Assignment 5

Initial parameters in the CNN Architecture were :
Each image size - 200*200
Convolution network - 1
Max Pool - 1
Fully Connected Network- 1
Loss function - CrossEntropyLoss()
Optimizer - SGD (Stochastic Gradient Descent)
Learning rate - 0.01
No of Epochs - 15

Different Approaches :

- Now, as the image size was very large, it was take more time to complete each epochs, so I did the resize of image from 200*200 to 32*32.
-
- As the learning rate was high it was overshooting the actual parameters, so I reduced the learning rate to 0.001
-
- By using 1 convolutional network, 1 max pooling and 1 fully connected network - the accuracy of the network was not good. So I created a model using 2 convolutional networks, 2 max pooling and 2 fully connected networks. This made the model complex but it increased the accuracy.
-
- Adam's optimizer was working more efficiently compared to SGD. So this also increased the accuracy.

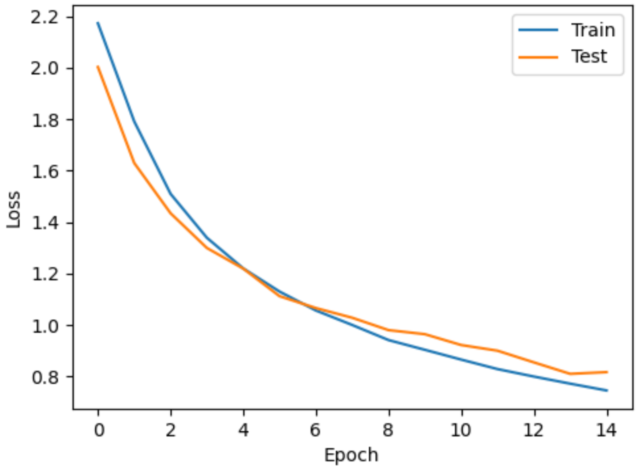
After doing all these trial and error methods, **the finalised parameters in the CNN Architecture were** :
Reduced image size - 32*32
Convolution network - 2
Max Pool - 2
Fully Connected Network- 2
Loss function - CrossEntropyLoss()
Optimizer - Adam's optimizer
Learning rate - 0.001
No of Epochs - 15

Design of the NN :

- Extracted all the data from the zip file and sorted the files according to the file name
- Converted the images and labels into tensor objects using dataloader and dataset
- Defined a CNN architecture using convolution networks, max pooling and fully connected networks

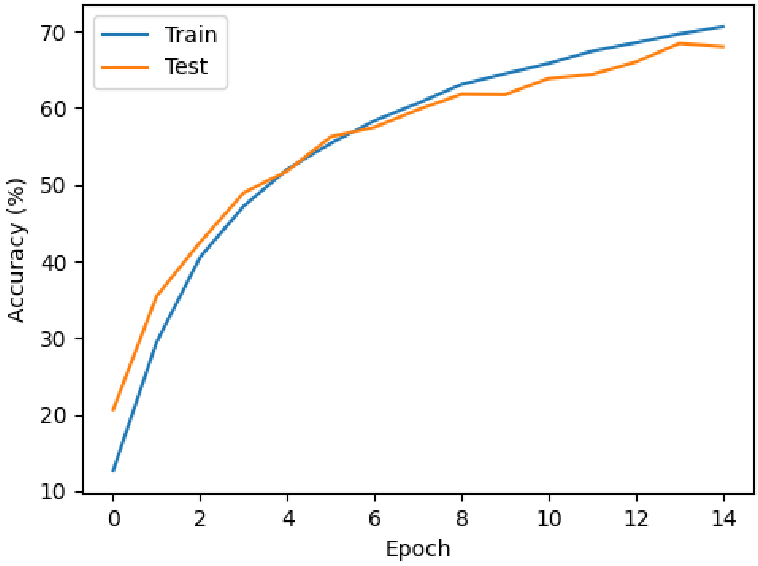
- Predicted the output for each element of test and train set, and calculated the loss and accuracy after each epoch
- Plotted the loss and accuracy vs epochs graphs on both test and train set
- Saved the train model in 0502-668575956Sheth.ZZZ file

A graph that shows epochs vs loss on training set, and on the same graph epoch vs loss on the Test set.



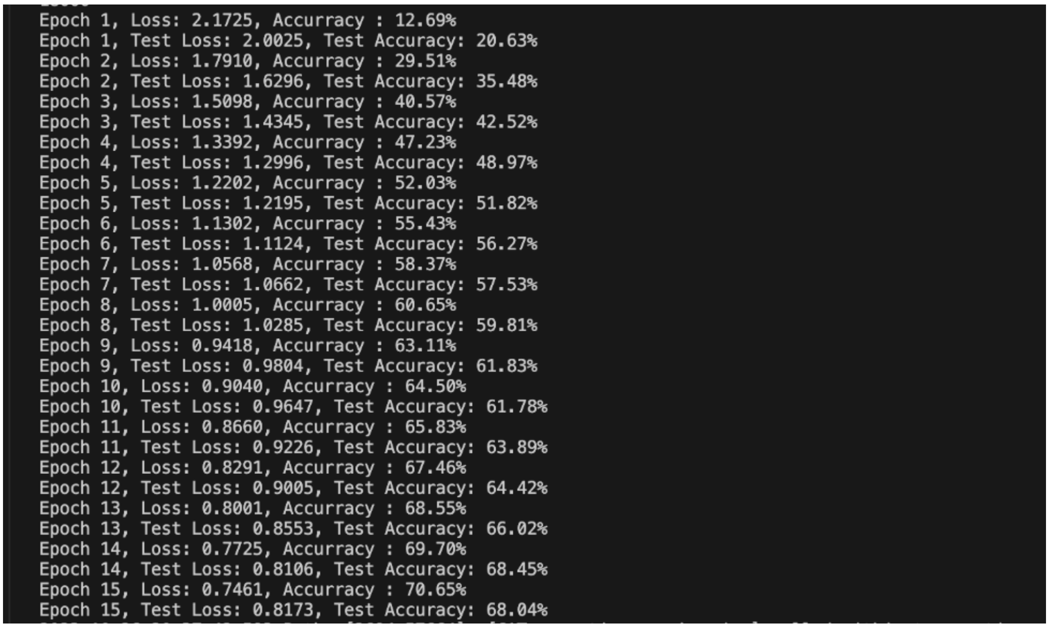
As we can see from the above graph that Loss reduces for both Test set and Training set as the number of epochs increases. But on epoch 14, there is an increase of loss in Test set compared to epoch 13 due to overfitting of Training set

A graph that shows epochs vs accuracy on training set, and on the same graph epoch vs accuracy on the Test set.



As we can see from the above graph, Accuracy increases for both Test set and Training set as the number of epochs increases. But on epoch 14, there is a decrease of accuracy in Test set compared to epoch 13 due to overfitting of Training set.

Output obtained on console after training the CNN in 0501-668575956Sheth.py



```
Epoch 1, Loss: 2.1725, Accurracy : 12.69%
Epoch 1, Test Loss: 2.0025, Test Accuracy: 20.63%
Epoch 2, Loss: 1.7910, Accurracy : 29.51%
Epoch 2, Test Loss: 1.6296, Test Accuracy: 35.48%
Epoch 3, Loss: 1.5098, Accurracy : 40.57%
Epoch 3, Test Loss: 1.4345, Test Accuracy: 42.52%
Epoch 4, Loss: 1.3392, Accurracy : 47.23%
Epoch 4, Test Loss: 1.2996, Test Accuracy: 48.97%
Epoch 5, Loss: 1.2202, Accurracy : 52.03%
Epoch 5, Test Loss: 1.2195, Test Accuracy: 51.82%
Epoch 6, Loss: 1.1302, Accurracy : 55.43%
Epoch 6, Test Loss: 1.1124, Test Accuracy: 56.27%
Epoch 7, Loss: 1.0568, Accurracy : 58.37%
Epoch 7, Test Loss: 1.0662, Test Accuracy: 57.53%
Epoch 8, Loss: 1.0005, Accurracy : 60.65%
Epoch 8, Test Loss: 1.0285, Test Accuracy: 59.81%
Epoch 9, Loss: 0.9418, Accurracy : 63.11%
Epoch 9, Test Loss: 0.9804, Test Accuracy: 61.83%
Epoch 10, Loss: 0.9040, Accurracy : 64.50%
Epoch 10, Test Loss: 0.9647, Test Accuracy: 61.78%
Epoch 11, Loss: 0.8660, Accurracy : 65.83%
Epoch 11, Test Loss: 0.9226, Test Accuracy: 63.89%
Epoch 12, Loss: 0.8291, Accurracy : 67.46%
Epoch 12, Test Loss: 0.9005, Test Accuracy: 64.42%
Epoch 13, Loss: 0.8001, Accurracy : 68.55%
Epoch 13, Test Loss: 0.8553, Test Accuracy: 66.02%
Epoch 14, Loss: 0.7725, Accurracy : 69.70%
Epoch 14, Test Loss: 0.8106, Test Accuracy: 68.45%
Epoch 15, Loss: 0.7461, Accurracy : 70.65%
Epoch 15, Test Loss: 0.8173, Test Accuracy: 68.04%
```