

Detección de signos de COVID-19 en radiografías de tórax a través del procesamiento digital de imágenes con redes neuronales convolucionales

Guido Sebastián Armoa¹, Nuria Isabel Vega Lencina¹ y Karina Beatriz Eckert¹

¹Universidad Gastón Dachary, Posadas, Misiones, Argentina.
{guidoarmao777, nurivega.nv, karinaeck}@gmail.com

Resumen. El presente trabajo de se vio motivado por la histórica pandemia que afectó a todo el mundo desde fines del 2019. El diagnóstico temprano de la enfermedad del COVID-19 es crucial para el tratamiento y control de la enfermedad. En este contexto, la radiografía de tórax juega un papel importante; precisamente este trabajo tiene como objetivo el desarrollo y análisis de un prototipo de software para el reconocimiento de signos de COVID-19 en radiografías de tórax, a partir del procesamiento de imágenes utilizando redes neuronales convolucionales. Se propone un modelo de red neuronal convolucional para detectar signos de COVID-19 en imágenes de radiografías de tórax. La metodología propuesta experimenta y analiza el comportamiento de la misma, mediante el entrenamiento de la red utilizando distintos conjuntos de datos disponibles públicamente. Los resultados experimentales demuestran la efectividad y las limitaciones de la metodología propuesta, logrando un 79% de exactitud en la clasificación.

Palabras clave. Procesamiento digital de imágenes, Redes neuronales artificiales, Redes neuronales convolucionales, Radiografía de tórax.

1 Introducción

El coronavirus 2019 (COVID-19) es una enfermedad infecciosa que ha afectado a todo el mundo y causado millones de muertes [1].

Un importante obstáculo para controlar la propagación de esta enfermedad es la falta de experiencia o conocimiento por la novedad de la enfermedad, y la escasez de pruebas. Hasta el momento, para saber si un paciente está o no infectado las técnicas más comúnmente utilizadas por los médicos son la reacción en Cadena de la Polimerasa de Transcripción Inversa (PCR, por sus siglas en inglés de Polymerase Chain Reaction) y los test rápidos con anticuerpos. Las primeras son consideradas las más fiables hasta el momento, pero se necesitan de 4 a 6 horas para obtener resultados. Sumado a esto, dada la enorme demanda que tienen los laboratorios autorizados, estos resultados pueden demorarse varios días y además de esto los kits de prueba de PCR son muy escasos [2]. Por otra parte, los test rápidos permiten conocer en 10 o 15 minutos si una persona está o no infectada, pero tienen sensibilidad inferior al 30% y no son aconsejables para una rutina de diagnóstico [3].

El diagnóstico temprano de la enfermedad del nuevo COVID-19 es crucial para el tratamiento y control de la enfermedad. Esto motiva a estudiar formas alternativas de prueba, como las Radiografías de Tórax (RXT). Las pruebas de imagen tienen un papel importante en la detección y manejo de los pacientes, se han utilizado para apoyar el diagnóstico, determinar la gravedad de la enfermedad, guiar el tratamiento y valorar la respuesta terapéutica [4].

La RXT generalmente es el estudio por imágenes de primera línea en la evaluación de pacientes con sospecha de COVID-19 por su utilidad, disponibilidad y bajo coste junto con la evaluación clínica y los exámenes de laboratorio, colabora en la evaluación inicial y en el seguimiento de esta enfermedad. Puede ser empleada como método para la organización de la atención de las personas según los recursos existentes y las necesidades de los individuos en determinados escenarios, acelerando su proceso de clasificación, ingreso hospitalario y tratamiento [3].

Actualmente se está viviendo un enorme desarrollo en la tecnología asociada a la Inteligencia Artificial (IA), la parte de la ciencia que se ocupa del diseño de sistemas de computación inteligentes, dando lugar a nuevas herramientas y aplicaciones. Los sistemas basados en IA integran algoritmos como el aprendizaje automático y aprendizaje profundo, en entornos complejos que permiten la automatización [4], [5].

El aprendizaje automático (ML, Machine Learning), tiene como objetivo desarrollar técnicas que permitan a las computadoras aprender. Esta trata de crear algoritmos capaces de generalizar comportamientos y reconocer patrones a partir de una información suministrada en forma de ejemplos. El aprendizaje profundo, es un subcampo dentro de ML, definido como un algoritmo automático estructurado o jerárquico que emula el aprendizaje humano utilizando distintas estructuras de redes neuronales con el fin de obtener ciertos conocimientos y lograr el aprendizaje. Destaca porque no requiere de reglas programadas previamente, sino que el propio sistema es capaz de “aprender” por sí mismo para efectuar una tarea a través de una fase previa de entrenamiento. La principal distinción del aprendizaje profundo se establece por su estructura y procesamiento de la información el cual imita las redes neuronales del cerebro humano [6], [7], [8].

Una red neuronal es un modelo de computación cuya estructura de capas se asemeja a la estructura interconectada de las neuronas en el cerebro, con capas de nodos conectados. Puede aprender de los datos, de manera que se puede entrenar para que reconozca patrones, clasifique datos y pronostique eventos futuros. Uno de los tipos más populares de Redes Neuronales Artificiales (RNA) son las conocidas como Redes Neuronales Convolucionales (CNN, por sus siglas en inglés Convolutional Neural Network). Las mismas, eliminan la necesidad de una extracción de características manual, por lo que no es necesario identificar las características utilizadas para clasificar las imágenes. Funciona mediante la extracción de características directamente de las imágenes. Las características relevantes no se entrenan previamente; se aprenden mientras la red se entrena con una colección de imágenes. La CNN dispone de decenas o cientos de capas ocultas que aprenden a detectar diferentes características de una imagen. Se aplican filtros a cada imagen de entrenamiento con distintas resoluciones, los filtros pueden variar desde características muy simples como el brillo y los bordes, hasta más complejas, como las características que definen el objeto de manera única.

Es decir, que las primeras capas pueden detectar líneas o curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o una silueta. La salida de cada imagen convolucionada se emplea como entrada para la siguiente capa. Estas redes son particularmente útiles para encontrar patrones en imágenes para reconocer objetos, caras y escenas [9].

Es por esto que teniendo en cuenta la situación en la que se vive la pandemia en la Argentina, se propone el desarrollo y entrenamiento de una CNN para poder analizar en forma automatizada radiografías de tórax de pacientes con sospecha COVID-19. Para así, en pocos minutos, contribuir a saber si un paciente presenta algún signo típico de esta afección y ayudar a los médicos a tomar decisiones sobre la internación y el tratamiento inmediato, mientras se esperan los resultados de laboratorio.

El desarrollo del presente trabajo se estructura en secciones, en cada una de ellas se detallan las actividades llevadas a cabo para lograr los objetivos planteados. En la sección 2 se describe la metodología propuesta y su desarrollo. Posteriormente, en la sección 3, se exponen las configuraciones y desarrollo de cada una de las pruebas realizadas, como así también, la comparación y análisis de los resultados obtenidos en cada una de ellas. Finalmente, en la sección 4, se presentan las conclusiones obtenidas y los trabajos a futuro propuestos.

2 Propuesta

Para la implementación de la CNN, se siguieron una serie de pasos, comenzando por la carga y configuración del dataset, luego se siguió con la creación del modelo de red con cada una de sus capas, incluyendo las capas de convolución, max pooling, aplanamiento y capa de densa. Por último, se procedió al entrenamiento de la red y las pruebas de la misma.

2.1 Obtención del dataset

Se decidió utilizar el conjunto de imágenes de una base de datos abierta creada por un equipo de investigadores y médicos, pertenecientes a la Universidad de Qatar, Doha, y la Universidad de Dhaka, Bangladesh, junto con sus colaboradores de Pakistán y Malasia [9]. La misma está compuesta de radiografías de tórax con casos positivos de COVID-19, casos Normales (pulmones sanos), casos de neumonía viral y casos de opacidad pulmonar. Todas las imágenes se encuentran en formato Portable Network Graphics (PNG), y con una resolución de 299x299 píxeles.

Para la realización de las pruebas, se generaron distintas distribuciones de imágenes en diferentes datasets. El dataset número 1 (formado por 10972 imágenes para la clase COVID y 10192 para la clase no COVID) considera como COVID a las imágenes con opacidad pulmonar, COVID positivo y neumonía viral. El dataset número 2 (4960 imágenes por cada clase) considera como COVID a las imágenes con COVID positivo y neumonía viral. El dataset número 3 (3615 imágenes por clase) considera COVID solo a las imágenes clasificadas con COVID positivo. El dataset número 4 (2462 imágenes por clase) considera COVID solo a las imágenes con COVID positivo y

además se encuentra depurado, es decir, que se eliminaron previamente las imágenes con errores o de mala calidad que pudieran sesgar los resultados. En todos los casos, solo se consideran como no COVID las imágenes de pulmones normales.

El objetivo de formar distintos grupos de datos, es encontrar el conjunto de imágenes que demuestre el mejor rendimiento en el aprendizaje de la red. Para ello, se determinó la realización de 3 entrenamientos con cada conjunto de prueba para evaluar la capacidad de aprendizaje del modelo propuesto.

Se tomó el 80% del conjunto de imágenes seleccionadas para que formen parte del conjunto de entrenamiento, el 10% para el conjunto de validación y el 10% restante para el conjunto de prueba.

2.2 Desarrollo del prototipo

El prototipo desarrollado fue implementado en Python, dado que es un lenguaje multiparadigma y multinivel, de código abierto y gratuito. El mismo fue seleccionado ya que es ideal para la implementación de técnicas de IA. A su vez, se utilizó la plataforma de Anaconda ya que puede crear e implementar modelos de aprendizaje profundo que utilizan redes neuronales. Además, la misma se integra fácilmente con herramientas como TensorFlow y Keras para poder crear y entrenar modelos de redes neuronales, incluidas redes neuronales convolucionales [10]–[13].

Para la creación de la red neuronal se utilizó la función Sequential() que agrupa una pila lineal de capas en un archivo tf.keras.Model [14].

Luego se procedió a añadir las capas que forman parte de la red. Para agregar cada capa se utilizó la función add(). Los parámetros que recibe esta función son los siguientes: Número de filtros utilizados, tamaño del filtro, la variable padding referente al filtro en las esquinas, la altura y longitud que tienen las imágenes de entrada y la función de activación utilizada. A su vez, se añadió una capa de MaxPooling, para la cual se utilizó un filtro de tamaño 2x2.

Con ésta última capa finaliza la primera convolución, en la figura 1 se observa de manera gráfica la arquitectura de la misma.

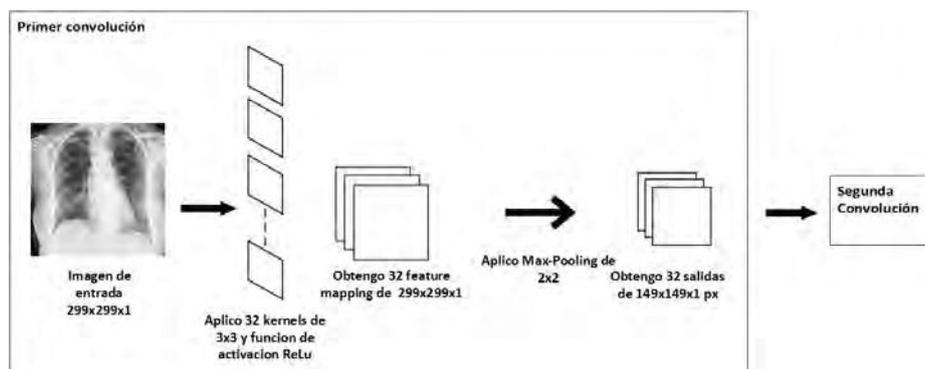


Figura 1. Arquitectura de la primera convolución.

A continuación, se agrega la segunda y tercera capa de convolución, en este caso, capas ocultas, por lo que no se especifica el tamaño de la imagen (inputShape). En ambos casos los parámetros que reciben son los siguientes: El número de filtros, el tamaño del filtro, la variable padding y la función de activación utilizada.

Al finalizar, se añade una capa de MaxPooling, para el cual se utilizó un filtro de tamaño 2x2, con esto finalizan las convoluciones.

Por último, se añade una capa de aplanamiento para transformar la red multidimensional a una dimensión para que luego se pase a una red neuronal tradicional. En la figura 2, se puede apreciar la arquitectura final de la red neuronal convolucional completa, formada por las 3 capas de convoluciones, el aplanamiento y el perceptrón multicapa. Las salidas de la MLP [15] (perceptrón multicapa, por sus siglas en inglés multilayer perceptrón) representan a las salidas de toda la red, dando como resultado un porcentaje de acierto para cada clase.

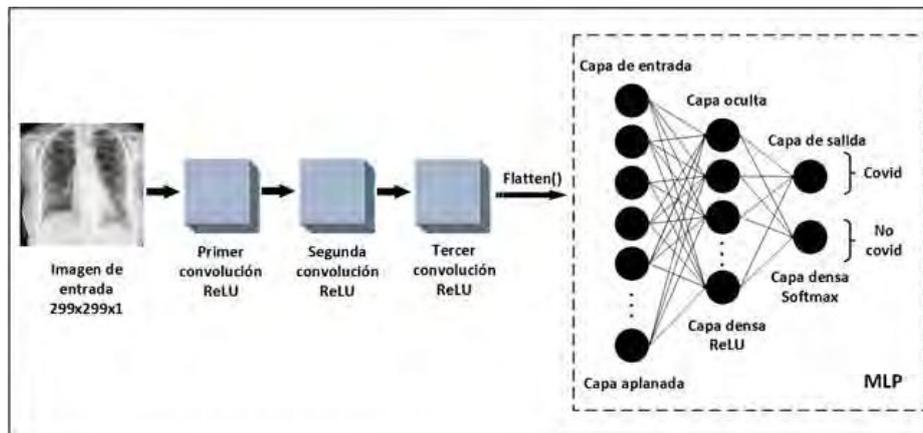


Figura 2. Arquitectura completa del prototipo con CNN.

2.3 Entrenamiento de la red

El siguiente paso es la especificación del entrenamiento y el dataset. Para ello se utilizó la función `fit_generator()`, que recibe los siguientes parámetros: conjunto de entrenamiento, número de pasos por épocas, número de épocas (en este caso 20), conjunto de validación y número de pasos de validación [16].

Una vez entrenada la red, el modelo se guarda en el directorio correspondiente, en este caso, en la carpeta específica llamada modelo, la estructura se guarda bajo el nombre "modelo.h5" y los pesos bajo el nombre de "pesos.h5".

Las cuatro métricas que devuelve el entrenamiento de la red tanto para el conjunto de entrenamiento como para el conjunto de validación son la exactitud del aprendizaje y el error.

3 Pruebas y Resultados

Con el fin de validar y analizar la funcionalidad del prototipo desarrollado con la metodología propuesta detallada en la sección anterior, se han planeado las siguientes pruebas:

Teniendo en cuenta el cuantioso número de imágenes de prueba, se recurrió al uso de una matriz de confusión, una herramienta útil para medir el desempeño en la clasificación de las imágenes de manera eficiente [17], [18].

Se realizaron tres pruebas por cada dataset, obteniendo un total de 12 pruebas. Es decir, cada prueba consistió en entrenar el modelo con su respectivo conjunto de datos (tabla 1). Una vez finalizado el entrenamiento, se utiliza la matriz de confusión, para poder evaluar el desempeño de la red ante el conjunto de prueba y analizar los valores de cada una de sus métricas (tabla 2). Probando automáticamente todas las imágenes del conjunto de prueba y comparando sus etiquetas reales con las predicciones hechas por la red.

A continuación, en la tabla 1 se presenta un cuadro comparativo con los resultados de error y exactitud obtenidos durante el entrenamiento de cada red. A su vez, en la tabla 2 se presentan los valores de las métricas obtenidas a partir de las matrices de confusión generadas en cada prueba.

Tabla 1. Resumen resultados de entrenamientos.

Dataset	Prueba	Error entrenamiento	Error validación	Exactitud entrenamiento	Exactitud validación
1	1	0,3840	0,3319	0,8292	0,8552
	2	0,3575	0,3030	0,8478	0,8725
	3	0,3478	0,2951	0,8511	0,8730
	4	0,2956	0,6591	0,8715	0,6099
2	5	0,2928	0,6354	0,8806	0,6069
	6	0,2939	0,5788	0,8752	0,6633
	7	0,3993	0,2759	0,8009	0,8920
3	8	0,3993	0,2759	0,8009	0,8920
	9	0,3602	0,2588	0,8387	0,9034
4	10	0,3510	0,3900	0,8410	0,8208
	11	0,3971	0,3815	0,8122	0,8562
	12	0,4212	0,3944	0,7909	0,8375

En la tabla 1 se puede observar que la gran mayoría de los entrenamientos tuvieron resultados que en primera instancia parecen prometedores. Se destaca el entrenamiento de la prueba 9 por lograr alcanzar una exactitud de 90% y un error de tan solo 0,25 sobre el conjunto validación. Por otra parte, la prueba 5 fue la prueba con los resultados menos favorables, contando con un error de 0,63 y una exactitud de 60% sobre el conjunto de validación.

Tabla 2. Resumen de las pruebas.

Dataset	Prueba	Precisión		Sensibilidad		F1-score		Exactitud
		Covid	No Covid	Covid	No Covid	Covid	No ovid	
1	1	0,6667	0,7757	0,8310	0,5845	0,7398	0,6667	70,78%
	2	0,5804	0,8560	0,9470	0,3153	0,7197	0,4609	63,11%
	3	0,5142	0,9118	0,9941	0,0609	0,6778	0,1142	52,75%
2	4	0,4995	0,5000	0,9455	0,0544	0,6536	0,0982	49,95%
	5	0,4945	0,4400	0,9152	0,0665	0,6421	0,1156	49,04%
	6	0,4995	0,0000	1,0000	0,0000	0,6662	0,0000	49,95%
3	7	0,5142	0,9545	0,9972	0,0580	0,6786	0,1094	52,76%
	8	0,5385	0,8378	0,9669	0,1713	0,6917	0,2844	56,91%
	9	0,4794	0,4727	0,5470	0,4061	0,5110	0,4368	47,65%
4	10	0,7912	0,7984	0,8008	0,7886	0,7960	0,7935	79,47%
	11	0,6579	0,6858	0,7114	0,6301	0,6836	0,6568	67,07%
	12	0,6667	0,6549	0,6423	0,6789	0,6542	0,6667	66,06%

Realizando un breve análisis de los resultados obtenidos con cada conjunto de datos de prueba, presentes en la tabla 2 se puede observar que todas las pruebas realizadas con un mismo dataset dieron resultados similares entre ellas.

El dataset 1, es el conjunto que presenta mayor variación en sus resultados, alcanzando una exactitud del 71% en la prueba 1 (tabla 2).

Las pruebas realizadas con el dataset número 2 se destacan por obtener los resultados menos favorables, en ningún caso logró superar el 50% de exactitud. El hecho de tener la misma cantidad de imágenes de cada clase, un 50% de exactitud indica que la red clasifica prácticamente a todas las imágenes como una misma clase.

Para el caso del dataset número 3, se ve un comportamiento similar a las pruebas realizadas con el dataset número 2, con resultados poco destacables. Es conveniente resaltar que la prueba 9 realizada con dicho conjunto de datos, presenta el mejor valor de exactitud alcanzando un 90% sobre el conjunto de validación durante el entrenamiento (tabla 1). Hay que tener en cuenta que, si bien el valor de exactitud en el entrenamiento es alto, no asegura un buen rendimiento en la clasificación de las imágenes, tal como se puede apreciar en los resultados de clasificación de la tabla 2.

Para finalizar, el dataset número 4 se destaca por obtener los mejores resultados, superando en todos los casos el 66% de exactitud sobre el conjunto de prueba. Resaltando la prueba 10 donde se obtuvo una exactitud del 79% en la clasificación de las imágenes del conjunto de prueba, en la figura 3 se observa la matriz de confusión generada.

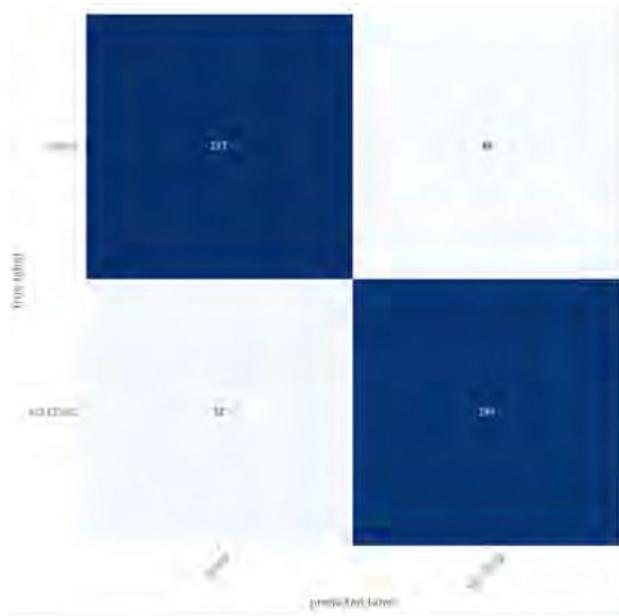


Figura 3. Matriz de confusión de la prueba 10

3.1 Validación de la red

Finalizadas las pruebas anteriores, se seleccionó la red que mejor desempeño demostró en la clasificación de imágenes, dicha red fue la obtenida en la prueba 10 con el dataset 4 (obteniendo un 79% de exactitud) y se procedió a la validación de la misma.

Ésta actividad fue realizada con la ayuda de un profesional en el área de diagnóstico por imágenes del Hospital Escuela de Agudos Dr. Ramón Madariaga.

Se suministró al profesional un dataset con 52 imágenes sin etiquetas provenientes del conjunto de prueba del dataset utilizado para el entrenamiento de la red, para las cuales se solicitó al mismo que las clasifique según su criterio. De las 52 imágenes de radiografías de tórax, 37 de ellas fueron clasificadas por el profesional de la misma manera que la red, 3 no pudieron ser definidas bajo ninguna clasificación por lo cual se decidió no tomarlas en cuenta, y las 12 imágenes restantes fueron clasificadas contradiciendo a las salidas obtenidas por la red.

Se obtuvo un 75% de coincidencia con la clasificación realizada por el médico. Es conveniente destacar, que el criterio de clasificación varía dependiendo de cada persona, dado que diferentes factores como la experiencia o el error humano pueden afectar a la decisión tomada. Por consiguiente, se consideran favorables los resultados obtenidos en la validación.

4 Conclusiones

Con el fin de cumplir con los objetivos de este trabajo fue necesario un exhaustivo proceso de investigación y análisis sobre los signos del COVID-19 en las radiografías de pulmón, profundizar y comprender la aplicación de las redes neuronales convolucionales en la clasificación de imágenes, diseñar la arquitectura de la red y los diferentes datasets, implementar el prototipo, entrenar y probar la red con cada conjunto de datos, y validar el funcionamiento del prototipo junto a un profesional. Para finalmente, analizar los resultados obtenidos y determinar la viabilidad de la propuesta.

Tras el análisis, se puede observar que las tablas de resultados y las matrices de confusión demuestran que el dataset número 4 presentó el mejor desempeño en la clasificación de imágenes, alcanzando un 79% de exactitud. En base a esto, se puede deducir que la depuración realizada sobre el mismo fue el factor clave para lograr esta mejora. En este sentido, las pruebas indican que la eliminación de imágenes de mala calidad produce una mejora exponencial en los resultados de la clasificación, demostrando la sensibilidad que poseen las CNN ante este tipo de imágenes y la importancia de tener un conjunto de datos representativo.

Dentro del análisis expuesto es posible observar que la red entrenada posee dos grandes limitaciones, la técnica utilizada para realizar la radiografía, lo cual puede afectar a la calidad de la imagen, y las afecciones crónicas que pueda presentar el paciente. Por otra parte, es importante destacar el gran costo computacional que demandan las CNN para su desarrollo.

Tal y como se ha podido comprobar, el prototipo obtenido resultó prometedor para el médico especialista en diagnóstico por imágenes, ya que, mediante el mismo se logró la clasificación de radiografías de pulmón con signos de COVID-19 de manera eficaz, en cuestión de segundos una vez obtenida la radiografía. De esta manera, se podría generar una alternativa para cooperar con los organismos de salud en el diagnóstico de COVID-19.

De acuerdo con lo expresado anteriormente, se demostró que con la aplicación de redes neuronales convolucionales se logró construir un prototipo capaz de reconocer eficazmente los signos de COVID-19 en radiografías de pulmón. De esta manera, queda en evidencia que el presente estudio logró cumplir satisfactoriamente los objetivos propuestos al inicio de la investigación.

El prototipo desarrollado puede presentar mejoras a futuro, una de ellas es implementar una red que, en vez de realizar una clasificación binaria como en este trabajo, realice una clasificación de múltiples clases. Además, se propone utilizar transferencia de aprendizaje y probar la eficiencia entre los distintos modelos disponibles (ResNet, VGG, etc.).

5 Referencias

- [1] “COVID-19 Visualizer.” <https://www.covidvisualizer.com/> (accessed Aug. 12, 2020).
- [2] E. M. Chamorro, A. D. Tascón, L. I. Sanz, S. O. Vélez, and S. B. Nacenta, “Diagnóstico

- radiológico del paciente con COVID-19,” *Radiologia*, vol. 63, no. 1, p. 56, Jan. 2021, doi: 10.1016/J.RX.2020.11.001.
- [3] E. Martínez Chamorro, A. Díez Tascón, L. Ibáñez Sanz, S. Ossaba Vélez, and S. Borruel Nacenta, “Radiologic diagnosis of patients with COVID-19,” *Radiologia*, vol. 63, no. 1, pp. 56–73, Jan. 2021, doi: 10.1016/j.rx.2020.11.001.
- [4] “¿Qué es la inteligencia artificial (IA)? - MATLAB & Simulink.” <https://la.mathworks.com/discovery/artificial-intelligence.html> (accessed Sep. 24, 2020).
- [5] *The Handbook of Artificial Intelligence*. Elsevier, 1981.
- [6] “Introducción al Aprendizaje Automático - Fernando Sancho Caparrini.” <http://www.cs.us.es/~fsancho/?e=75> (accessed Sep. 07, 2020).
- [7] “¿Qué es el Deep Learning? | SmartPanel.” <https://www.smartpanel.com/que-es-deep-learning/> (accessed Sep. 07, 2020).
- [8] “Deep Learning - Libro online de IAAR.” <https://iaarbook.github.io/deeplearning/> (accessed Nov. 17, 2021).
- [9] “COVID-19 Radiography Database | Kaggle.” <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> (accessed May 21, 2021).
- [10] “Anaconda | Use Cases.” <https://www.anaconda.com/use-cases> (accessed Mar. 09, 2022).
- [11] “Librerías ML: TensorFlow, Scikit-learn, Pytorch y Keras - Platzi.” https://platzi.com/blog/librerias-de-machine-learning-tensorflow-scikit-learn-pytorch-y-keras/?gclid=CjwKCAiA4KaRBhBdEiwAZi1zzm5QrcLNP_R6BqpM9DZj0H6v9yvzsHEXltymGQzgu3FfBQaImjc_hoCmHMQAvD_BwE&gclidsrc=aw.ds (accessed Mar. 10, 2022).
- [12] “Keras: the Python deep learning API.” <https://keras.io/> (accessed Jul. 04, 2022).
- [13] “TensorFlow.” <https://www.tensorflow.org/?hl=es-419> (accessed Jul. 04, 2022).
- [14] “The Sequential class.” <https://keras.io/api/models/sequential/> (accessed Mar. 02, 2022).
- [15] J. Hilerá and V. Martínez, “Redes neuronales artificiales: fundamentos, modelos y aplicaciones,” *Madrid: Ra-ma*, no. January, p. 9, 1995, [Online]. Available: <http://en.scientificcommons.org/7007722>.
- [16] “fit_generator function - RDocumentation.” https://www.rdocumentation.org/packages/keras/versions/2.4.0/topics/fit_generator (accessed Jul. 04, 2022).
- [17] “La matriz de confusión y sus métricas – Inteligencia Artificial –.” <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/> (accessed Mar. 15, 2022).
- [18] “Evaluando los modelos de Clasificación en Aprendizaje Automático: La matriz de confusión. | profesorDATA.com.” <https://profesordata.com/2020/08/07/evaluando-los-modelos-de-clasificacion-en-aprendizaje-automatico-la-matriz-de-confusion-claramente-explicada/> (accessed Apr. 04, 2022).