

INTRODUCCIÓN

La pandemia del COVID-19 continúa teniendo un efecto devastador sobre la salud y el bienestar de la población. Un paso importante para hacerle frente a esta situación, es darse cuenta que pacientes se ven afectados por el virus, y tomar las medidas necesarias para que sean tratados. Esta detección se realiza mediante tests y diversos mecanismos. Dentro de estos métodos tenemos el hisopado, y las radiografías de tórax.

Se descubrió en los primeros estudios de COVID-19 que los pacientes presentan anomalías en las imágenes de radiografía de tórax las cuales son características de las personas infectadas. Entrenando modelos convolucionales seríamos capaces de poder predecir la condición del paciente, en función a estas anomalías y diferencias presentes entre una radiografía de una persona normal, con neumonía y con COVID-19.

En este proyecto, tomamos un dataset de más de 15000 imágenes de radiografías de tórax, las cuales fueron trabajadas por nosotros mediante diferentes herramientas de Pandas.

Nos valimos de dos modelos pre entrenados, los cuales fueron ResNet50 y MobileNet. Utilizamos el servicio de Google Collaboratory para poder llevar a cabo este proyecto, puesto que nos veíamos limitados por nuestro Hardware.

Es sabido que mientras más opciones de herramientas y mecanismos de detección de COVID-19, mejor es para el bienestar de la población y la prosperidad de la misma. Con este proyecto buscamos intensificar esta posibilidad y ofrecer a personal de salud y médico la posibilidad de valerse con estos modelos de predicción y utilizarlos para detectar de forma precisa la presencia o no de COVID-19 o neumonía en un paciente.

Tendremos como referencia los resultados obtenidos por la red Open Source COVID-NET.

OBJETIVO

- Entrenar un modelo lo suficientemente preciso para predecir si una persona tiene o no COVID-19 o neumonía en función a una radiografía de tórax.
- Dar soporte y nuevas herramientas al personal de salud y médico para poder combatir la pandemia que azota al mundo.

METODOLOGÍA

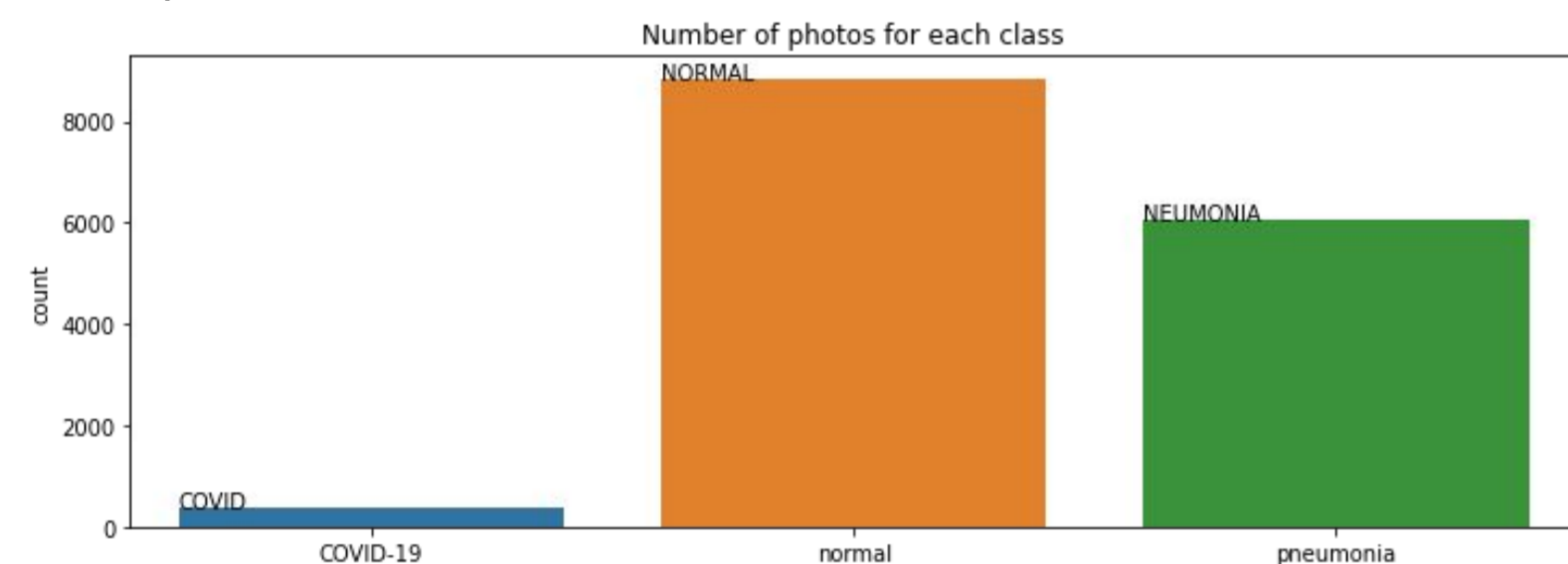
1 PREPARACIÓN DE DATOS

Importamos las imágenes de nuestro dataset a google collaboratory. Hicimos esto para poder valernos de las herramientas de HW que el mismo ofrece. Importamos el CSV con los datos respectivos a cada imagen e hicimos un análisis de previo de los mismos

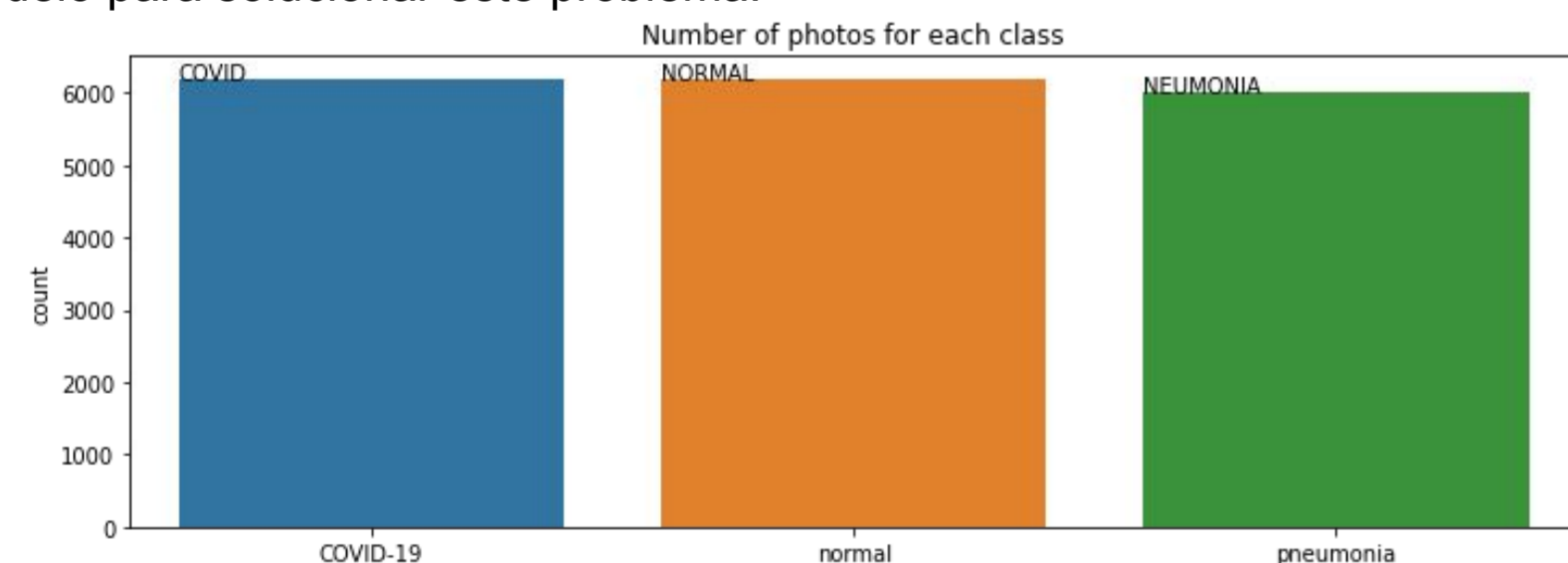
| | td | xcol | ycol | dataset |
|---|----|---|-----------|---------|
| 0 | 3 | SARS-10.1148rg.242035193-g04nr34g0-Flg8a-day0... | pneumonia | cohen |
| 1 | 3 | SARS-10.1148rg.242035193-g04nr34g0-Flg8b-days5... | pneumonia | cohen |
| 2 | 3 | SARS-10.1148rg.242035193-g04nr34g0-Flg8c-day10... | pneumonia | cohen |
| 3 | 7 | SARS-10.1148rg.242035193-g04nr34g04a-Flg4a-day... | pneumonia | cohen |
| 4 | 7 | SARS-10.1148rg.242035193-g04nr34g04b-Flg4b-day... | pneumonia | cohen |
| 5 | 9 | SARS-10.1148rg.242035193-g04nr34g07a-Flg7a-day... | pneumonia | cohen |
| 6 | 9 | SARS-10.1148rg.242035193-g04nr34g07b-Flg7b-day... | pneumonia | cohen |

2 DISTRIBUCIÓN DE LOS DATOS

En este apartado nos encontramos con un dataset sumamente desbalanceado. Como se puede observar en la imagen el mismo presenta un número muy bajo de imágenes con Covid-19. Para que el modelo pudiese entrenar correctamente fue necesario un balanceo de las clases.



Nuestra estrategia para resolver este problema fue realizar una estrategia híbrida entre **Over Sampling** y **Under Sampling**. Lo que hicimos fue multiplicar las imágenes de covid hasta llegar a aproximadamente 6000 de ellas, por otro lado quitamos el excedente de imágenes de la clase Normal, con el fin de obtener un dataset balanceado en 6000 imágenes por clase. Esta estrategia nos diferencia de COVID-NET, ya que ellos simplemente utilizaron los weights del modelo para solucionar este problema.



3

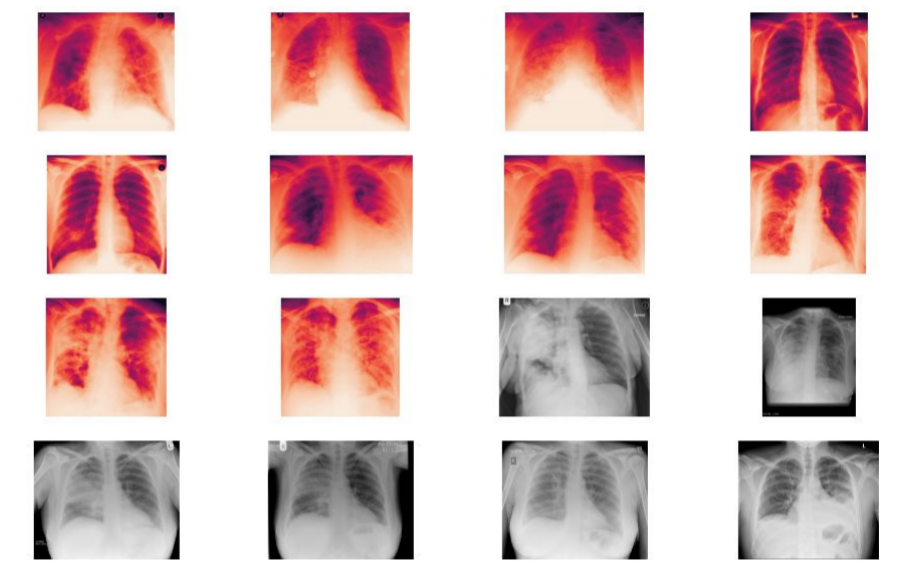
TRATAMIENTO DE IMÁGENES

Como vemos hay algunas imágenes que presentan color, mientras que otras parecen estar en escala de grises.

Para comprobar que modo de color tienen realmente las imágenes, realizamos un análisis por cada clase.

Como resultado obtuvimos, que en las tres clases poseen imágenes con ambos modos de color, es decir, con un solo canal de color (coloreadas de rojo/amarillo) y con 3 canales (las imágenes grises).

Con esta información recabada decidimos utilizar la función 'color_mode' la cual convierte todas las imágenes al modo de color seleccionado por nosotros. En nuestro caso, RGB.



4

ARQUITECTURAS SELECCIONADAS

Utilizamos 2 modelos para entrenar nuestros datos. Nos valimos de transfer learning para poder realizarlo

El primer modelo es **ResNet50**, una arquitectura de red neuronal que aprovecha los principios de diseño residual y los patrones de diseño livianos

El segundo modelo es **MobileNet**. El principal motivo de este modelo, es su liviandad y facilidad para ser ejecutado en dispositivos móviles o con limitaciones de HW.

5

ENTRENAMIENTO

Separamos los datos en tres sectores. **Validación, entrenamiento y testeo.**

El entrenamiento consistió en un total de 15 Epochs por arquitectura, donde el tiempo medio de entrenamiento fue de aproximadamente 2 horas.

La distribución de imágenes para entrenar es del 33% aproximadamente en cada clase. Previamente separamos un total de 400 imágenes las cuales corresponden al grupo de testeo. Aclaramos que estas imágenes nunca fueron vista por los modelos en entrenamiento.

CONCLUSIÓN

Para comparar ambos modelos entrenados calculamos el accuracy de ambos, así como también la sensibilidad de los mismos. El valor de la accuracy junto con la complejidad del modelo, expresada en la cantidad de parámetros, se pueden observar en la siguiente tabla.

| PERFORMANCE | | | |
|--------------|------------|----------|----------|
| Architecture | Params (M) | Acc. (%) | Loss (%) |
| MobileNet | 3.2 | 0.90 | 0.60 |
| ResNet50 | 31.98 | 0.95 | 0.37 |

Como podemos ver, ambos modelos obtuvieron un buen valor de exactitud, siendo superior por 5% el modelo basado en ResNet, de todas formas, es destacable el desempeño del modelo MobileNet, el cual con 10 veces menos parámetros obtuvo un muy buen desempeño.

Estos resultados son sumamente prometedores. Recordemos que la utilidad final de nuestro trabajo es mejorar la efectividad y eficiencia humana frente a la detección de nuevos posibles casos de Covid-19. Bajo ningún aspecto busca reemplazar el criterio humano, al contrario, buscar ser una **herramienta precisa y fiable** para ayudar en la detección de la enfermedad

Incluimos además los datos de Sensibilidad de cada arquitectura. Nuevamente podemos observar que ambas presentan resultados muy similares y prometedores.

| SENSITIVITY OR RECALL (%) | | | |
|---------------------------|----------|--------|----------|
| Architecture | COVID-19 | NORMAL | NEUMONÍA |
| MobileNet | 0.98 | 0.92 | 0.83 |
| Resnet50 | 1.00 | 0.98 | 0.85 |

Presentamos a continuación las matrices de confusión de ambos modelos.



En este trabajo logramos entrenar dos modelos basados en redes neuronales convulsionadas para la detección de casos de COVID-19 en radiografías de tórax. Se evaluó, además la viabilidad de llevar esta solución a entornos móviles, como por ejemplo un smartphone. En este apartado los resultados con la red MobileNet, son prometedores, la misma ocupa un espacio de 25MB y le toma aproximadamente unos 30ms por foto. Desde nuestra postura como ingenieros, vemos de suma utilidad esta propuesta y esperamos sea llevada a la práctica, para que entre todos le demos solución a esta situación que tanto nos atormenta.

A diferencia de nuestro trabajo de referencia, nuestro modelo superó la precisión que ellos habían obtenido, COVID-Net obtuvo una precisión del 92.4% mientras que nuestro modelo ResNet50 obtuvo 95.0%, puede que las estrategias para tomar los datos desbalanceados haya marcado una pequeña diferencia, además nuestra cantidad de parámetros es menor, 116.6M contra nuestro modelo ResNet que posee 31.98M.

Nuestro trabajo continuará con un análisis mas preciso y detallado de los modelos obtenidos, con el fin de poder brindar una solución aún mas completa y documentada a los posibles interesados.