Covid-19 Detection through X-Rays

INTRODUCTION:

In this pandemic which spread very quickly, it was difficult to test every person for the Covid-19 virus. Then came some variants which were not detectable early through RT PCR. Thus X-Rays and CT Scans were used to detect the virus. X-Rays were more popularly used to detect the Covid-19 virus.

As the cases rose at a very high rate during the 2nd wave, it was physically not possible to analyze each X-Ray and detect the Covid virus. Thus many Data Scientists and Radiologists came together to prepare different CNN models to detect Covid through X-Rays.

In this project, we also tried to create a model which will predict the Covid virus through X-Ray inputs.

Project Description:

Deep learning has gained great importance in the field of Machine Learning. It has improved the performance of the model as well as increased the fields in which we can apply machine learning.

We have used Deep Learning to achieve our model. We have used Python inbuilt libraries in order to create our model.

Libraries used :

1. Keras
2. Matplotlib

3. Numpy
4. Pandas

The model is created is as follows:

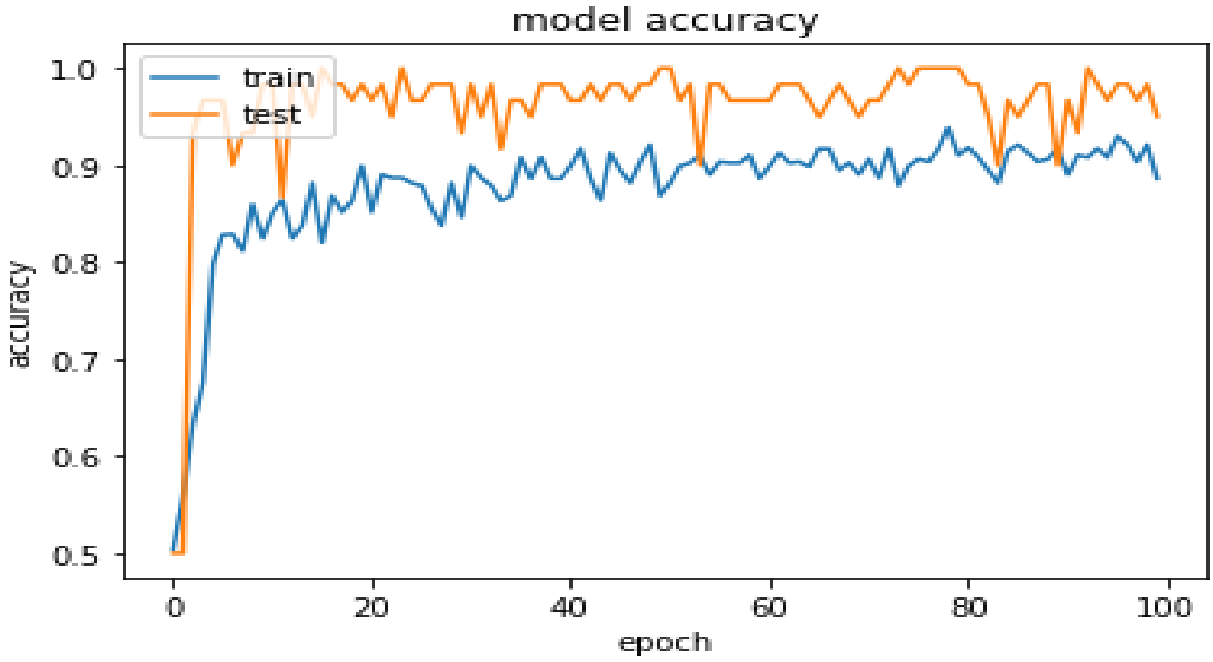
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 32)      896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 220, 220, 64)      18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 110, 110, 64)      0
_________________________________________________________________
dropout (Dropout)            (None, 110, 110, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 108, 108, 64)      36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 52, 52, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 26, 26, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 24, 24, 128)       147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 12, 12, 128)       0
_________________________________________________________________
dropout_3 (Dropout)          (None, 12, 12, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 18432)             0
_________________________________________________________________
dense (Dense)                (None, 64)                1179712
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 1,457,537
Trainable params: 1,457,537
Non-trainable params: 0
```

The model needs to be trained on a dataset. Thus we used two sources to gain the dataset.
Source:

1. https://github.com/muhammedtalo/COVID-19/tree/master/X-Ray%20Image%20DataSet
2. https://www.kaggle.com/fusicfenta/chest-xray-for-covid19-detection

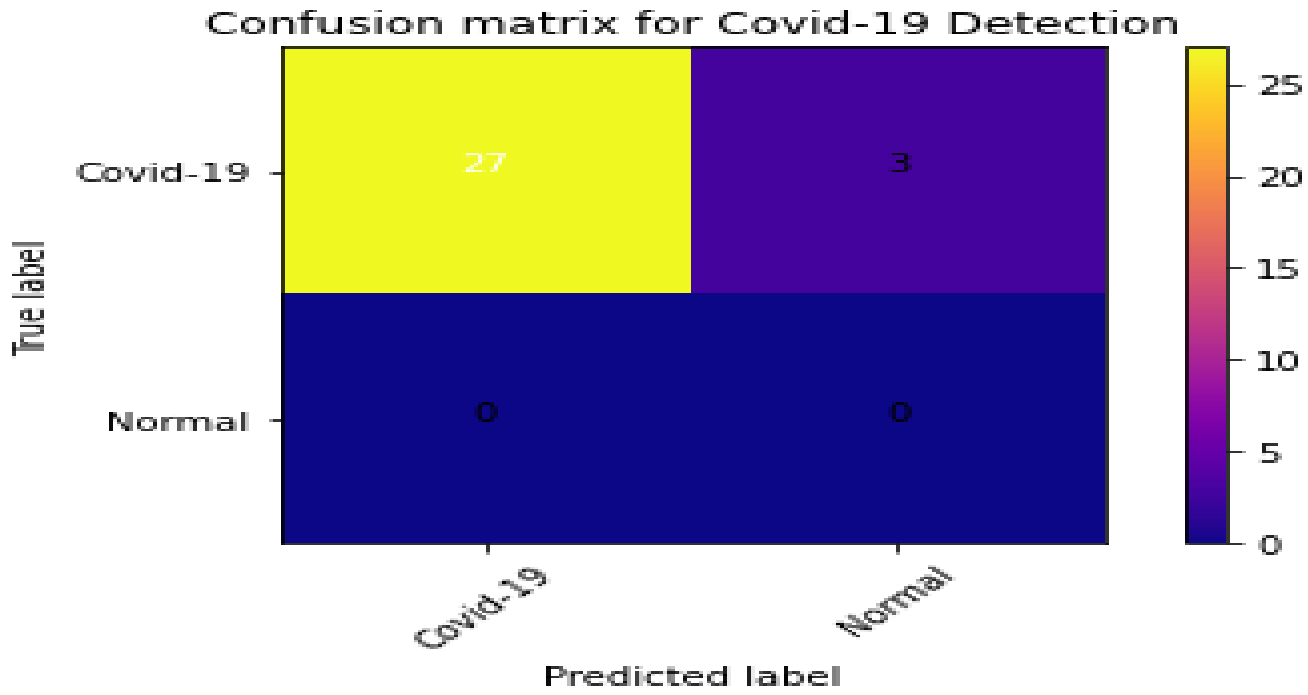
Now we used 178 images of COVID-19 affected X-Rays and 178 images of Normal X-Rays for training our model.
In order to validate our model we used 30 images of COVID-affected X-Rays and 30 images of Normal X-Rays.

The model trained for 100 epochs gave the following graph of training and validation accuracy:



After the training we validated our model and reached this confusion matrix. This confusion matrix shows that out of 30 Covid-19 affected X-Rays, 27

Covid-19 X-Rays were detected as Covid-19 affected X-Rays by our model while only 3 images were detected Normal.



Also we used an Image of Covid-19 affected X-Ray to predict whether the person has Covid-19 or not. And the model predicted it rightly as Covid-19 affected X-Ray.

In [16]:
```
import numpy as np
# from google.colab.patches import cv2_imshow
import cv2
from keras.preprocessing import image
xtest_image = image.load_img('./Dataset/Prediction/ryct.2020200034.fig5-day7.jpeg', target_size = (224, 224,3))
xtest_image = image.img_to_array(xtest_image)
xtest_image = np.expand_dims(xtest_image, axis = 0)
results = model.predict_classes(xtest_image)
# training_set.class_indices
imggg = cv2.imread('./Dataset/Prediction/ryct.2020200034.fig5-day7.jpeg')
print("This Xray Image is of positive covid-19 patient")
imggg = np.array(imggg)
imggg = cv2.resize(imggg,(400,400))
plt.imshow(imggg)
# cv2_imshow(imggg)
# print(results)
if results[0][0] == 0:
    prediction = 'Positive For Covid-19'
else:
    prediction = 'Negative for Covid-19'
print("Prediction Of Our Model : ",prediction)
```

```
C:\Users\asus\anaconda3\lib\site-packages\keras\engine\sequential.py:450: UserWarning: `model.predict_classes()` is deprecated and will be removed after 2021-01-01. Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it uses a `sigmoid` last-layer activation).
  warnings.warn('`model.predict_classes()` is deprecated and '
```

```
This Xray Image is of positive covid-19 patient
Prediction Of Our Model :  Positive For Covid-19
```

Conclusion:

We have created a model which can predict Covid-19 X-Ray with an accuracy > 90.