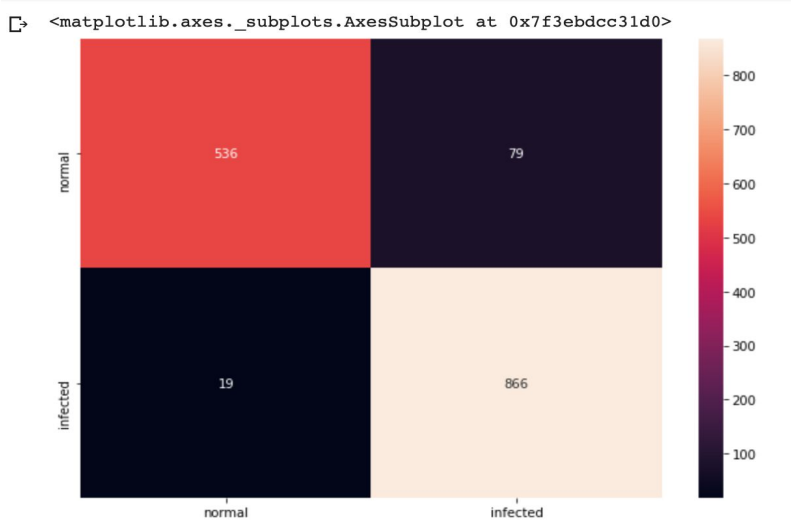
Task1:

1 - Freeze all Layers except FC Layer
2 - replace FC layer with 2 FC layers (1st 1input = 25088, out = 790, 2nd in = 790, out 2)

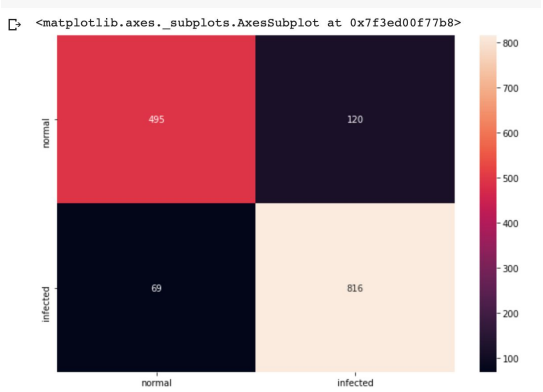
VGG-16
Confusion Matrix of results

Learning Rate = 0.001
Batch Size = 200
Epochs = 5

Trained on entire data
Test accuracy = 94%
Validation Accuracy = 87%



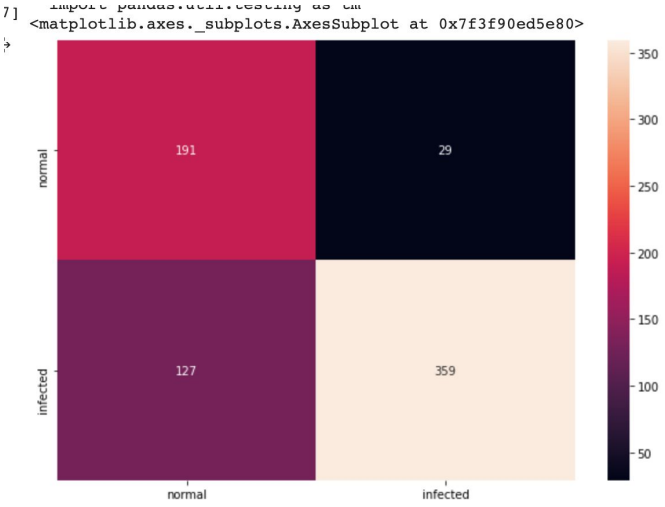
Validation Confusion MAtrix



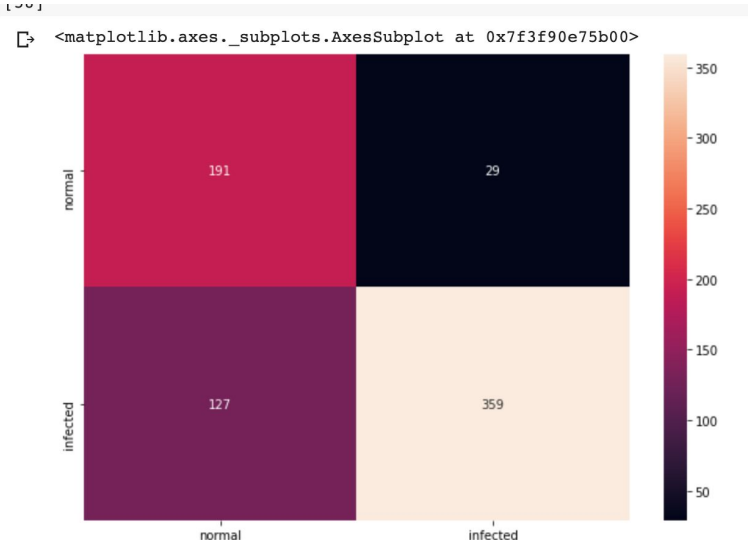
ResNet18

LAyers manipulation same like VGG
Confusion Matrix of results

Learning Rate = 0.001
Batch Size = 200
Epochs = 20
Training: class accuracy = `[81.41361257 83.54114713]`



Validation= class accuracy = `[normal = 86.81818182 inffected = 73.86831276]`



Task 2:

VGG
Unfreeze every fifth layer in network
And extended existing FC layers with 2 new layers same as fc layer in task 1 vgg.

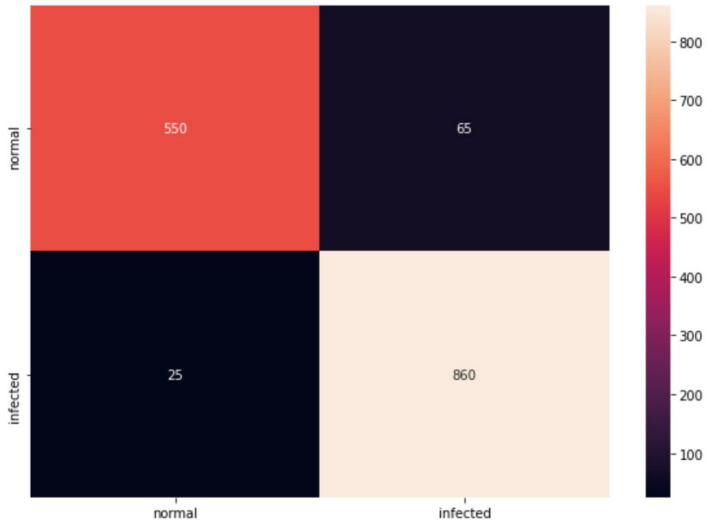
Training data:

Learning Rate = 0.001
Batch Size = 200
Epochs = 5

class accuracy = [89.43089431 97.17514124]

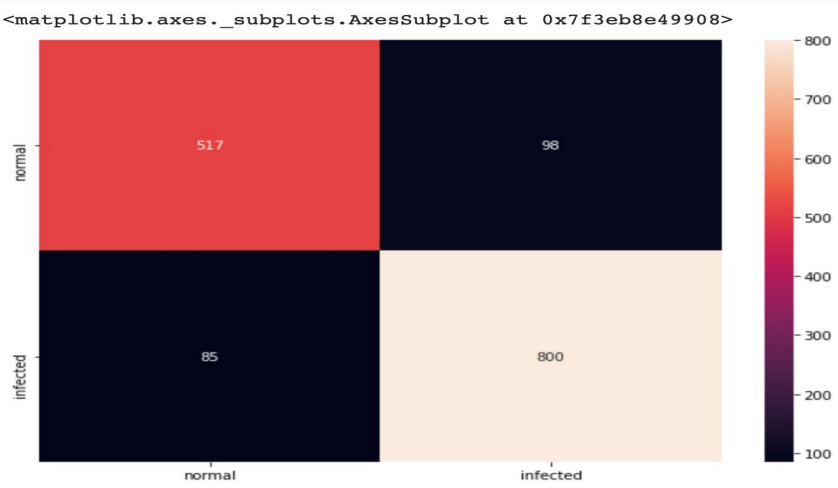
```
]  sn.heatmap(df_cm, annot=True, fmt='g')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f2dd06320>
```




```
Validation: class accuracy [84.06504065 90.39548023]
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3eb8e49908>
```

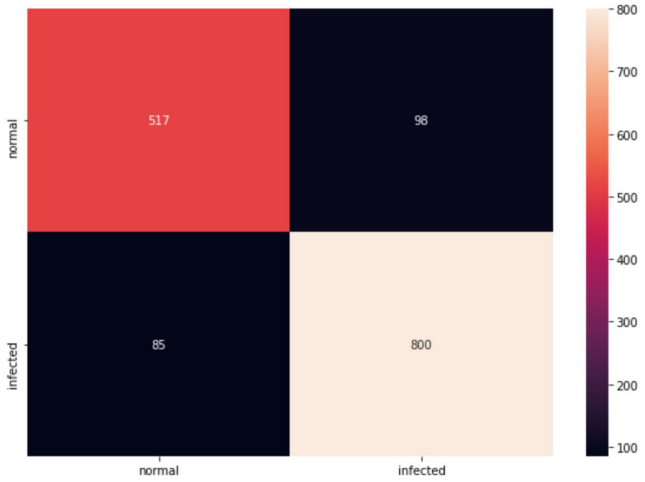


ResNet

Learning Rate = 0.001
Batch Size = 200
Epochs = 20

Training class accuracy = `[86.91099476 90.02493766]`

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3eb8e49908>
```




```
Validation test accuracy :  [87.43455497 90.27431421]
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f70fd44e0>
```