

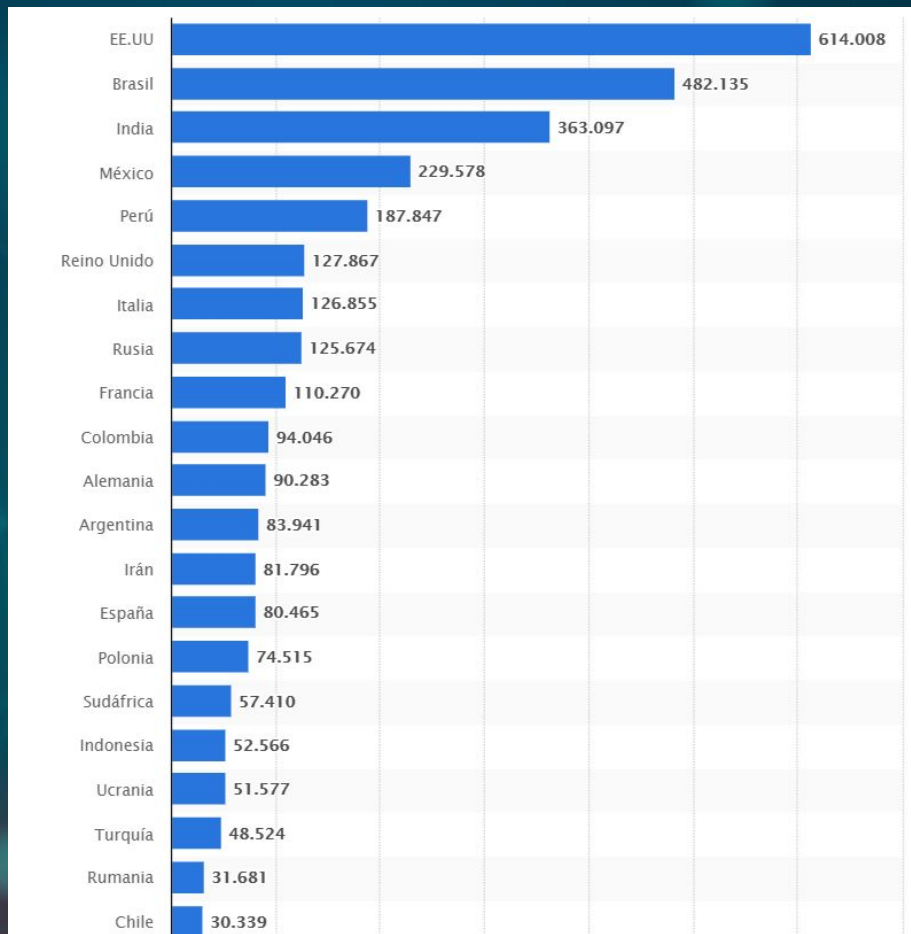
Clasificación de Imágenes X-Ray Covid-19

Tania Delgado & Felipe Quisel

Es la enfermedad causada por el nuevo coronavirus conocido como SARS-CoV-2. La OMS (Organización Mundial de la Salud) tuvo noticia por primera vez de la existencia de este nuevo virus el 31 de diciembre de 2019, al ser informada de un grupo de casos de neumonía vírica que se habían declarado en Wuhan (República Popular China).

¿Qué es Covid-19?





El covid-19 (SARS-CoV-2) ha producido inmensas consecuencias a nivel mundial sea en las personas como también en la economía de los países, como se puede observar en la imagen (datos reales, muertes por Covid-19) en un periodo de estudio del 11 de junio 2021.

- El proyecto aborda un problema de clasificación, mediante una entrada de radiografías de tórax, las que poseen un diagnóstico positivo en Covid-19 o negativo.
- La clasificación aplicada es de tipo Supervisada.

Descripción del problema

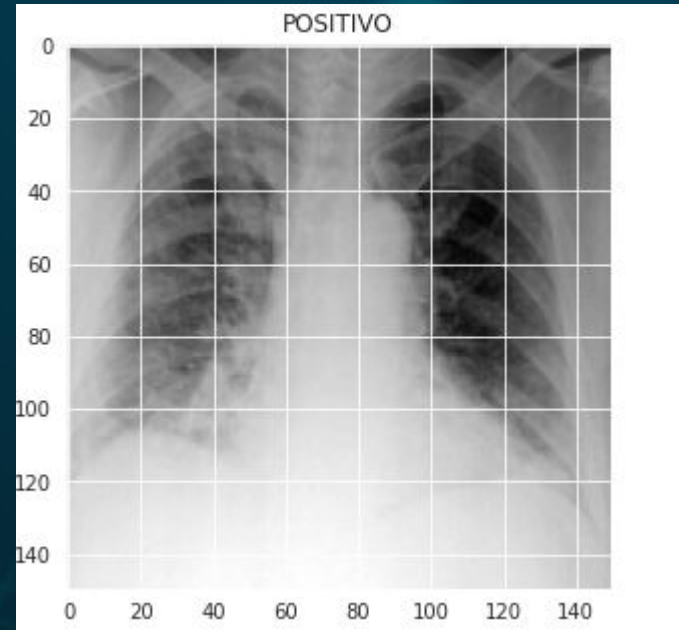
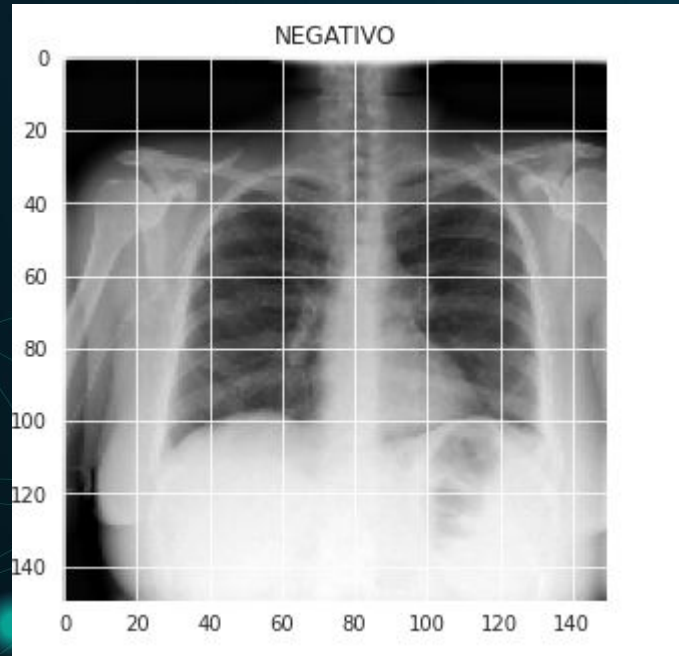
Sobre el DataSet

El dataset a utilizar tiene su origen en Kaggle, el cual contiene una recopilación de imágenes de radiografía de tórax de pacientes COVID-19 positivos y negativos correspondientes a una competencia para obtener el modelo en resultados. Este conjunto de datos comprende aproximadamente 16.351 imágenes de rayos x.

Por otro lado se extrajeron 500 imágenes de clase positiva correspondiente a otro dataset.

Type	COVID-19 Negative	COVID-19 Positive	Total
train	13793	2158	15951
test	200	200	400

Diagnósticos (Ejemplos)



Conjunto de Entrenamiento

El conjunto de entrenamiento se compone de:

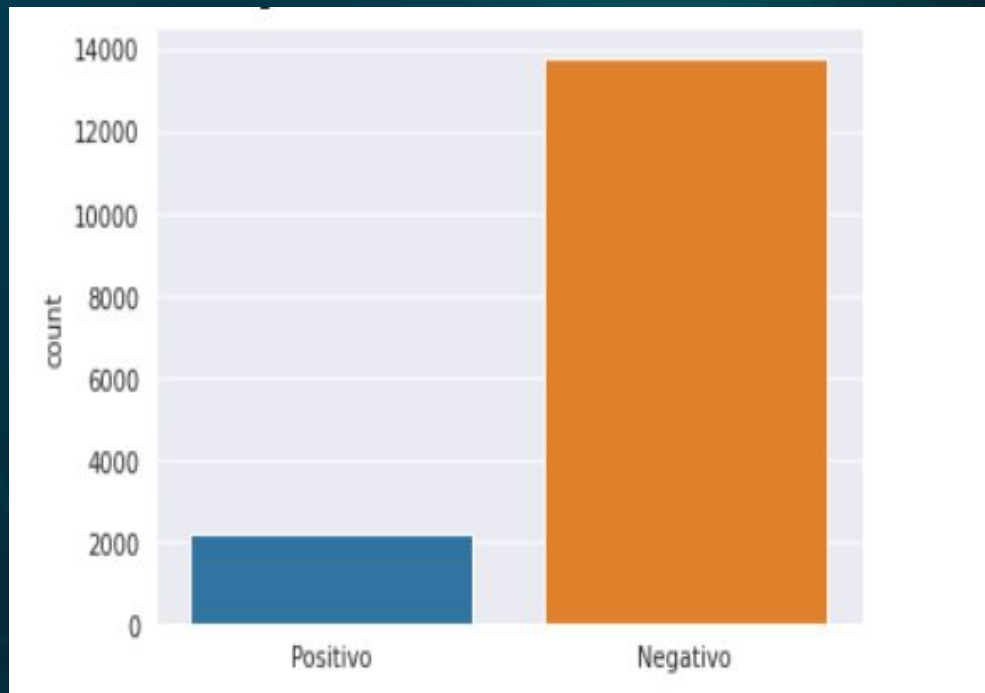
- 2158 imágenes de la clase Positiva
- 13793 de la clase Negativa.

Por lo tanto está desbalanceado.

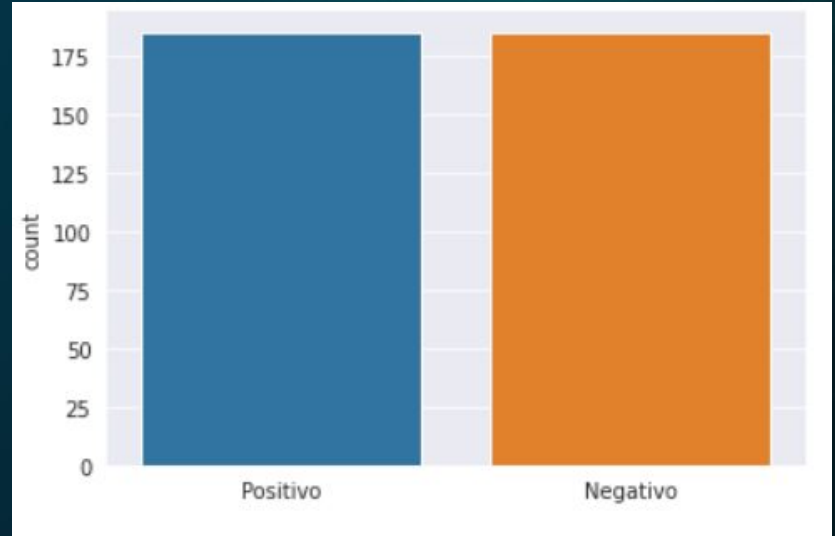
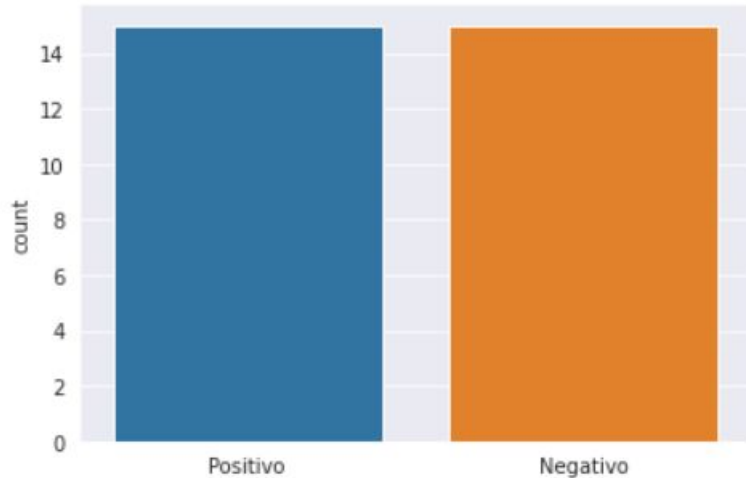
Experimento 2 se utiliza Data Augmentation

Experimento 3 se añaden 500 imágenes Positivas

Experimento 4 se duplican las imágenes Positivas



Conjunto de Prueba y Validación



El archivo val cuenta con 15 para las clases Positivo y Negativo y la carpeta test cuenta con 185 imágenes de cada clases.

Python Keras



Esta es una librería de código abierto con licencia MIT escrita en Python se basa principalmente en el trabajo de Francois Chollet



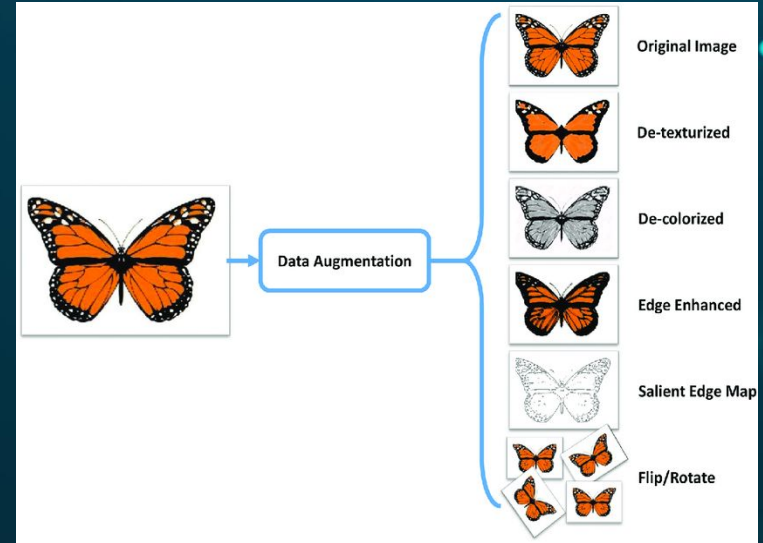
La primera versión de este software multiplataforma se lanzó el 28 de marzo de 2015. El objetivo de la biblioteca es acelerar la creación de redes neuronales: para ello, Keras no funciona como un framework independiente.

Data Augmentation

Exp. 2, 3 y 4

Para el aumento de datos, se aplicó:

- Rotación aleatoriamente algunas imágenes de entrenamiento 30 grados
- Zoom aleatorio en un 20% algunas imágenes de entrenamiento
- Cambiar las imágenes de forma aleatoria horizontalmente en un 10% del ancho
- Cambiar las imágenes verticalmente al azar en un 10% de la altura
- Giro en las imágenes horizontalmente al azar.



CNN (Convolutional Neural Network)

Aplicación: COMPUTER VISION (DNN)

- Tipo especializado de red neuronal diseñada para trabajar en la clasificación de imágenes en múltiples dimensiones
- Las imágenes están formadas por píxeles
- Cada píxel está representado por un número entre 0 y 255

Funcionamiento CNN

Es el bloque de construcción central de una CNN, y es donde ocurre la mayor parte de los cálculos.



**Convolutional
Layer**

Lleva a cabo una reducción de dimensionalidad, lo que reduce el número de parámetros en la entrada.



Pooling Layer

Realiza la conexión de cada nodo de la capa de salida directamente a un nodo de la capa anterior.

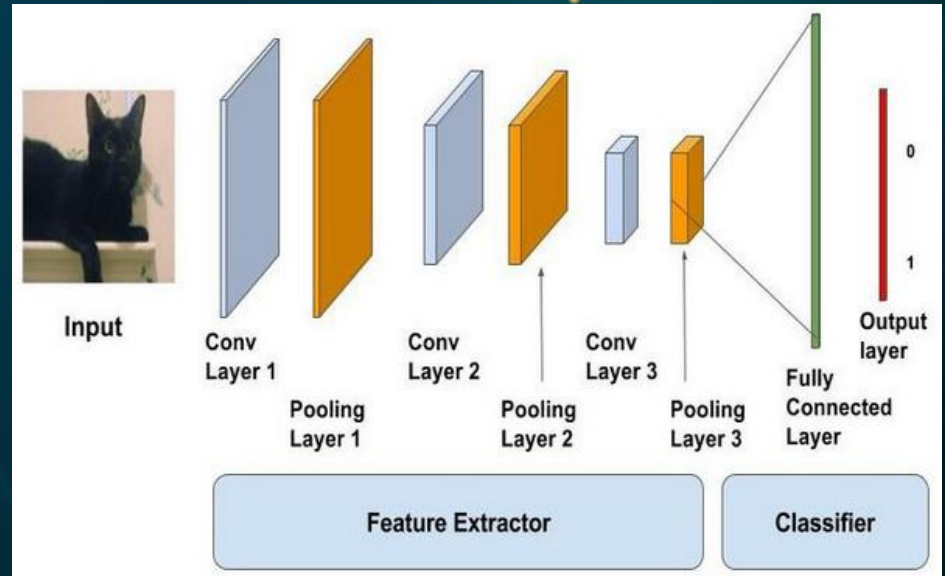


**Fully-Connected
Layer**

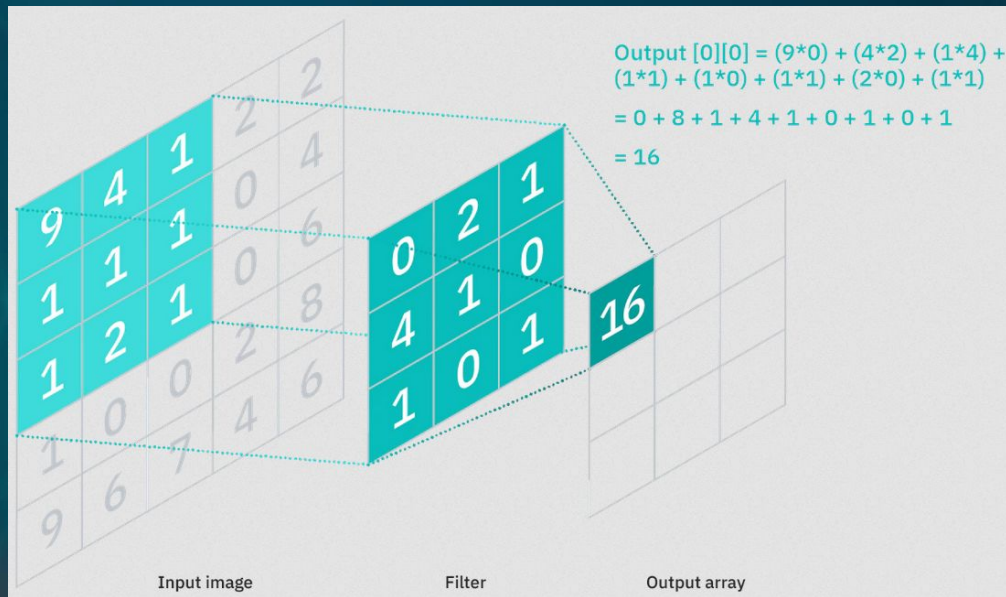
Configuraciones del Modelo

El modelo se compone de:

- 5 Capas convolucionales
- 5 capas MaxPool



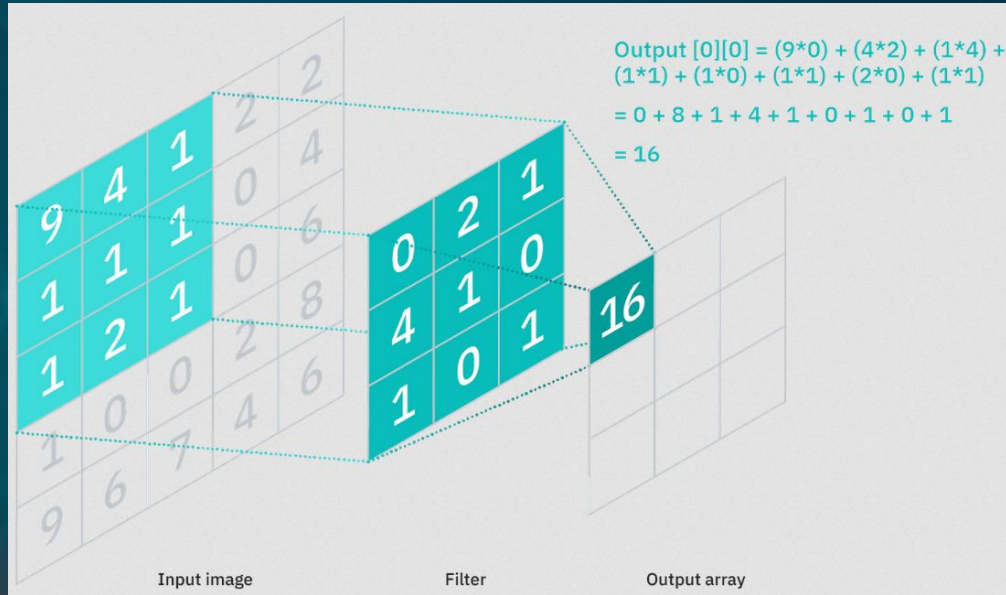
Convolución



Define el número de filtros de salida en la convolución (la dimensionalidad) aplica 32 para la primera capa de convolución, mientras que en las siguientes implementa 64, 65, 128 y 256.

Filters (Number of filters)

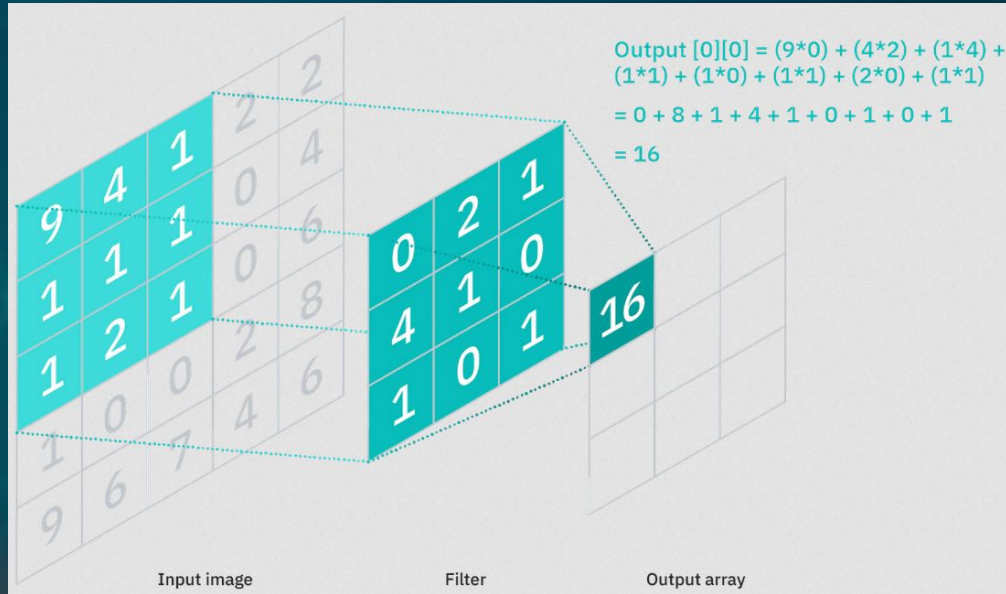
Convolución



Filtro convolucional que realiza una validación cruzada y especifica la altura y el ancho de la ventana de convolución 2D que se aplica en una matriz (3,3) para las 5 capas convolucionales

Kernel Size

Convolución

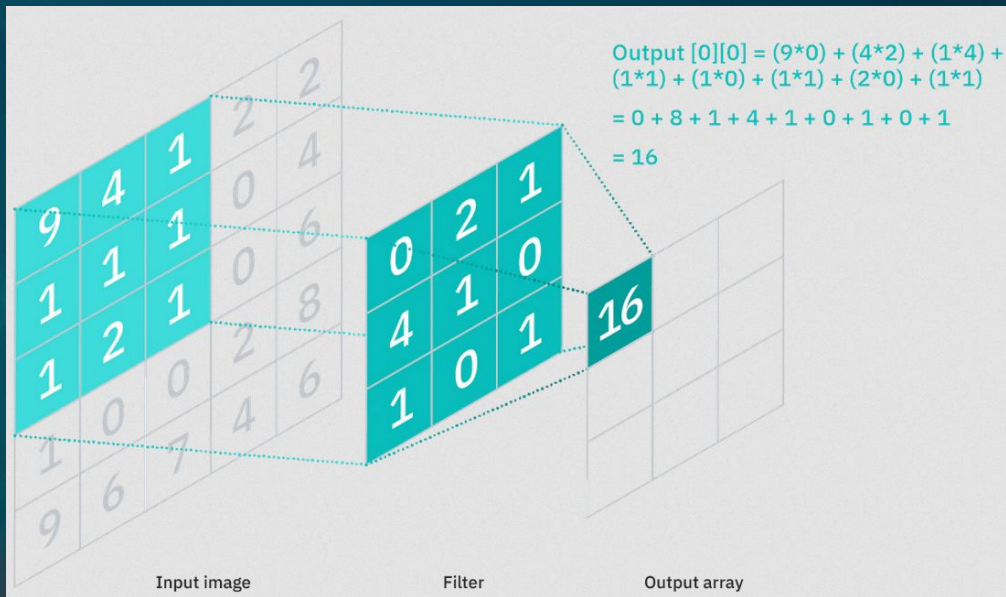


Distancia, o número de píxeles, en la que el núcleo se moverá sobre la matriz de entrada.

Para las 5 capas se aplicará en 1, es decir pixel por pixel.

Stride

Convolución



Utilizado cuando los filtros no se ajustan a la imagen de entrada.

Se utiliza Same, es decir, que para el stride 1 la salida será la misma que la entrada y se aplica padding a la imagen de entrada para que la imagen de entrada quede completamente cubierta por el filtro y el stride especificado.

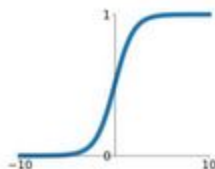
Padding

Funciones de Activación

Activation: Alimenta a la neurona actual y su salida que va a la siguiente capa, donde deciden si la neurona debe activarse o no, para las 5 capas de convolución se aplica una transformación de unidad lineal rectificada(ReLU) al mapa de características, lo que introduce la no linealidad.

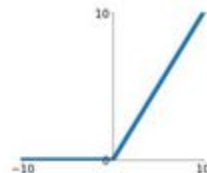
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



ReLU

$$\max(0, x)$$



Otras aplicaciones en el modelo

(>1 Capa) Transforma las entradas para que estén estandarizadas, lo que significa que tendrán una media de cero y una desviación estándar de uno.

Luego se aplica después de cada Dropout, exceptuando el último.

BatchNormalization

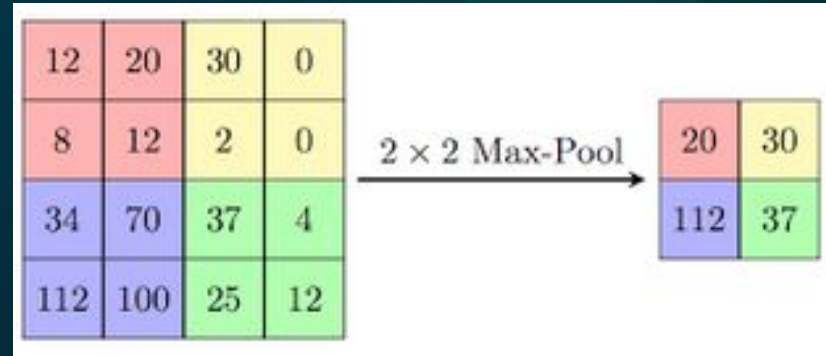
Actúa para prevenir el sobreajuste en el entrenamiento ocasionando que algunas neuronas no propaguen su aprendizaje en esta etapa, definido con una tasa de desactivación de 10% de las neuronas en la primera capa y un 20% en las siguientes.

DropOut

Max-Pool

Se ejecuta MaxPool2D:

- Resume todos los valores de la entrada por píxel en una nueva matriz con los máximos de los filtros establecidos en una matriz de salida.
- En el proyecto, el tamaño de estos filtros (size) se define en una matriz de (2,2) y con un stride de 2

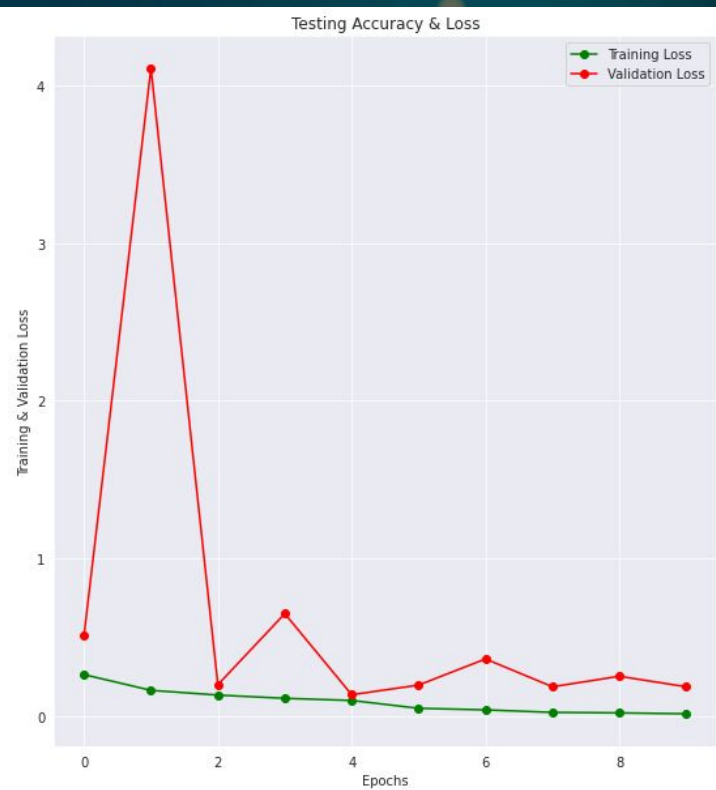
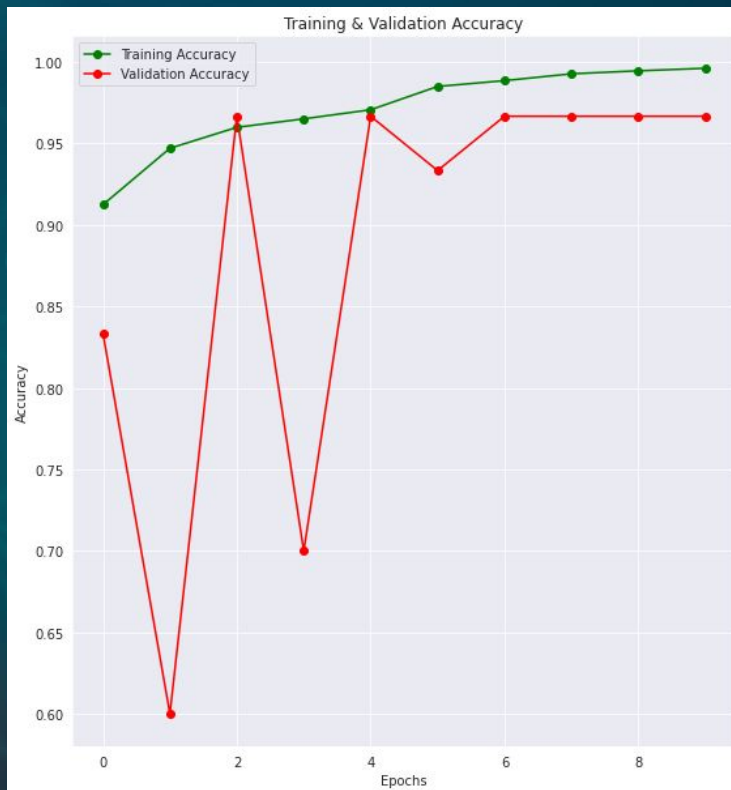


- Una vez lo anterior, se realiza Flattening para transformar toda la matriz del mapa de características en un vector que será entrada de la red neuronal.
- Por último se compone la capa Fully Connected y para construir la red neuronal se utiliza Dense, quien define la capa oculta de la red con 128 neuronas, para después definir la capa de salida, la cuál es la predicción que se forma en conjunto a la función de activación, la que depende de la cantidad de clases a predecir.
 - Sigmoid

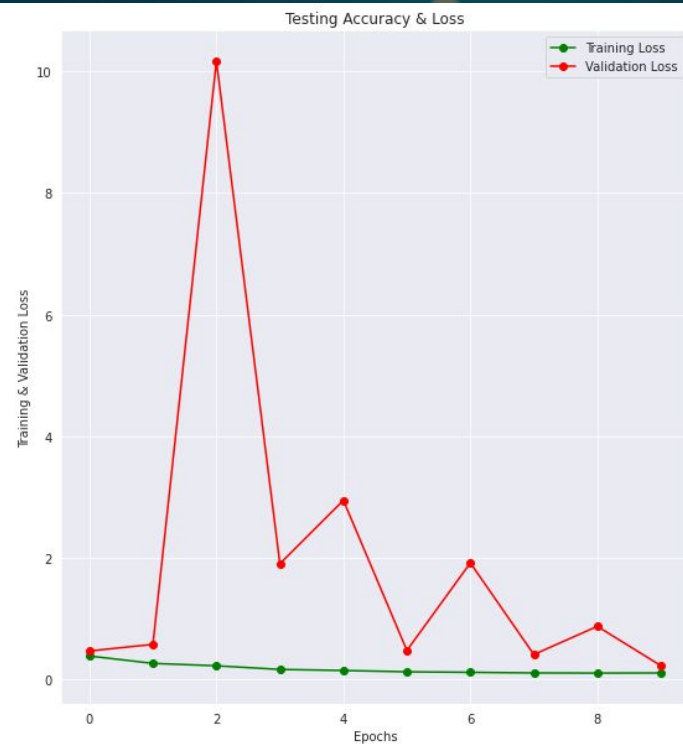
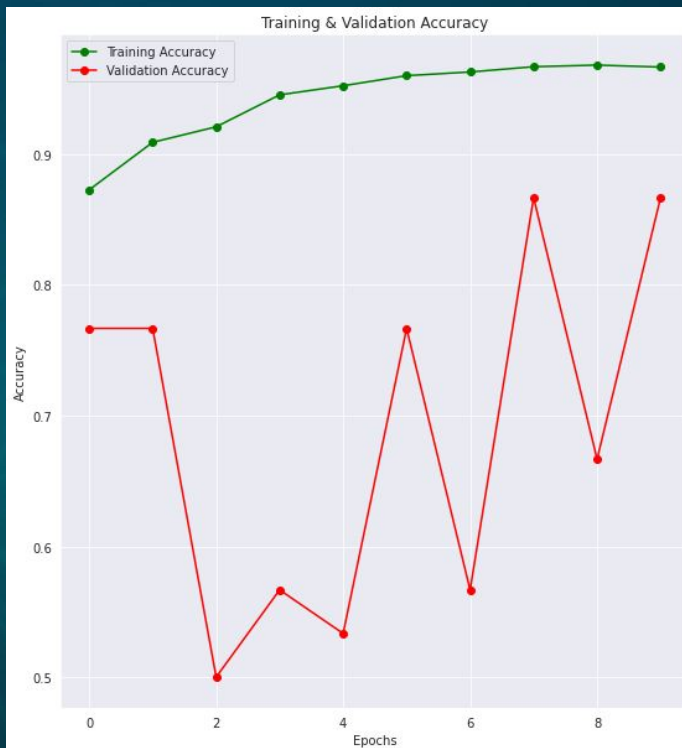
Optimizador y Entrenamiento

- La función de pérdida se define para medir el rendimiento del modelo en imágenes con etiquetas conocidas, lo que se traduce en una tasa de error entre las etiquetas observadas y las predichas.
 - El proyecto utiliza *binarycrossentropy* para las clasificaciones binarias (2 clases) con un valor de salida entre 0 y 1.
- El optimizador mejorará iterativamente los parámetros (filtra los valores del kernel, los pesos y el sesgo de las neuronas) para minimizar la pérdida.
 - El proyecto utiliza *RMSprop* (con valores predeterminados), considerado generalmente un optimizador muy efectivo.

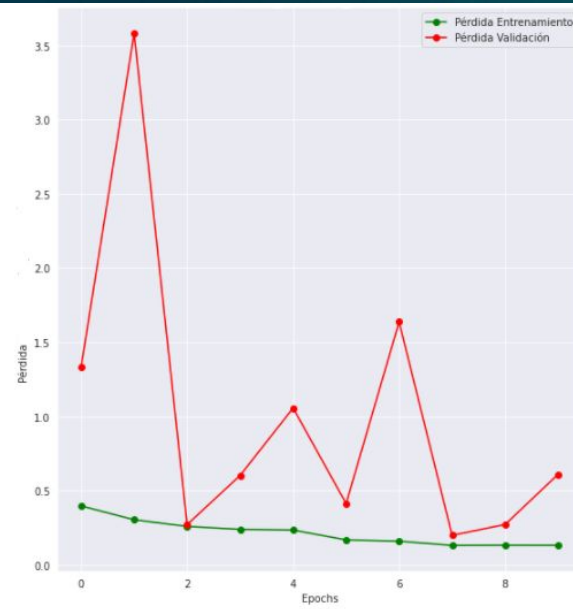
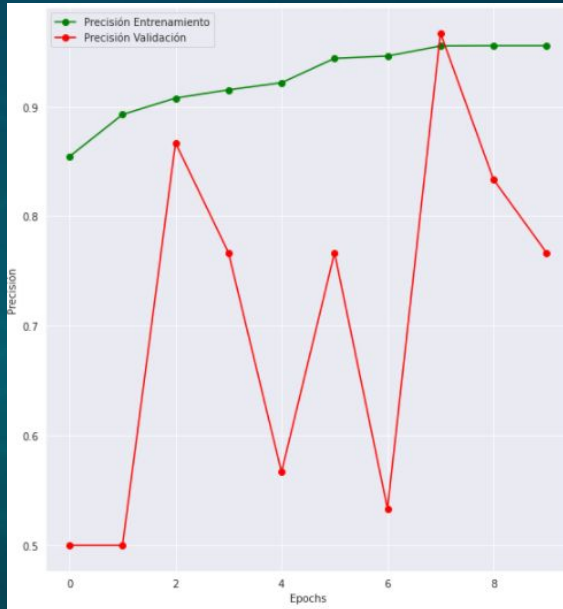
Evolución (10 epoch) Exp. 1



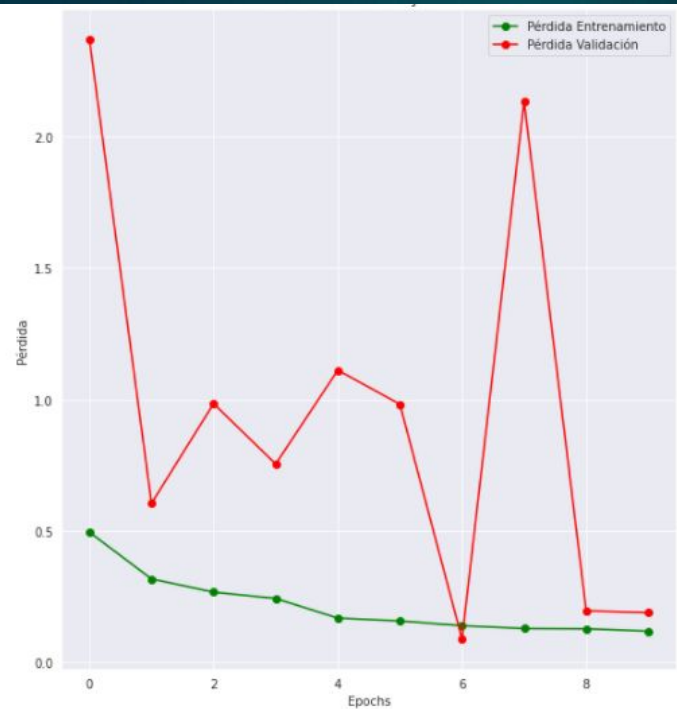
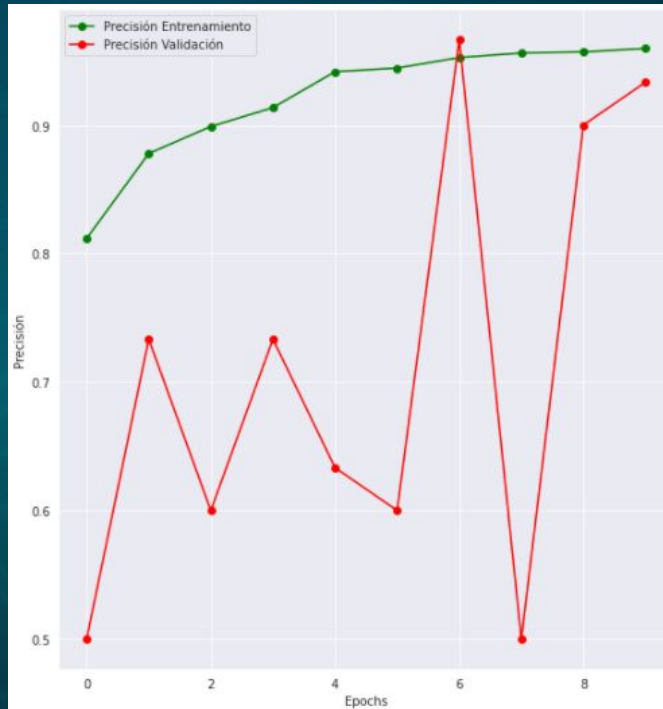
Evolución (10 epoch) Exp. 2



Evolución (10 epoch) Exp. 3



Evolución (10 epoch) Exp. 4



Predicciones y evaluación del Modelo Exp. 1 y 2

	precision	recall	f1-score	support
Positivo (Class 0)	0.49	0.73	0.59	185
Negativo (Class 1)	0.47	0.24	0.32	185
accuracy			0.48	370
macro avg	0.48	0.48	0.45	370
weighted avg	0.48	0.48	0.45	370

	precision	recall	f1-score	support
Positivo (Class 0)	0.71	1.00	0.83	185
Negativo (Class 1)	1.00	0.60	0.75	185
accuracy			0.80	370
macro avg	0.86	0.80	0.79	370
weighted avg	0.86	0.80	0.79	370

El reporte de clasificación dicta que precisión del modelo obtiene un porcentaje de efectividad bastante alto para el testing.

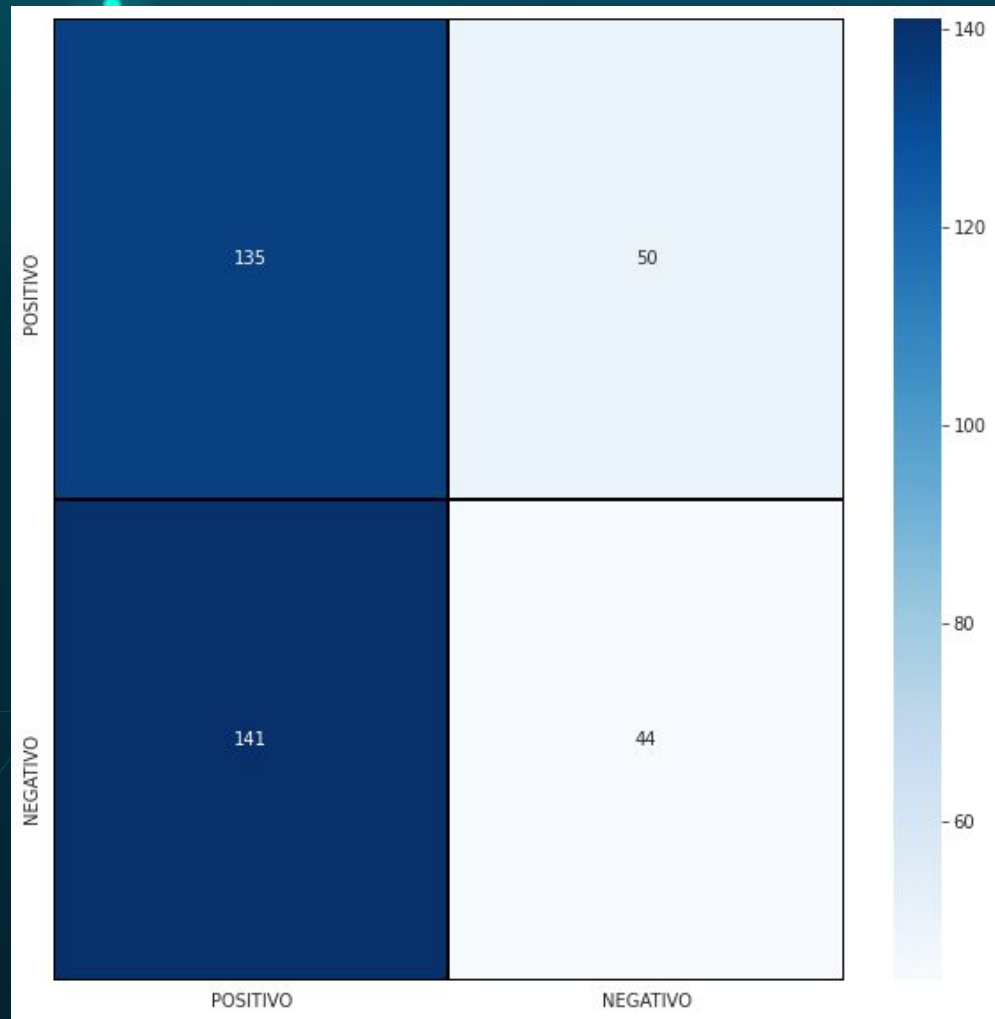
Predicciones y evaluación del Modelo Exp. 3 y 4

	precision	recall	f1-score	support
POSITIVO (Class 0)	0.91	0.99	0.95	185
NEGATIVO (Class 1)	0.99	0.90	0.94	185
accuracy			0.94	370
macro avg	0.95	0.94	0.94	370
weighted avg	0.95	0.94	0.94	370

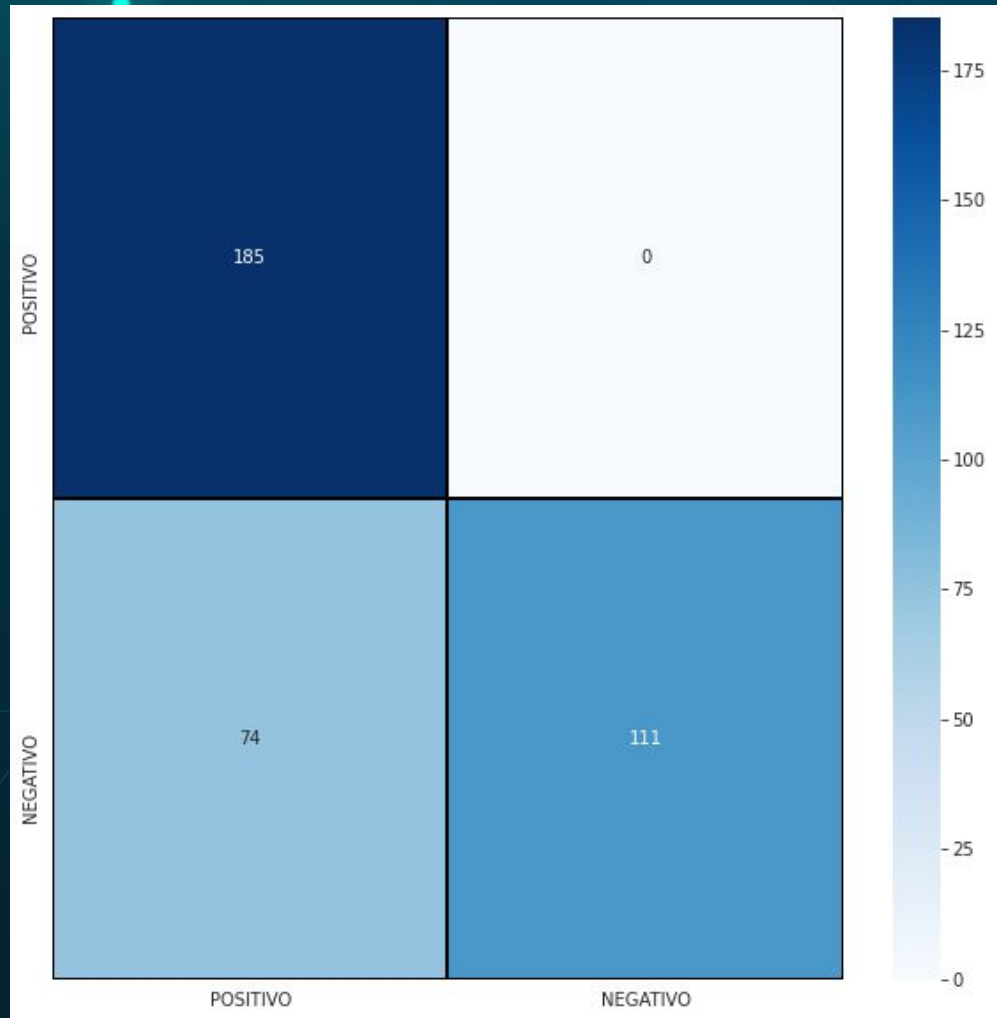
POSITIVO (Class 0)	0.80	0.99	0.88	185
NEGATIVO (Class 1)	0.99	0.75	0.85	185
accuracy			0.87	370
macro avg	0.89	0.87	0.87	370
weighted avg	0.89	0.87	0.87	370

El reporte de clasificación dicta que precisión del modelo obtiene un porcentaje de efectividad bastante alto para el testing.

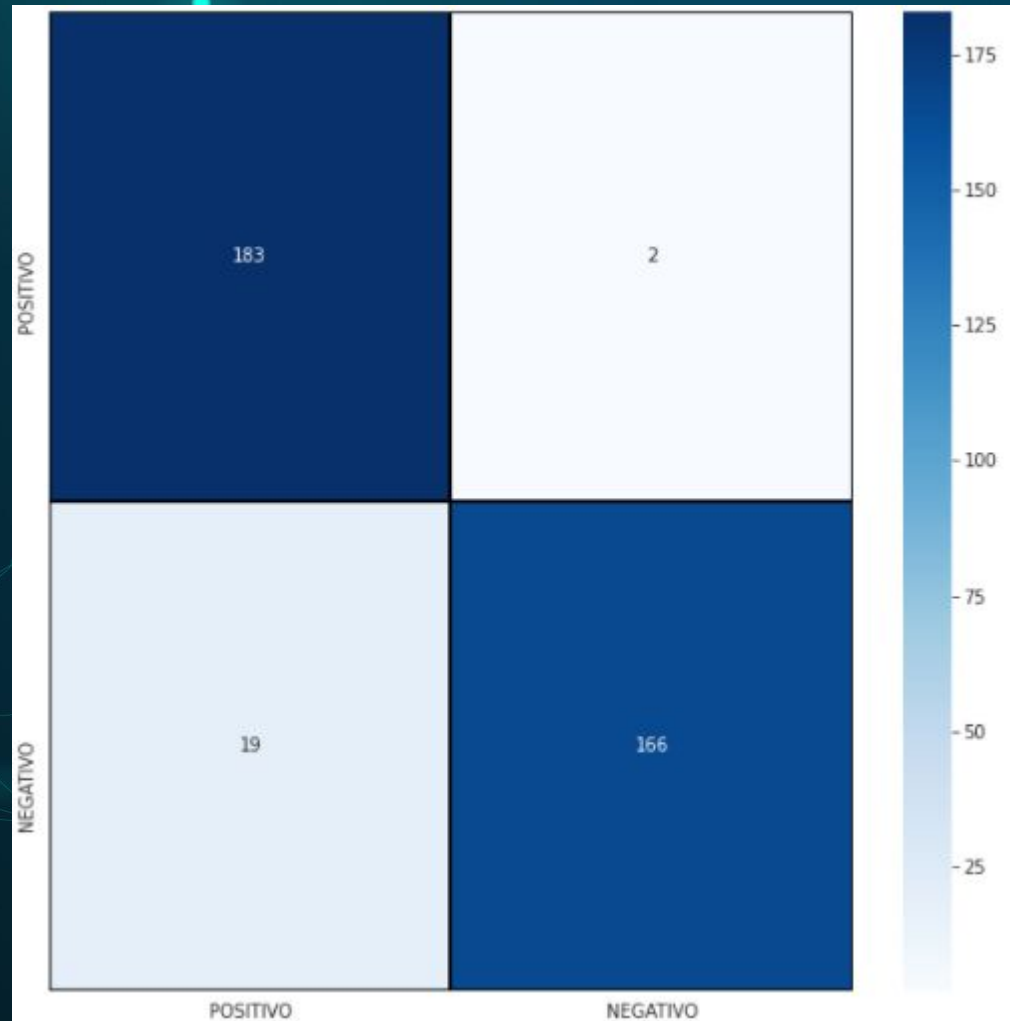
Matrix Confusion Exp. 1



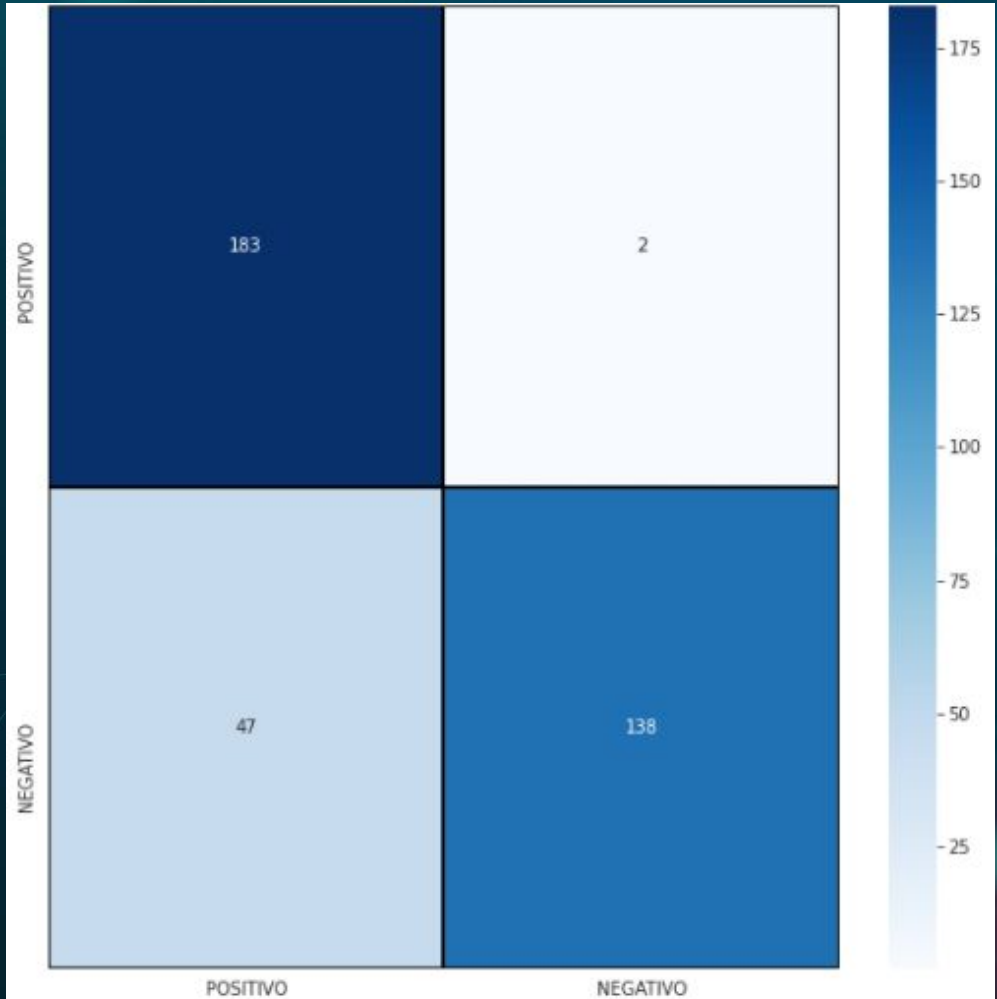
Matrix Confusion Exp. 2



Matrix Confusion Exp. 3



Matrix Confusion Exp. 4



Conclusión

Con la creciente demanda de atención médica, en particular respecto de un médico en tiempos de pandemia, el poder identificar a un paciente con un diagnóstico positivo en Covid-19 tendría un impacto positivo en el sistema de salud, solo basta obtener una precisión en los modelos que permitan generar una alta confiabilidad para la comunidad médica y optimizar recursos humanos, recordando además que la radiografía de tórax tiene una gran disponibilidad, con bajo coste, rápida y muy útil en el diagnóstico y tratamiento de emergencia.

Gracias!

