

INFORME FINAL: Proyecto de clasificación de neumonía en Radiografías de Tórax

1. INTRODUCCIÓN:

El análisis de radiografías de tórax mediante Machine Learning se ha convertido en un eje estratégico dentro de la analítica clínica moderna, especialmente en aquellos escenarios donde la interpretación manual depende del criterio de expertos, introduce variabilidad diagnóstica, y no escala eficientemente para grandes volúmenes. Este proyecto propone abordar la detección y tipificación de neumonía usando Deep Learning, con arquitectura CNN secuencial implementada en un entorno reproducible, empleando datos públicos y una red preentrenada eficiente para extracción visual de patrones pulmonares.

Las imágenes utilizadas provienen originalmente del dataset público “Chest X-Ray Images (Pneumonia)” del usuario Kaggle, creado por Paul Mooney, que permite identificar infiltrados, opacidades pulmonares y alteraciones estructurales, categorizadas según casos sanos, neumonía bacteriana y neumonía viral.

2. OBJETIVO DEL PROYECTO:

Desarrollar un modelo de Deep Learning capaz de **clasificar automáticamente entre 3 clases clínicas** a partir de radiografías: *Paciente sano, Neumonía bacteriana, Neumonía viral*. Todo esto mediante la **clasificación multiclase** donde la entrada son las imágenes, y la salida es la probabilidad por clase aprendida por la CNN.

3. DATASET

El dataset usado proviene del repositorio público de Paul Mooney, descargado desde Kaggle, el cual está compuesto por radiografías de tórax (Rayos X) organizadas en **3 clases clínicas**: NORMAL, BACTERIAL y VIRAL. Se obtuvieron 5.216 imágenes para el training y 624 para test. En test hay 242 casos bacterianos, 234 normales y 148 virales, un desbalance moderado que conlleva al uso de pesos de clase durante el entrenamiento, pero no impide aprender patrones visuales útiles para la clasificación.

Este conjunto se reorganizó de tal forma que las infecciones bacterianas y virales se identifican a partir del nombre del archivo original, buscando sub-strings como "bacteria" o "virus" y separándolos en carpetas dedicadas para el entrenamiento.

4. MÉTRICAS DE DESEMPEÑO

Durante el desarrollo del proyecto se construyeron **tres versiones del modelo**, todas bajo el marco de **redes neuronales convolucionales (CNN)** en formato **secuencial**. Los modelos fueron implementados en el repositorio GitHub, mientras que el dataset fue obtenido desde la plataforma Kaggle, creado por Paul Mooney. Los modelos que se implementaron fueron:

4.1. Transfer Learning Rápido :

El primer modelo desarrollado es una **CNN con transfer Learning**, usando MobileNetV2 en modo congelado para actuar como extractor visual fijo. Con esta elección se priorizó la eficiencia: la red no ajusta los 2.2 millones de pesos convolucionales, sino únicamente la cabeza de clasificación, reduciendo de manera drástica el costo computacional y el riesgo de sobreajuste en las iteraciones iniciales. Con **Global Average Pooling** se redujeron las activaciones, que simplemente resume la información espacial en 1280 características sin añadir parámetros. Sobre ese vector compacto se entrenó una cabeza liviana compuesta por una capa densa de 32 neuronas con activación **ReLU** y un **Dropout bajo (15%)** para evitar co-adaptación, más una capa Softmax de 3 neuronas para clasificación multiclase. Se dejó el optimizador Adam por defecto, aprovechando su ajuste adaptativo de learning rate para facilitar una convergencia sólida sin configuraciones adicionales. En test, el modelo alcanzó 76% de exactitud y F1-macro de 0.73, mostrando **muy buen reconocimiento de neumonía bacteriana** y un desempeño razonable en las tres clases para ser una línea base puramente visual, lo que confirma que la red logró aprender fronteras de decisión útiles con un costo computacional mínimo.

4.2. Transfer Learning Robusto:

El Modelo 2 mantiene el enfoque del primero como classifier secuencial en Keras sobre TensorFlow, pero se diferencia en que **robustece la cabeza densa** en lugar de usar solo 32 neuronas. Aquí se amplió el bloque fully-connected para aprender relaciones no lineales más complejas desde las 1,280 features extraídas por el backbone congelado, y se añadieron **capas de Batch Normalization y Dropout más alto (30% y 25%)** para estabilizar gradientes y reducir sobreajuste, algo que el primero no tenía. También se incluyó EarlyStopping, ReduceLROnPlateau y ModelCheckpoint para frenar aprendizaje cuando el modelo deja de mejorar, ajustar la tasa de aprendizaje y guardar la mejor versión observada en validación. Estos cambios fueron implementados para pasar de una línea base mínima a una **frontera más generalizable y defendible**, conservando rapidez al no descongelar la backbone. Gracias a esto, logró **81% de exactitud y 0.79 de F1-macro en test**, superando al primero y ofreciendo un balance más estable entre las 3 clases. El repositorio de entrega y reproducibilidad sigue alojado en GitHub.

4.3. CNN secuencial con Fine-Tuning parcial:

El Modelo 3 corresponde a una **CNN secuencial** construida con Keras ejecutado sobre TensorFlow. A diferencia de los modelos previos, aquí se activó un **fine-tuning parcial**, liberando las últimas 40 capas del backbone MobileNetV2 para que la red pudiera ajustar internamente los pesos convolucionales y no depender solo de características fijas. La cabeza densa se amplió a 128 y 64 neuronas con **ReLU**, incluyendo Batch Normalization y **Dropout fuerte (30% y 25%)**, lo que estabiliza gradientes y regulariza el aprendizaje mientras se optimizan pesos internos de la CNN. La capa final usa **Softmax de 3 neuronas** para decidir entre NORMAL, BACTERIAL y VIRAL mediante probabilidades aprendidas. Este modelo se eligió como iteración final porque permite **mejor adaptación visual a neumonías complejas**, especialmente las infecciones virales, que son menos representadas en los datos. Se redujo el learning rate a **1e-5** para evitar que el ajuste del backbone destruyera las representaciones preentrenadas. Además, se aplicaron **pesos de clase balanceados** para forzar atención a la clase viral, la más minoritaria, sin requerir aumentar el tamaño de la CNN.

5. MATRIZ DE CONFUSIÓN

Los tres modelos fueron entrenados como CNN secuenciales en TensorFlow, usando representaciones visuales de radiografías curadas por Paul Mooney en Kaggle, reorganizadas en tres clases. A medida que iteramos, cada matriz de confusión nos mostró cómo el modelo aprendía y dónde estaban las fronteras difíciles.

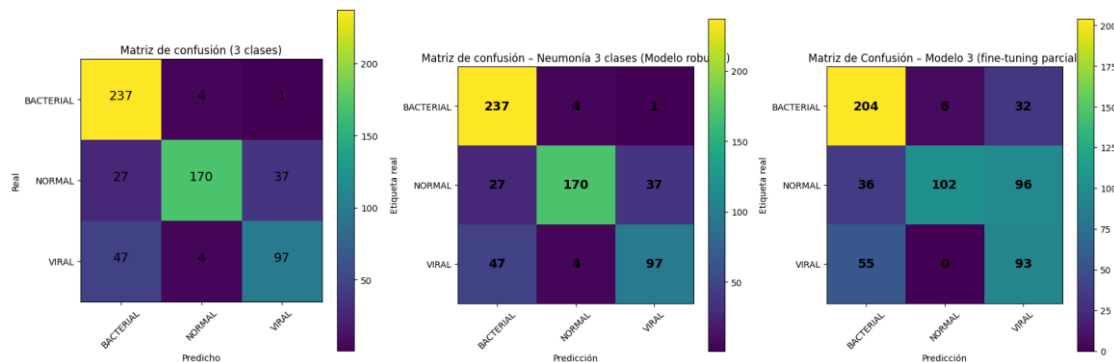


Figura 1. Comparación de Matrices de Confusión – Clasificación Multiclase de Neumonía (Modelos 1, 2 y 3)

El Modelo 1 (Ver figura 1.a)) reveló que la CNN detectó muy bien la neumonía bacteriana, con 204 de 242 casos clasificados correctamente. También reconoció 93 de 148 virales en la clase correcta, pero mostró dificultad para confirmar sanos: 152 de 234 individuos normales fueron bien identificados, mientras que 44 fueron confundidos como bacterianos y 38 como virales. Esta dispersión fuera de la diagonal anticipó que NORMAL y VIRAL comparten patrones visuales menos evidentes con una cabeza tan pequeña y sin callbacks. El Modelo 2 mostrado en la Figura 1.b) mostró un patrón mucho más compacto en la diagonal: 237 de 242 bacterianos, 170 de 234 sanos y 97 de 148 virales quedaron en la clase correcta. Los errores siguieron existiendo, pero fueron menos caóticos: solo 27 normalidades se vieron como bacteria y 37 como virus, mientras que los virales mal clasificados se dividieron entre normal (47) y la bacteria (4). Esta matriz confirmó que el mejora del Modelo 2 no vino por hacer la CNN enorme, sino por **hacerla más estable en su optimización y regularización** de la cabeza. El tercer modelo evidenció que, aunque bacterial sigue fuerte (204 de 242 OK y solo 6→NORMAL), el sano sufrió más: solo 102 de 234 normalidades quedaron bien, con 96 confusiones hacia VIRAL. Viral también mantuvo recall superior a Normal, pero con precisión baja.

6. DISCUSIÓN COMPARATIVA ENTRE MODELOS

El proyecto comparó tres CNN secuenciales con backbone basado en MobileNetV2. Los Modelos 1 y 2 usaron este backbone como **extractor congelado**, lo que permitió aprender representaciones visuales generales y optimizar únicamente la cabeza densa, obteniendo fronteras de decisión más estables. Esto se reflejó en sus matrices de confusión: el Modelo 1 logró 238/242 aciertos en BACTERIAL pero tuvo 44 confusiones VIRAL→BACTERIAL y 63→NORMAL, con 152/234 NORMAL correctos. El Modelo 2, en cambio, redujo la

dispersión y **mejoró el equilibrio global**, conservando sensibilidad a neumonía (237/242 BACTERIAL OK, 170/234 NORMAL OK, 97/148 VIRAL OK) y errores menos caóticos, lo que elevó su exactitud test al 81% y F1-macro a 0.79.

El Modelo 3 introdujo **fine-tuning parcial** (40 capas finales entrenables) con learning rate muy bajo y pocas épocas; aunque es metodológicamente válido, su matriz mostró degradación en NORMAL (102/234 OK, 96 confusiones hacia VIRAL) y VIRAL con menor precisión, bajando el F1-macro a 0.62. Este comportamiento evidencia que la neumonía **es detectable visualmente con alta sensibilidad**, pero la tipificación viral exige más ajuste o datos complementarios.

En el ámbito hospitalario, estos resultados son aplicables hoy: el Modelo 2, por su balance, es idóneo para **triage automático** de rayos X, donde lo crítico es no perder casos (sensibilidad alta en PNEUMONIA). Integrado a flujos de PACS o a consultas de Telesalud, un modelo así puede priorizar lecturas urgentes, apoyar zonas rurales, y reducir la variabilidad diagnóstica. A futuro, incluso en hospitales que avanzan hacia analítica clínica con IA en salud, este tipo de CNN podría escalar a alertar brotes locales, optimizar tiempos de lectura radiológica, y asistir la toma de decisiones tempranas, convirtiéndose en un componente valioso de apoyo diagnóstico visual efectivo y eficiente.