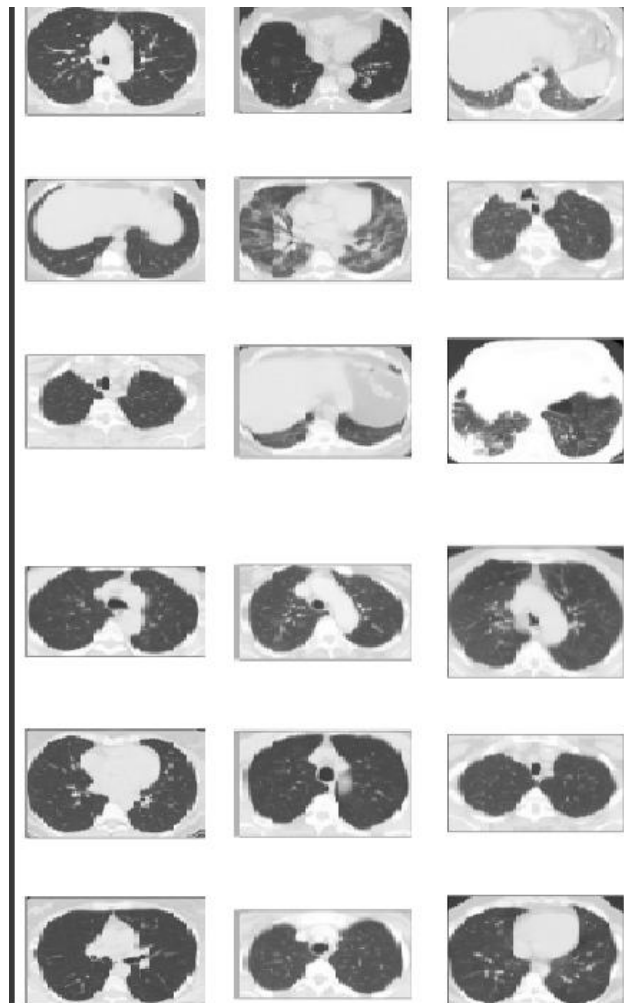
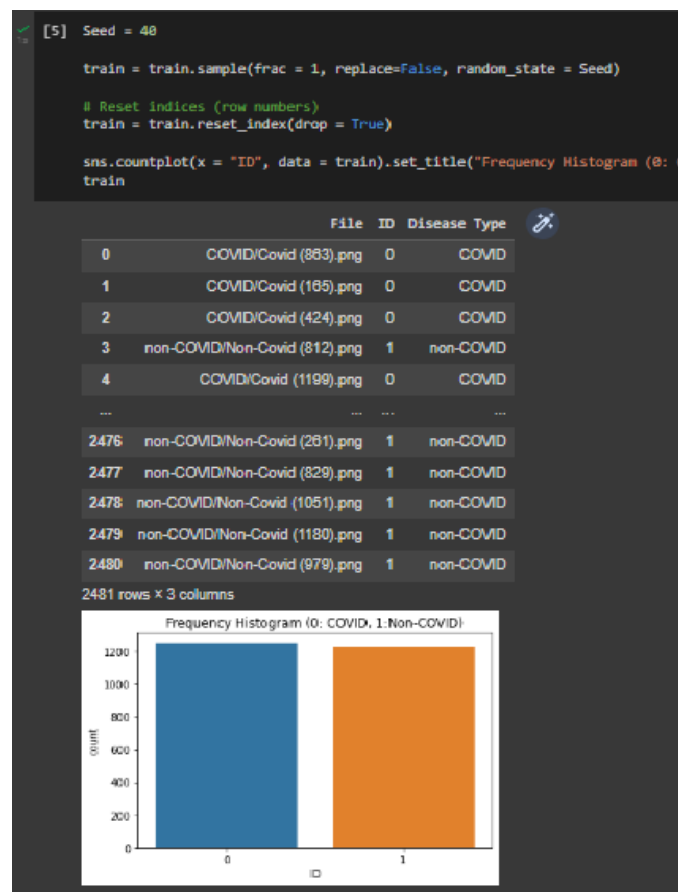


1) Load the images from the dataset.



2) Plot Histogram to see count of images



3) Build the ResNet50 model

```
# Architectural function for Resnet50
def build_resnet50(IMAGE_SIZE, channels):

    resnet50 = ResNet50(weights = 'imagenet', include_top = False)

    input = Input(shape = (IMAGE_SIZE, IMAGE_SIZE, channels))
    x = Conv2D(3, (3, 3), padding = 'same')(input)
    x = resnet50(x)
    x = GlobalAveragePooling2D()(x)
    x = BatchNormalization()(x)
    x = Dense(64, activation = 'relu')(x)
    x = BatchNormalization()(x)

    output = Dense(2, activation = 'softmax')(x)

    # model
    model = Model(input, output)

    optimizer = Adam(learning_rate = 0.003, beta_1 = 0.9, beta_2 = 0.999, epsilon = 0.1, decay = 0.0)
    model.compile(loss = 'categorical_crossentropy', # minimize the negative multinomial log-likelihood also known as the cross-entropy.
                  optimizer = optimizer,
                  metrics = ['accuracy'])
    model.summary()

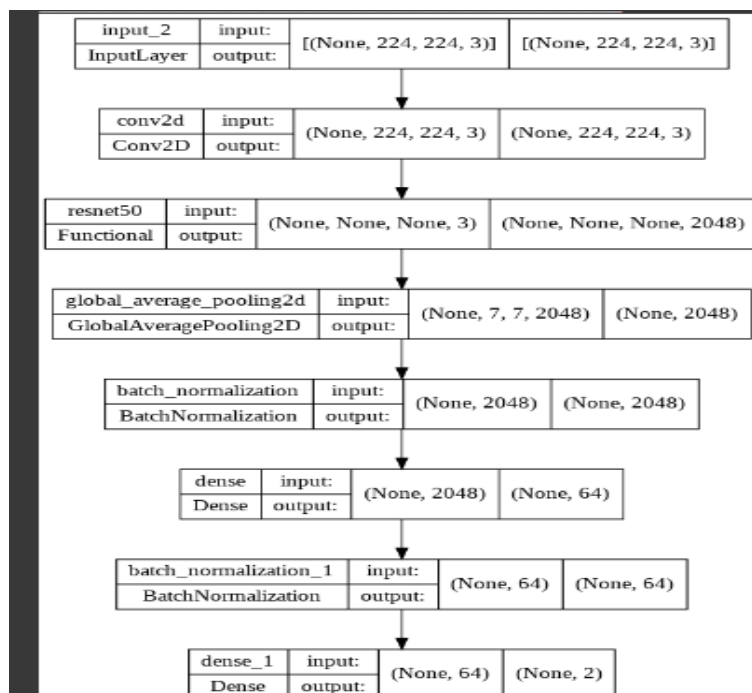
    return model
```

4) Compile and fit the model

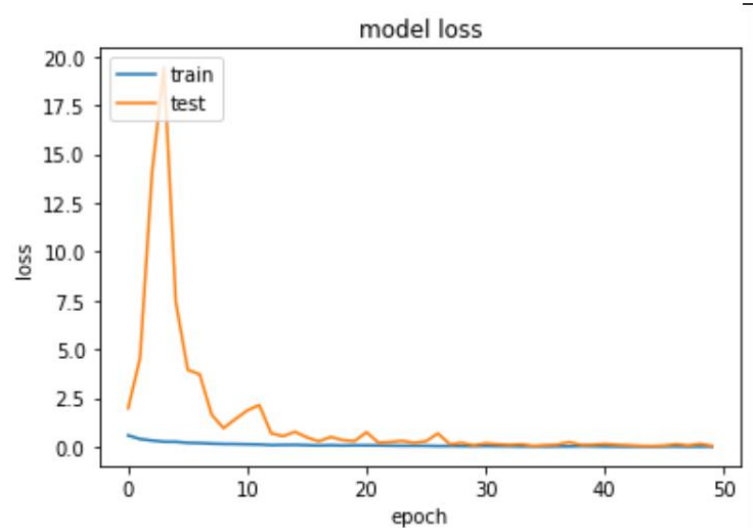
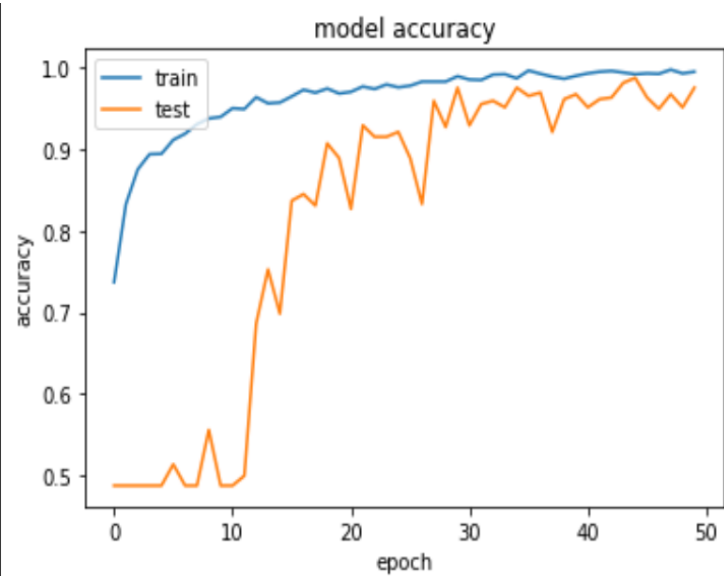
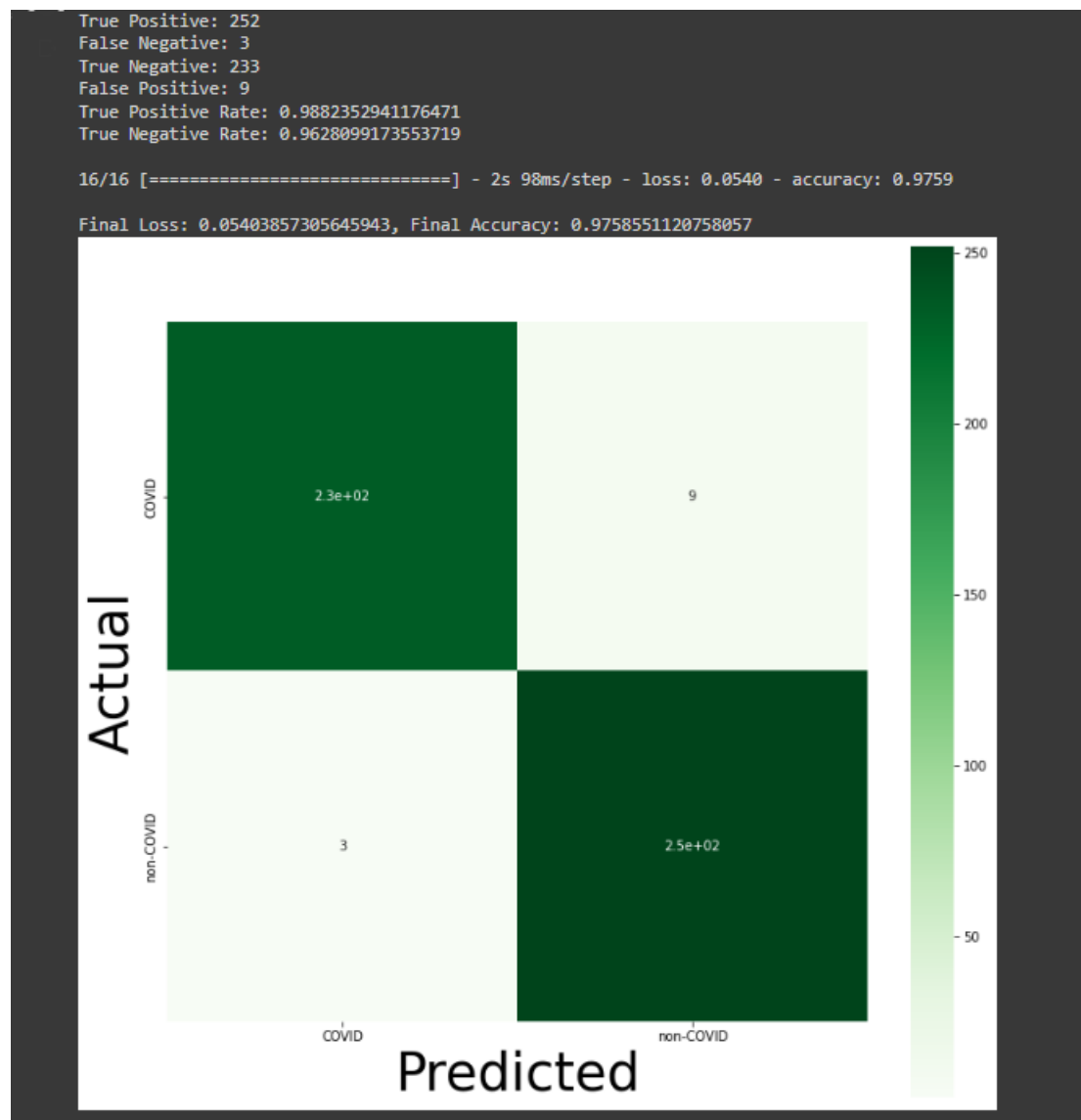
```
[*] Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5
94773248/94765736 [=====] - 1s 0us/step
94781440/94765736 [=====] - 1s 0us/step
Model: "model"
```

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
conv2d (Conv2D)	(None, 224, 224, 3)	84
resnet50 (Functional)	(None, None, None, 2048)	23587712
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
batch_normalization (BatchNormalization)	(None, 2048)	8192
dense (Dense)	(None, 64)	131136
batch_normalization_1 (BatchNormalization)	(None, 64)	256
dense_1 (Dense)	(None, 2)	130

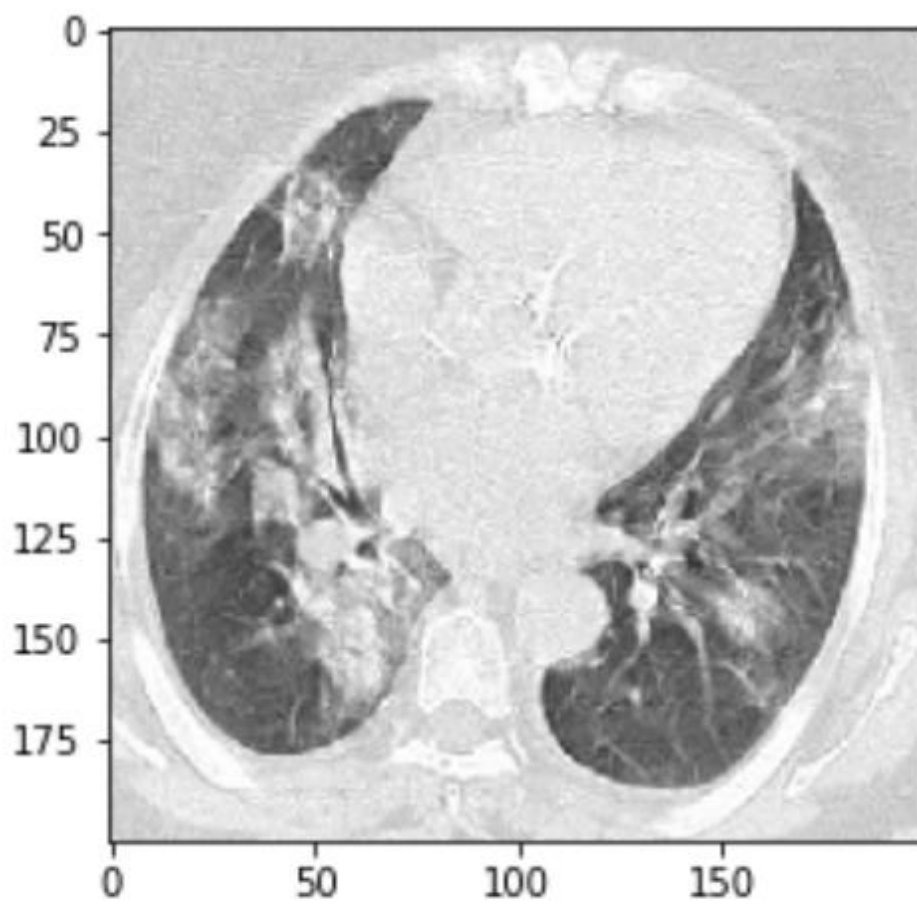
```
Total params: 23,727,510
Trainable params: 23,678,166
Non-trainable params: 57,344
```



5) Confusion matrix , model loss and accuracy



6) Predicted image



Prediction: Covid-19