Muhammad Junaid Ahmad

BSCS 17012

Deep Learning

Mohsen Ali

Assignment – 5

Task 1:

I tried various learning rates and momentum but the values I have given in the report gave the best results. The VGG-16 took much longer to train on the same number of epochs than ResNet-18.

VGG-16 FC-Layer Tuning

The training accuracy of **99.66%**, testing accuracy of **92.01%** and validation accuracy of **85.42%** was achieved by fine tuning the FC-Layers with drop out of 50% and ReLu activation with **learning rate = 0.0001 in 50 epochs and momentum of 0.8**

```
F1 Score = 0.919387875220718
```

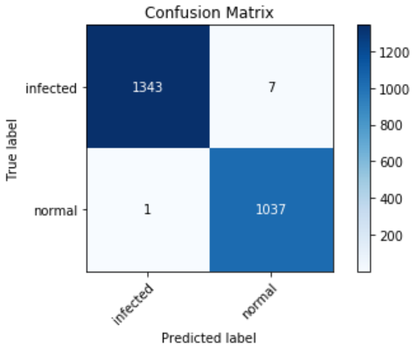


Figure 1 Training

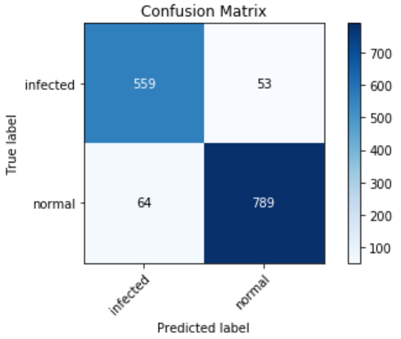


Figure 2 Testing

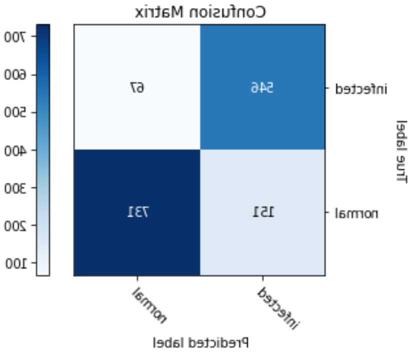
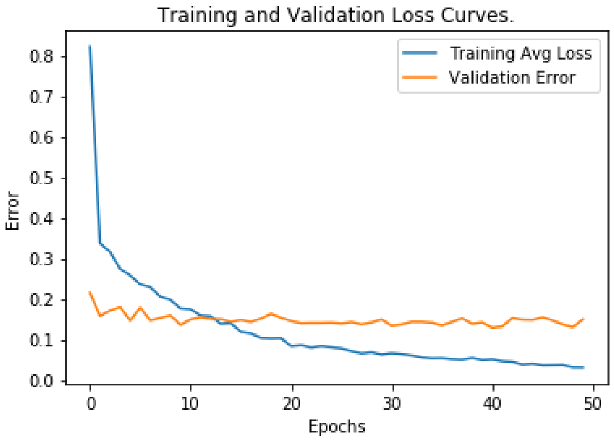
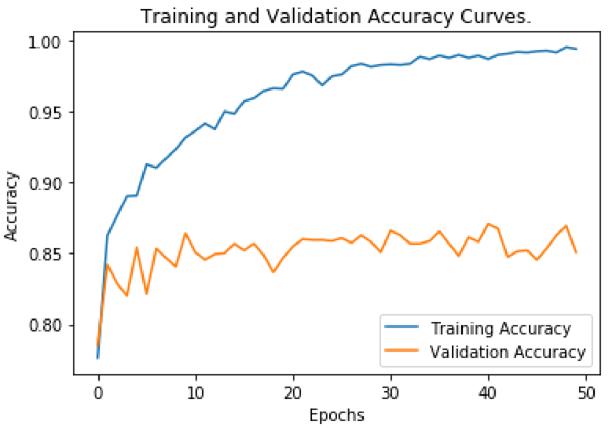
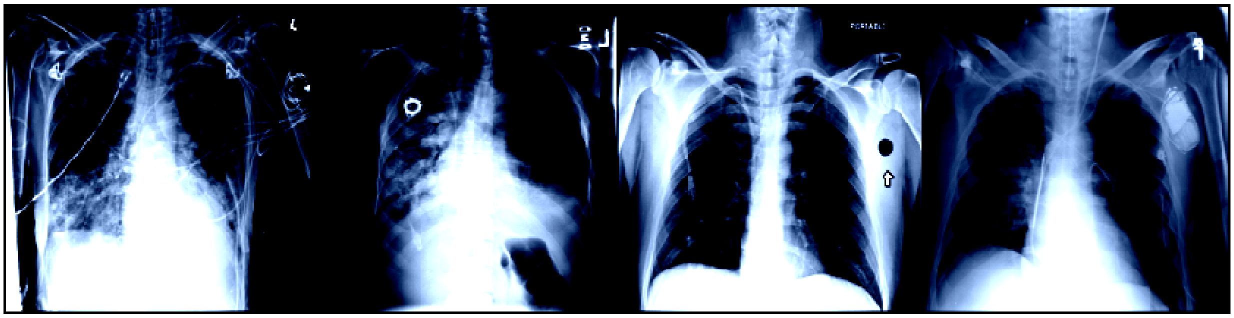


Figure 3 Validation







```
[['P=33.78, acutal=inf, pred=inf'], ['P=33.28, actual=inf, pred=inf'], ['P=26.08, actual=nor, pred=inf'], ['P=21.00, actual=inf, pred=nor']]
```

ResNet18 FC-Layer Tuning

The training accuracy of **86.52%**, testing accuracy of **84.64%** and validation accuracy of **79.06%** was achieved by fine tuning the FC-Layers with drop out of 50% and ReLu activation with **learning rate = 0.0001 in 50 epochs and momentum of 0.9**

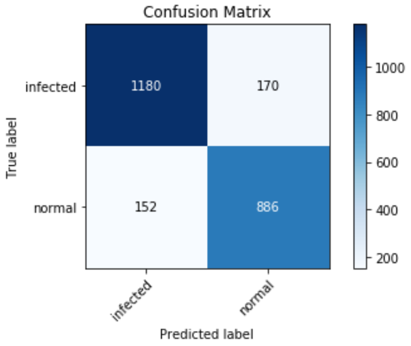


Figure 2 Training

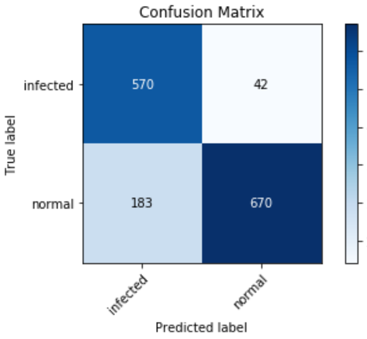


Figure 2 Testing

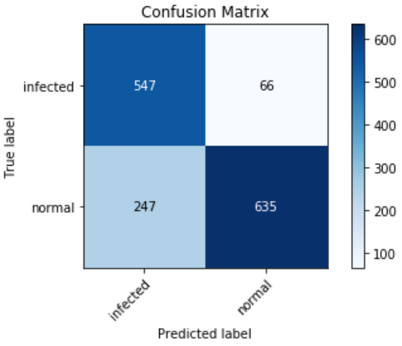
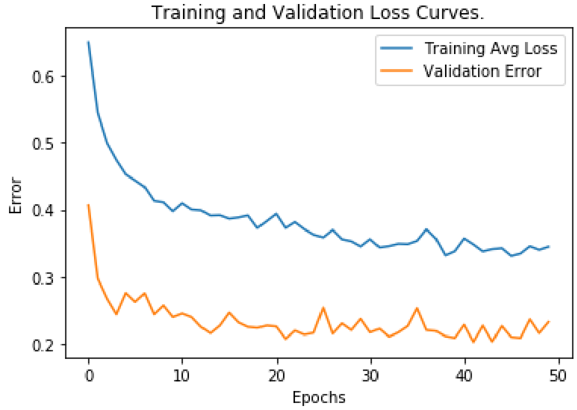
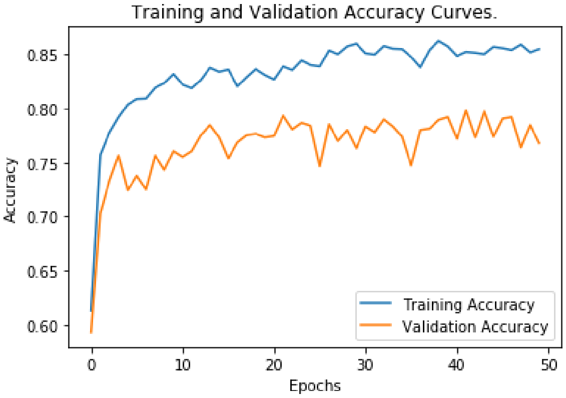
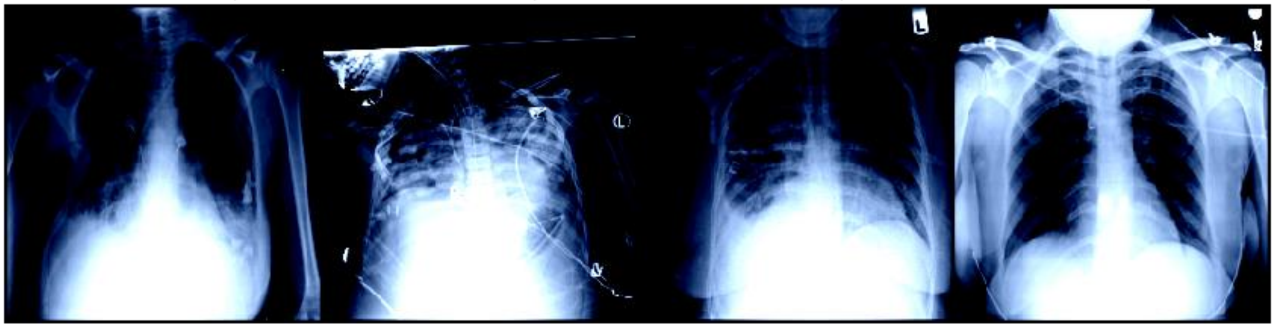


Figure 3 Validation





The VGG16 model with only FC-Layers tuned gave a lot higher accuracy than the ResNet18 model with FC-Layers tuned because the VGG16 gives much higher number of features to the FC layer on which actual training is happening. ResNet18 model depends largely on Convu Layers and only give 512 feature vector which can not be classified as properly as 25088 feature vector in case of VGG-16.



[['P=4.45, acutal=inf, pred=inf'], ['P=3.78, actual=inf, pred=inf'], ['P=3.38, actual=nor, pred=inf'], ['P=2.31, actual=nor, pred=inf']]

Task 2:

VGG-16 last Conv and FC-layer tuning

The training accuracy of **99.79%**, testing accuracy of **91%** and validation accuracy of **86.89%** was achieved by fine tuning the all layers with drop out of 50% in FC layer and ReLu activation with **learning rate = 0.0001 in 50 epochs and momentum of 0.8**

```
F1 Score = 0.914357187522718
```

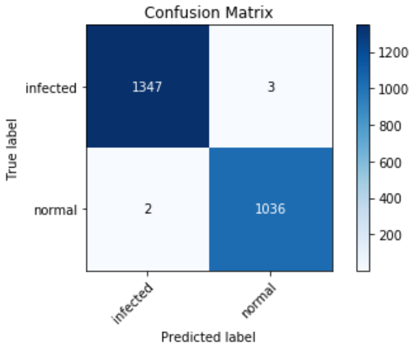


Figure 3 Training

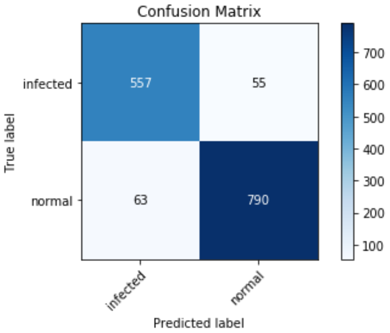


Figure 2 Testing

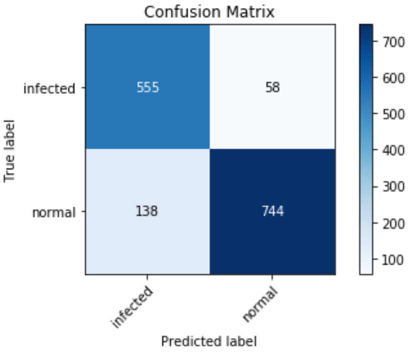
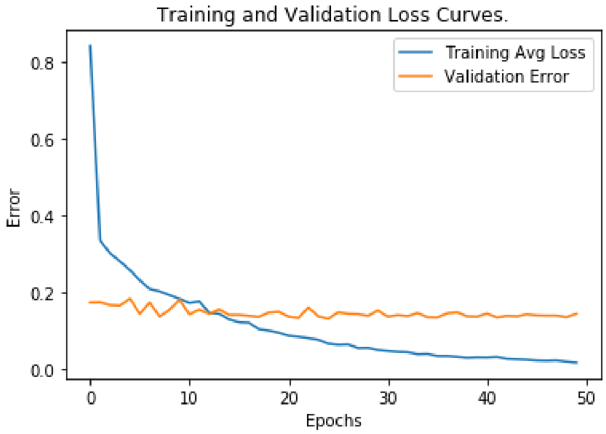
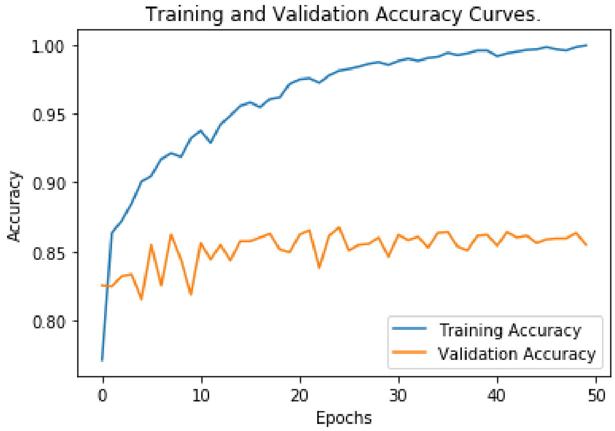


Figure 3 Validation





VGG-16 all layers tuning

The training accuracy of **100%**, testing accuracy of **95%** and validation accuracy of **89.23%** was achieved by fine tuning the all layers with drop out of 50% in FC layer and ReLu activation with **learning rate = 0.0001 in 50 epochs and momentum of 0.8,**

```
F1 Score = 0.959387875220718
```

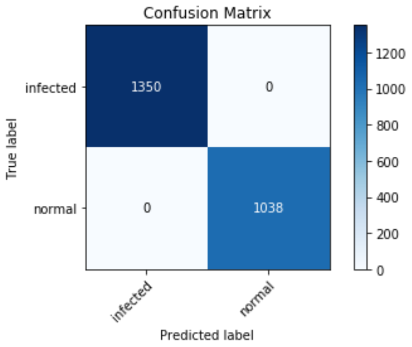


Figure 4 Training

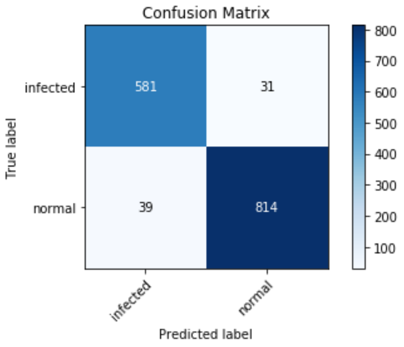


Figure 2 Testing

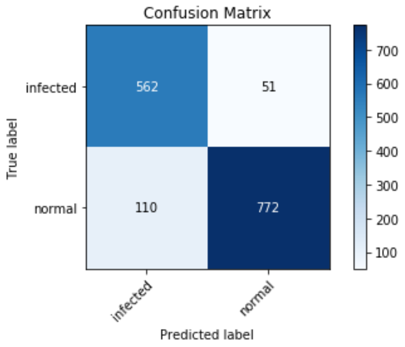
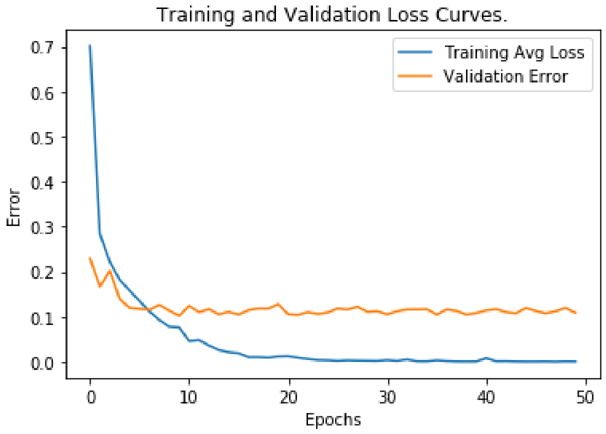
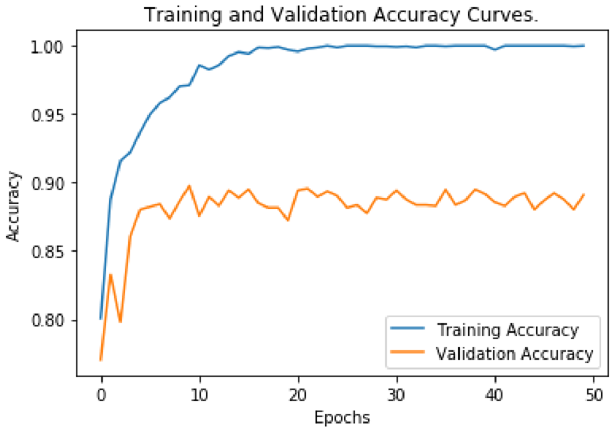
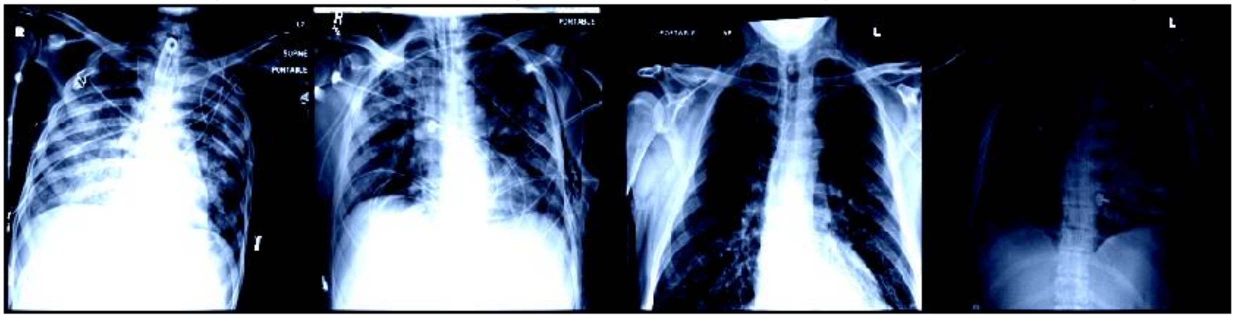


Figure 3 Validation





[['P=61.45, acutal=inf, pred=inf'], ['P=60.24, actual=inf, pred=inf'], ['P=37.59, actual=nor, pred=inf'], ['P=36.04, actual=nor, pred=inf']]



ResNet-18 last Conv (layer4) and FC-layer tuning

The training accuracy of **98.83%**, testing accuracy of **89.62%** and validation accuracy of **83.21%** was achieved by fine tuning the all layers with drop out of 50% in FC layer and ReLu activation with **learning rate = 0.0001 in 50 epochs and momentum of 0.9**

```
F1 Score = 0.897364341085270
```

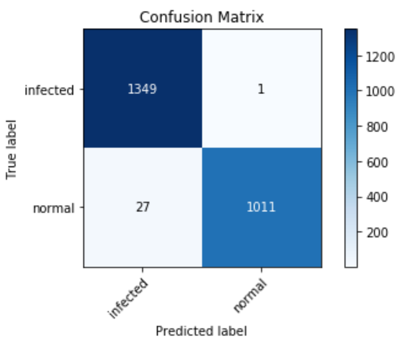


Figure 5 Training

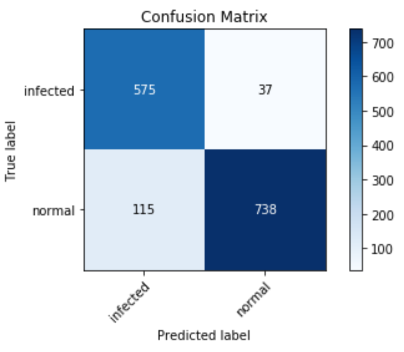


Figure 2 Testing

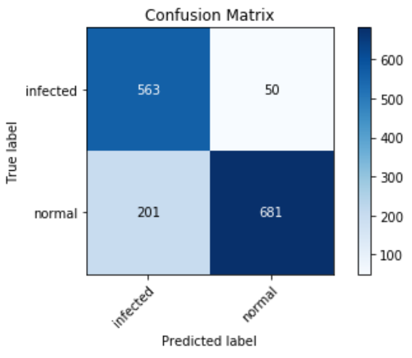
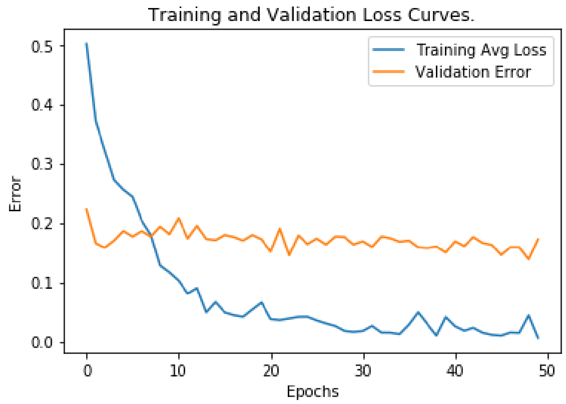
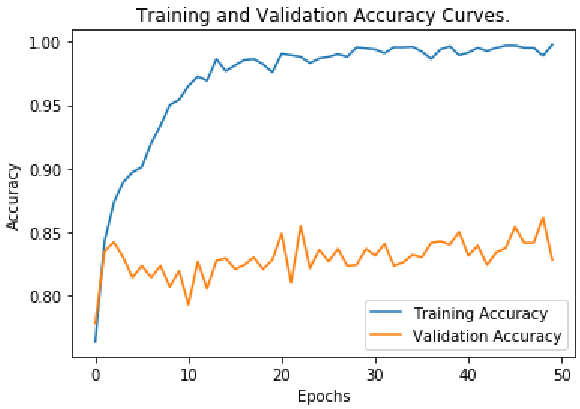


Figure 3 Validation

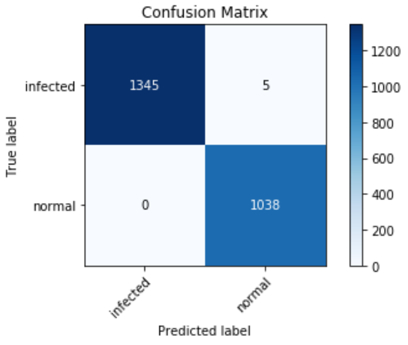


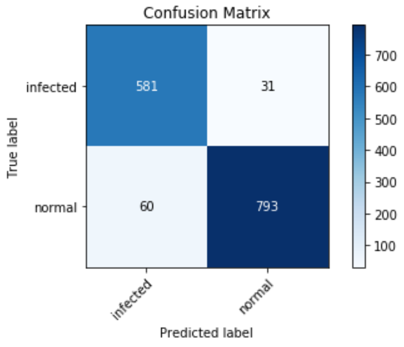


ResNet-18 all layers tuning

The training accuracy of **99.79%**, testing accuracy of **93.79%** and validation accuracy of **88.23%** was achieved by fine tuning the all layers with drop out of 50% in FC layer and ReLu activation with **learning rate = 0.0001 in 50 epochs and momentum of 0.9**

```
F1 Score = 0.945736434108527
```





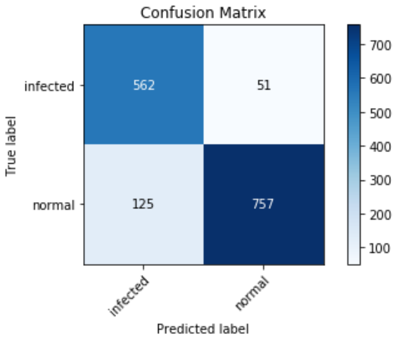
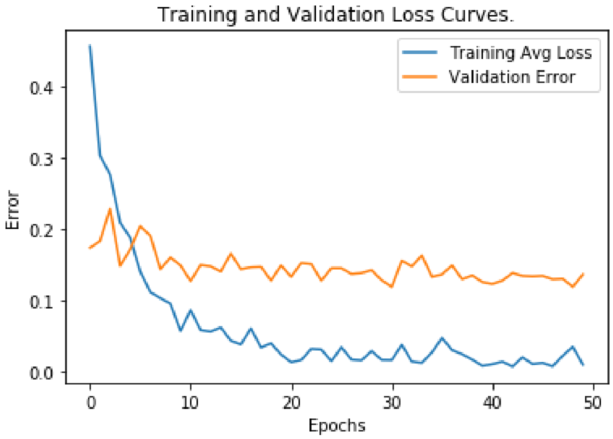
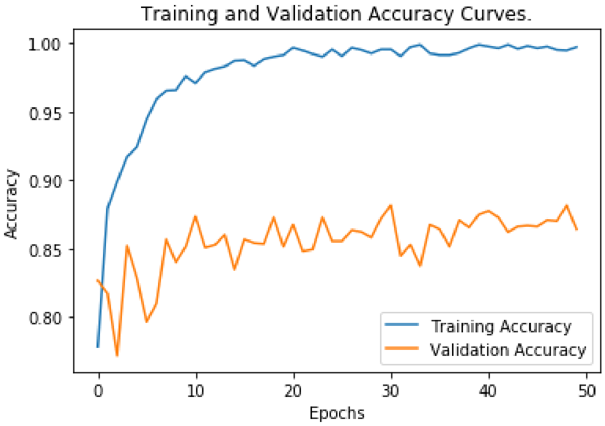
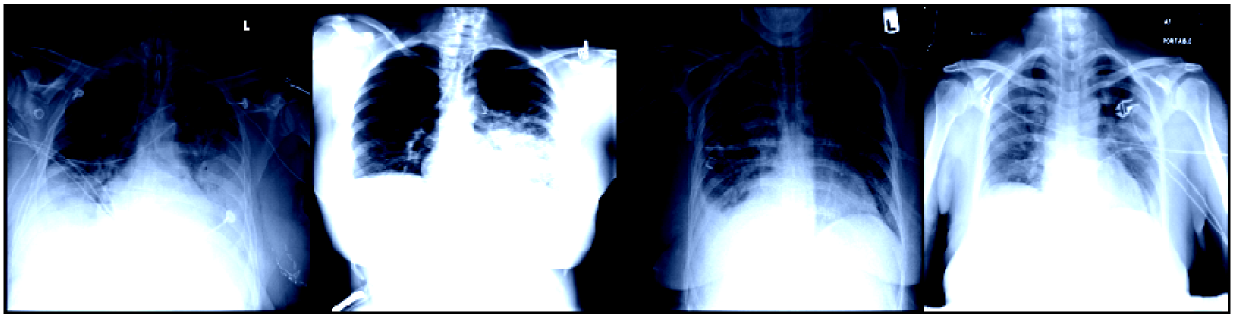


Figure 6 Training *Figure 2 Testing* *Figure 3 Validation*





```
[['P=22.23, acutal=inf, pred=inf'], ['P=14.11, actual=inf, pred=inf'], ['P=9.24, actual=nor, pred=inf'], ['P=8.63, actual=nor, pred=inf']]
```



As seen in the above two tasks there is a big improvement when we unfreeze convolutional layers and allowed them to be tuned as well on our data. As there are some features that must be learned to efficiently separate/classify our data. So, when we let the feature learning layer i.e. convolutional layers to learn. This layer then start learning to extract features particular to our dataset which helps improve the accuracy.

When we allowed more convolutional layers to be tuned the accuracy improved more becaus now more features can be tuned to fit our data.

GitHub Repository Link:

https://github.com/Muhammad-Junaid-Ahmad/BSCS17012_COVID19_DLSpring2020