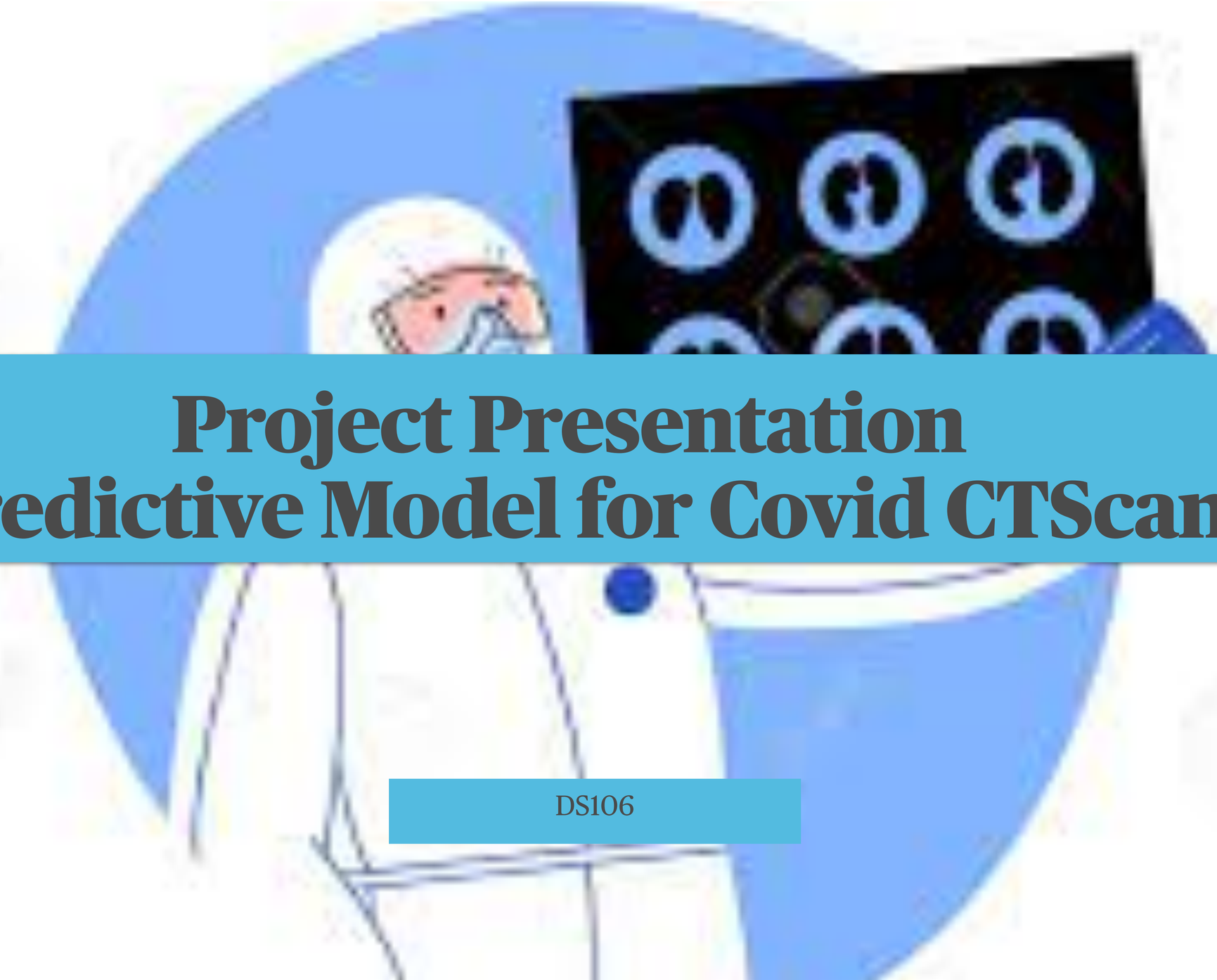
ANG WEE HWA 20/5/2021



Project Presentation (Predictive Model for Covid CTScan)

DS106

ANG WEE HWA 20/5/2021

Why?

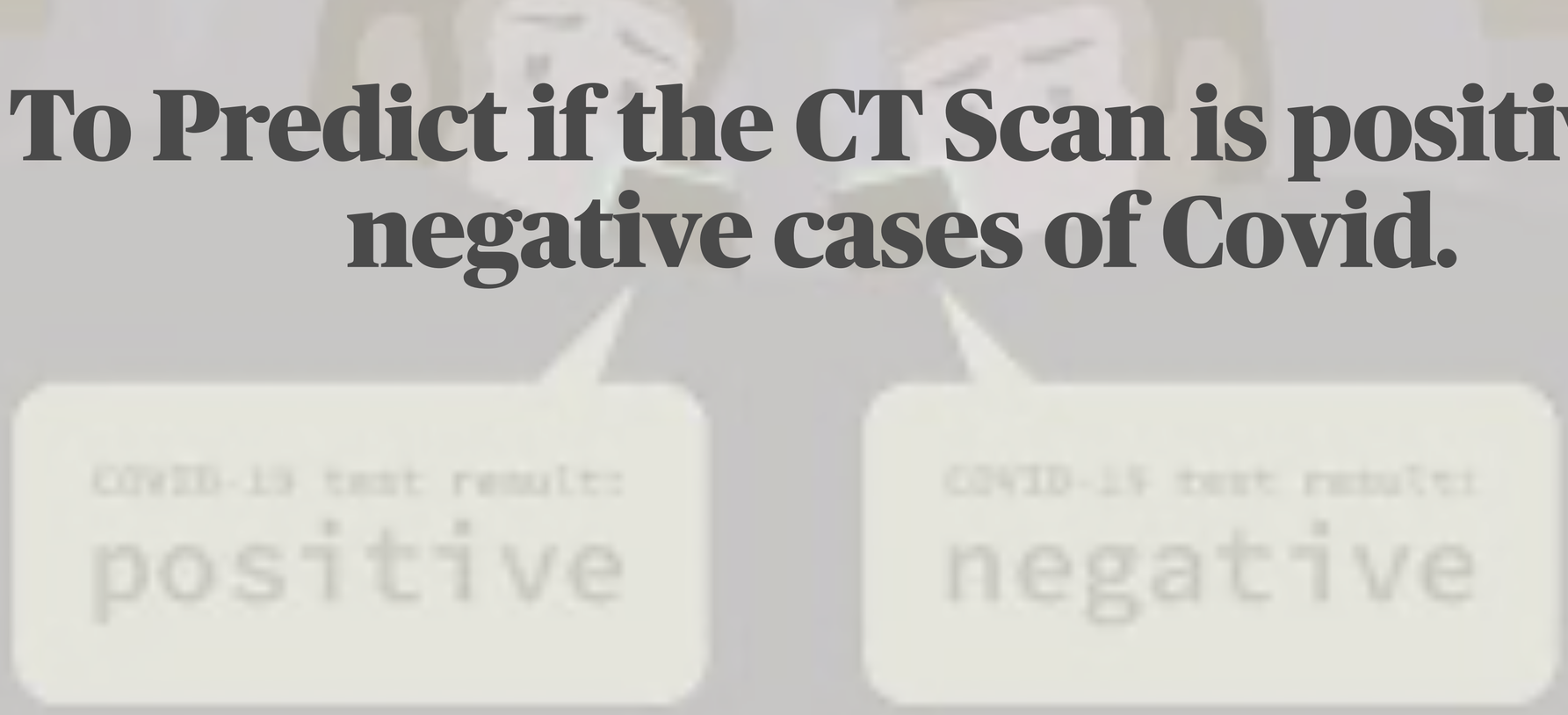
- Covid19 had been a issue with the world, as the virus had affected varies countries around the global. Businesses and individuals are losing customers as governments are issuing stricter measures to counter and prevent the spread of the virus.

What do we want to achieve?

- To Create a model to predict if the CTScan image is positive or negative case for Covid

Problem Statement

To Predict if the CT Scan is positive or negative cases of Covid.



Difficulties

- From the dataset we might be able to spot the difference however I am without medical knowledge it might not be the only factor or feature
- There is only CT Scan of positive and negative omitting the other types of lung diseases which might cause the image to change hence unable to get a complete model

Steps

WE WILL BREAK INTO 3 PARTS FOR THIS PROJECT NAMELY:

PART A : OBSERVE THE CT SCAN AND TO DIFFERENTIATE THE DIFFERENCE VISUALLY.

PART B : WE WILL TEST OUR 3 MODELS NAMELY:

1) VGG16

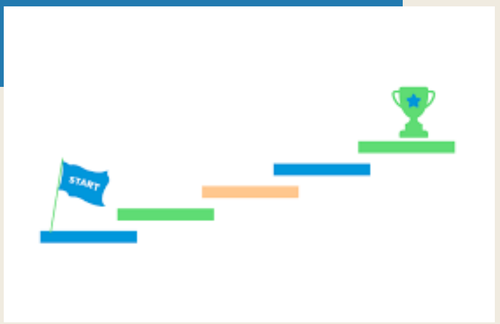
2) INCEPTIONV3

3) RESNET50

AND SELECT THE BEST ONE FOR HYPERPARAMETER TUNING.

PART C : WE WILL USE THE BEST MODEL TO FINE TUNE THE TRAINING RATE AND USE THE BEST MODEL.

TO AVOID BIAS, WE WILL BE USING 50 EPOCH TO TRAIN ALL THE MODELS AND CONFUSION METRIC FROM PREDICTION OF THE TEST DATASET. ACCURACY AND VALIDITY ACCURACY ARE THE MEASUREMENTS FOR THE PERFORMANCE OF THE MODEL.



Dataset

FOR THIS PREDICTIVE MODEL WE DOWNLOADED THE DATA FROM KAGGLE
REF: HTTPS://WWW.KAGGLE.COM/C/COVIDCT

WE HAVE A TOTAL OF:
COVID CT SCAN IMAGES: 1252 (PNG)
NON-COVID CT SCAN IMAGES: 1230 (PNG)

WE WILL DIVIDE THEM INTO 3 FOLDERS MANUALLY:

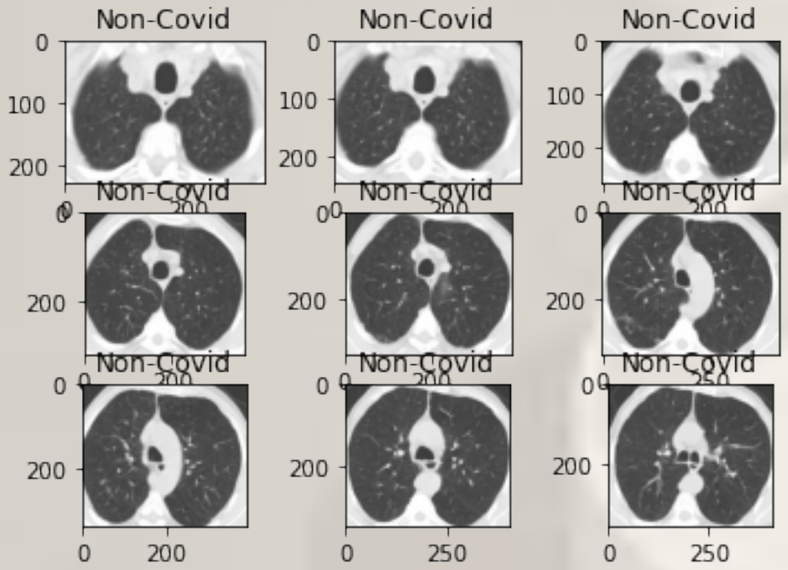
1) TRAIN: (TO TRAIN THE MODEL)
COVID: 992 IMAGES
NON-COVID: 975 IMAGES

2) VAL: (TO VALIDATE THE MODEL AFTER EACH EPOCH)
COVID: 248 IMAGES
NON-COVID: 244 IMAGES

3) TEST: (TO MANUALLY TEST AND OBSERVE THE RESULT)
COVID : 12 IMAGES
NON-COVID : 10 IMAGES

Difference

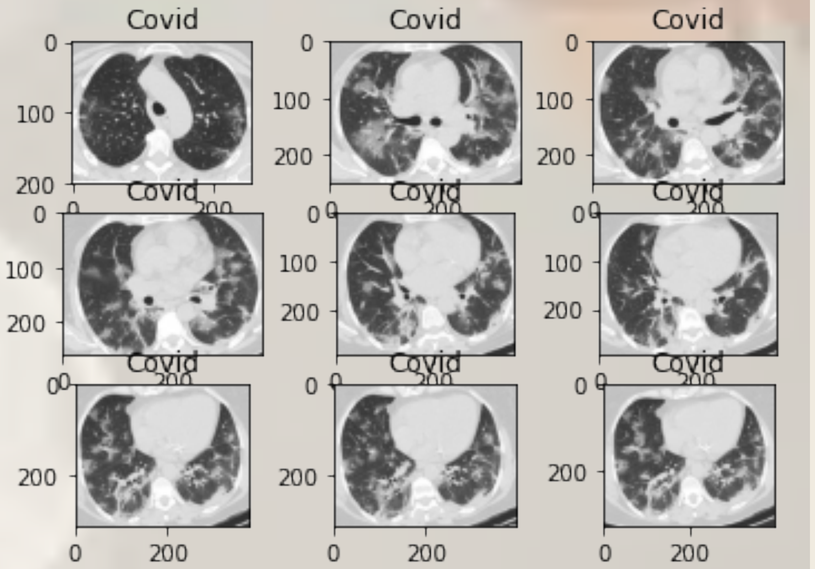
NORMAL LUNGS



OBSERVATION

- CTSCAN OF THE NORMAL LUNGS ARE CLEAR
- CTSCAN OF COVID PATIENTS LUNG ARE WITH WHITE AREAS AND LUNG SEEMS TO BE SMALLER

LUNGS INFECTED WITH COVID



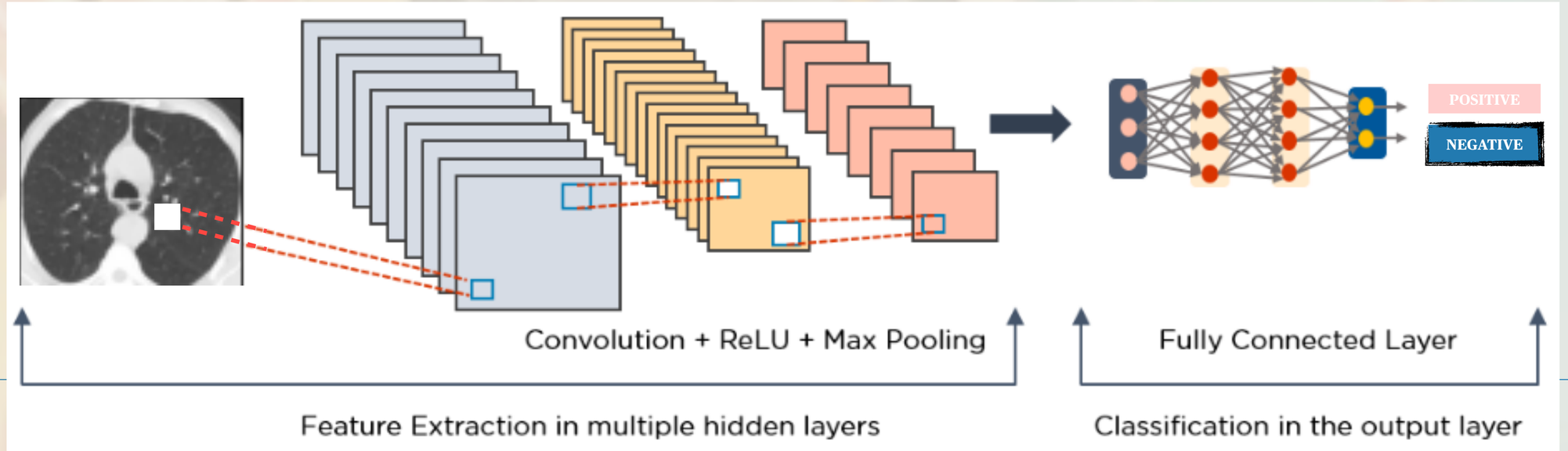
Machine Learning Model

For this problem statement I will be using deep learning method Convolutional neural network(CNN)

Will prepare and fit the respective model

- VGG 16
- ResNet50
- Inception V3

After comparing the performance of the 3 models will select the one with the best prediction to do hyper parameter tuning

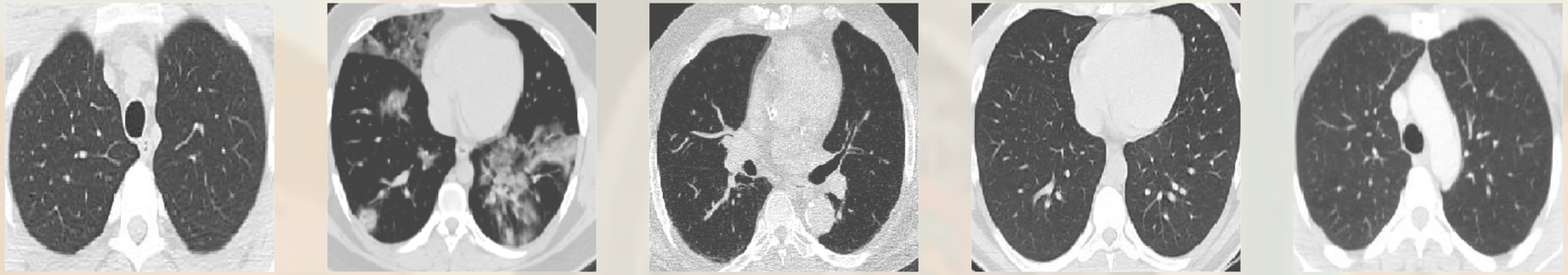


ANG WEE HWA 20/5/2021

Pre Processing (Augmentation)

BEFORE

[0. 1.] [1. 0.] [1. 0.] [0. 1.] [0. 1.]



AFTER

Shift up and Left [1. 0.]

Shift Left [1. 0.]

Shift up [1. 0.]

Width Shift [1. 0.]

Horizontal Shift [1. 0.]

To make slight alteration to the image for better machine learning

ANG WEE HWA 20/5/2021

Pre Processing (Data fitting)

```
# use image generator to preprocess target size to (224,244) with batch size 32
# prepare data for train, test and predict where predict shuffle=False,
# so we can observe the result after prediction
train_res = keras.preprocessing.image.ImageDataGenerator(featurewise_center=True,
                                                         featurewise_std_normalization=True,
                                                         rotation_range=20,
                                                         width_shift_range=0.2,
                                                         height_shift_range=0.2,
                                                         horizontal_flip=True,
                                                         preprocessing_function=tf.keras.applications.resnet50.preprocess_input)\
                                                         .flow_from_directory(folder + 'train/',target_size = (224,224), batch_size=32)
val_res = keras.preprocessing.image.ImageDataGenerator(preprocessing_function=tf.keras.applications.resnet50.preprocess_input)\
.flow_from_directory(folder + 'val/',target_size = (224,224), batch_size=32)
test_res = keras.preprocessing.image.ImageDataGenerator(preprocessing_function=tf.keras.applications.resnet50.preprocess_input)\
.flow_from_directory(folder + 'test/',target_size = (224,224), batch_size=32,shuffle=False)
```

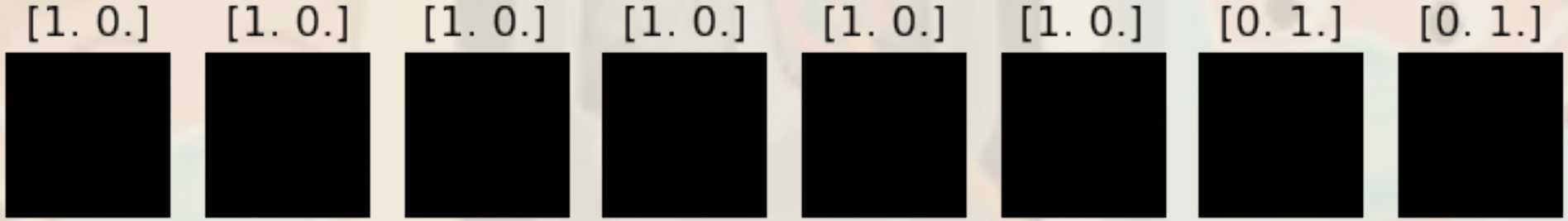
- keras application library to use varies model fitting
- Use batch size 32 (test 32 batches per steps per epoch to prevent overfitting
- Use Flow to extract data from directory

ANG WEE HWA 20/5/2021

Pre Processing (Varies Model)

VGG 16

INCEPTION V3



RESNET50

ANG WEE HWA 20/5/2021

Transfer Learning

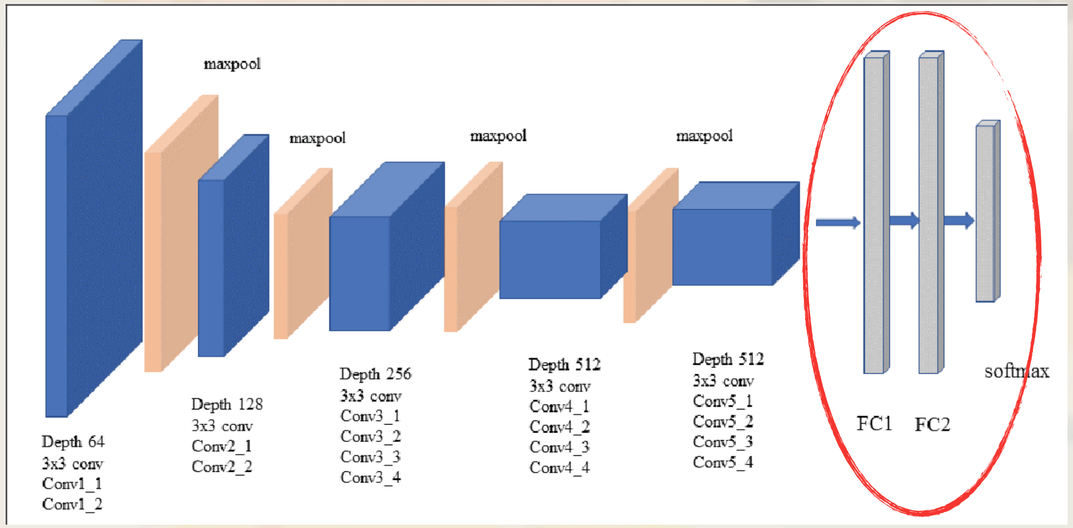
```python
# identify classifier
classifier_vgg = tf.keras.applications.VGG16(
                include_top=False,
                weights="imagenet",
                input_shape=(224,224,3),
                pooling='avg',
                classes=2,
                classifier_activation='softmax',
)
```

```python
# add new classifier layers
flat1 = Flatten()(classifier_vgg.layers[-1].output)
class1 = Dense(256, activation='relu')(flat1)
output = Dense(2, activation='softmax')(class1)
# define new model
classifier_vgg = Model(inputs=classifier_vgg.inputs, outputs=output)

```

- For the transfer learning we excludes all the top layers (output layers)
- We add our own top layer (FC)
- Flatten the output to 1D array
- We add in Dense (output layer)
- Final we will have 2 classes for output

SAMPLE MODEL

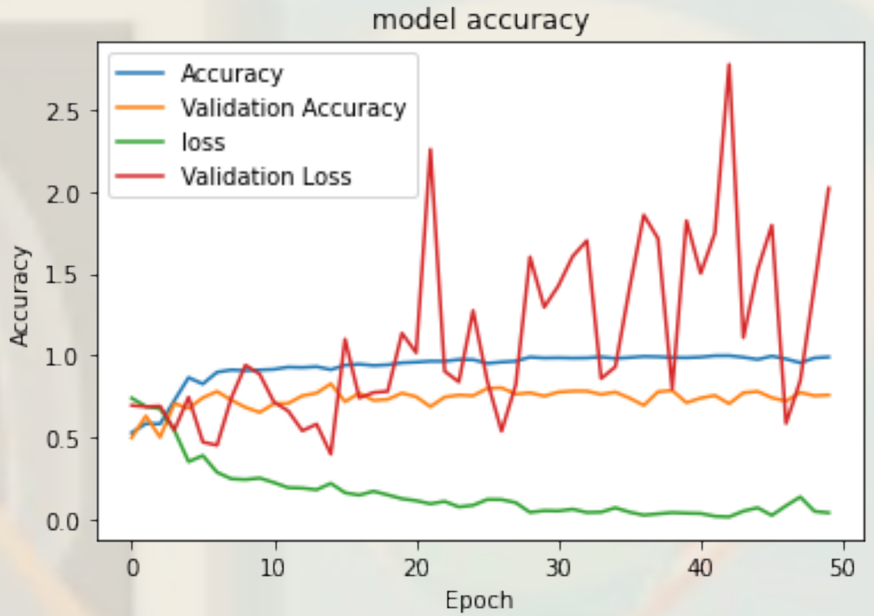


flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 256)	131328
dense_1 (Dense)	(None, 2)	514

```
Total params: 14,846,530
Trainable params: 14,846,530
Non-trainable params: 0
```

Model 1: VGG16

```
val_loss: 1.2725 - val_accuracy: 0.7520
Epoch 26/50
62/62 [==============================] - 32s 523ms/step - loss: 0.1303 - accuracy: 0.9433 -
val_loss: 0.8425 - val_accuracy: 0.7967
Epoch 27/50
62/62 [==============================] - 32s 519ms/step - loss: 0.0989 - accuracy: 0.9620 -
val_loss: 0.5352 - val_accuracy: 0.8008
Epoch 28/50
62/62 [==============================] - 32s 511ms/step - loss: 0.1296 - accuracy: 0.9519 -
val_loss: 0.8193 - val_accuracy: 0.7622
Epoch 29/50
62/62 [==============================] - 32s 508ms/step - loss: 0.0441 - accuracy: 0.9877 -
val_loss: 1.6008 - val_accuracy: 0.7703
Epoch 30/50
62/62 [==============================] - 32s 508ms/step - loss: 0.0333 - accuracy: 0.9880 -
val_loss: 1.2934 - val_accuracy: 0.7500
Epoch 31/50
62/62 [==============================] - 31s 505ms/step - loss: 0.0282 - accuracy: 0.9905 -
val_loss: 1.4248 - val_accuracy: 0.7744
```



Observation:

From the graph we can observe that on the 29 epoch the model is performing the best with

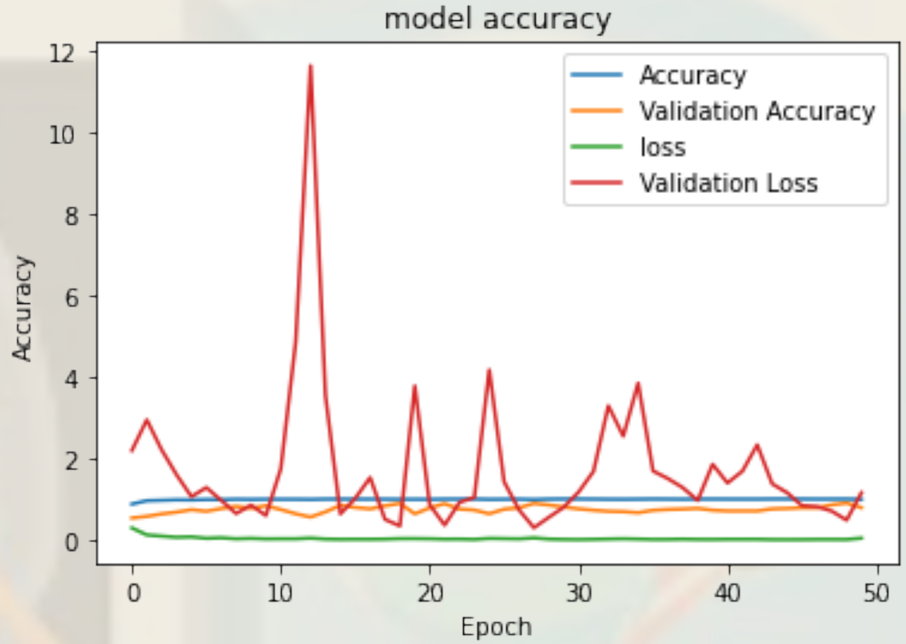
Accuracy: 98.77%

Val_accuracy: 77.03%

ANG WEE HWA 20/5/2021

Inception V3

```
val_loss: 1.3728 - val_accuracy: 0.7561
Epoch 45/50
62/62 [==============================] - 50s 803ms/step - loss: 3.3373e-04 - accuracy: 1.00
00 - val_loss: 1.1557 - val_accuracy: 0.7663
Epoch 46/50
62/62 [==============================] - 50s 803ms/step - loss: 0.0012 - accuracy: 0.9994 -
val_loss: 0.8364 - val_accuracy: 0.7825
Epoch 47/50
62/62 [==============================] - 51s 817ms/step - loss: 0.0016 - accuracy: 0.9990 -
val_loss: 0.8133 - val_accuracy: 0.7805
Epoch 48/50
62/62 [==============================] - 50s 813ms/step - loss: 0.0054 - accuracy: 0.9976 -
val_loss: 0.7023 - val_accuracy: 0.8557
Epoch 49/50
62/62 [==============================] - 50s 811ms/step - loss: 9.4007e-04 - accuracy: 0.99
99 - val_loss: 0.4791 - val_accuracy: 0.9004
Epoch 50/50
62/62 [==============================] - 51s 824ms/step - loss: 0.0177 - accuracy: 0.9935 -
val_loss: 1.1556 - val_accuracy: 0.7866
```

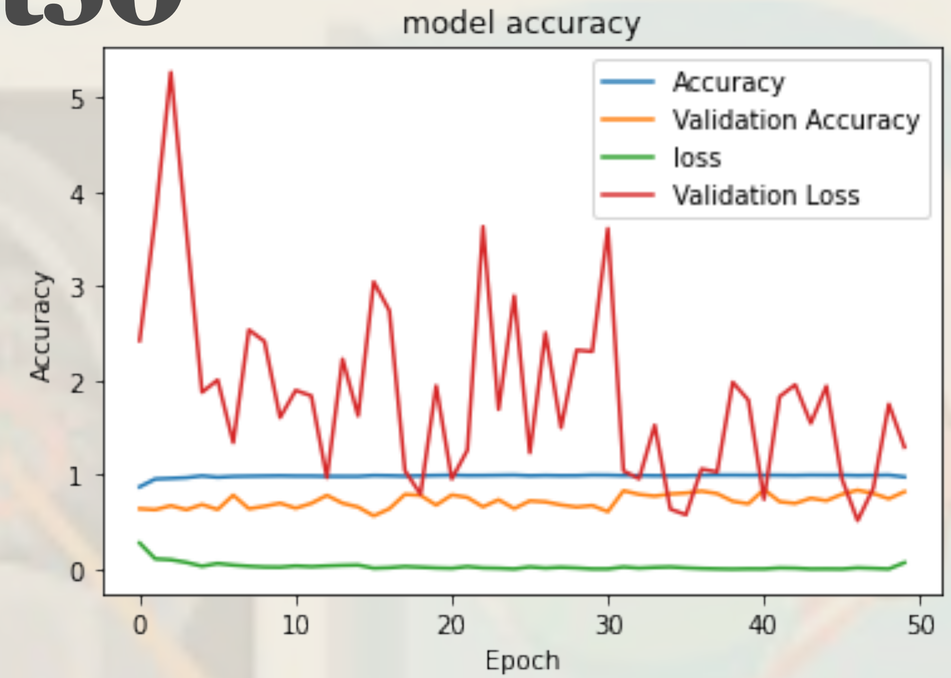


Observation:

From the graph we can observe that on the 49 epoch the model is performing the best with
Accuracy: 99.99%
Val_accuracy: 90.04%

ResNet50

```
Epoch 44/50
62/62 [==============================] - 34s 541ms/step - loss: 0.0042 - accuracy: 0.9999 -
val_loss: 1.5483 - val_accuracy: 0.7541
Epoch 45/50
62/62 [==============================] - 33s 532ms/step - loss: 0.0070 - accuracy: 0.9978 -
val_loss: 1.9407 - val_accuracy: 0.7276
Epoch 46/50
62/62 [==============================] - 33s 534ms/step - loss: 0.0019 - accuracy: 0.9995 -
val_loss: 0.9565 - val_accuracy: 0.7967
Epoch 47/50
62/62 [==============================] - 33s 528ms/step - loss: 0.0245 - accuracy: 0.9929 -
val_loss: 0.5174 - val_accuracy: 0.8394
Epoch 48/50
62/62 [==============================] - 33s 530ms/step - loss: 0.0145 - accuracy: 0.9947 -
val_loss: 0.8566 - val_accuracy: 0.8049
Epoch 49/50
62/62 [==============================] - 33s 529ms/step - loss: 0.0038 - accuracy: 0.9992 -
val_loss: 1.7496 - val_accuracy: 0.7480
Epoch 50/50
62/62 [==============================] - 33s 531ms/step - loss: 0.0365 - accuracy: 0.9890 -
```

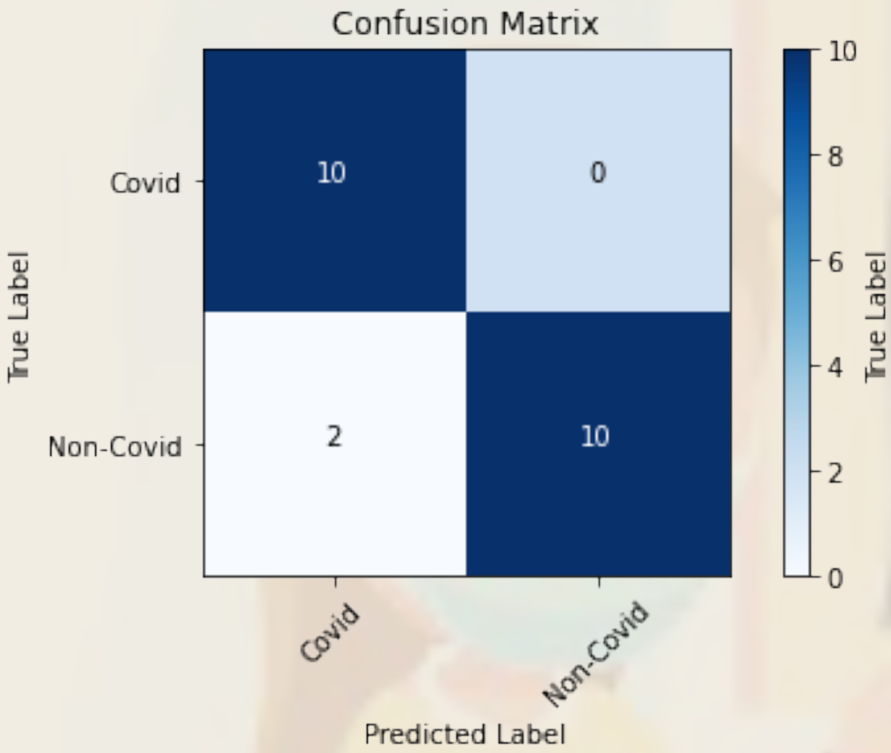


Observation:

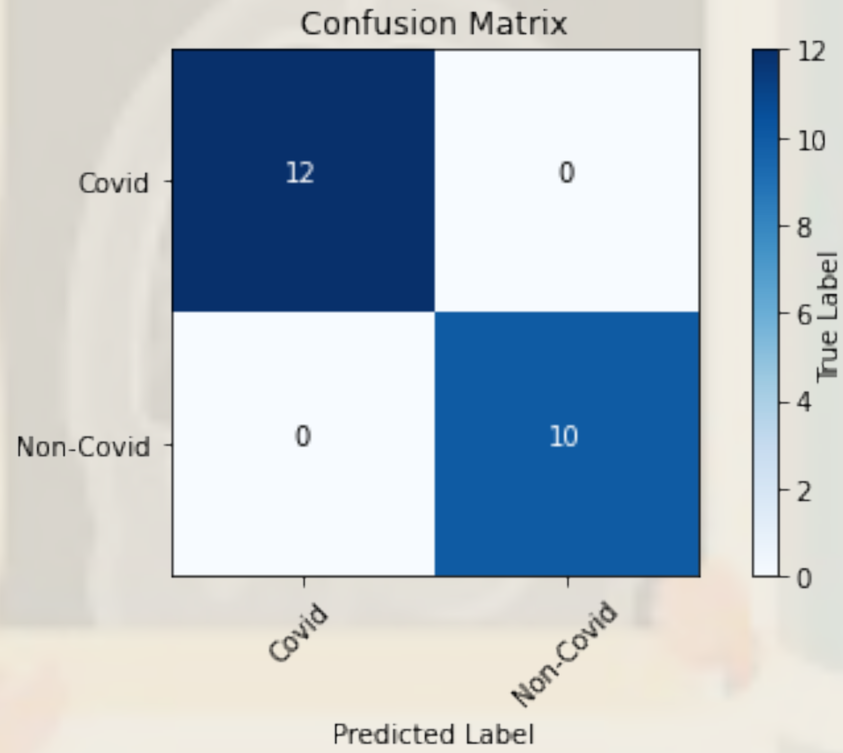
From the graph we can observe that on the 47 epoch the model is performing the best with
Accuracy: 99.29%
Val_accuracy: 83.94%

Test

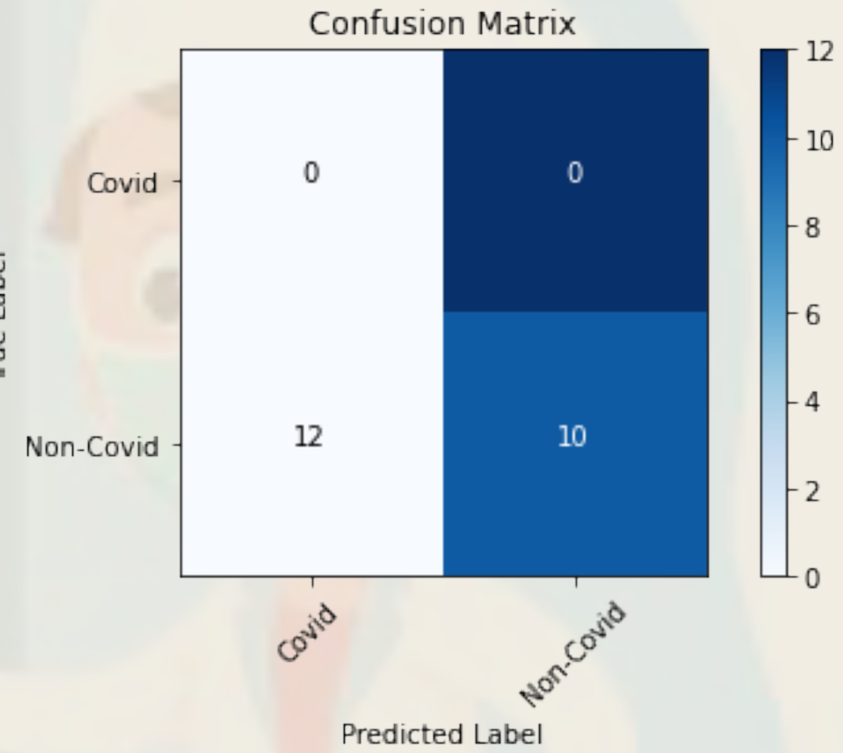
VGG16



InceptionV3



ResNet50

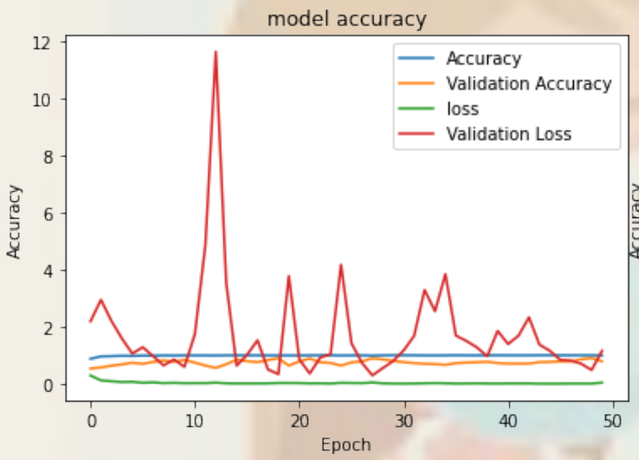


Hyperparameter Tuning

From the observation we can see that Inception model performs better for this study with 50 epoch.

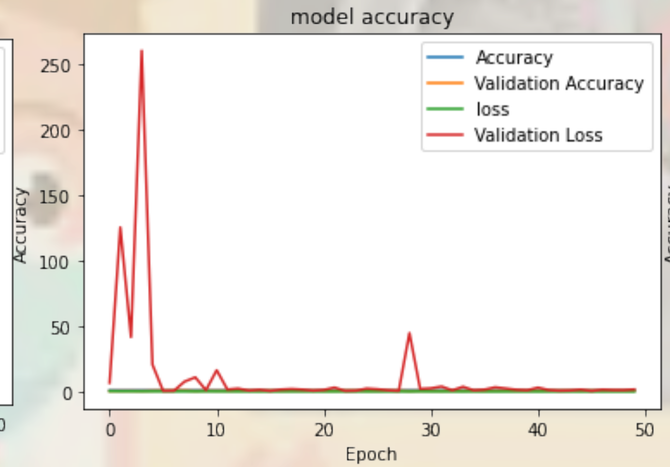
Next we will fine tune the learning rate and observe the difference.

Training Rate: 0.0001



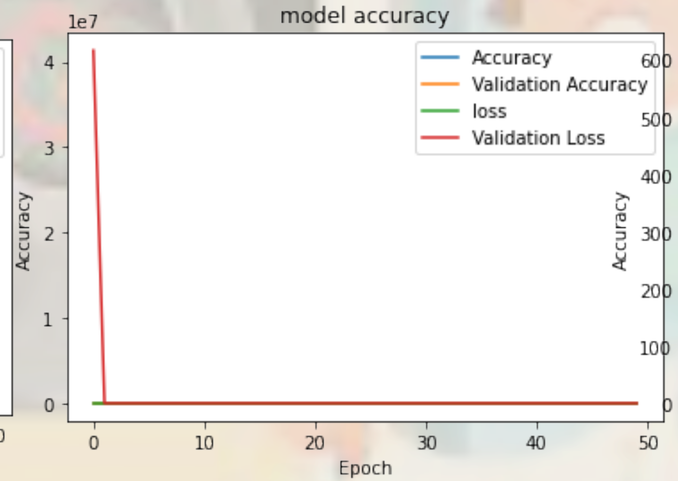
Best: 49 epoch
Accuracy: 99.99%
Val_accuracy: 90.04%

Training Rate: 0.001



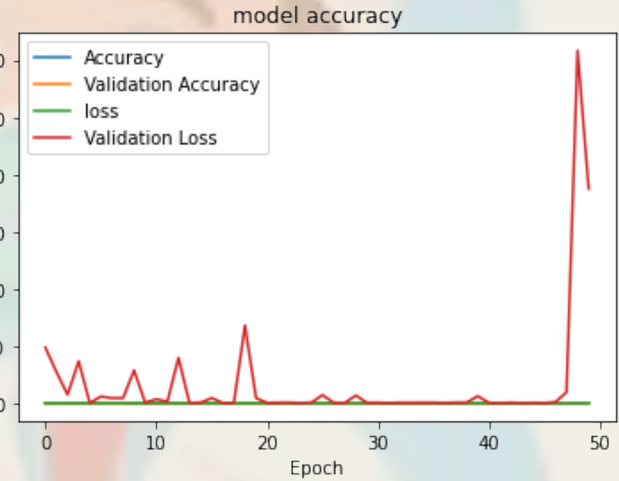
Best: 46 epoch
Accuracy: 99.18%
Val_accuracy: 86.99%

Training Rate: 0.01



Best: 50 epoch
Accuracy: 97.37%
Val_accuracy: 80.49%

Training Rate: 0.1



Best: 50 epoch
Accuracy: 90.32%
Val_accuracy: 80.23%

Hyperparameter Tuning

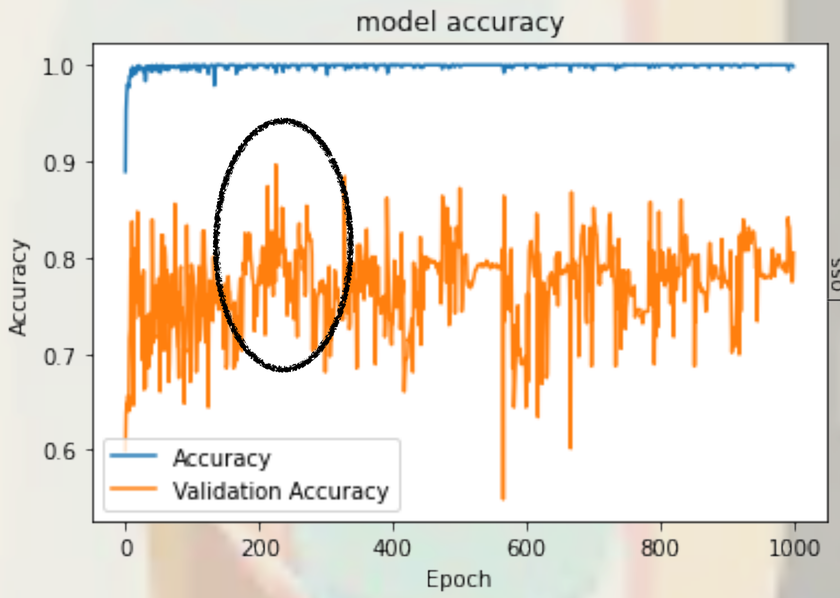
With Training Rate of 0.0001, we will train for 1000 epoch to observe and fine tune the model

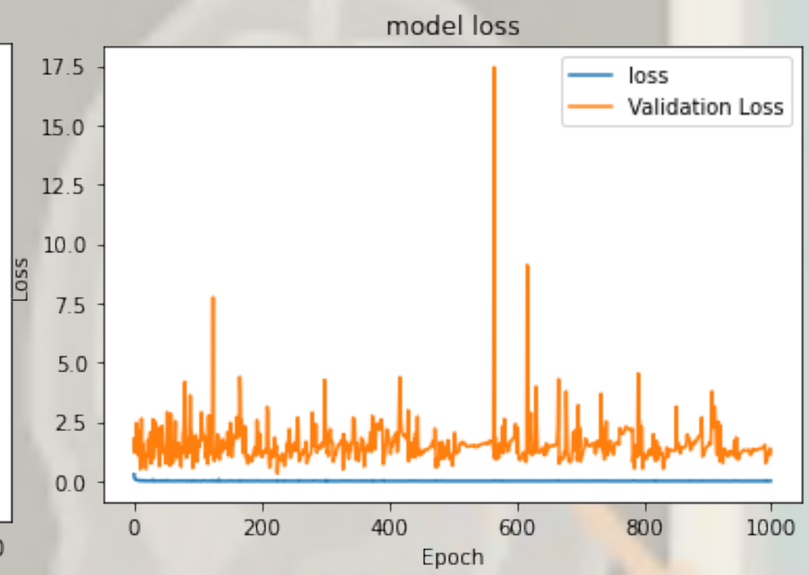
```
accuracy: 0.8313
Epoch 995/1000
62/62 [==============================] - 47s 755ms/step - loss: 6.0495e-04 - accuracy: 1.0000 - val_loss: 0.9955 -
val_accuracy: 0.8110
Epoch 996/1000
62/62 [==============================] - 47s 757ms/step - loss: 0.0011 - accuracy: 1.0000 - val_loss: 1.1327 - val_
accuracy: 0.7927
Epoch 997/1000
62/62 [==============================] - 47s 758ms/step - loss: 6.4484e-04 - accuracy: 1.0000 - val_loss: 1.2769 -
val_accuracy: 0.7846
Epoch 998/1000
62/62 [==============================] - 47s 755ms/step - loss: 6.8964e-04 - accuracy: 1.0000 - val_loss: 1.3271 -
val_accuracy: 0.7744
Epoch 999/1000
62/62 [==============================] - 47s 761ms/step - loss: 1.7510e-04 - accuracy: 1.0000 - val_loss: 1.3764 -
val_accuracy: 0.7764
Epoch 1000/1000
62/62 [==============================] - 47s 755ms/step - loss: 0.0081 - accuracy: 0.9979 - val_loss: 1.1390 - val_
accuracy: 0.8049
```

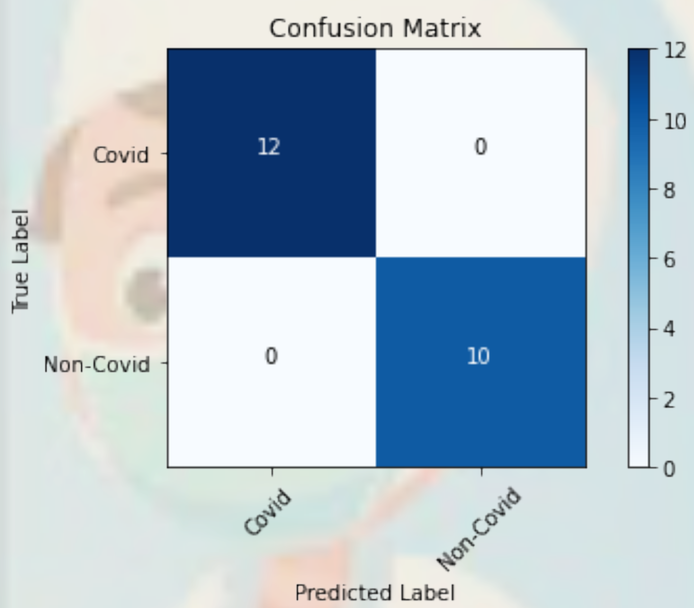
Although from observation we can see that the accuracy has dropped but after each epoch the validity result is stabilising and had had been consistent.

Hyperparameter Tuning

With Training Rate of 0.0001, we will train for 1000 epoch to observe and fine tune the model







Validity Accuracy is at above 80 percent and is increasing gradually. From the graph we can see that at 200 - 250 epoch the performance is maximised.

Dataset: https://drive.google.com/drive/folders/1QkZNjk6UVmUIMZ4vwRAdNOjzF_j4yR0b?usp=sharing
Notebook: https://colab.research.google.com/drive/1OhkOshybbhs-NFSC_fwFucmDBUwJi8Mt3?usp=sharing
Github: https://github.com/weehwa/ctscan_cnn_model_covid

“

The doctor of the future will give no medicine but will interest his patients in the care of the human frame, in diet and in the cause and prevention of disease.

- Thomas Edison