DEEP LEARNING MINOR 1 REPORT

Google colab link:
https://colab.research.google.com/drive/1cIk5hGZFiWStVR8nkGVx86ntUstNl3FJ?usp=sharing

QUESTION 1:

Training Loss at the end of epoch 1 is 1.062

Training Loss at the end of epoch 2 is 0.839

Training Loss at the end of epoch 3 is 0.707

Training Loss at the end of epoch 4 is 0.613

Training Loss at the end of epoch 5 is 0.538

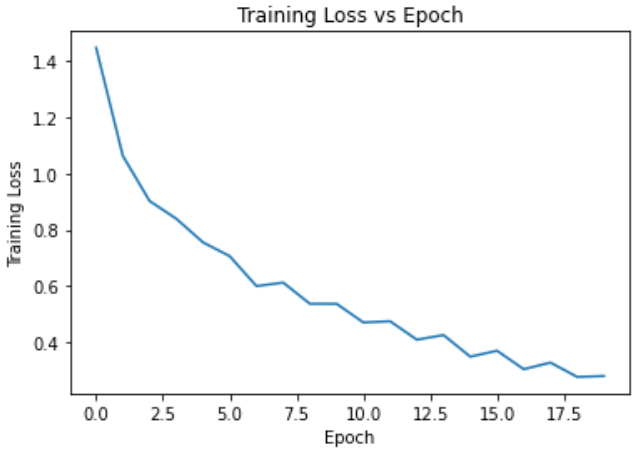
Training Loss at the end of epoch 6 is 0.476

Training Loss at the end of epoch 7 is 0.427

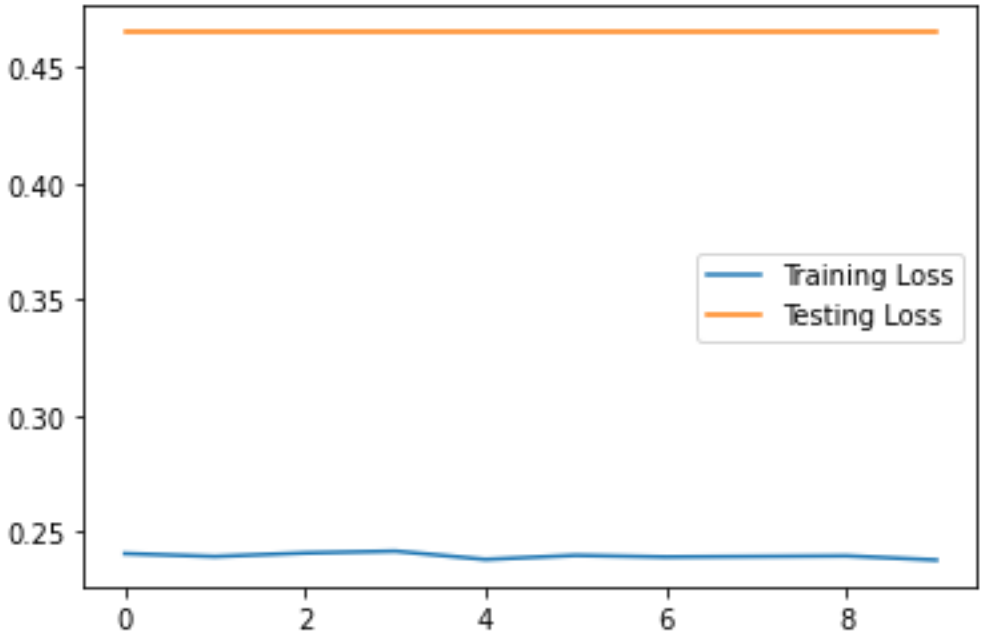
Training Loss at the end of epoch 8 is 0.372

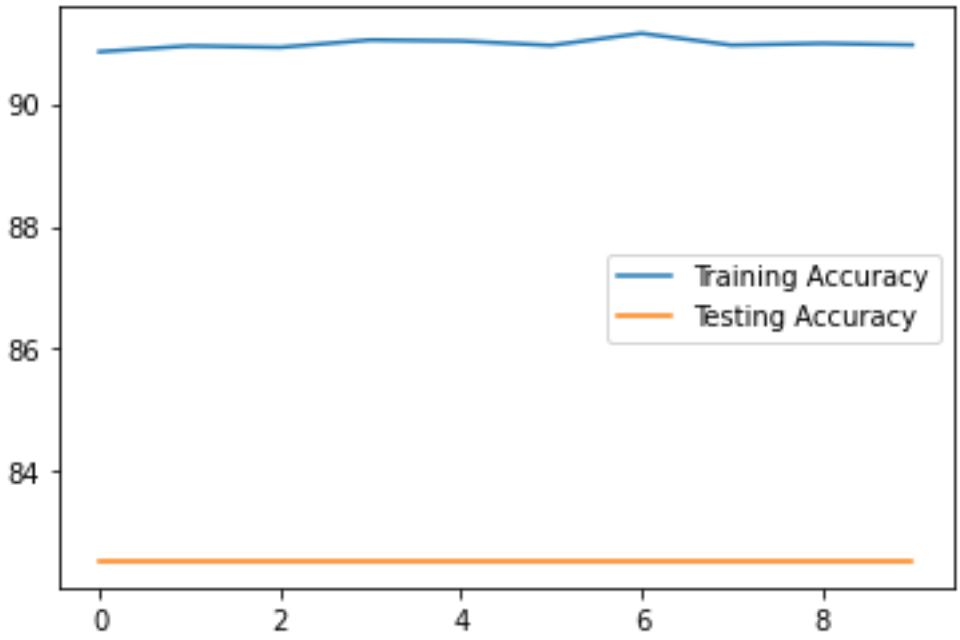
Training Loss at the end of epoch 9 is 0.330

Training Loss at the end of epoch 10 is 0.282



Accuracy of network on test images = 82%





QUESTION 2:

GOOGLE COLAB:
https://colab.research.google.com/drive/1Rl70rFDSngS6AauSFYbBZjtnGeYMoX2R?usp=sharing

Hyperparameters chosen were

1)number of epochs is taken ten since it is given

2)learning rate is used, which can be adjusted to increase the speed of the training and also used to increase the performance of the model.

```
        # Combine the losses
        loss = loss_recon + classification_loss_weight * loss_class
```

3) The classification_loss_weight (multiplier to loss_class) is chosen to be 0.1, which means that the classification loss is ten times more important than the reconstruction loss.

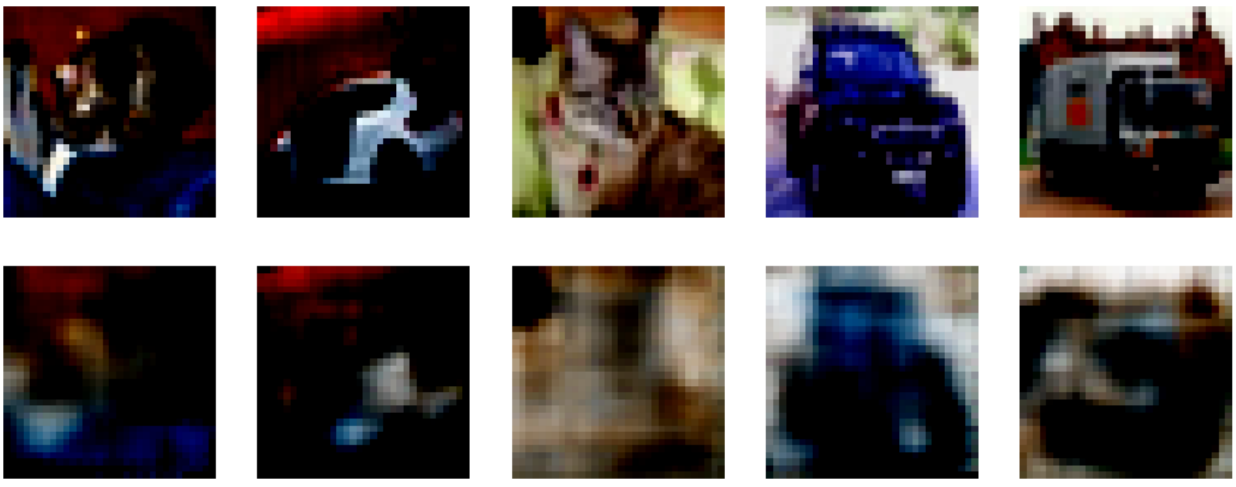
By manipulating this value, we can give importance to either classification or reconstruction of an image depending on need. when the classification task is more important than the reconstruction, the classification loss weight can be increased to give more importance to the classification task.

When the reconstruction is more important than the classification, we will provide less value to the classification_loss_weight. This will give more importance to the reconstruction than the classification.

When you want to give equal importance to both classification and reconstruction, put classification_loss_weight =1.

LOSSES AFTER EVERY EPOCH:

	Reconstruction loss	Classification loss	Total loss
Epoch 1	0.115	1.709	0.286
Epoch 2	0.093	1.672	0.260
Epoch 3	0.080	1.653	0.245
Epoch 4	0.074	1.635	0.237
Epoch 5	0.069	1.593	0.228
Epoch 6	0.0612	1.623	0.223
Epoch 7	0.061	1.662	0.227
Epoch 8	0.058	1.638	0.222
Epoch 9	0.056	1.631	0.220
Epoch 10	0.059	1.614	0.220



1st row: original images

2nd row: predicted images

Results on validation set:

Reconstruction loss	0.0445
Classification loss	0.0146
Reconstruction accuracy	15.74%
Classification accuracy	0.59%

COMPARISON:

The accuracy of the CNN model (82%) is more than the Accuracy of the autoencoder (15.74). This is because we are dealing with a classification problem.

Another reason why CNN performance is better is due to image classification. CNN is specifically designed for image recognition tasks.

Another reason why CNN is more accurate than the autoencoder because the number of layers used in the autoencoder is less(3) than the CNN (6 layers).

Autoencoder performs well in image denoising or compression but not suitable for image classification.