

Proyecto 2: Redes Neuronales

1° Aldo Cambronero Ureña
Ingeniería en Computadores
Tecnológico de Costa Rica
Cartago, Costa Rica
aldocambr@estudiantec.cr

2° Aguero Sandí Johnny
Ingeniería en Computadores
Tecnológico de Costa Rica
Cartago, Costa Rica
johnny.zaet08@estudintec.cr

Resumen—En este trabajo presenta un análisis de dos tipos de redes neuronales, sobre un mismo conjunto de datos. La primera es una red MLP donde se comparara los resultados del entrenamiento de está con los datos crudos y con los datos aplicando el algoritmo LBP, y para la segunda red CNN se realiza la misma metodología solo que aplicando el filtro de con los datos crudos y aplicando el filtro bilateral. Los resultados sugieren que las arquitecturas planteadas para procesar los datos son ideales para hacer una clasificación de datos no lineales.

Palabras clave—MLP, CNN, Redes Neuronales, Salud, ROC, AUC, Matriz de confusión.

I. INTRODUCCIÓN

Las redes neuronales son modelos matemáticos que se asemejan al funcionamiento de las redes de neuronas biológicas. Están compuestas por capas interconectadas de nodos, llamados neuronas artificiales, que procesan la información mediante cálculos numéricos. Cada neurona toma entradas, las pondera y aplica una función de activación para producir una salida. A través del ajuste de los pesos sinápticos, las redes neuronales son capaces de aprender patrones complejos y realizar tareas de manera más eficiente [1].

En el presente trabajo, exploramos dos tipos de redes neuronales con diferentes arquitecturas para tratar un mismo set de datos, con el objetivo de observar y analizar las métricas relevantes a nuestras arquitecturas. El set de datos, son un conjunto de imágenes de rayos x sobre el tórax de diferentes pacientes con covid, opacidad pulmonar, neumonía y pacientes normales.

Estos datos se amolda de buena manera para entrenar una red neuronal que me dicta si un paciente posee una de estas cuatro condiciones, para conseguir esto emplearemos la red neuronal MLP, una red neuronal MLP consiste en múltiples capas de neuronas interconectadas, donde cada neurona en una capa está conectada a todas las neuronas de la capa siguiente. La primera capa se conoce como capa de entrada, la última como capa de salida, y las capas intermedias se denominan capas ocultas. Cada neurona en la red toma una serie de entradas, las pondera mediante conexiones sinápticas y las procesa a través de una función de activación no lineal para generar una salida [2].

Para está primera red neuronal, se emplearon tres tipos de arquitecturas para analizar cual de estas presentaba mejores resultados en sus métricas, ahora bien a los datos de antemano se le debían de hacer el correspondiente análisis para observar

que no estuvieran sesgados o que estos se optimizarán al máximo para que la duración del entrenamiento no fuera tan extensa. Ya una vez realizado este proceso, guardamos estas imágenes para ser entrenadas en la red neuronal, y haremos una bifurcación con los datos para aplicarles un algoritmo de *feature extractor* como LBP, Histogramas de Color, SIFT, ORB, entre otros. Con el fin de reducir la dimensionalidad de las imágenes y extraer los datos más relevantes para que la red neuronal sea entrenada con estos datos y poder compararlos con los datos sin este proceso.

Cabe destacar que se empleó el algoritmo de LBP, ya que este se amolda a las condiciones del proyecto y del set de datos, teniendo imágenes de poca resolución y que la mayor información está en la “textura” de las imágenes. Ya que, el algoritmo LBP se basa en la idea de capturar la apariencia local de una textura en una imagen. En lugar de considerar las intensidades de los píxeles en sí, LBP se centra en las relaciones de vecindad entre los píxeles. Para cada píxel en la imagen, se compara su valor con el de sus vecinos circundantes, generando un patrón binario que codifica estas comparaciones. Estos patrones binarios locales reflejan la distribución de las transiciones de intensidad en la vecindad de cada píxel, lo que proporciona información valiosa sobre la textura presente en esa región [3].

Ahora la segunda red neuronal que se trabajará es la red CNN, a diferencia de las redes neuronales tradicionales, las CNN están diseñadas específicamente para el procesamiento de imágenes y aprovechan la estructura espacial de los datos visuales. La arquitectura de una CNN se basa en capas convolucionales, que aplican filtros convolutivos a las imágenes para extraer características locales. Estas capas convolucionales se combinan con capas de agrupación (pooling), que reducen la dimensionalidad de las características y preservan su información más relevante. Además, las CNN también pueden contener capas completamente conectadas al final de la red para realizar la clasificación o la regresión en base a las características extraídas [4].

Para está arquitectura se hará una comparación similar al punto anterior donde se entrenará la red con las imágenes crudas y con las imágenes cuando le aplicamos el algoritmo del filtro bilateral. El funcionamiento del filtro bilateral se basa en dos componentes principales: el peso espacial y el peso de similitud. El peso espacial mide la cercanía de un píxel con respecto al píxel central en términos de su posición en la

imagen. El peso de similitud, por otro lado, mide la similitud en los valores de los píxeles entre el píxel central y los píxeles vecinos. Estos pesos se utilizan para ponderar la contribución de cada píxel vecino al cálculo del valor del píxel filtrado, de modo que los píxeles más cercanos y más similares tengan un mayor impacto en el resultado final [5].

A todas estas arquitecturas se les sacarán las métricas necesarias para sus análisis, como la precisión, exactitud, recuperación, AUC, ROC y matriz de confusión para cada clase del modelo.

II. DETALLES DEL DISEÑO DEL PROGRAMA DESARROLLO

En lo que respecta al diseño al diseño del programa implementado, se tuvo que seguir una línea de análisis un poco distinta ya que estamos empleando un conjunto de datos que son imágenes; por lo cual, requerimos de un análisis un poco distinto para este. De primera mano hay que contabilizar con cuantas imágenes estamos trabajando, en total con las cuatro tipos de imágenes se contabilizan más de veinte mil imágenes, con un sesgo detectado con respecto a los pacientes que no presentan ningún síntoma a los que presentan covid, opacidad pulmonar o neumonía, por lo cual se decide no tomar en cuenta la mitad de las imágenes de está categoría, para reducir este sesgo.

Ahora bien, se realiza un proceso de escaneo sobre la imagenes restante para observar que no esté corruptas, luego se reduce su dimensionalidad de 299 x 299 a 244 x 244 ya que está dimensionalidad es un buen punto medio entre perder información de la imagen y mantener un modelo de entrenamiento rápido. A su vez, se normalizan los pixeles de las imágenes para eliminar información irrelevante, con estos análisis se termina el proceso de *feature engineering*, se genera un nuevo archivo csv con las rutas de las imágenes y con sus etiquetas para facilitar el proceso de montaje en las redes neuronales.

Después de este proceso, a las imágenes que se tiene en este csv se les aplica el filtro LBP y bilateral tanto para la red neuronal MLP y CNN. Los datos del csv se parten en dos listas para las rutas de las imágenes y sus etiquetas, a las etiquetas se convierten en una representación numérica y las imágenes se convierten en arreglos de np. Para luego ser partidas en listas de pruebas y testeo, que serán entrenadas en las redes neuronales y luego analizar las métricas que estas devuelven.

II-A. Diagrama de Flujo

A continuación se presenta el diagrama de flujo para los algoritmos empleados en CNN y MLP, ya que en general poseen el mismo *modus operandi*.

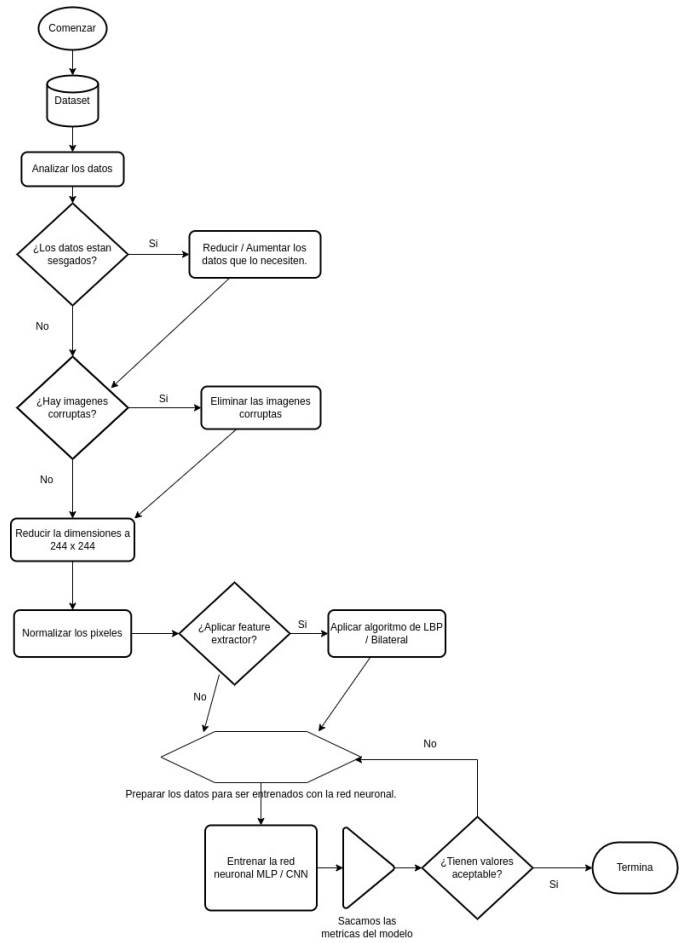


Figura 1. Diagrama de flujo de los algoritmos MLP y CNN

II-B. Diagrama de arquitectura

A continuación se presentan las arquitecturas de la red neuronal MLP, ya que esta requiere que se realizarán tres diferentes para el análisis de esta.

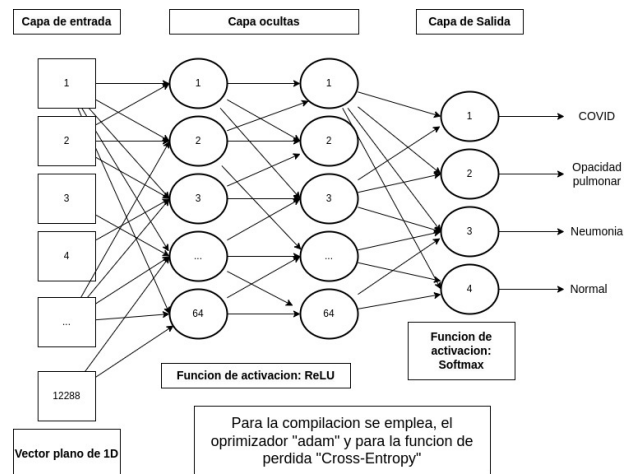


Figura 2. Diagrama de arquitectura 1, de la red neuronal MLP

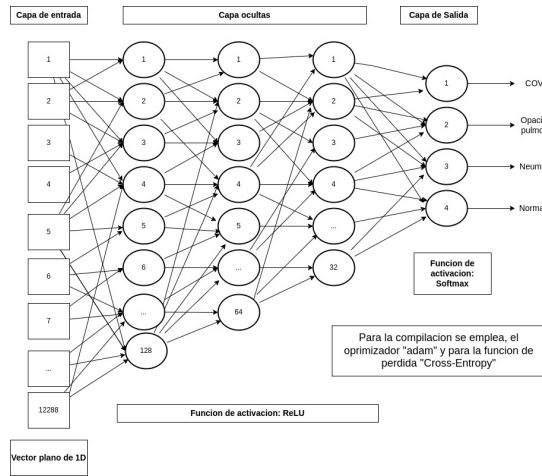


Figura 3. Diagrama de arquitectura 1, de la red neuronal MLP

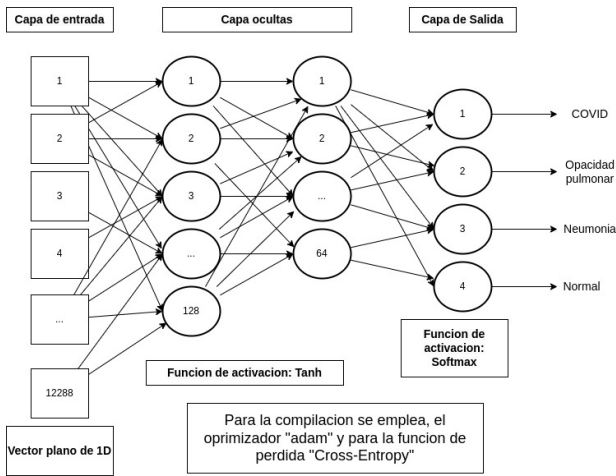


Figura 4. Diagrama de arquitectura 1, de la red neuronal MLP

Posteriormente, se presenta el diagrama de arquitectura utilizado para la red CNN, esta red varía de las anteriores puesto que

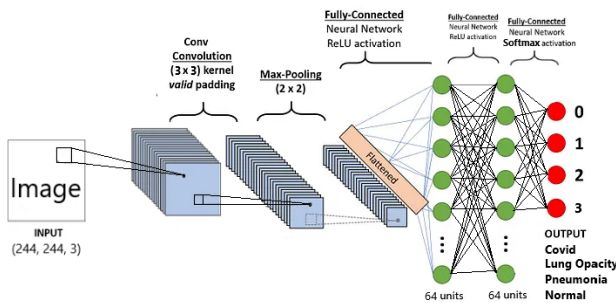


Figura 5. Diagrama de arquitectura 1, de la red convolucional (CNN)

III. RESULTADOS Y ANÁLISIS

III-A. Resultados

A continuación se presentan las gráficas de ROC para cada clase, su matriz de confusión y a su vez las tablas con los resultados de las métricas obtenidas para cada una de las tres redes neuronales propuestas en la sección de MLP. Cabe destacar que se empleó los datos con el *feature extractor* para entrenar los modelos.

	Arquitectura 1			
Métricas	Precision	Recall	F1-score	AUC
Covid	0.53	0.25	0.34	0.83
Lung opacity	0.86	0.36	0.51	0.73
Normal	0.41	0.96	0.58	0.84
Pneumonia	0.00	0.00	0.00	0.87

Tabla I

TABLA DE COMPARACIÓN DE MÉTRICAS DE LA ARQUITECTURA 1 DE LA RED MLP, CON *feature extractor*

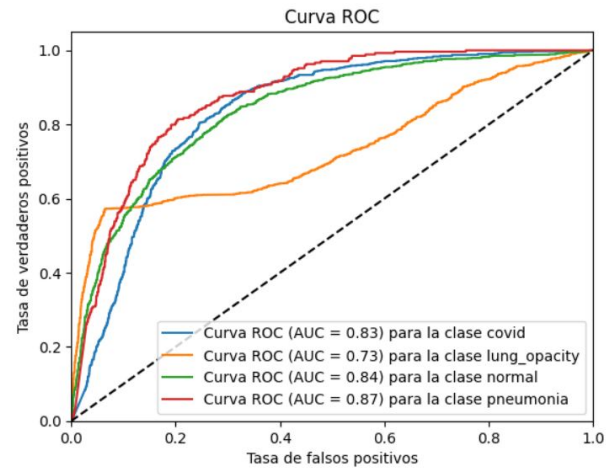


Figura 6. Gráficas de ROC para la arquitectura 1, con *feature extractor*

Matriz de Confusión

	covid	lung_opacity	normal	pneumonia
Etiqueta Verdadera covid	179	36	508	0
Etiqueta Verdadera lung_opacity	9	433	761	0
Etiqueta Verdadera normal	9	31	979	0
Etiqueta Verdadera pneumonia	139	1	129	0
Etiqueta Predicha	covid	lung_opacity	normal	pneumonia

Figura 7. Matriz de confucion para la arquitectura 1, con *feature extractor*

	Arquitectura 2			
Métricas	Precision	Recall	F1-score	AUC
Covid	0.36	0.64	0.46	0.75
Lung opacity	0.78	0.55	0.65	0.87
Normal	0.84	0.28	0.42	0.82
Pneumonia	0.36	0.99	0.53	0.98

Tabla II

TABLA DE COMPARACIÓN DE MÉTRICAS DE LA ARQUITECTURA 2 DE LA RED MLP, CON *feature extractor*

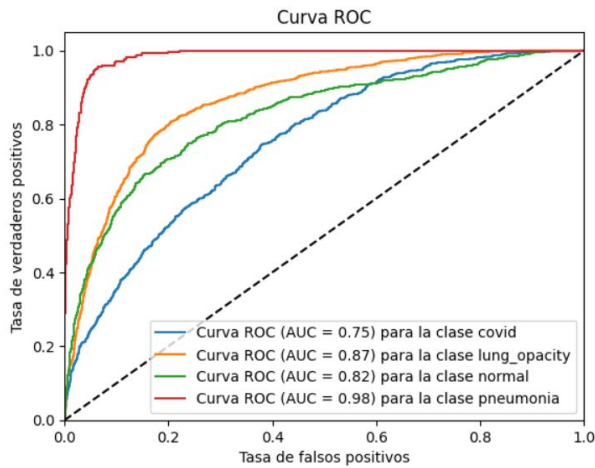


Figura 8. Gráficas de ROC para la arquitectura 2, con *feature extractor*

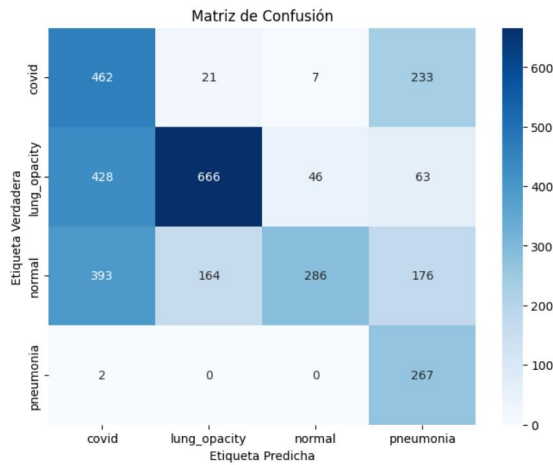


Figura 9. Matriz de confusion para la arquitectura 2, con *feature extractor*

	Arquitectura 3			
Métricas	Precision	Recall	F1-score	AUC
Covid	0.00	0.00	0.00	0.49
Lung opacity	0.37	1.00	0.54	0.5
Normal	0.00	0.00	0.00	0.5
Pneumonia	0.00	0.00	0.00	0.5

Tabla III

TABLA DE COMPARACIÓN DE MÉTRICAS DE LA ARQUITECTURA 3 DE LA RED MLP, CON *feature extractor*

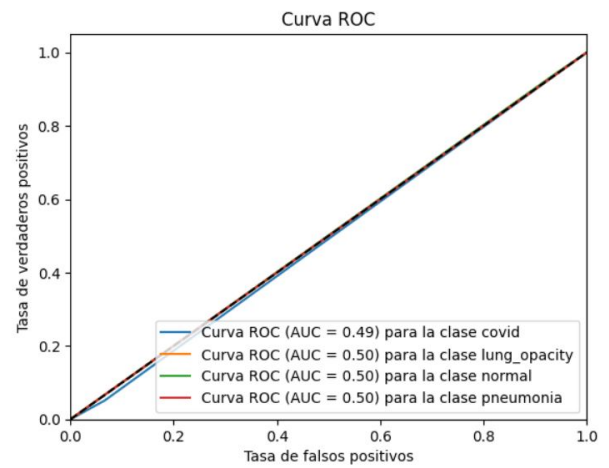


Figura 10. Gráficas de ROC para la arquitectura 3, con *feature extractor*

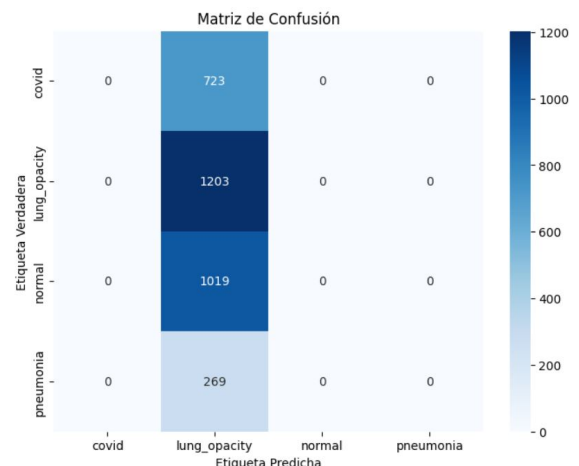


Figura 11. Matriz de confusion para la arquitectura 3, con *feature extractor*

Ahora se presenta la tabla con las métricas de la arquitectura 1, solo que empleando los datos sin el *feature extractor*, para su análisis.

	Arquitectura			
Métricas	Precision	Recall	F1-score	AUC
Covid	0.78	0.38	0.51	0.85
Lung opacity	0.59	0.74	0.66	0.87
Normal	0.69	0.83	0.75	0.87
Pneumonia	0.81	0.88	0.84	0.99

Tabla IV

TABLA DE COMPARACIÓN DE MÉTRICAS DE LA ARQUITECTURA 1 DE LA RED MLP, SIN *feature extractor*

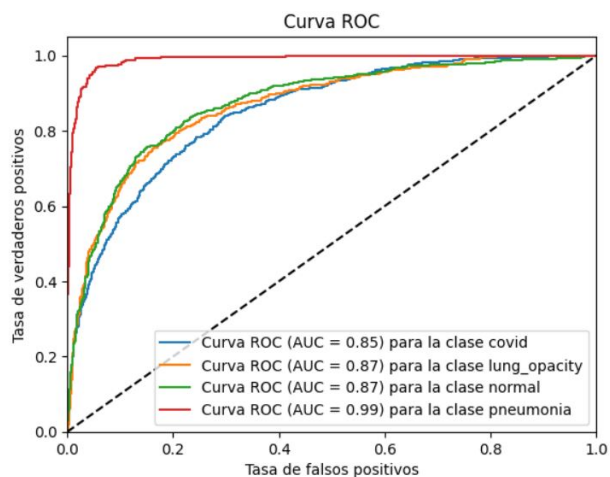


Figura 12. Gráficas de ROC para la arquitectura 1, sin *feature extractor*

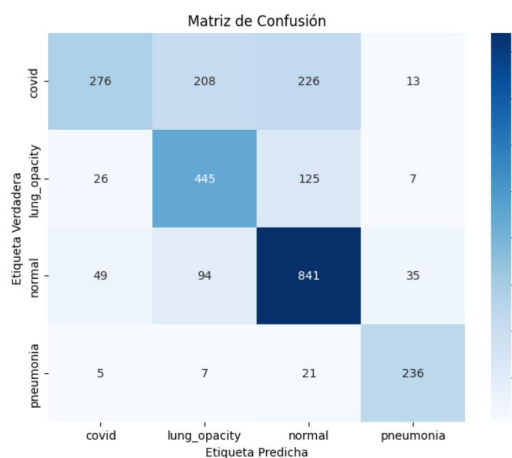


Figura 13. Matriz de confusión para la arquitectura 1, sin *feature extractor*

Posteriormente, se presenta una imagen con su mapa de calor, esta imagen es extraída del modelo más aceptado.

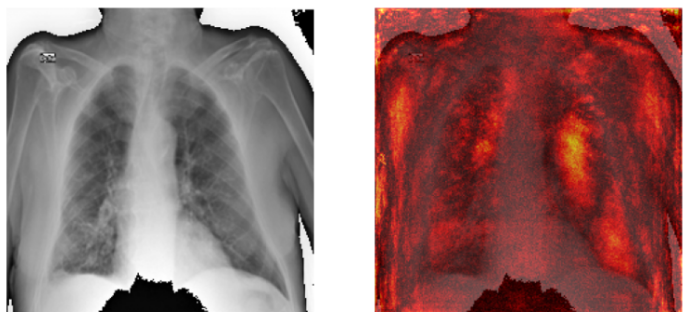


Figura 14. Mapa de calor, del modelo MLP

A continuación se presentan las tablas con los resultados de las métricas obtenidas de las arquitecturas de las redes CNN propuestas. Se presentan los datos con las imágenes crudas y los datos aplicando el filtro bilateral a cada imagen.

—	Arquitectura			
Métricas	Precision	Recall	F1-score	AUC
Covid	0.90	0.84	0.87	0.98
Lung opacity	0.79	0.75	0.77	0.91
Normal	0.83	0.89	0.86	0.93
Pneumonia	0.95	0.88	0.92	1.00

Tabla V

TABLA DE COMPARACIÓN DE MÉTRICAS DE LA RED CNN, SIN FILTRO BILATERAL.

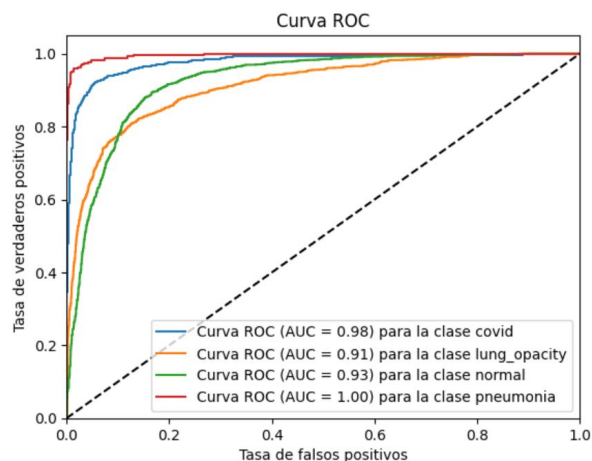


Figura 15. Gráficas de ROC para red convolucional, sin *filtro bidirrecional*

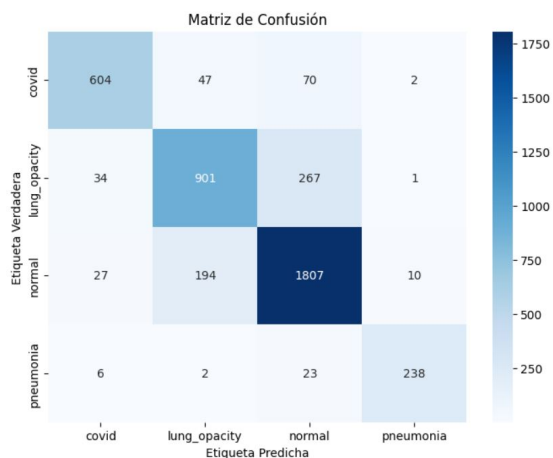


Figura 16. Gráficas de matriz de confusión para red convolucional, sin *filtro bidirrecional*

—	Arquitectura			
Métricas	Precision	Recall	F1-score	AUC
Covid	0.74	0.86	0.80	0.97
Lung opacity	0.82	0.66	0.73	0.91
Normal	0.84	0.89	0.86	0.93
Pneumonia	0.94	0.89	0.91	1.00

Tabla VI

TABLA DE COMPARACIÓN DE MÉTRICAS DE LA RED CNN, CON FILTRO BILATERAL.

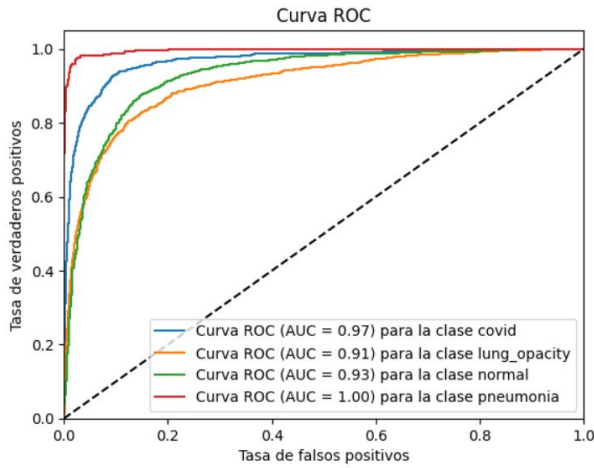


Figura 17. Gráficas de ROC para red convolucional, con *filtro bidirrecional*

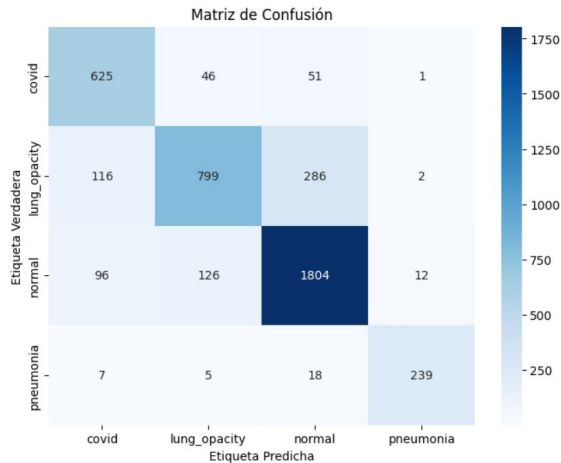


Figura 18. Gráficas de matriz de confusión para red convolucional, con *filtro bidirrecional*

III-B. Análisis

Respecto a la subsección de análisis de datos iniciaremos con la parte de la arquitectura MLP, hay mucho que abarcar debido a que se presentan tres tipos de arquitecturas con el mismo conjuntos de datos y una de las tres arquitecturas anteriores pero con otro conjuntos de datos.

Con respecto a los valores vistos en las tablas I, II, III, IV o en las gráficas 6, 8, 10 y 12 podemos observar que las métricas obtenidas con respecto a las diferentes arquitecturas empleando el conjunto de datos con *feature extractor*, no indican una estabilidad muy buena en este. Primero que nada, la arquitectura 1 no converge para la clase de neumonía y el resto de clases presentan los valores de precisión y recall muy dispares entre sí. Esto se evidencia más con la arquitectura 2 que sus métricas andan más dispersas entre sí y ya para la arquitectura 3 el modelo no converge.

Esto es el caso contrario con respecto a la arquitectura 1 que emplea el conjunto de datos sin el *feature extractor*, ya que presenta todas sus métricas para las cuatro clases con

valores relativamente altos y sin una diferencia entre estos alta. Considerando esta arquitectura como la predilecta, no presenta overfitting o underfitting, caso contrario con las otras tres que si llegan a presentar esto, su valor de AUC para las cuatro clases es mayor a 0.7, en la matriz de confusión presenta la mejor relación entre etiquetas predichas y etiquetas verdaderas y no se redujo un poco las imágenes de pacientes con opacidad pulmonar para reducir el sesgo del conjunto de datos.

Con respecto al conjunto de datos a manera, se detectó que este estaba sesgado debido a que habían más de diez mil imágenes de pacientes normales en comparación a las menos de dos mil para los pacientes con neumonía. Este modelo posee un tamaño, de cuatro capas, con un total de 11 413 396 parámetros totales, la capa de entrada plana posee 178608 neuronas, con dos capas internas de 64 neuronas con una función de activación “ReLU”, una de salida de 4 neuronas con la función “softmax” y se compila con una función de optimización “adam” y de pérdida Cross-Entropy.

Como se puede evidenciar con los resultados presentados, el *feature extractor* no ayuda realmente al modelo a poseer unas métricas más eficiente; ya que, el algoritmo LBP al generar las texturas de las imágenes no ayuda a resaltar la información de los pacientes con enfermedades, pero sí reduce su tiempo de ejecución alrededor de 20 minutos y con esto la arquitectura número 3 es la que más rápido lo hace debido a que no converge.

Como se puede observar en la figura 14, la zona que más contribuyó al modelo fue la zona pulmonar superior

Por otra parte, se pueden analizar los resultados de la red CNN con la arquitectura que se observa en la figura 5. Según la tabla V, se presentan las métricas de precisión, recall, F1-score y AUC para cada una de las cuatro clases: Covid, Lung opacity, Normal y Pneumonia. Estas métricas son indicadores importantes para evaluar el rendimiento de tu modelo en la tarea de clasificación. A continuación, analizaré los resultados para la arquitectura sin filtro bilateral.

Para la clase covid según la tabla V estos valores indican que el modelo tiene una precisión del 90 % al predecir casos de Covid. El recall del 84 % muestra que el modelo identifica correctamente el 84 % de las imágenes de Covid en el conjunto de datos. El F1-score, que combina precisión y recall, es de 0.87. Un valor de AUC de 0.98 sugiere un buen rendimiento general para la clase Covid. Posteriormente el modelo tiene una precisión del 79 % para la clase Lung opacity. El recall del 75 % indica que el modelo identifica correctamente el 75 % de las imágenes de Lung opacity. El F1-score es de 0.77, lo que refleja un equilibrio entre precisión y recall. El valor de AUC de 0.91 indica un rendimiento razonable para esta clase.

Para la clase Normal, el modelo logra una precisión del 83 %. El recall del 89 % muestra que el modelo identifica correctamente el 89 % de las imágenes normales. El F1-score de 0.86 indica un buen equilibrio entre precisión y recall. Además, el valor de AUC de 0.93 sugiere un rendimiento sólido para la clase Normal. La clase Pneumonia muestra una alta precisión del 95 %. El recall del 88 % indica que el modelo identifica correctamente el 88 % de las imágenes

de Pneumonia. El F1-score de 0.92 refleja un buen equilibrio entre precisión y recall. El valor de AUC de 1.00 indica un rendimiento excelente para esta clase.

En general, la arquitectura del modelo de redes convolucionales muestra buenos resultados en la clasificación de las cuatro clases mencionadas. Sin embargo, es importante tener en cuenta que estos resultados son específicos para imágenes crudas. A continuación, se proporcionan los resultados correspondientes a las imágenes con filtro bilateral para realizar una comparación.

Según la tabla VI para la clase Covid tenemos que comparando con los resultados anteriores, la precisión ha disminuido ligeramente, mientras que el recall ha aumentado. El F1-score también ha disminuido en comparación con la arquitectura sin filtro bilateral. Sin embargo, el valor de AUC se mantiene alto, indicando un buen rendimiento general para la clase Covid. En este caso, la precisión ha aumentado, pero el recall ha disminuido en comparación con la arquitectura sin filtro bilateral. El F1-score también ha disminuido, lo que indica un equilibrio inferior entre precisión y recall. El valor de AUC se mantiene en un nivel similar al anterior.

Los resultados para la clase Normal son similares a los obtenidos en la arquitectura sin filtro bilateral. La precisión y el recall se mantienen prácticamente iguales, al igual que el F1-score y el valor de AUC. Al igual que en la clase Normal, los resultados para la clase Pneumonia son similares a los de la arquitectura sin filtro bilateral. La precisión y el recall se mantienen prácticamente iguales, así como el F1-score y el valor de AUC.

En general, se observan algunas diferencias en las métricas al utilizar imágenes con filtro bilateral en comparación con imágenes crudas. Algunas clases experimentan cambios significativos, mientras que otras se mantienen estables. Es importante tener en cuenta estos cambios al evaluar el rendimiento del modelo con diferentes tipos de imágenes.

IV. CONCLUSIONES

Por lo tanto, las arquitecturas del perceptrón multicapa (MLP) que utilizan el conjunto de datos sin el "feature extractor" presentan mejores métricas en comparación con aquellas que emplean el conjunto de datos con el "feature extractor". Las arquitecturas 1, 2 y 3 que utilizan el "feature extractor" muestran una falta de estabilidad en las métricas, con problemas de convergencia y valores dispersos de precisión y recall. En contraste, la arquitectura 1 sin el "feature extractor" presenta métricas relativamente altas y equilibradas para todas las clases, sin evidencia de sobreajuste o infravaloración. Esto indica que el "feature extractor" no proporciona beneficios significativos en términos de rendimiento del modelo.

Por otra parte, el conjunto de datos utilizado en el análisis muestra un sesgo significativo, con una gran cantidad de imágenes de pacientes normales en comparación con las imágenes de pacientes con neumonía. Este desequilibrio puede afectar el rendimiento del modelo, ya que puede estar más inclinado a clasificar las imágenes como normales debido a la mayor presencia de ejemplos de esa clase. A pesar de

este sesgo en el conjunto de datos, la arquitectura 1 sin el "feature extractor" demuestra un rendimiento sólido, con un valor de AUC mayor a 0.7 para todas las clases y una matriz de confusión que muestra una buena relación entre las etiquetas predichas y las etiquetas verdaderas. Esto sugiere que esta arquitectura es preferible en este contexto y se ha evitado el sesgo en el conjunto de datos mediante la no reducción de las imágenes de pacientes con opacidad pulmonar.

El modelo de redes convolucionales muestra un rendimiento sólido en la clasificación de las cuatro clases evaluadas (Covid, Lung opacity, Normal y Pneumonia). En ambas arquitecturas (sin filtro bilateral y con filtro bilateral), se obtienen resultados favorables en términos de precisión, recall, F1-score y AUC.

Al comparar los resultados entre las imágenes crudas y las imágenes con filtro bilateral, se observan diferencias en las métricas para algunas clases. Por ejemplo, en la clase Covid, el recall mejora significativamente con el uso del filtro bilateral, lo que indica que el modelo puede identificar más casos positivos correctamente. Sin embargo, en la clase Lung opacity, el recall disminuye con el filtro bilateral, lo que sugiere una disminución en la capacidad del modelo para detectar casos positivos. Estas diferencias resaltan la importancia de considerar las características de preprocesamiento de las imágenes al evaluar el rendimiento del modelo.

El análisis de las métricas de evaluación (precisión, recall, F1-score y AUC) proporciona una visión completa del rendimiento del modelo en la clasificación de diferentes clases. Estas métricas permiten una comparación objetiva y cuantitativa de los resultados, lo que es esencial al evaluar y comunicar el rendimiento de un modelo de redes convolucionales en tareas de clasificación médica.

REFERENCIAS

- [1] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
- [2] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.
- [3] M. Pietikäinen, T. Ojala, and Z. Xu, "Rotation-invariant texture classification using feature distributions," *Pattern recognition*, vol. 33, no. 1, pp. 43–52, 2000.
- [4] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.
- [5] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," in *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Proceedings, Part IV 9*. Springer, 2006, pp. 568–580.