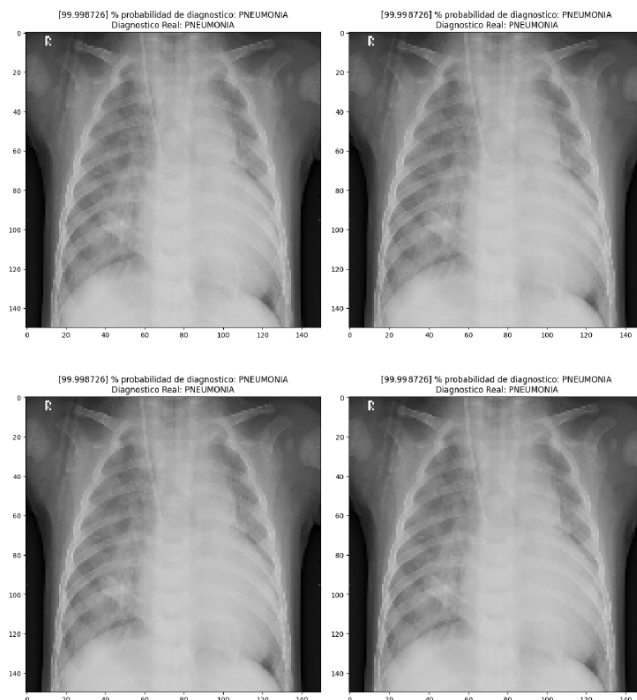


Figura 9: Prueba del modelo para identificar a pacientes con neumonía.

Después de realizar las comprobaciones correspondientes se construye la matriz de confusión y las curvas ROC-AUC y Precisión-Recall.

		Predicción	
		Paciente Sano	Paciente con Neumonía
Real	Paciente Sano	183	51
	Paciente con Neumonía	7	383

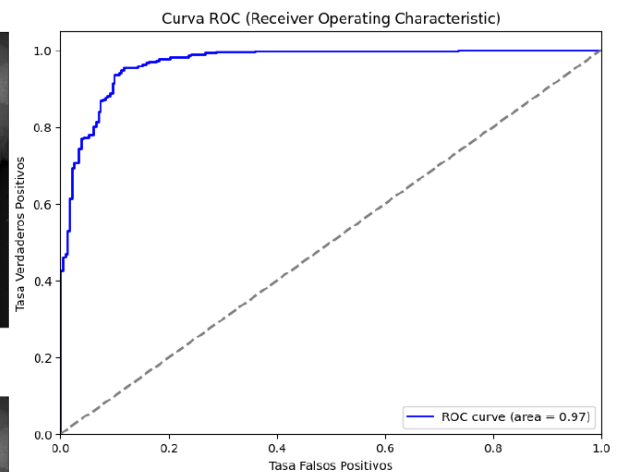


Figura 10: Curvas ROC-AUC.

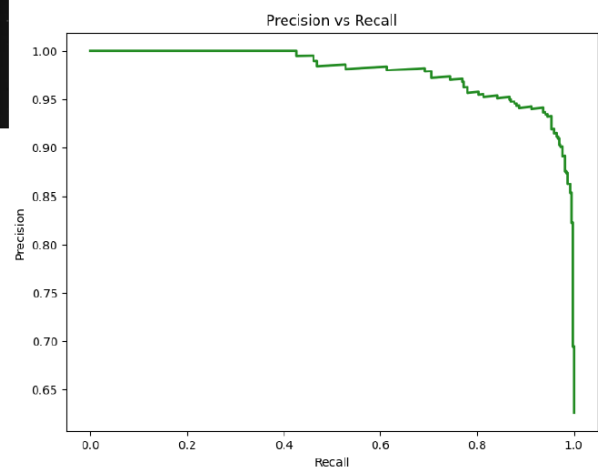


Figura 11: Curvas Precisión-Recall.

Como podemos observar en ambas gráficas, el comportamiento es cercano al ideal y al mejor desempeño descrito en el marco teórico. Esto corrobora que el modelo tiene una buena precisión al realizar las predicciones.

Por otro lado se va a evaluar como varia la precisión del modelo a medida de avanza la época y también la función de pérdida.

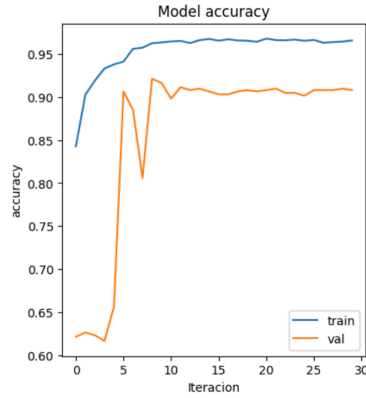


Figura 12: *Precisión del Modelo.*

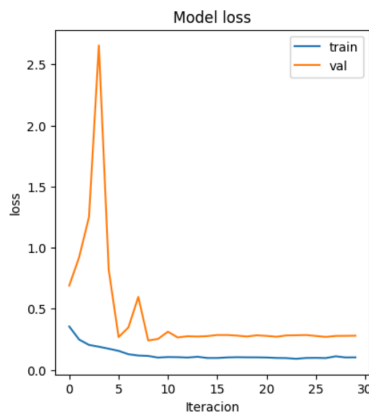


Figura 13: *Perdida del Modelo.*

5. Conclusiones

- La base de datos utilizada en el entrenamiento de la red neuronal, tiene una cantidad significativamente más grande de imágenes clasificadas como neumonía, lo que se traduce en la red neuronal realizando predicciones en neumonía, más veces que en normal, esto también se observa en la matriz de confusión, donde hay alrededor de 7 veces más falsos positivos que falsos negativos, lo que se traduce en que el algoritmo es más probable a declarar a alguien sano como enfermo que lo contrario.
- La optimización de los hiperparámetros de la red neuronal es un paso importante para lograr resultados tanto acertados como eficientes, como se observa en la gráfica de la precisión en las iteraciones o épocas, el valor de esta métrica no varía significativamente después de la época número 15, lo que abre las puertas para seguir optimizando esta red neuronal.
- AUC-ROC de 0.97 confirma que el modelo es altamente preciso y tiene un excelente rendimiento en la clasificación, lo que está alineado con el comportamiento observado en las gráficas y el marco teórico.