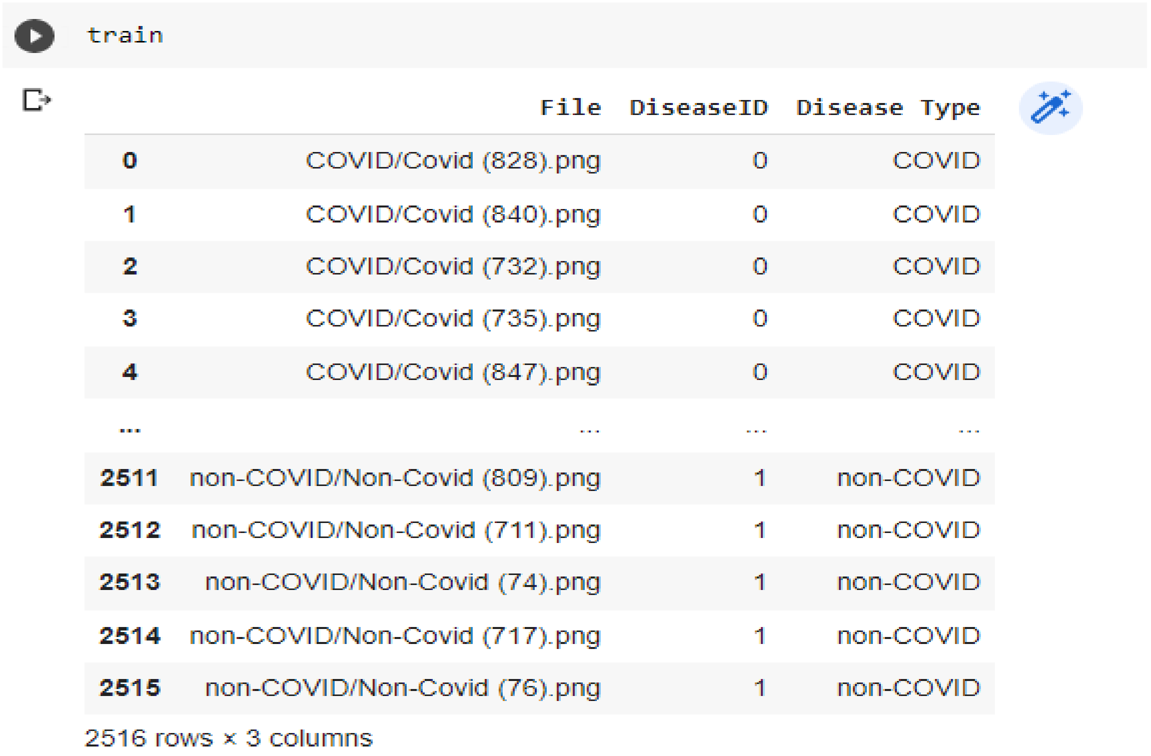
Explanation and Screen Shots

Problem Statement: Dataset contains 1252 CT scans that are positive for SARS-CoV-2 infection (COVID-19) and 1230 CT scans for patients non-infected by SARS-CoV-2, 2482 CT scans in total. These data have been collected from real patients in hospitals from Sao Paulo, Brazil. The aim is to identify if a person is infected by SARS-CoV-2 through the analysis of his/her CT scans.

Data Mining: Data was collected from https://drive.google.com/drive/folders/1WOeodRmv1Mw5Cswuip3nUIi6ViQWKpo_?usp=sharing

And following data frame has been generated:

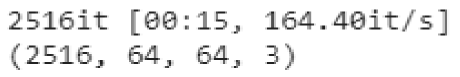


```
train
```

	File	DiseaseID	Disease Type
0	COVID/Covid (828).png	0	COVID
1	COVID/Covid (840).png	0	COVID
2	COVID/Covid (732).png	0	COVID
3	COVID/Covid (735).png	0	COVID
4	COVID/Covid (847).png	0	COVID
...	...	...	...
2511	non-COVID/Non-Covid (809).png	1	non-COVID
2512	non-COVID/Non-Covid (711).png	1	non-COVID
2513	non-COVID/Non-Covid (74).png	1	non-COVID
2514	non-COVID/Non-Covid (717).png	1	non-COVID
2515	non-COVID/Non-Covid (76).png	1	non-COVID

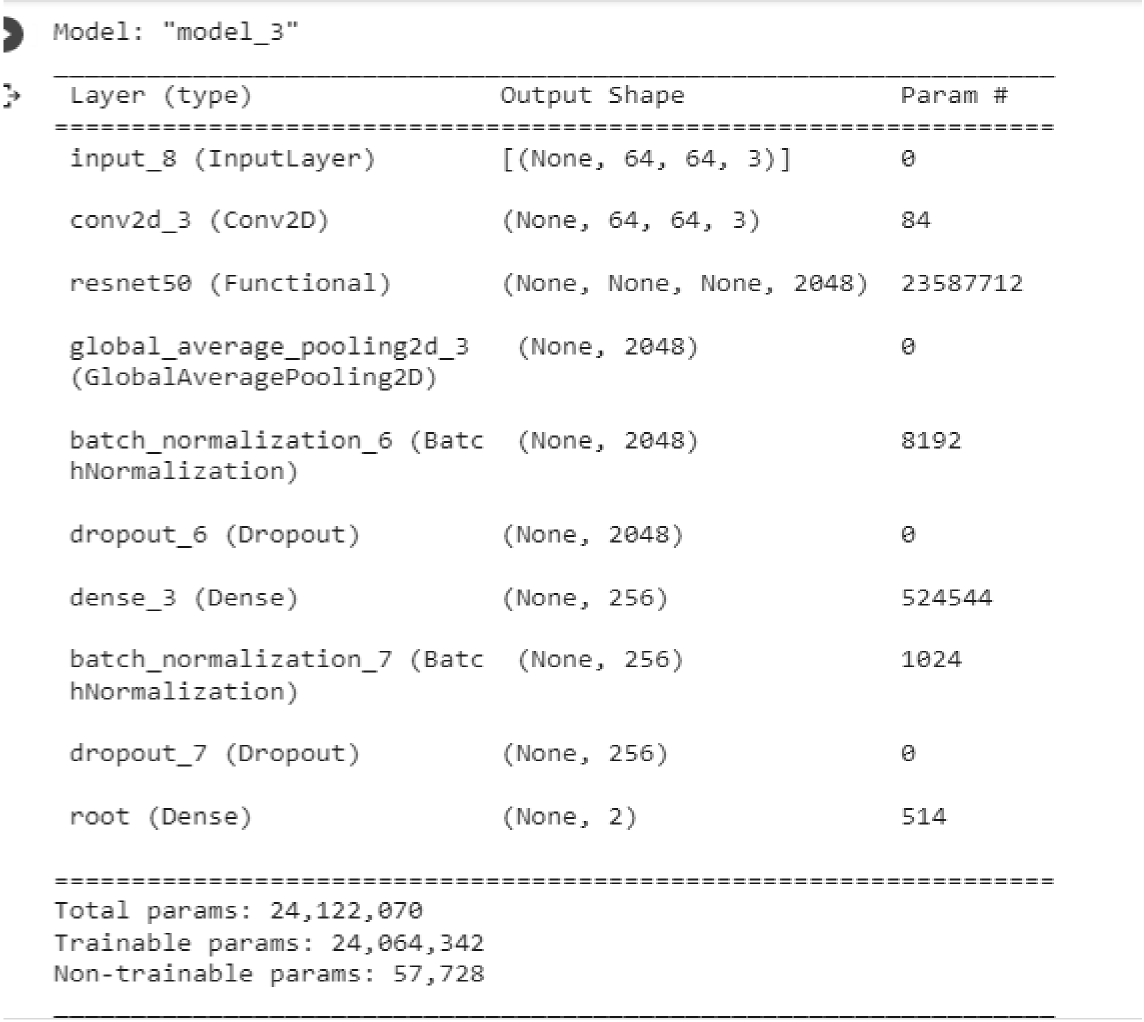
2516 rows × 3 columns

As mentioned images were in different sizes need to fix a size to work on with it:



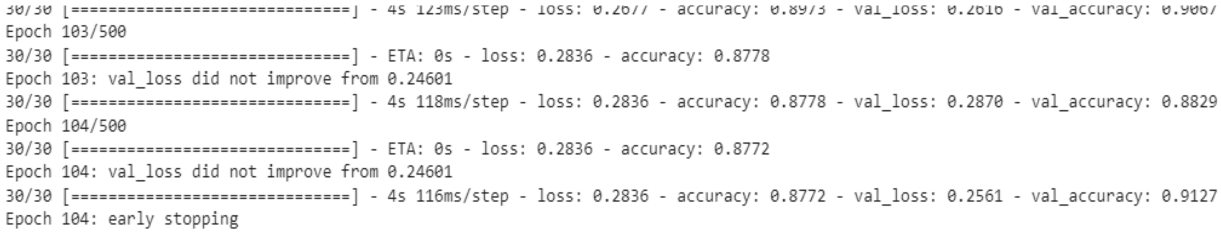
```
2516it [00:15, 164.40it/s]
(2516, 64, 64, 3)
```

Data Augmentation: for training model resnet50() is used, data augmentation is done by setting the degree range for random rotations, range for random horizontal shifts, range for random vertical shifts, zoom range, random input flips horizontally and random input flips vertically.



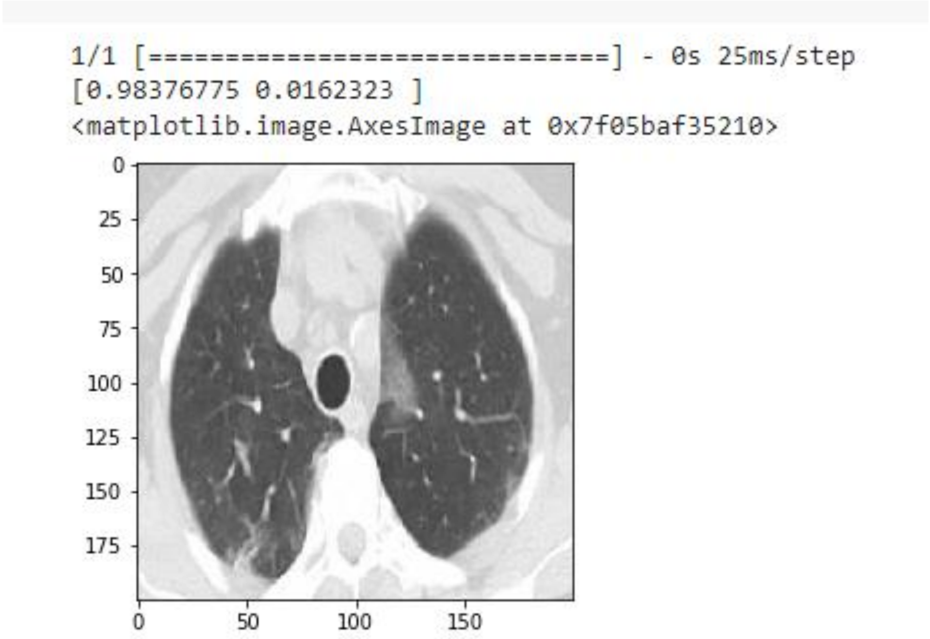
```
Model: "model_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_8 (InputLayer)        [(None, 64, 64, 3)]       0

 conv2d_3 (Conv2D)           (None, 64, 64, 3)         84

 resnet50 (Functional)       (None, None, None, 2048)  23587712

 global_average_pooling2d_3   (None, 2048)             0
 (GlobalAveragePooling2D)

 batch_normalization_6 (Batc  (None, 2048)             8192
 hNormalization)

 dropout_6 (Dropout)         (None, 2048)              0

 dense_3 (Dense)             (None, 256)               524544

 batch_normalization_7 (Batc  (None, 256)              1024
 hNormalization)

 dropout_7 (Dropout)         (None, 256)               0

 root (Dense)                (None, 2)                 514

=================================================================
Total params: 24,122,070
Trainable params: 24,064,342
Non-trainable params: 57,728
_________________________________________________________________
```

Training model: During training model, checkpoints and early stopping is also applied.



```
30/30 [==============================] - 4s 123ms/step - loss: 0.2677 - accuracy: 0.8973 - val_loss: 0.2616 - val_accuracy: 0.9067
Epoch 103/500
30/30 [==============================] - ETA: 0s - loss: 0.2836 - accuracy: 0.8778
Epoch 103: val_loss did not improve from 0.24601
30/30 [==============================] - 4s 118ms/step - loss: 0.2836 - accuracy: 0.8778 - val_loss: 0.2870 - val_accuracy: 0.8829
Epoch 104/500
30/30 [==============================] - ETA: 0s - loss: 0.2836 - accuracy: 0.8772
Epoch 104: val_loss did not improve from 0.24601
30/30 [==============================] - 4s 116ms/step - loss: 0.2836 - accuracy: 0.8772 - val_loss: 0.2561 - val_accuracy: 0.9127
Epoch 104: early stopping
```

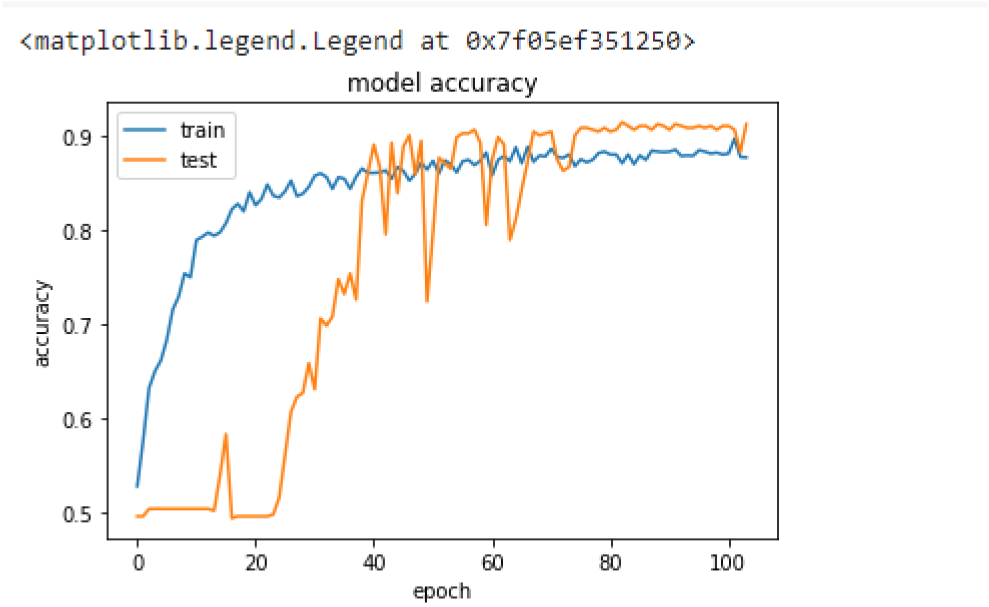
Model Prediction result:



```
1/1 [==============================] - 0s 25ms/step
[0.98376775 0.0162323 ]
<matplotlib.image.AxesImage at 0x7f05baf35210>
```

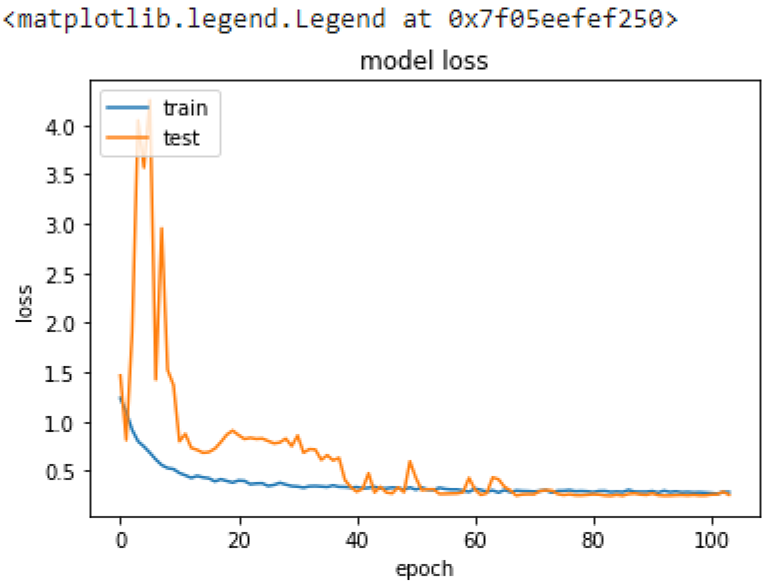
Various Performance metrics results:

1) Accuracy versus Epoch



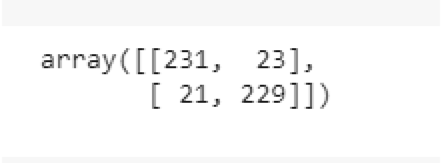
```
<matplotlib.legend.Legend at 0x7f05ef351250>
```

2) Loss versus Epochs



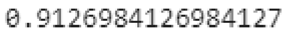
```
<matplotlib.legend.Legend at 0x7f05eefef250>
```

3) Confusion matrix



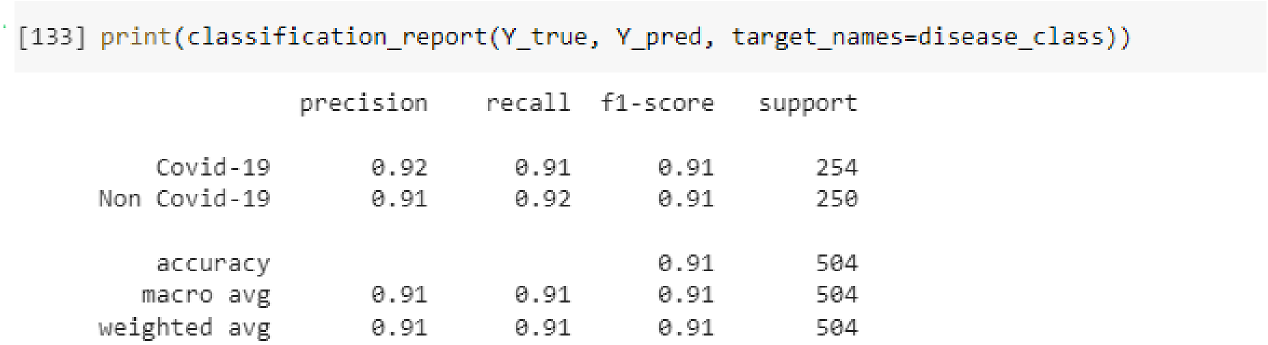
```
array([[231,  23],
       [ 21, 229]])
```

4) Accuracy score



```
0.9126984126984127
```

5) Classification report



```
[133] print(classification_report(Y_true, Y_pred, target_names=disease_class))

                  precision    recall  f1-score   support

        Covid-19       0.92      0.91      0.91       254
    Non Covid-19       0.91      0.92      0.91       250

        accuracy                           0.91       504
       macro avg       0.91      0.91      0.91       504
    weighted avg       0.91      0.91      0.91       504
```