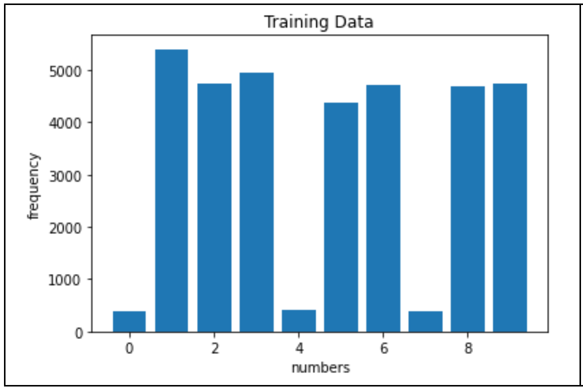
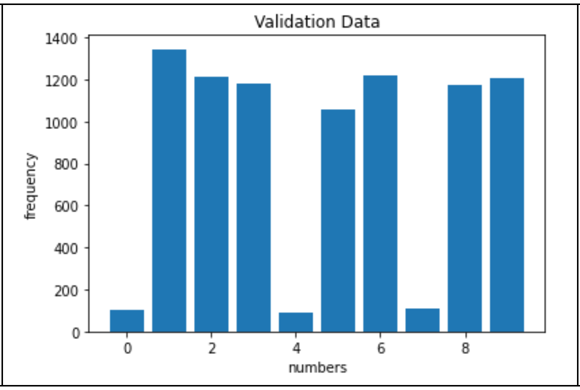
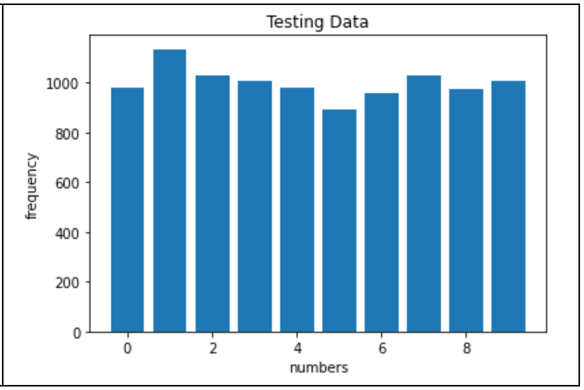
**Muhammad Muneeb Ur Rahman BSCS19057**

**The initial distribution of each class of data is shown below in the figures.**



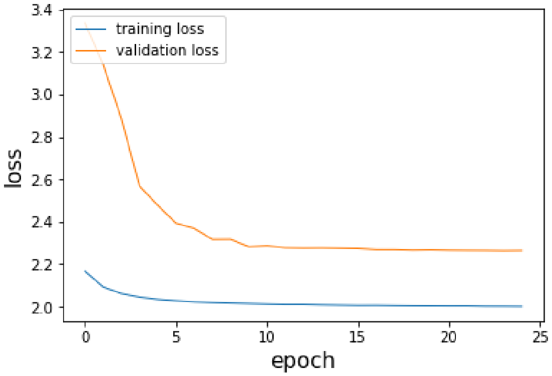




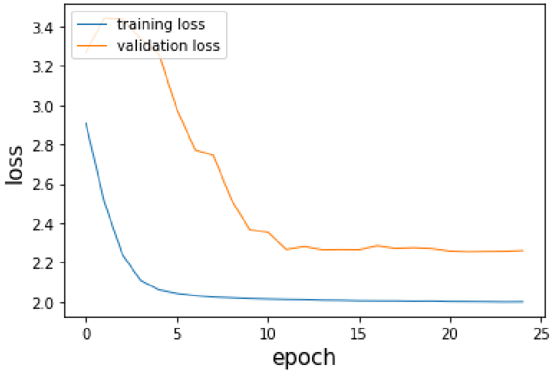
The distribution of training data is unbalance, the class 0, 4 and 7 have less training examples in the dataset.

1. **Compare results of simple convolution block base architecture vs MobileNet block-based architecture and report results with reasoning.**

<table>
<tr><td>Architecture a) With MobileNet blocks</td><td>Architecture b) With simple Conv Network</td></tr>
<tr><td colspan="2"><b>Epochs</b> = 25, <b>Lr</b> = 0.01, <b>Loss</b> = CrossEntropyLoss, <b>Optimizer</b> = Adam, <b>Val</b> = 20 % training data provided.</td></tr>
<tr><td>

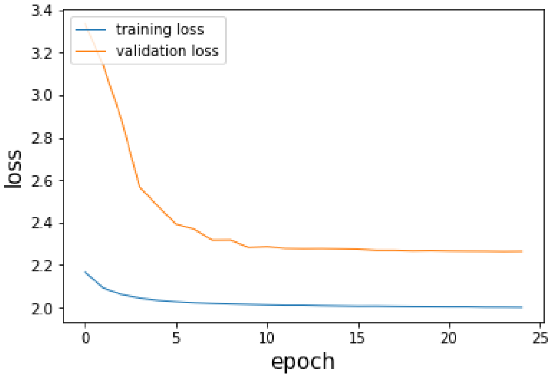
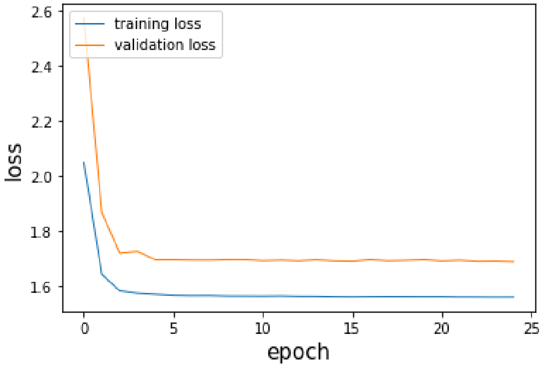
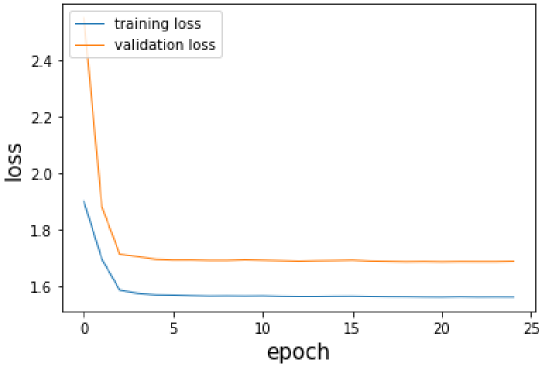


</td><td>



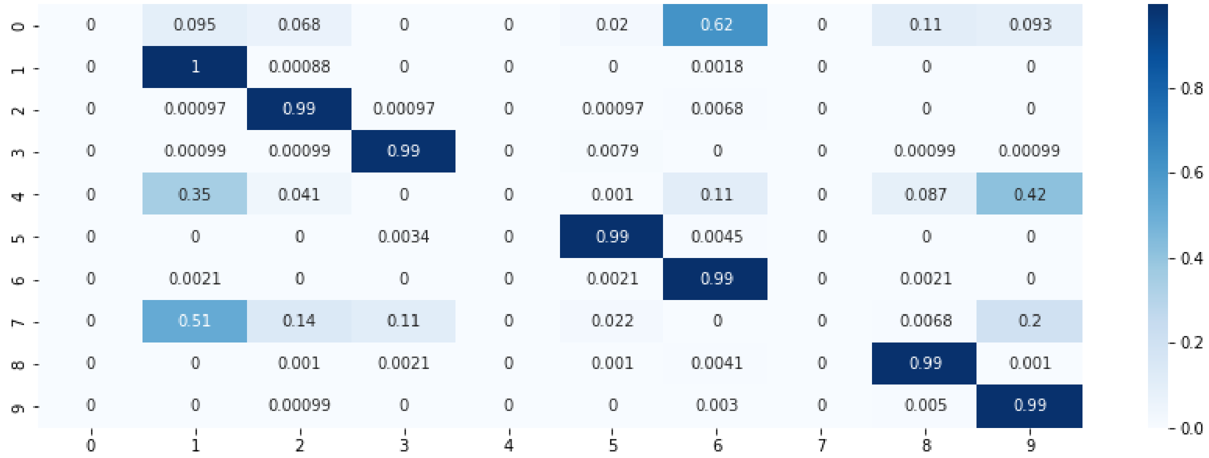
</td></tr>
<tr><td>test dataset accuracy is 69.29693603515625</td><td>test dataset accuracy is 69.44694519042969</td></tr>
<tr><td colspan="2">The loss curve and the accuray is same for the both achitectures because the number of total blocks are equal.<br>• Simple Convolution Network has the 4 Conv Blocks.<br>• Network with MobileNet Block has 1 Conv Block and 3 MobileNet Blocks.<br>• The capability that the MobileNet Blocks add is that there is lesser parameters to train due to which the training time is less and and running time is less with few decrease in the accuracy. As in this case the difference is 0.2% which is affordable as compared to the time efficency.</td></tr>
</table>

## 2. Use no of different conv block and MobileNet block and compare its accuracy.

<table>
<tr><td>1 Conv Block + 3 MobileNet Blocks</td><td>3 Conv Blocks + 3 MobileNet Blocks</td><td>3 Conv Blocks + 5 MobileNet Blocks</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td>test dataset accuracy is 69.29693603515625</td><td>test dataset accuracy is 69.7369766235351</td><td>test dataset accuracy is 69.64696502685547</td></tr>
<tr><td colspan="3">There is not much increase in the accuracy by increasing the number of layers which means the all three models have capacity to represent the and learn the features of data. But the accuracy could not improve because of imbalanced data.</td></tr>
</table>

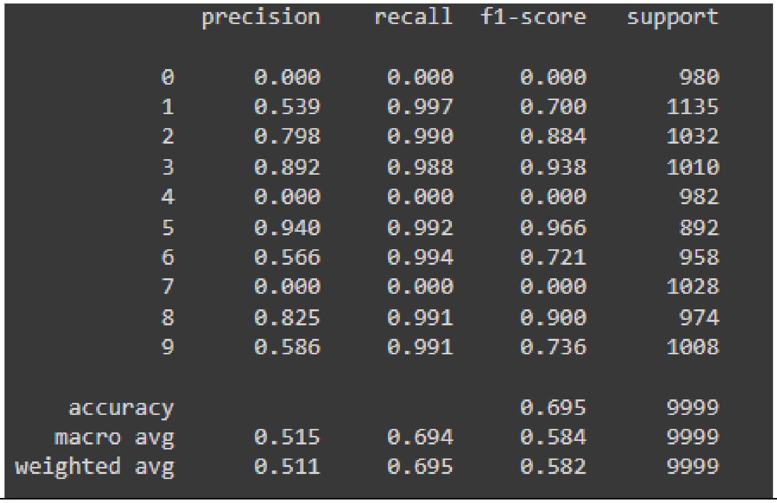
## 3. Confusion matrices, Recall, and Accuracy for the testing set.

Confusion Matrix



|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0.095 | 0.068 | 0 | 0 | 0.02 | 0.62 | 0 | 0.11 | 0.093 |
| 1 | 0 | 1 | 0.00088 | 0 | 0 | 0 | 0.0018 | 0 | 0 | 0 |
| 2 | 0 | 0.00097 | 0.99 | 0.00097 | 0 | 0.00097 | 0.0068 | 0 | 0 | 0 |
| 3 | 0 | 0.00099 | 0.00099 | 0.99 | 0 | 0.0079 | 0 | 0 | 0.00099 | 0.00099 |
| 4 | 0 | 0.35 | 0.041 | 0 | 0 | 0.001 | 0.11 | 0 | 0.087 | 0.42 |
| 5 | 0 | 0 | 0 | 0.0034 | 0 | 0.99 | 0.0045 | 0 | 0 | 0 |
| 6 | 0 | 0.0021 | 0 | 0 | 0 | 0.0021 | 0.99 | 0 | 0.0021 | 0 |
| 7 | 0 | 0.51 | 0.14 | 0.11 | 0 | 0.022 | 0 | 0 | 0.0068 | 0.2 |
| 8 | 0 | 0 | 0.001 | 0.0021 | 0 | 0.001 | 0.0041 | 0 | 0.99 | 0.001 |
| 9 | 0 | 0 | 0.00099 | 0 | 0 | 0 | 0.003 | 0 | 0.005 | 0.99 |

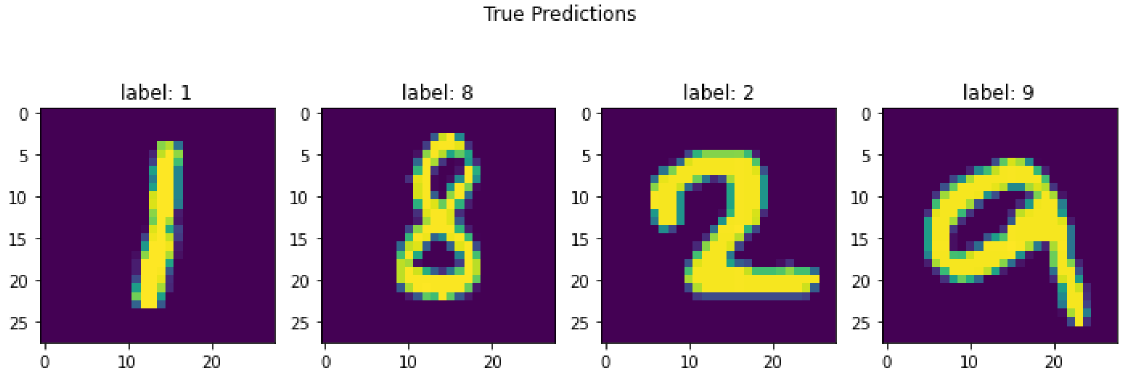
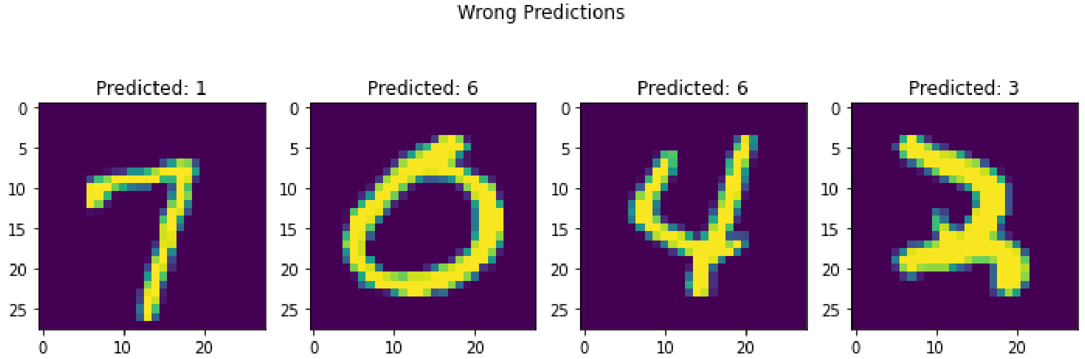
Precision
Recall
F1-Score



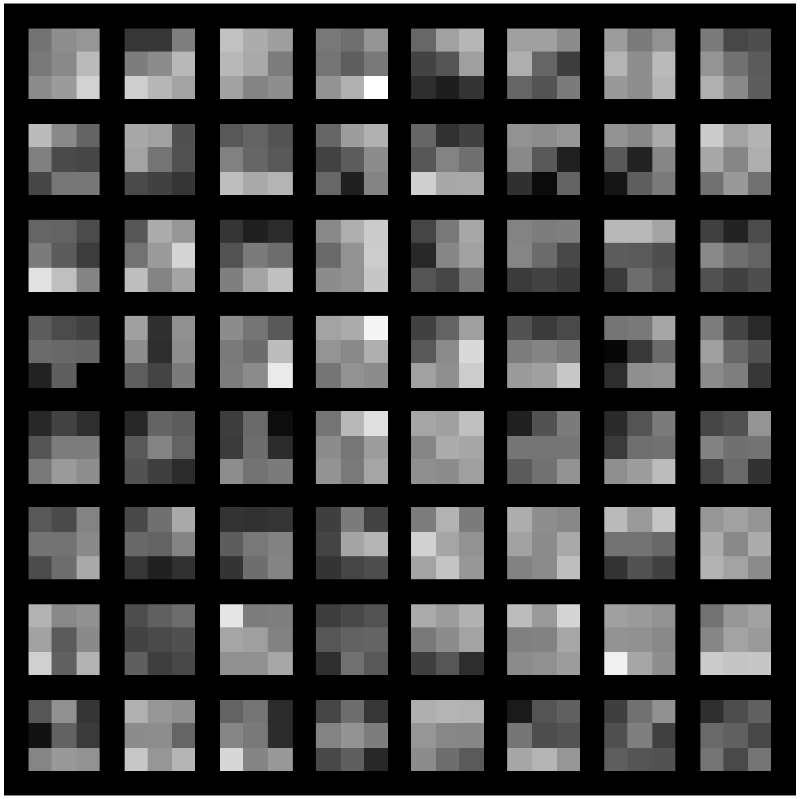
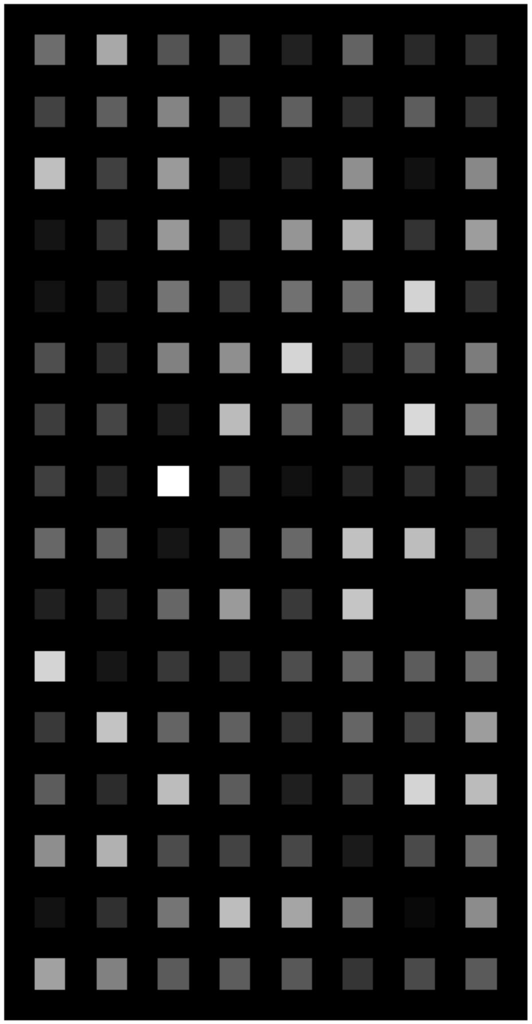
```
              precision    recall  f1-score   support

           0      0.000     0.000     0.000       980
           1      0.539     0.997     0.700      1135
           2      0.798     0.990     0.884      1032
           3      0.892     0.988     0.938      1010
           4      0.000     0.000     0.000       982
           5      0.940     0.992     0.966       892
           6      0.566     0.994     0.721       958
           7      0.000     0.000     0.000      1028
           8      0.825     0.991     0.900       974
           9      0.586     0.991     0.736      1008

    accuracy                          0.695      9999
   macro avg      0.515     0.694     0.584      9999
weighted avg      0.511     0.695     0.582      9999
```

- These results are for the architecture with 1 Conv Block and 3 MobileNet Blocks
- Here, we can see that the precision, recall and f1 score for class 0, 4, 7 is 0. This is because the True Positive is 0 for these classes. The training set contains 6 – 7% of these classes as compared to others. So, the network was unable to identify them and differentiate them.

## 4. Figures along with labels for correct predictions and wrong ones.

| **Architecture Used = 1 Conv Block + 3 MobileNet Blocks, hyperparameters as given in 1.** | |
|---|---|
| |  |
| |  |

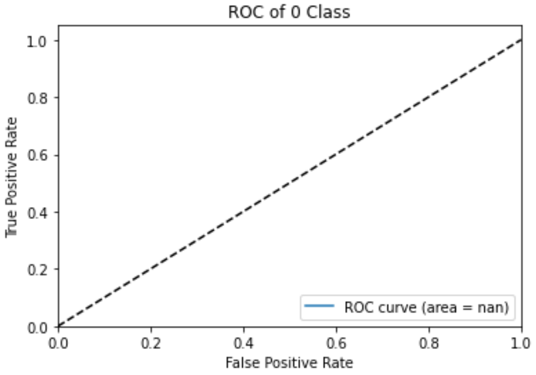
## 5. Plot learned filters of your last convolution layer using matplotlib.

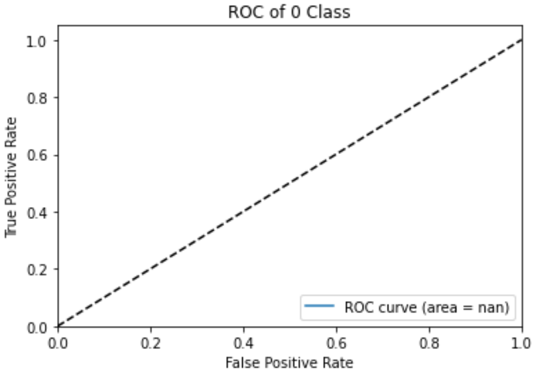
| **Architecture Used = 1 Conv Block + 3 MobileNet Blocks, hyperparameters as given in 1.**<br>**Last Block is MobileNet 3$^{rd}$ Layer:** | |
|---|---|
| Depth wise layer learned 64 Filters<br>torch.Size([64, 1, 3, 3]) | Point wise learned 128 filters<br>torch.Size([128, 64, 1, 1]) |
|  |  |

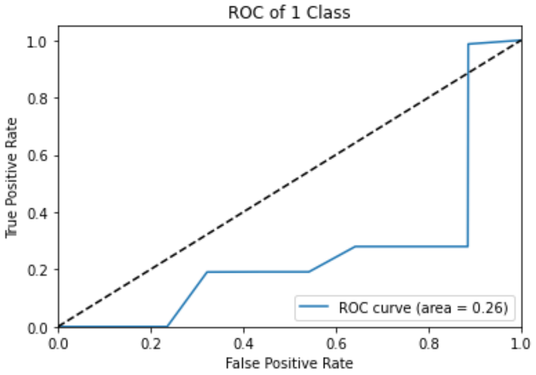
6. **Show what happens when we do not use MobileNet block and when we use MobileNet block in your architecture. Show ROC curves, accuracy/loss curve, confusion matrix, and tsne plot.**

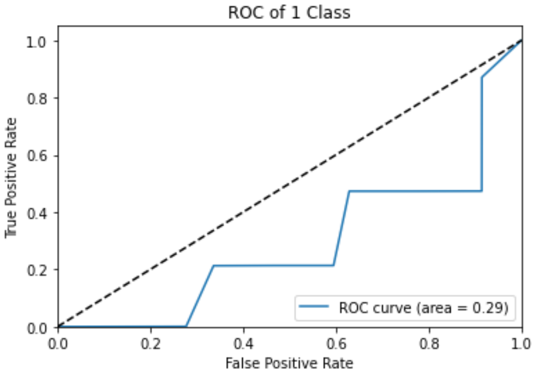
**Architecture Used = 1 Conv Block + 3 MobileNet Blocks, hyperparameters as given in 1.**

a. **ROC curves**

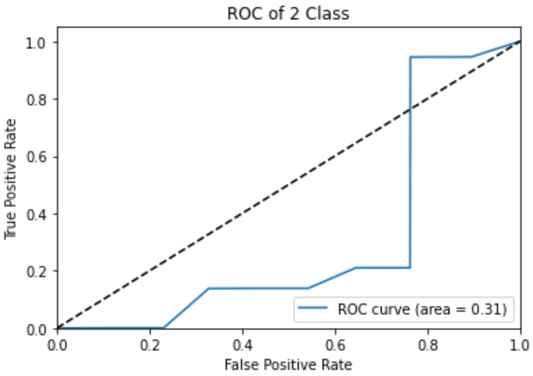
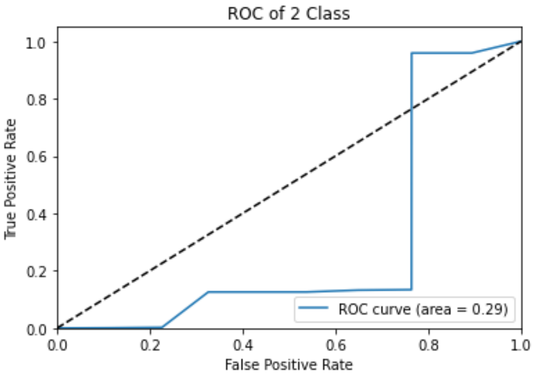
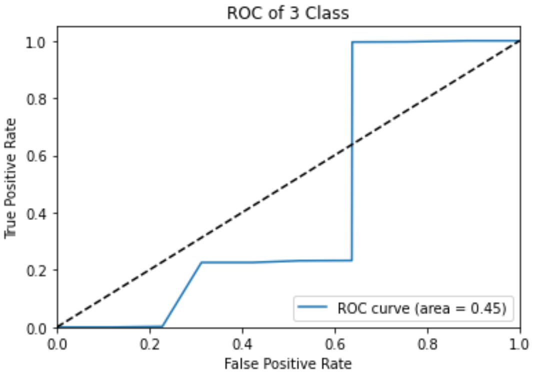
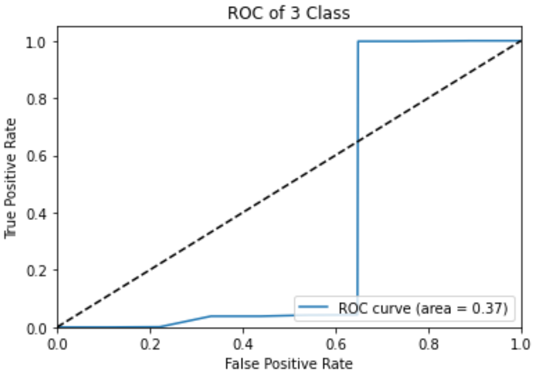
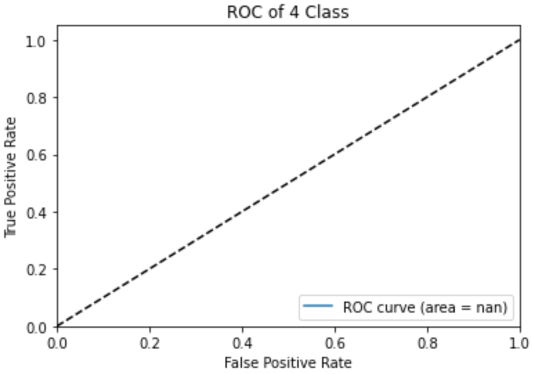
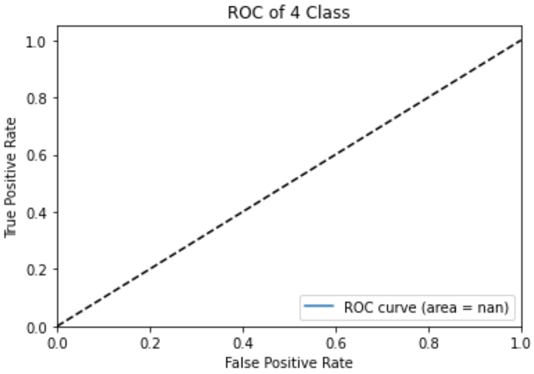
<table>
<tr><th>Classes</th><th>With MobileNet</th><th>Without MobileNet</th></tr>
<tr><td></td><td>


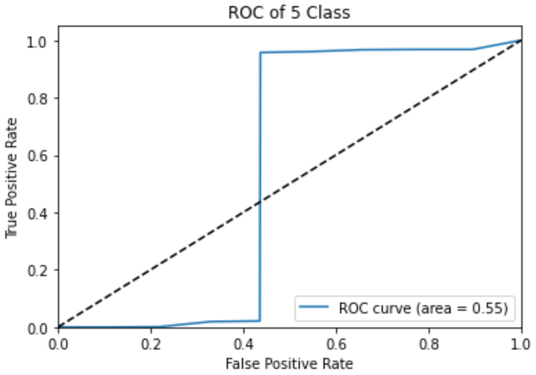
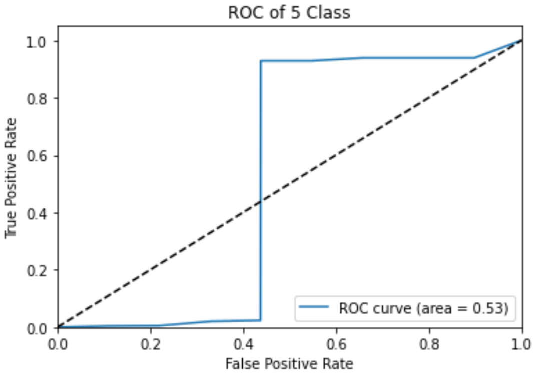
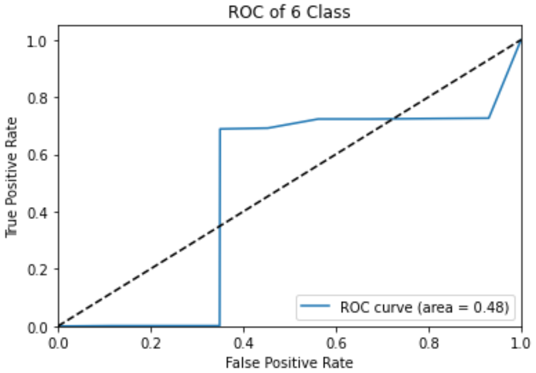
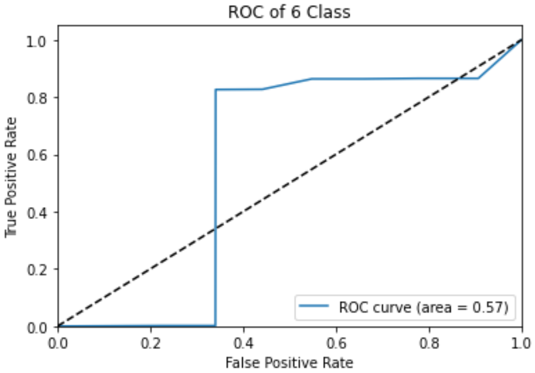
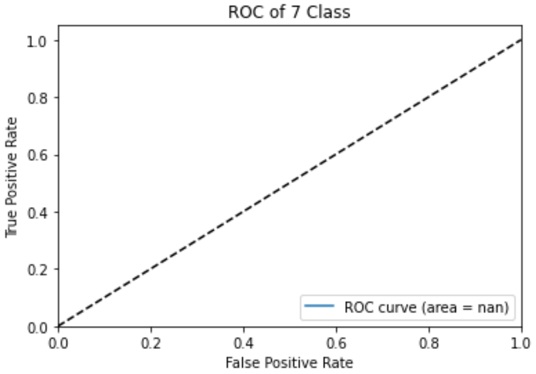
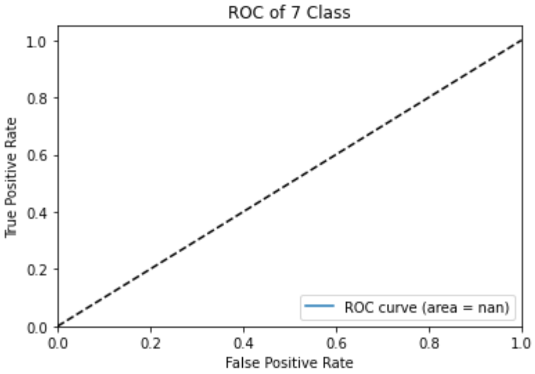
</td><td>


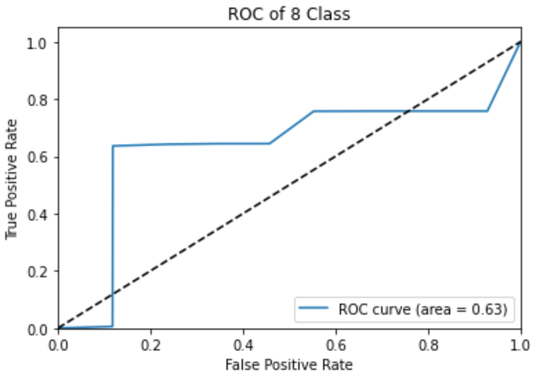
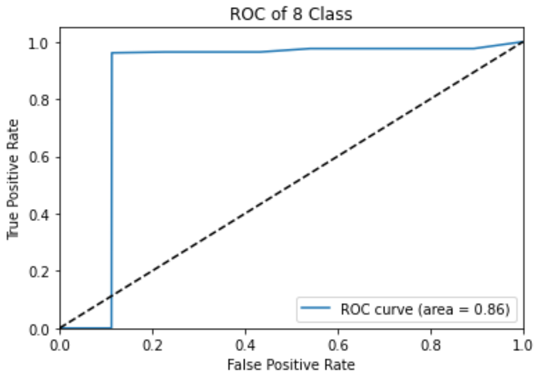
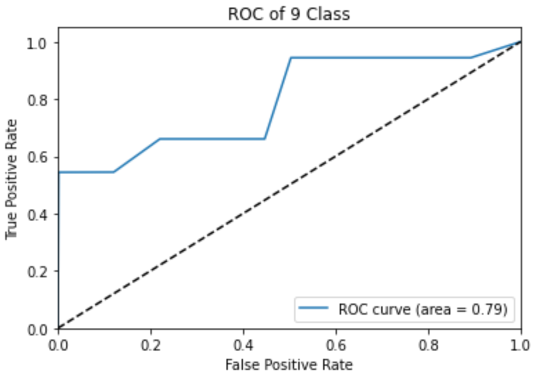
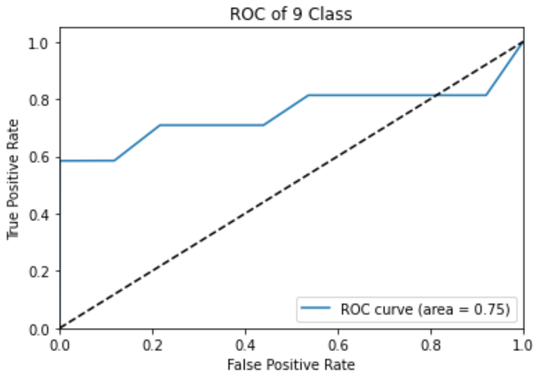
</td></tr>
<tr><td></td><td>


</td><td>


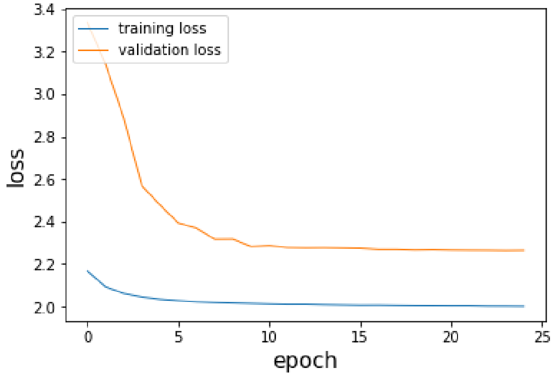
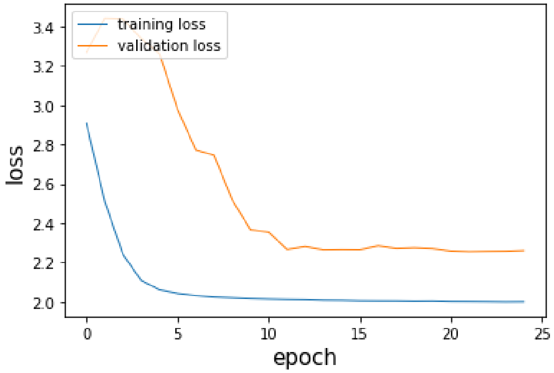
</td></tr>
</table>

| | | |
|---|---|---|
| | ROC of 2 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = 0.31) | ROC of 2 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = 0.29) |
| | ROC of 3 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = 0.45) | ROC of 3 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = 0.37) |
| | ROC of 4 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = nan) | ROC of 4 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = nan) |

| | | |
|---|---|---|
| | ROC of 5 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = 0.55) | ROC of 5 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = 0.53) |
| | ROC of 6 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = 0.48) | ROC of 6 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = 0.57) |
| | ROC of 7 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = nan) | ROC of 7 Class<br>True Positive Rate<br>False Positive Rate<br>0.0<br>0.2<br>0.4<br>0.6<br>0.8<br>1.0<br>ROC curve (area = nan) |

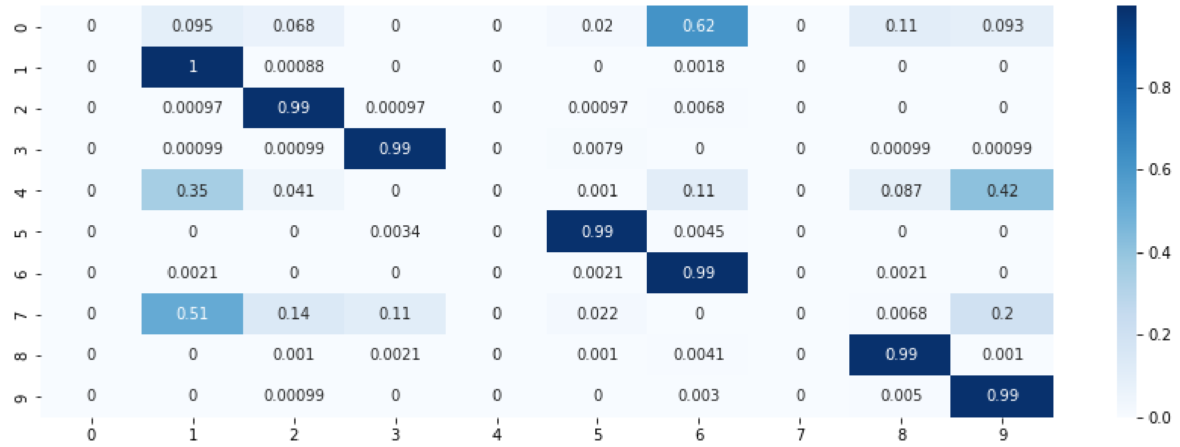
|  |  |  |
|---|---|---|
|  |  |  |
|  |  |  |

## b. Loss curve

| With MobileNet | Without MobileNet |
|---|---|
|  |  |

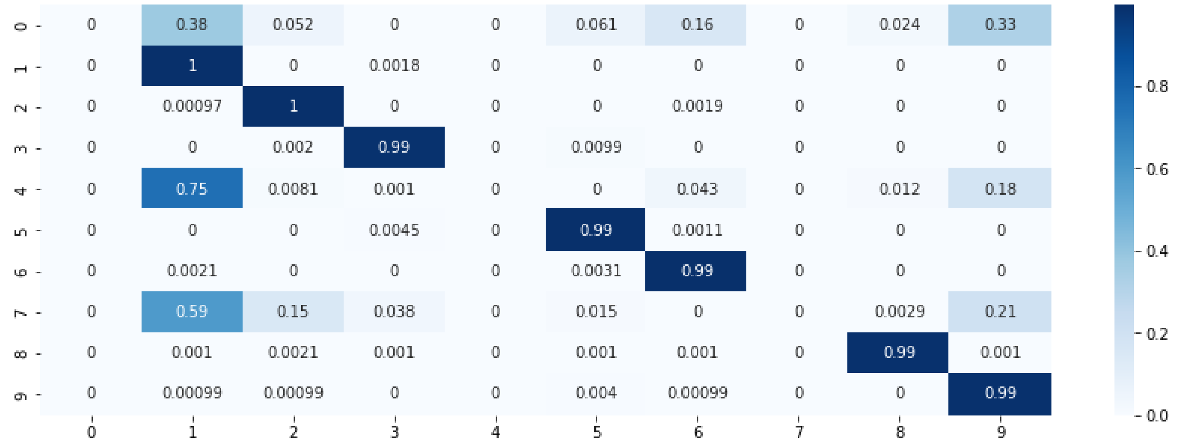
## c. Confusion matrix

**With MobileNet**



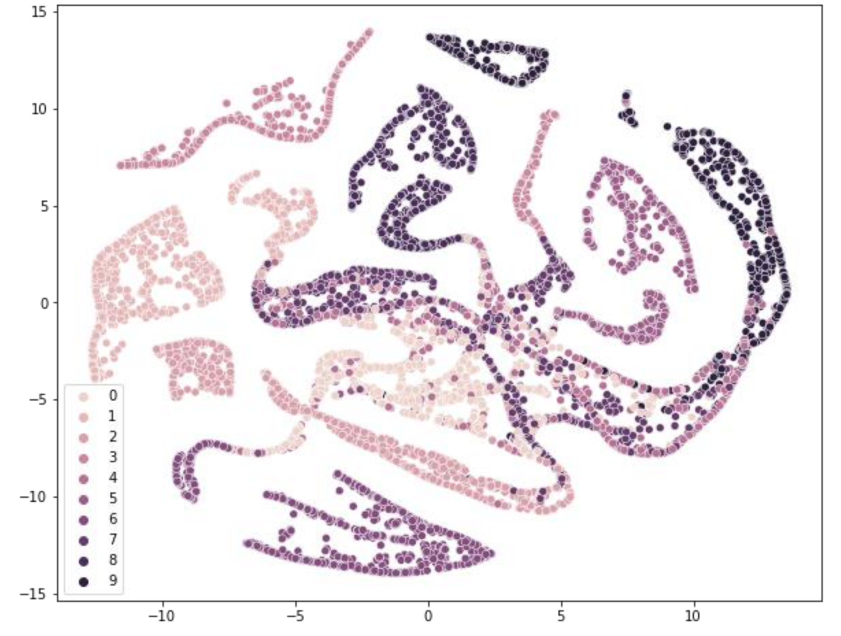
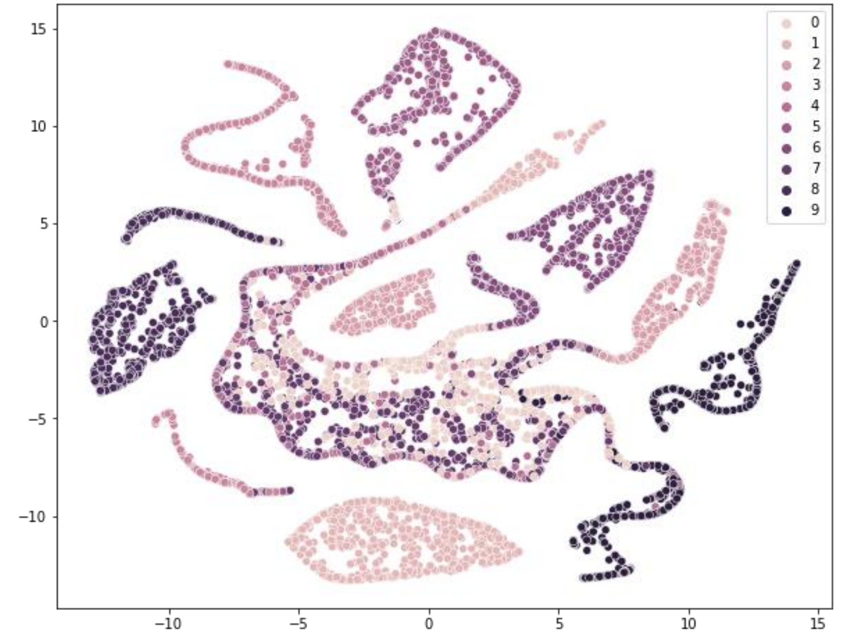
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0.095 | 0.068 | 0 | 0 | 0.02 | 0.62 | 0 | 0.11 | 0.093 |
| 1 | 0 | 1 | 0.00088 | 0 | 0 | 0 | 0.0018 | 0 | 0 | 0 |
| 2 | 0 | 0.00097 | 0.99 | 0.00097 | 0 | 0.00097 | 0.0068 | 0 | 0 | 0 |
| 3 | 0 | 0.00099 | 0.00099 | 0.99 | 0 | 0.0079 | 0 | 0 | 0.00099 | 0.00099 |
| 4 | 0 | 0.35 | 0.041 | 0 | 0 | 0.001 | 0.11 | 0 | 0.087 | 0.42 |
| 5 | 0 | 0 | 0 | 0.0034 | 0 | 0.99 | 0.0045 | 0 | 0 | 0 |
| 6 | 0 | 0.0021 | 0 | 0 | 0 | 0.0021 | 0.99 | 0 | 0.0021 | 0 |
| 7 | 0 | 0.51 | 0.14 | 0.11 | 0 | 0.022 | 0 | 0 | 0.0068 | 0.2 |
| 8 | 0 | 0 | 0.001 | 0.0021 | 0 | 0.001 | 0.0041 | 0 | 0.99 | 0.001 |
| 9 | 0 | 0 | 0.00099 | 0 | 0 | 0 | 0.003 | 0 | 0.005 | 0.99 |

**Without MobileNet**



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0.38 | 0.052 | 0 | 0 | 0.061 | 0.16 | 0 | 0.024 | 0.33 |
| 1 | 0 | 1 | 0 | 0.0018 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0.00097 | 1 | 0 | 0 | 0 | 0.0019 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0.002 | 0.99 | 0 | 0.0099 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0.75 | 0.0081 | 0.001 | 0 | 0 | 0.043 | 0 | 0.012 | 0.18 |
| 5 | 0 | 0 | 0 | 0.0045 | 0 | 0.99 | 0.0011 | 0 | 0 | 0 |
| 6 | 0 | 0.0021 | 0 | 0 | 0 | 0.0031 | 0.99 | 0 | 0 | 0 |
| 7 | 0 | 0.59 | 0.15 | 0.038 | 0 | 0.015 | 0 | 0 | 0.0029 | 0.21 |
| 8 | 0 | 0.001 | 0.0021 | 0.001 | 0 | 0.001 | 0.001 | 0 | 0.99 | 0.001 |
| 9 | 0 | 0.00099 | 0.00099 | 0 | 0 | 0.004 | 0.00099 | 0 | 0 | 0.99 |

d. **tsne plot**

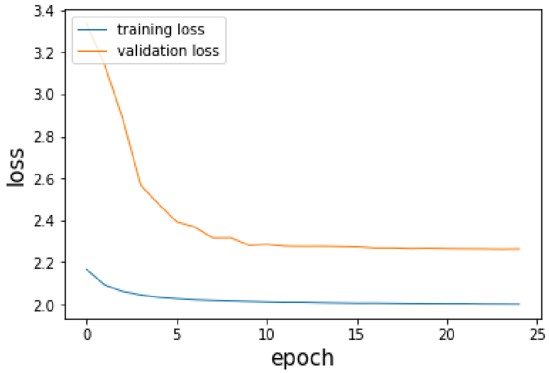
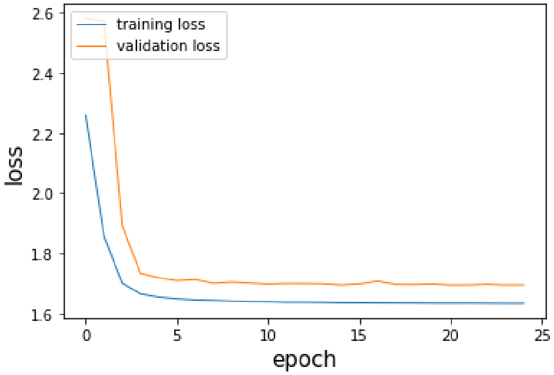
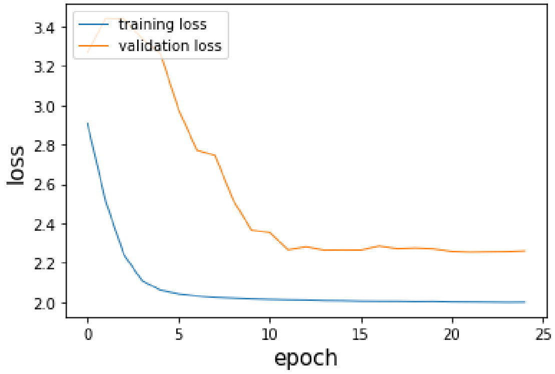
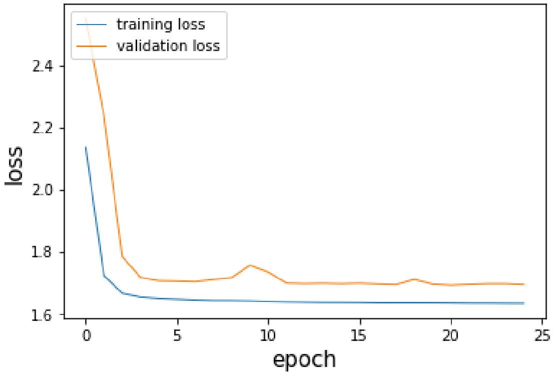
| **With MobileNet** | **Without MobileNet** |
|---|---|
|  |  |
| • For both architecture the TSNE plot shows that the classes are not well separated. | |

7. **Report what you learned from this assignment, your analysis, and if you find something innovative or interesting in the conclusion section.**

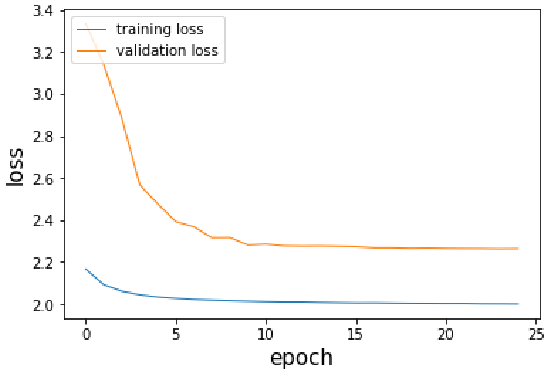
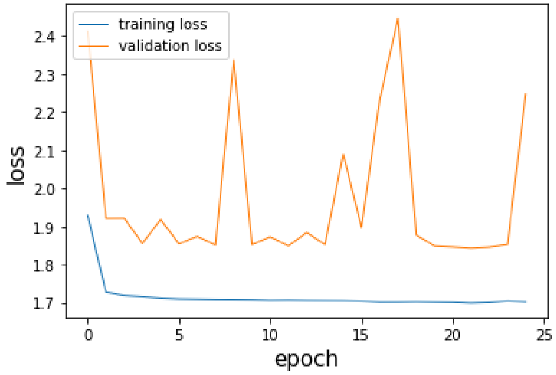
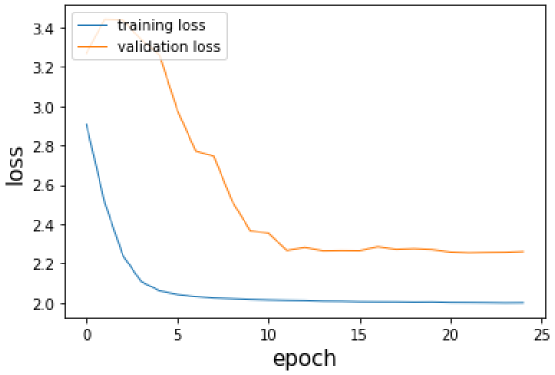
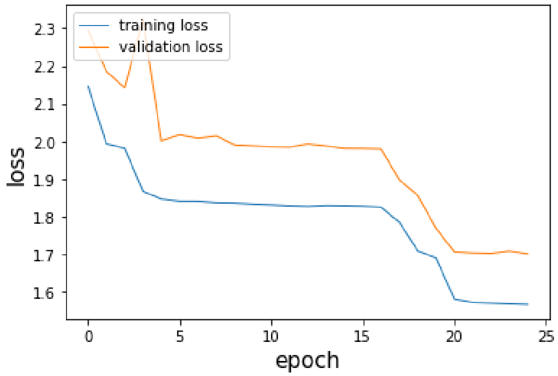
The important point is that the distribution of data classes in dataset results in biased training due to which the accuracy suffers. Here, for the training data we can do Data Augmentation in which we can duplicate some of the samples as it and some with a bit of image transformations like rotation and crop. This would make our training data more balanced and hence improve the accuracy.

## 8. Report the accuracy by changing these hyperparameters.

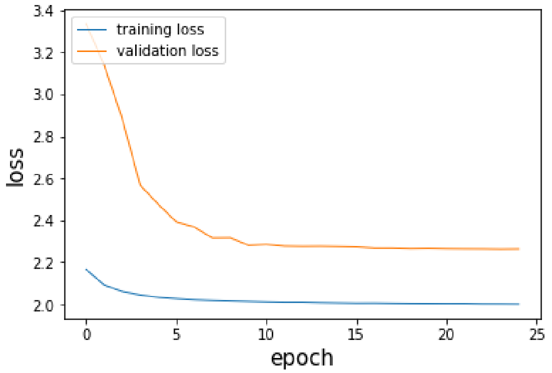
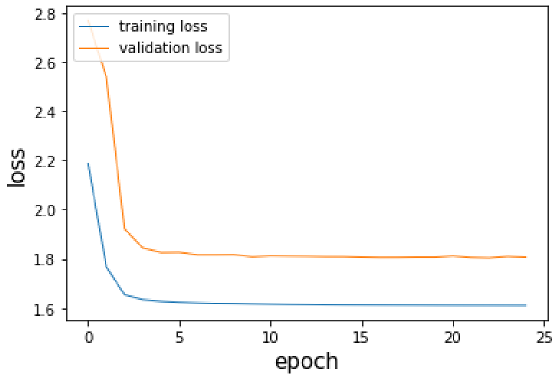
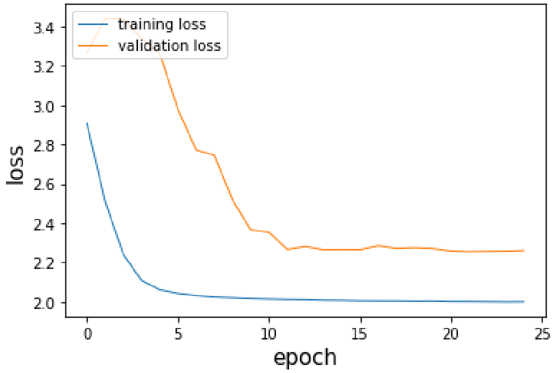
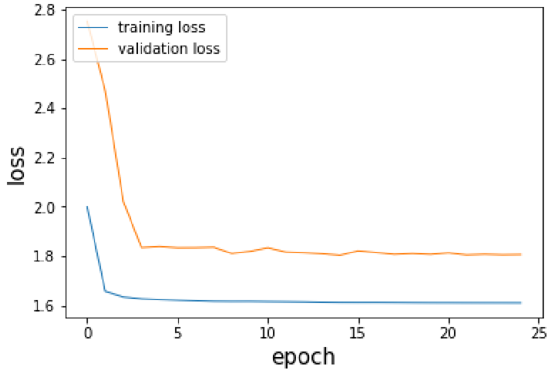
### a. Batch Size

| Structure | TRAIN_BS = 1500 | TRAIN_BS = 3000 |
|---|---|---|
| With MobileNet (C1+M3) |  <br>test dataset accuracy is 69.29693603515625 |  <br>test dataset accuracy is 69.45694732666016 |
| Without MobileNet (C4) |  <br>test dataset accuracy is 69.44694519042969 |  <br>test dataset accuracy is 69.58695983886719 |

### b. Learning Rate

| Structure | Lr = 0.01 | Lr = 0.1 |
|---|---|---|
| With MobileNet (C1+M3) |  <br>test dataset accuracy is 69.29693603515625 |  <br>test dataset accuracy is 34.05340576171875 |
| Without MobileNet (C4) |  <br>test dataset accuracy is 69.44694519042969 |  <br>test dataset accuracy is 69.08690643310547 |
| • The architecture with MobileNet block could not converge due to high learning rate, this resulted in low accuracy. | | |

**c.** Ratio of training and testing data

| Structure | Ratio = Training/Testing = 3.4 | Ratio = Training/Testing = 2.0 |
|---|---|---|
| With MobileNet (C1+M3) | <br> <br>test dataset accuracy is 69.29693603515625 | <br> <br>test dataset accuracy is 69.39693450927734 |
| Without MobileNet (C4) | <br> <br>test dataset accuracy is 69.44694519042969 | <br> <br>test dataset accuracy is 69.62696075439453 |
| • There is a slightly increase in the accuracy by decrease the Training/Testing Ratio this is because the classes with lesser example have higher occurance as compared to others. | | |