Deep Learning For Visual Analytics

Aksheit Saxena

Assignment-2

Problem Statement

A. *Self-Supervised Pre-Training & Classification*
B. *Vision Transformer Implementation*

Hardware & Environment Details

1. Intel I9 processor
2. RTX 4080 GPU -12 GB (Self sourced)
3. Anaconda environment with VS Code

Self-Supervised Pre-Training & Classification Algorithm Details

1. Take the CIFAR-10 dataset, each class has 5000 samples and there are 10 classes.
2. Split the dataset in 2 parts (A) 40000 and (B) 10000 each with equal number of samples per class in each split.
3. Discard the labels of the samples in the first set.
4. Take a resnet-18 (initialised) and strip the ImageNet classification layer with a 4 way classification layer.
5. Train this network on the self-training task of classifying the rotation of the image.
6. Once this self-supervised pretraining is done, strip the classification layer and add a classification layer for CIFAR-10 classification this is finetuned on the set B for the task of image classification.
7. Log the loss (cross entropy) and accuracies for both the pre-training task and classification task.

Vision Transformer Algorithm Details

1. Convert the image into patches.
2. Vectorize the patches d1 X d2 X d3 --> d1d2d3 X 1 (One vector per patch)
3. Apply Dense layer to these vectors. All have same W and same b. Dense layer takes input of positions to create Z_1, Z_2, …, Z_n positional embeddings
4. CLS token input to an embedding layer to create Z_0 vector (same shape as other z's).
5. Output of transformer here is used for classification.
6. Z_0, ..., Z_n are inputs to multiheaded self-attention (n+1 vectors output)
7. Apply a dense layer
8. Add as many as multi-headed self-attention plus dense layers as u want (jointly called transformer encoder network)
9. At the last layer we focus on c_0 vector and feed to a SoftMax classifier. Output (say p) has shape equal to number of classes (10 on our case)
10. During training Loss is CE of p and GT

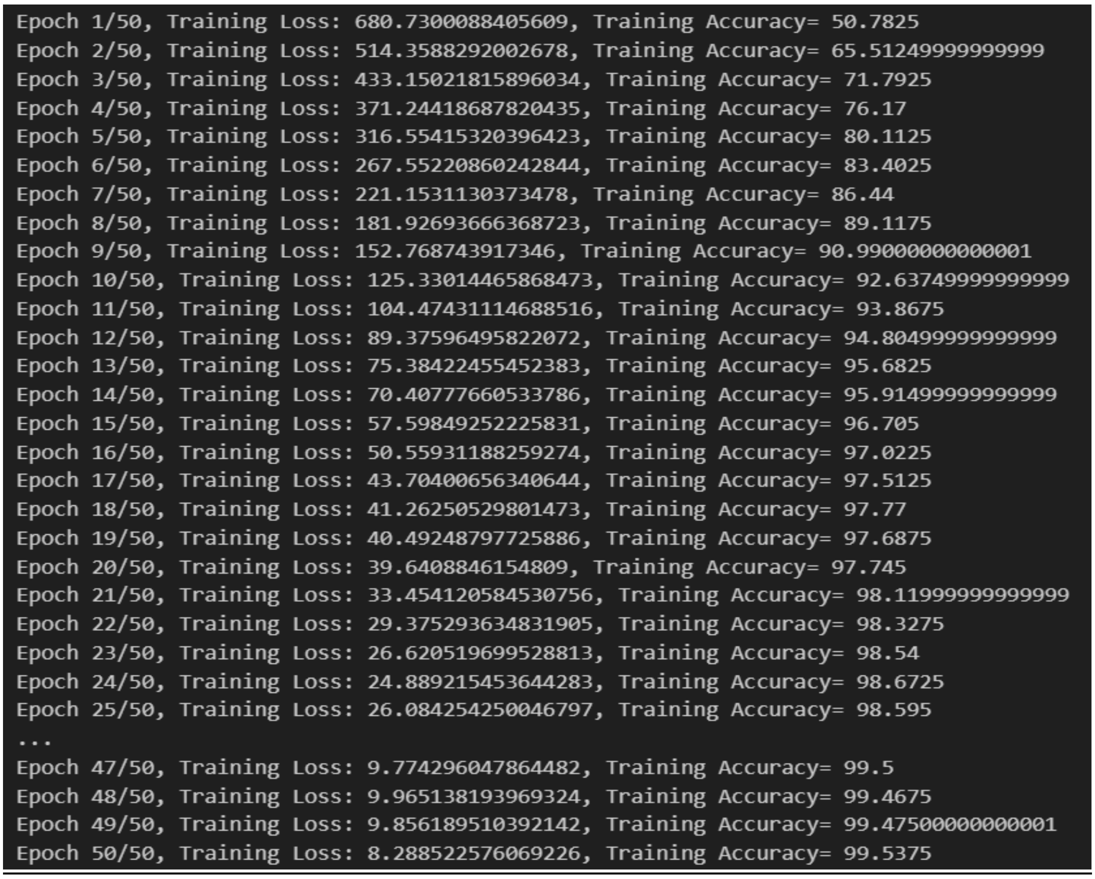
11. Perform loss optimisation and update.

Deviations from Vision Transformer paper

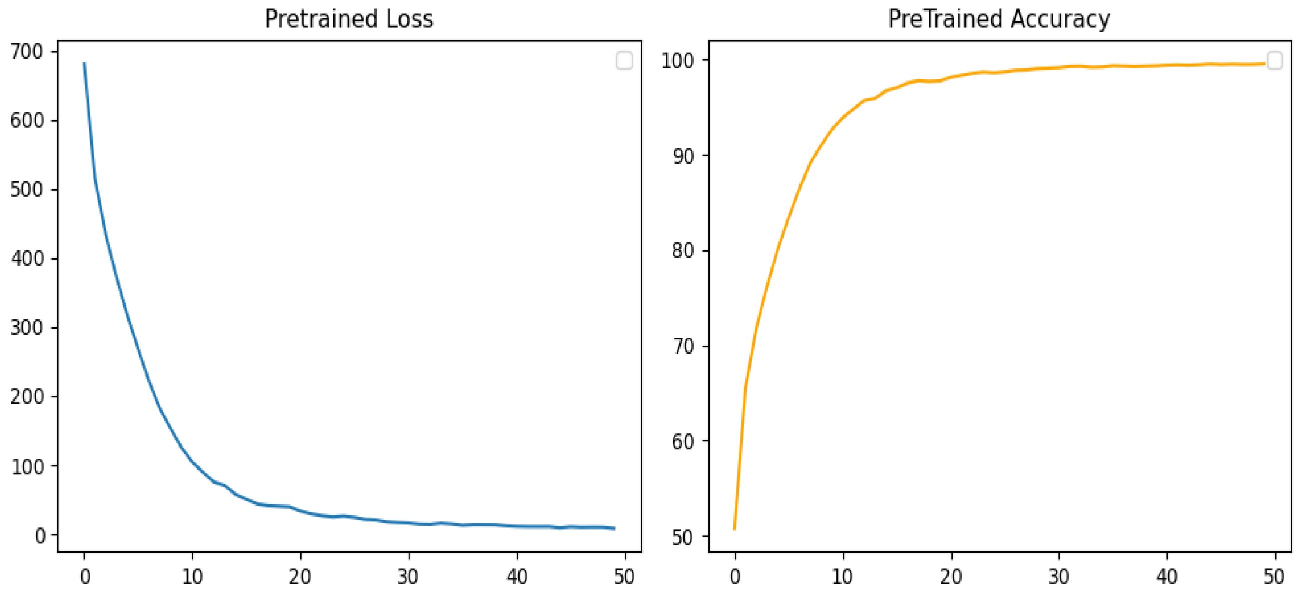
1. The paper mentions using a batch-size of 4096, but due to memory & computation constraints, I am using batch-size of 300 and 64 for train and test respectively.
2. I am not using the learning rate, weight decay and beta values as suggested by the paper. My algorithm is using default values.
3. The number of epochs =25 (standard used for all experiments) is different from the paper.

Self-Supervised Pre-Training & Classification Results

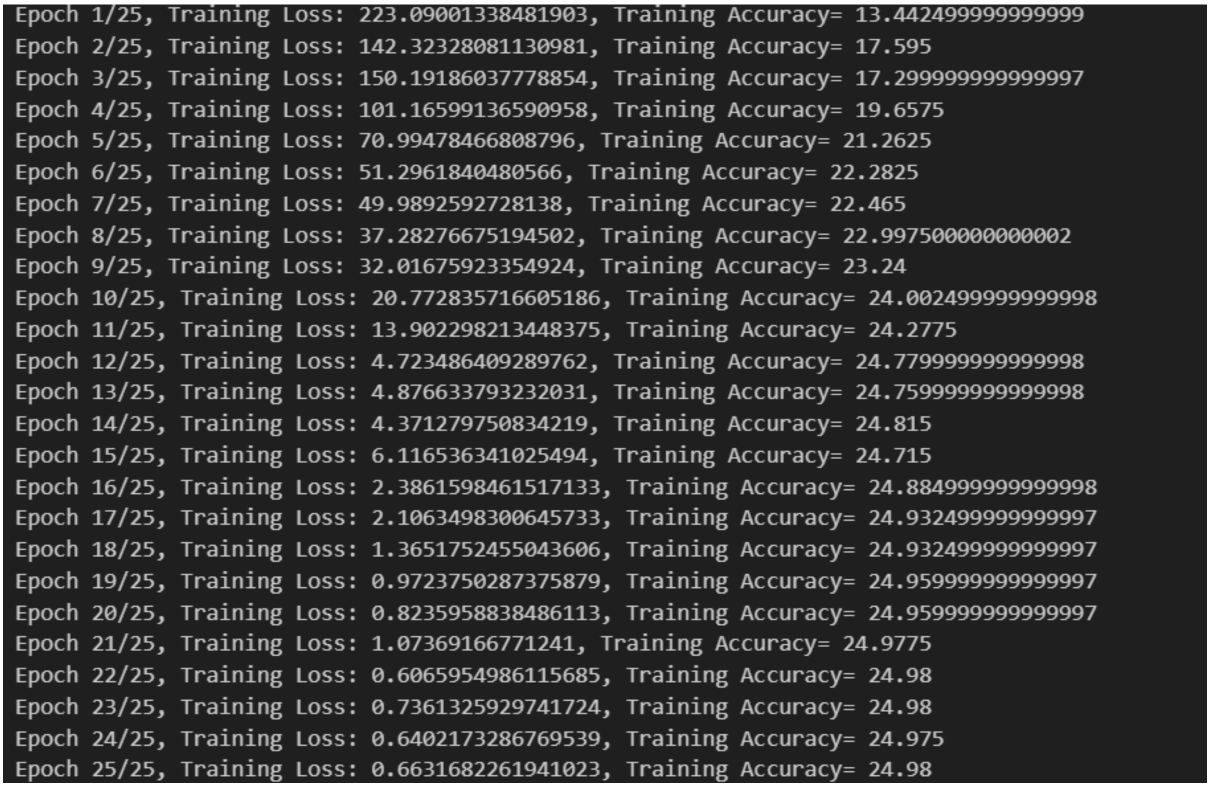
a. Pretraining Rotation Classification Results



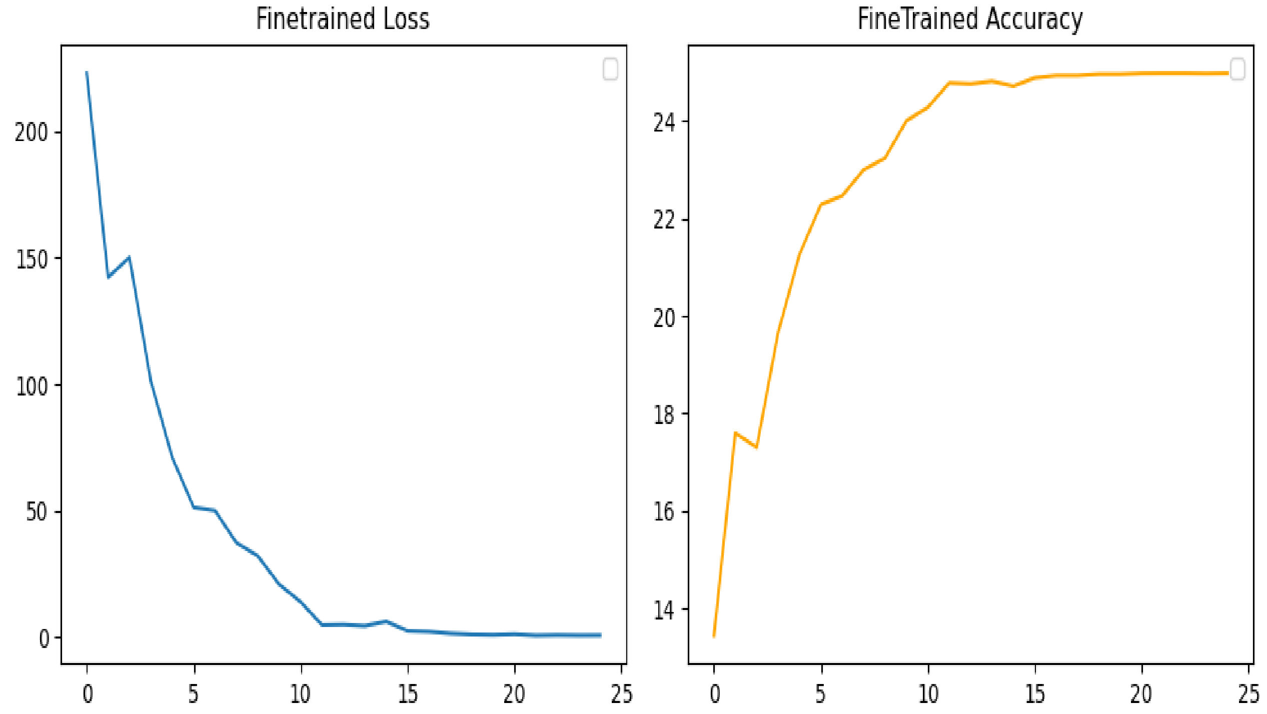
```
Epoch 1/50, Training Loss: 680.7300088405609, Training Accuracy= 50.7825
Epoch 2/50, Training Loss: 514.3588292002678, Training Accuracy= 65.51249999999999
Epoch 3/50, Training Loss: 433.15021815896034, Training Accuracy= 71.7925
Epoch 4/50, Training Loss: 371.24418687820435, Training Accuracy= 76.17
Epoch 5/50, Training Loss: 316.55415320396423, Training Accuracy= 80.1125
Epoch 6/50, Training Loss: 267.55220860242844, Training Accuracy= 83.4025
Epoch 7/50, Training Loss: 221.1531130373478, Training Accuracy= 86.44
Epoch 8/50, Training Loss: 181.92693666368723, Training Accuracy= 89.1175
Epoch 9/50, Training Loss: 152.768743917346, Training Accuracy= 90.99000000000001
Epoch 10/50, Training Loss: 125.33014465868473, Training Accuracy= 92.63749999999999
Epoch 11/50, Training Loss: 104.47431114688516, Training Accuracy= 93.8675
Epoch 12/50, Training Loss: 89.37596495822072, Training Accuracy= 94.80499999999999
Epoch 13/50, Training Loss: 75.38422455452383, Training Accuracy= 95.6825
Epoch 14/50, Training Loss: 70.40777660533786, Training Accuracy= 95.91499999999999
Epoch 15/50, Training Loss: 57.59849252225831, Training Accuracy= 96.705
Epoch 16/50, Training Loss: 50.55931188259274, Training Accuracy= 97.0225
Epoch 17/50, Training Loss: 43.70400656340644, Training Accuracy= 97.5125
Epoch 18/50, Training Loss: 41.26250529801473, Training Accuracy= 97.77
Epoch 19/50, Training Loss: 40.49248797725886, Training Accuracy= 97.6875
Epoch 20/50, Training Loss: 39.6408846154809, Training Accuracy= 97.745
Epoch 21/50, Training Loss: 33.454120584530756, Training Accuracy= 98.11999999999999
Epoch 22/50, Training Loss: 29.375293634831905, Training Accuracy= 98.3275
Epoch 23/50, Training Loss: 26.620519699528813, Training Accuracy= 98.54
Epoch 24/50, Training Loss: 24.889215453644283, Training Accuracy= 98.6725
Epoch 25/50, Training Loss: 26.084254250046797, Training Accuracy= 98.595
...
Epoch 47/50, Training Loss: 9.774296047864482, Training Accuracy= 99.5
Epoch 48/50, Training Loss: 9.965138193969324, Training Accuracy= 99.4675
Epoch 49/50, Training Loss: 9.856189510392142, Training Accuracy= 99.47500000000001
Epoch 50/50, Training Loss: 8.288522576069226, Training Accuracy= 99.5375
```



b. Fine-Tuning CIFAR-10 Classification Results



```
Epoch 1/25, Training Loss: 223.09001338481903, Training Accuracy= 13.442499999999999
Epoch 2/25, Training Loss: 142.32328081130981, Training Accuracy= 17.595
Epoch 3/25, Training Loss: 150.19186037778854, Training Accuracy= 17.299999999999997
Epoch 4/25, Training Loss: 101.16599136590958, Training Accuracy= 19.6575
Epoch 5/25, Training Loss: 70.99478466808796, Training Accuracy= 21.2625
Epoch 6/25, Training Loss: 51.2961840480566, Training Accuracy= 22.2825
Epoch 7/25, Training Loss: 49.9892592728138, Training Accuracy= 22.465
Epoch 8/25, Training Loss: 37.28276675194502, Training Accuracy= 22.997500000000002
Epoch 9/25, Training Loss: 32.01675923354924, Training Accuracy= 23.24
Epoch 10/25, Training Loss: 20.772835716605186, Training Accuracy= 24.002499999999998
Epoch 11/25, Training Loss: 13.902298213448375, Training Accuracy= 24.2775
Epoch 12/25, Training Loss: 4.723486409289762, Training Accuracy= 24.779999999999998
Epoch 13/25, Training Loss: 4.876633793232031, Training Accuracy= 24.759999999999998
Epoch 14/25, Training Loss: 4.371279750834219, Training Accuracy= 24.815
Epoch 15/25, Training Loss: 6.116536341025494, Training Accuracy= 24.715
Epoch 16/25, Training Loss: 2.3861598461517133, Training Accuracy= 24.884999999999998
Epoch 17/25, Training Loss: 2.1063498300645733, Training Accuracy= 24.932499999999997
Epoch 18/25, Training Loss: 1.3651752455043606, Training Accuracy= 24.932499999999997
Epoch 19/25, Training Loss: 0.9723750287375879, Training Accuracy= 24.959999999999997
Epoch 20/25, Training Loss: 0.8235958838486113, Training Accuracy= 24.959999999999997
Epoch 21/25, Training Loss: 1.07369166771241, Training Accuracy= 24.9775
Epoch 22/25, Training Loss: 0.6065954986115685, Training Accuracy= 24.98
Epoch 23/25, Training Loss: 0.7361325929741724, Training Accuracy= 24.98
Epoch 24/25, Training Loss: 0.6402173286769539, Training Accuracy= 24.975
Epoch 25/25, Training Loss: 0.6631682261941023, Training Accuracy= 24.98
```

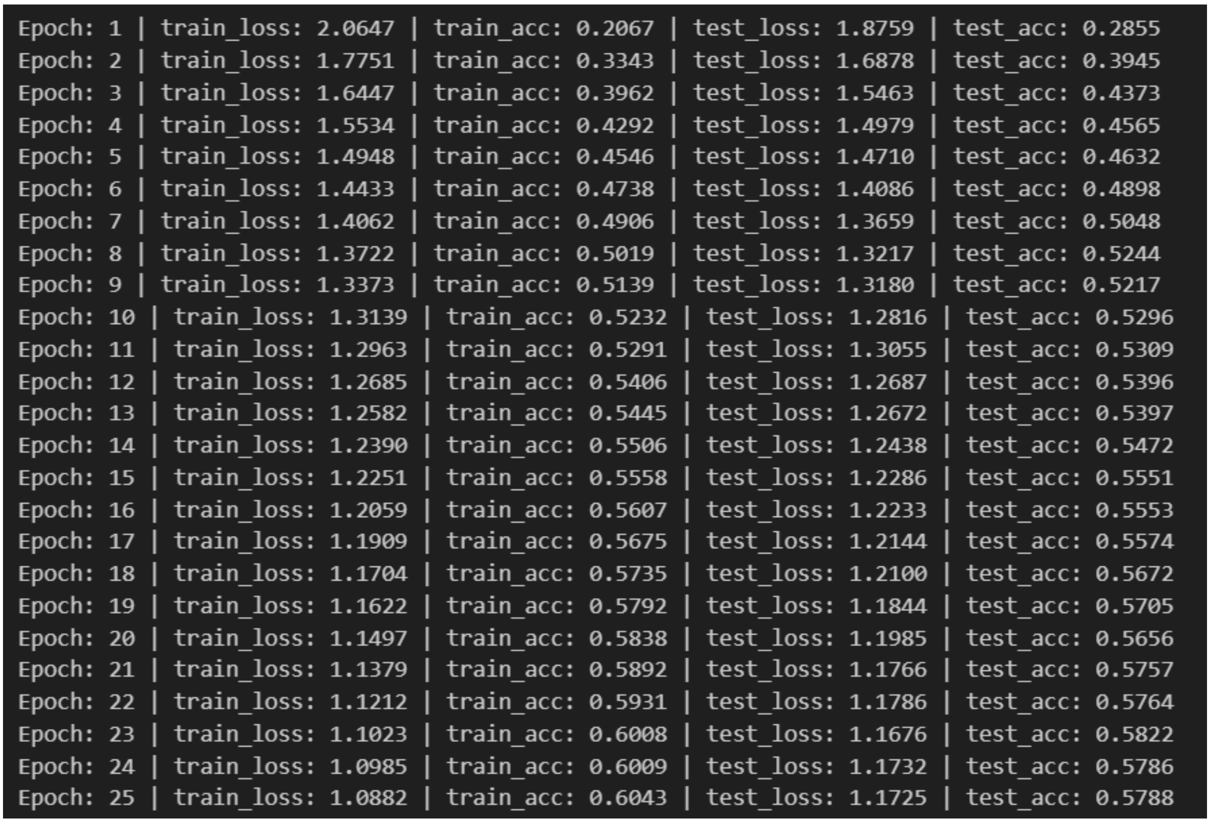


Vision transformer Experiments- Results & Analysis

1. Train this model on the CIFAR-10 dataset for 10-class classification. Keep the number of attention heads to be 4 for all the experiments.

Experiment 1 Hyperparameters	**Values Used**
No. of Training Epochs	25
Patch size	4 X 4
Number of attention heads	4
Overlapping Used	No

Experiment 1 Results



```
Epoch: 1 | train_loss: 2.0647 | train_acc: 0.2067 | test_loss: 1.8759 | test_acc: 0.2855
Epoch: 2 | train_loss: 1.7751 | train_acc: 0.3343 | test_loss: 1.6878 | test_acc: 0.3945
Epoch: 3 | train_loss: 1.6447 | train_acc: 0.3962 | test_loss: 1.5463 | test_acc: 0.4373
Epoch: 4 | train_loss: 1.5534 | train_acc: 0.4292 | test_loss: 1.4979 | test_acc: 0.4565
Epoch: 5 | train_loss: 1.4948 | train_acc: 0.4546 | test_loss: 1.4710 | test_acc: 0.4632
Epoch: 6 | train_loss: 1.4433 | train_acc: 0.4738 | test_loss: 1.4086 | test_acc: 0.4898
Epoch: 7 | train_loss: 1.4062 | train_acc: 0.4906 | test_loss: 1.3659 | test_acc: 0.5048
Epoch: 8 | train_loss: 1.3722 | train_acc: 0.5019 | test_loss: 1.3217 | test_acc: 0.5244
Epoch: 9 | train_loss: 1.3373 | train_acc: 0.5139 | test_loss: 1.3180 | test_acc: 0.5217
Epoch: 10 | train_loss: 1.3139 | train_acc: 0.5232 | test_loss: 1.2816 | test_acc: 0.5296
Epoch: 11 | train_loss: 1.2963 | train_acc: 0.5291 | test_loss: 1.3055 | test_acc: 0.5309
Epoch: 12 | train_loss: 1.2685 | train_acc: 0.5406 | test_loss: 1.2687 | test_acc: 0.5396
Epoch: 13 | train_loss: 1.2582 | train_acc: 0.5445 | test_loss: 1.2672 | test_acc: 0.5397
Epoch: 14 | train_loss: 1.2390 | train_acc: 0.5506 | test_loss: 1.2438 | test_acc: 0.5472
Epoch: 15 | train_loss: 1.2251 | train_acc: 0.5558 | test_loss: 1.2286 | test_acc: 0.5551
Epoch: 16 | train_loss: 1.2059 | train_acc: 0.5607 | test_loss: 1.2233 | test_acc: 0.5553
Epoch: 17 | train_loss: 1.1909 | train_acc: 0.5675 | test_loss: 1.2144 | test_acc: 0.5574
Epoch: 18 | train_loss: 1.1704 | train_acc: 0.5735 | test_loss: 1.2100 | test_acc: 0.5672
Epoch: 19 | train_loss: 1.1622 | train_acc: 0.5792 | test_loss: 1.1844 | test_acc: 0.5705
Epoch: 20 | train_loss: 1.1497 | train_acc: 0.5838 | test_loss: 1.1985 | test_acc: 0.5656
Epoch: 21 | train_loss: 1.1379 | train_acc: 0.5892 | test_loss: 1.1766 | test_acc: 0.5757
Epoch: 22 | train_loss: 1.1212 | train_acc: 0.5931 | test_loss: 1.1786 | test_acc: 0.5764
Epoch: 23 | train_loss: 1.1023 | train_acc: 0.6008 | test_loss: 1.1676 | test_acc: 0.5822
Epoch: 24 | train_loss: 1.0985 | train_acc: 0.6009 | test_loss: 1.1732 | test_acc: 0.5786
Epoch: 25 | train_loss: 1.0882 | train_acc: 0.6043 | test_loss: 1.1725 | test_acc: 0.5788
```

Observations

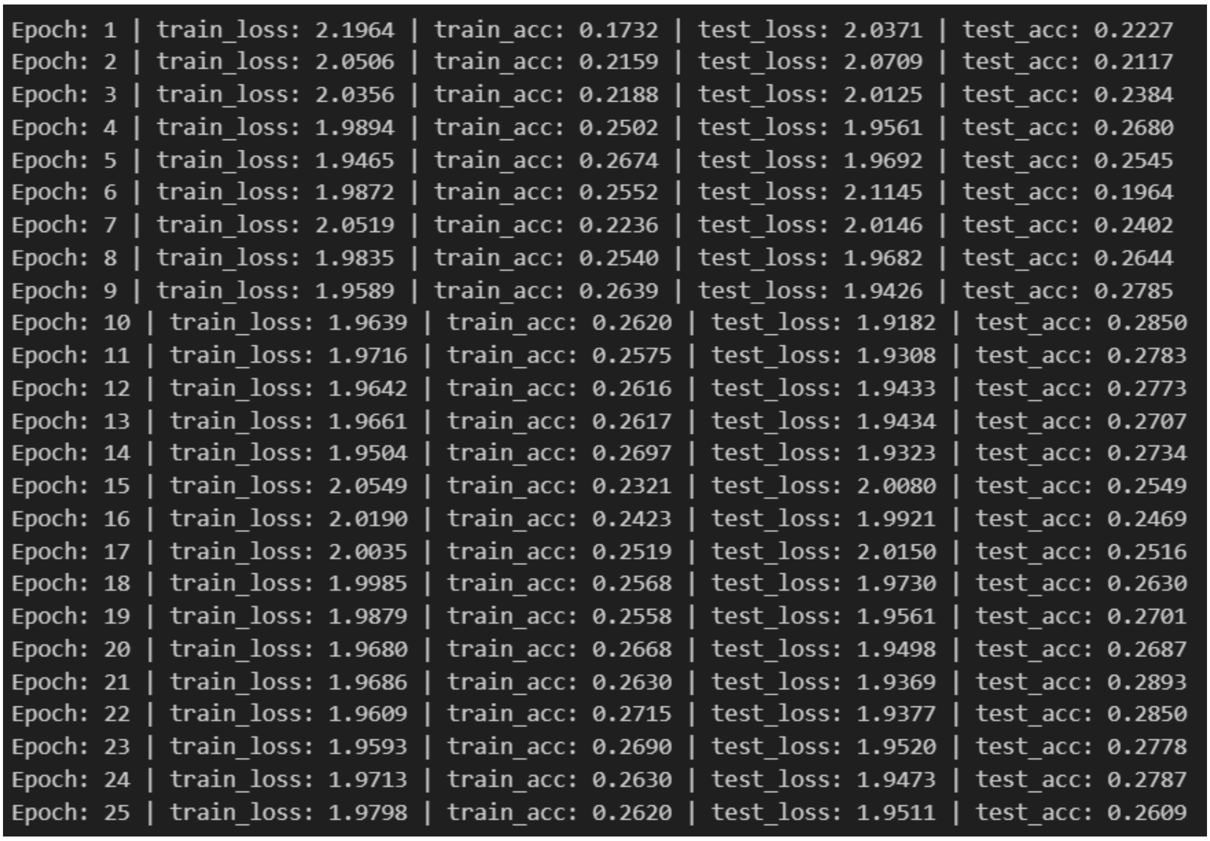
- The model execution took 20 minutes for 25 epochs.
- The model is performing sufficiently well with improving training and testing accuracies (& thereby decreasing training and testing losses respectively) seen with increasing number of epochs.
- The training accuracy improved from ~20% (epoch 1) to ~60% (epoch 25)
- The testing accuracy improved from ~29 (epoch 1) to ~58% (epoch 25)
- Observing the trend, it is expected that the model will improve with increased number of epochs. For now, I have used the epoch=25 as standard for model comparison against all other experiments as mentioned further in the document.

2. Try out different patch sizes (like 4x4, 8x8, 16x16). You can divide the image into both overlapping and non-overlapping patches.

Experiment 2 Hyperparameters	**Values Used**
No. of Training Epochs	25
Patch size	8 X 8 , 16 x 16
Number of attention heads	4
Overlapping Used	No

Experiment 2 Results

Results for patch size=8 X 8

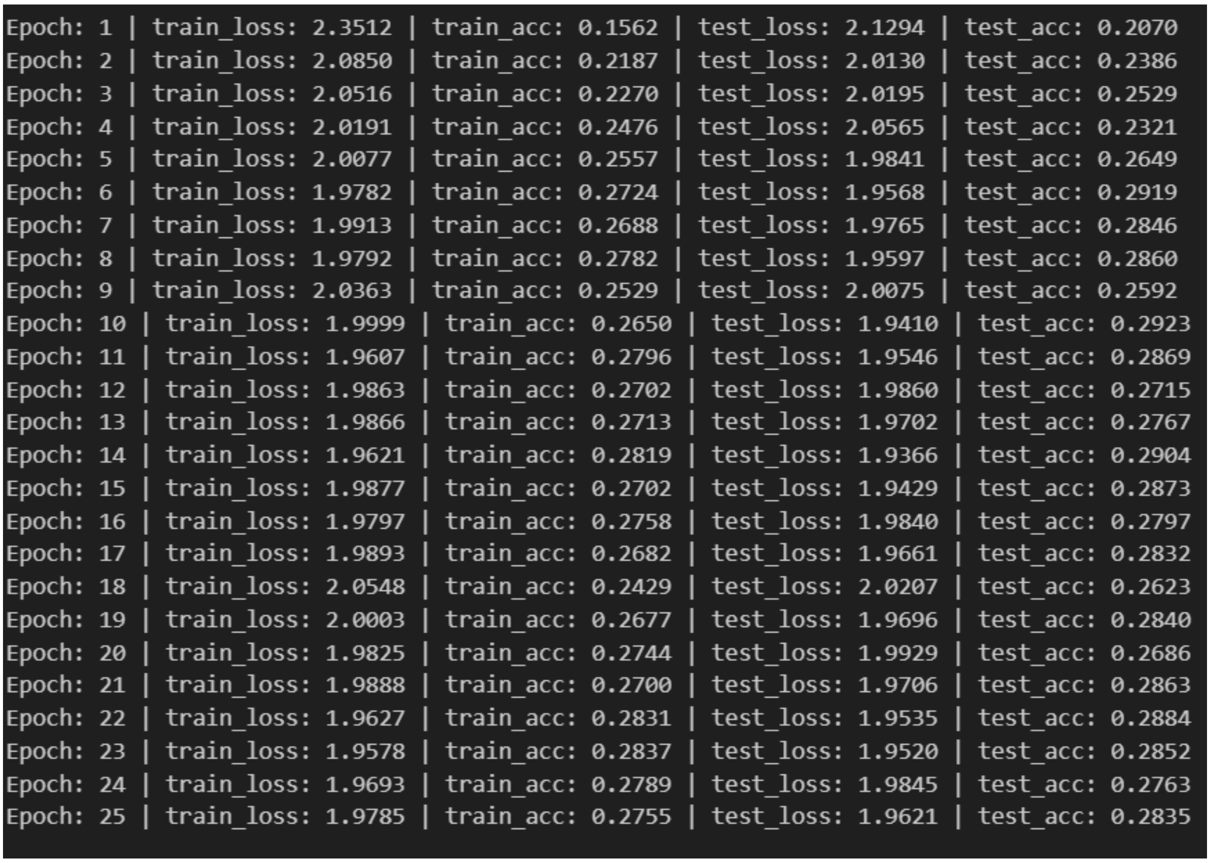


```
Epoch: 1 | train_loss: 2.1964 | train_acc: 0.1732 | test_loss: 2.0371 | test_acc: 0.2227
Epoch: 2 | train_loss: 2.0506 | train_acc: 0.2159 | test_loss: 2.0709 | test_acc: 0.2117
Epoch: 3 | train_loss: 2.0356 | train_acc: 0.2188 | test_loss: 2.0125 | test_acc: 0.2384
Epoch: 4 | train_loss: 1.9894 | train_acc: 0.2502 | test_loss: 1.9561 | test_acc: 0.2680
Epoch: 5 | train_loss: 1.9465 | train_acc: 0.2674 | test_loss: 1.9692 | test_acc: 0.2545
Epoch: 6 | train_loss: 1.9872 | train_acc: 0.2552 | test_loss: 2.1145 | test_acc: 0.1964
Epoch: 7 | train_loss: 2.0519 | train_acc: 0.2236 | test_loss: 2.0146 | test_acc: 0.2402
Epoch: 8 | train_loss: 1.9835 | train_acc: 0.2540 | test_loss: 1.9682 | test_acc: 0.2644
Epoch: 9 | train_loss: 1.9589 | train_acc: 0.2639 | test_loss: 1.9426 | test_acc: 0.2785
Epoch: 10 | train_loss: 1.9639 | train_acc: 0.2620 | test_loss: 1.9182 | test_acc: 0.2850
Epoch: 11 | train_loss: 1.9716 | train_acc: 0.2575 | test_loss: 1.9308 | test_acc: 0.2783
Epoch: 12 | train_loss: 1.9642 | train_acc: 0.2616 | test_loss: 1.9433 | test_acc: 0.2773
Epoch: 13 | train_loss: 1.9661 | train_acc: 0.2617 | test_loss: 1.9434 | test_acc: 0.2707
Epoch: 14 | train_loss: 1.9504 | train_acc: 0.2697 | test_loss: 1.9323 | test_acc: 0.2734
Epoch: 15 | train_loss: 2.0549 | train_acc: 0.2321 | test_loss: 2.0080 | test_acc: 0.2549
Epoch: 16 | train_loss: 2.0190 | train_acc: 0.2423 | test_loss: 1.9921 | test_acc: 0.2469
Epoch: 17 | train_loss: 2.0035 | train_acc: 0.2519 | test_loss: 2.0150 | test_acc: 0.2516
Epoch: 18 | train_loss: 1.9985 | train_acc: 0.2568 | test_loss: 1.9730 | test_acc: 0.2630
Epoch: 19 | train_loss: 1.9879 | train_acc: 0.2558 | test_loss: 1.9561 | test_acc: 0.2701
Epoch: 20 | train_loss: 1.9680 | train_acc: 0.2668 | test_loss: 1.9498 | test_acc: 0.2687
Epoch: 21 | train_loss: 1.9686 | train_acc: 0.2630 | test_loss: 1.9369 | test_acc: 0.2893
Epoch: 22 | train_loss: 1.9609 | train_acc: 0.2715 | test_loss: 1.9377 | test_acc: 0.2850
Epoch: 23 | train_loss: 1.9593 | train_acc: 0.2690 | test_loss: 1.9520 | test_acc: 0.2778
Epoch: 24 | train_loss: 1.9713 | train_acc: 0.2630 | test_loss: 1.9473 | test_acc: 0.2787
Epoch: 25 | train_loss: 1.9798 | train_acc: 0.2620 | test_loss: 1.9511 | test_acc: 0.2609
```

Observations (8 X 8 patch size)

- The model execution took ~18 minutes for 25 epochs.
- Model performance is poor.
- The training and testing accuracy do not seem to improve much through the epochs. The current learning rate used (0.005) does not seem to agree with the model. The learning rate needs to be changed to find a better convergence for the model. Either case, the model would require more than 25 epochs to get to a considerable range of accuracy.

Results for patch size=16 X 16



```
Epoch: 1 | train_loss: 2.3512 | train_acc: 0.1562 | test_loss: 2.1294 | test_acc: 0.2070
Epoch: 2 | train_loss: 2.0850 | train_acc: 0.2187 | test_loss: 2.0130 | test_acc: 0.2386
Epoch: 3 | train_loss: 2.0516 | train_acc: 0.2270 | test_loss: 2.0195 | test_acc: 0.2529
Epoch: 4 | train_loss: 2.0191 | train_acc: 0.2476 | test_loss: 2.0565 | test_acc: 0.2321
Epoch: 5 | train_loss: 2.0077 | train_acc: 0.2557 | test_loss: 1.9841 | test_acc: 0.2649
Epoch: 6 | train_loss: 1.9782 | train_acc: 0.2724 | test_loss: 1.9568 | test_acc: 0.2919
Epoch: 7 | train_loss: 1.9913 | train_acc: 0.2688 | test_loss: 1.9765 | test_acc: 0.2846
Epoch: 8 | train_loss: 1.9792 | train_acc: 0.2782 | test_loss: 1.9597 | test_acc: 0.2860
Epoch: 9 | train_loss: 2.0363 | train_acc: 0.2529 | test_loss: 2.0075 | test_acc: 0.2592
Epoch: 10 | train_loss: 1.9999 | train_acc: 0.2650 | test_loss: 1.9410 | test_acc: 0.2923
Epoch: 11 | train_loss: 1.9607 | train_acc: 0.2796 | test_loss: 1.9546 | test_acc: 0.2869
Epoch: 12 | train_loss: 1.9863 | train_acc: 0.2702 | test_loss: 1.9860 | test_acc: 0.2715
Epoch: 13 | train_loss: 1.9866 | train_acc: 0.2713 | test_loss: 1.9702 | test_acc: 0.2767
Epoch: 14 | train_loss: 1.9621 | train_acc: 0.2819 | test_loss: 1.9366 | test_acc: 0.2904
Epoch: 15 | train_loss: 1.9877 | train_acc: 0.2702 | test_loss: 1.9429 | test_acc: 0.2873
Epoch: 16 | train_loss: 1.9797 | train_acc: 0.2758 | test_loss: 1.9840 | test_acc: 0.2797
Epoch: 17 | train_loss: 1.9893 | train_acc: 0.2682 | test_loss: 1.9661 | test_acc: 0.2832
Epoch: 18 | train_loss: 2.0548 | train_acc: 0.2429 | test_loss: 2.0207 | test_acc: 0.2623
Epoch: 19 | train_loss: 2.0003 | train_acc: 0.2677 | test_loss: 1.9696 | test_acc: 0.2840
Epoch: 20 | train_loss: 1.9825 | train_acc: 0.2744 | test_loss: 1.9929 | test_acc: 0.2686
Epoch: 21 | train_loss: 1.9888 | train_acc: 0.2700 | test_loss: 1.9706 | test_acc: 0.2863
Epoch: 22 | train_loss: 1.9627 | train_acc: 0.2831 | test_loss: 1.9535 | test_acc: 0.2884
Epoch: 23 | train_loss: 1.9578 | train_acc: 0.2837 | test_loss: 1.9520 | test_acc: 0.2852
Epoch: 24 | train_loss: 1.9693 | train_acc: 0.2789 | test_loss: 1.9845 | test_acc: 0.2763
Epoch: 25 | train_loss: 1.9785 | train_acc: 0.2755 | test_loss: 1.9621 | test_acc: 0.2835
```

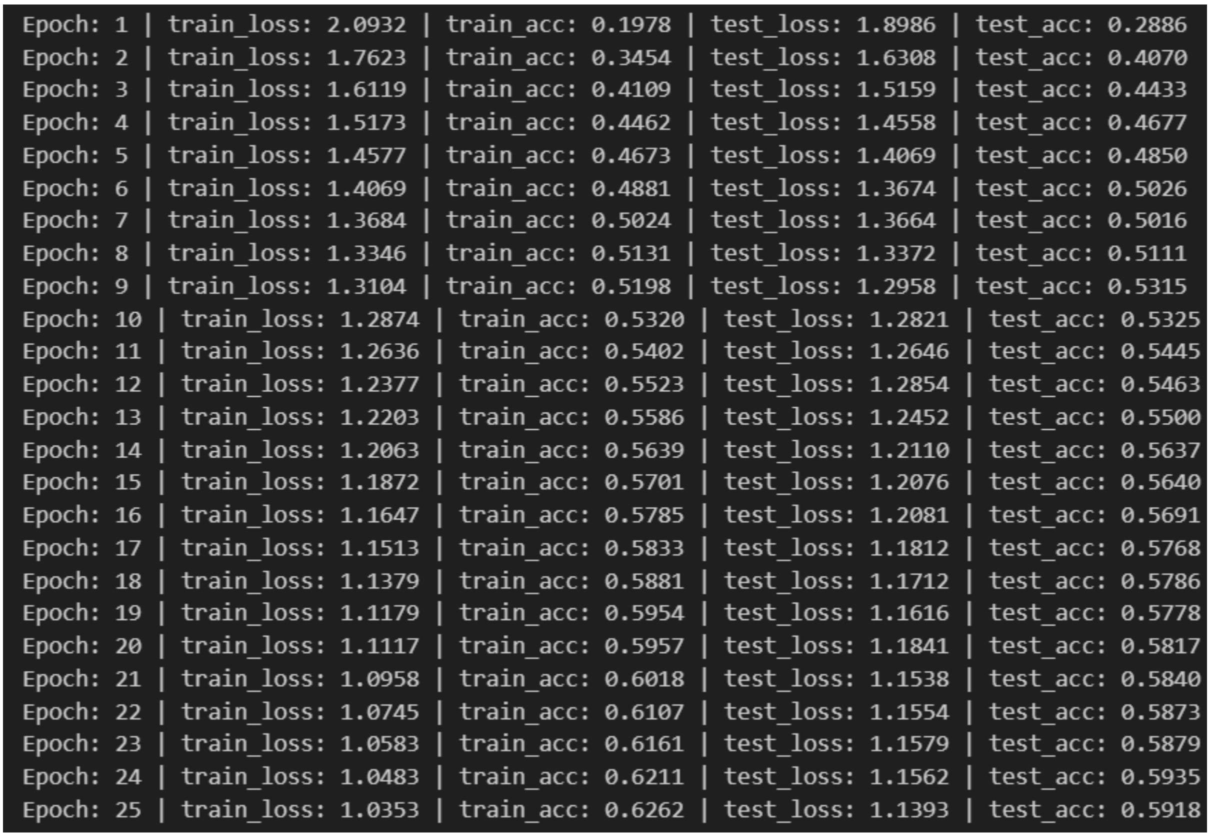
Observations (16 X 16 patch size)

- The model execution took 16.5 minutes for 25 epochs.
- Model performance is poor.
- The training and testing accuracy do not seem to improve much through the epochs. The current learning rate used (0.005) does not seem to agree with the model. The learning rate needs to be changed to find a better convergence for the model. Either case, the model would require more than 25 epochs to get to a considerable range of accuracy.
- This model seems to have marginally better training as well as testing accuracies than the 8 X 8 version.

3. How does model performance change if you vary the number of attention heads?

Experiment 3 Hyperparameters	**Values Used**
No. of Training Epochs	25
Patch size	4 X 4
Number of attention heads	6,8,12
Overlapping Used	No

Results for attention heads=6

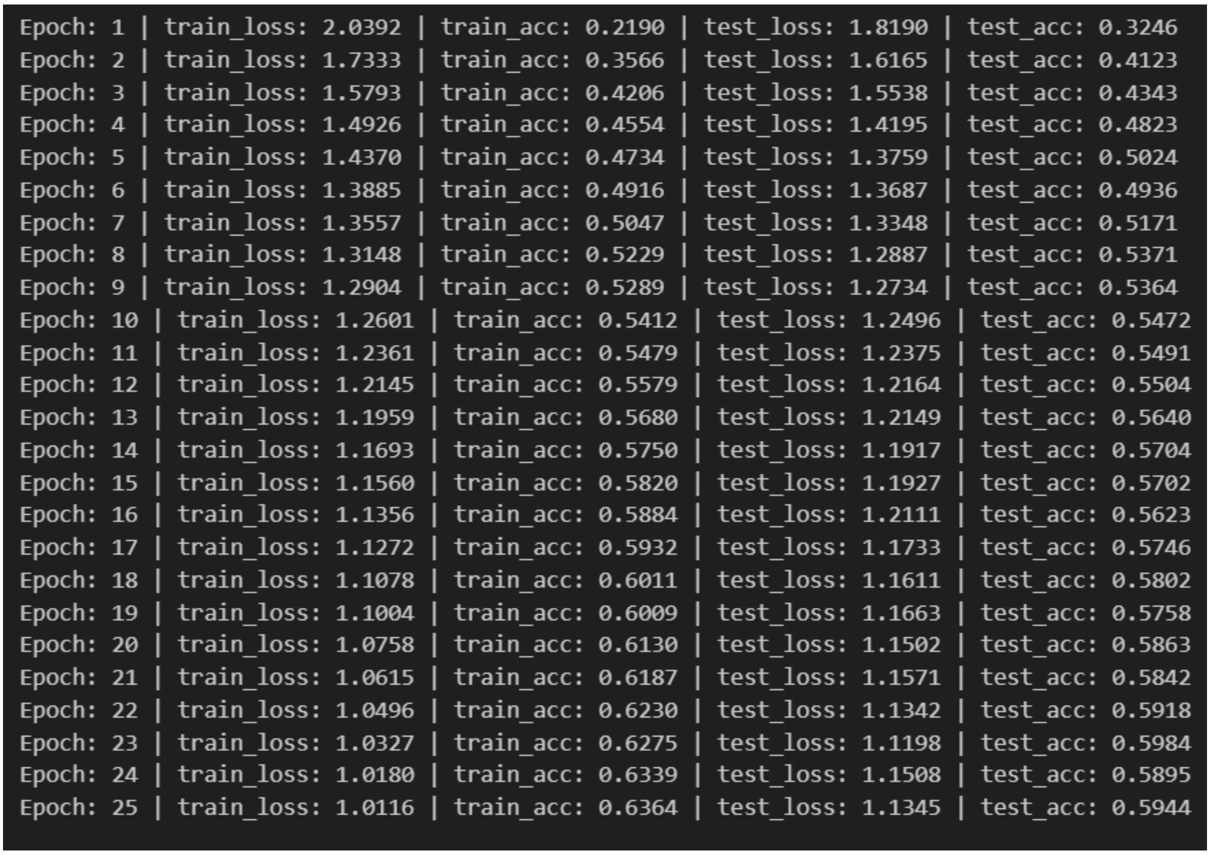


```
Epoch: 1 | train_loss: 2.0932 | train_acc: 0.1978 | test_loss: 1.8986 | test_acc: 0.2886
Epoch: 2 | train_loss: 1.7623 | train_acc: 0.3454 | test_loss: 1.6308 | test_acc: 0.4070
Epoch: 3 | train_loss: 1.6119 | train_acc: 0.4109 | test_loss: 1.5159 | test_acc: 0.4433
Epoch: 4 | train_loss: 1.5173 | train_acc: 0.4462 | test_loss: 1.4558 | test_acc: 0.4677
Epoch: 5 | train_loss: 1.4577 | train_acc: 0.4673 | test_loss: 1.4069 | test_acc: 0.4850
Epoch: 6 | train_loss: 1.4069 | train_acc: 0.4881 | test_loss: 1.3674 | test_acc: 0.5026
Epoch: 7 | train_loss: 1.3684 | train_acc: 0.5024 | test_loss: 1.3664 | test_acc: 0.5016
Epoch: 8 | train_loss: 1.3346 | train_acc: 0.5131 | test_loss: 1.3372 | test_acc: 0.5111
Epoch: 9 | train_loss: 1.3104 | train_acc: 0.5198 | test_loss: 1.2958 | test_acc: 0.5315
Epoch: 10 | train_loss: 1.2874 | train_acc: 0.5320 | test_loss: 1.2821 | test_acc: 0.5325
Epoch: 11 | train_loss: 1.2636 | train_acc: 0.5402 | test_loss: 1.2646 | test_acc: 0.5445
Epoch: 12 | train_loss: 1.2377 | train_acc: 0.5523 | test_loss: 1.2854 | test_acc: 0.5463
Epoch: 13 | train_loss: 1.2203 | train_acc: 0.5586 | test_loss: 1.2452 | test_acc: 0.5500
Epoch: 14 | train_loss: 1.2063 | train_acc: 0.5639 | test_loss: 1.2110 | test_acc: 0.5637
Epoch: 15 | train_loss: 1.1872 | train_acc: 0.5701 | test_loss: 1.2076 | test_acc: 0.5640
Epoch: 16 | train_loss: 1.1647 | train_acc: 0.5785 | test_loss: 1.2081 | test_acc: 0.5691
Epoch: 17 | train_loss: 1.1513 | train_acc: 0.5833 | test_loss: 1.1812 | test_acc: 0.5768
Epoch: 18 | train_loss: 1.1379 | train_acc: 0.5881 | test_loss: 1.1712 | test_acc: 0.5786
Epoch: 19 | train_loss: 1.1179 | train_acc: 0.5954 | test_loss: 1.1616 | test_acc: 0.5778
Epoch: 20 | train_loss: 1.1117 | train_acc: 0.5957 | test_loss: 1.1841 | test_acc: 0.5817
Epoch: 21 | train_loss: 1.0958 | train_acc: 0.6018 | test_loss: 1.1538 | test_acc: 0.5840
Epoch: 22 | train_loss: 1.0745 | train_acc: 0.6107 | test_loss: 1.1554 | test_acc: 0.5873
Epoch: 23 | train_loss: 1.0583 | train_acc: 0.6161 | test_loss: 1.1579 | test_acc: 0.5879
Epoch: 24 | train_loss: 1.0483 | train_acc: 0.6211 | test_loss: 1.1562 | test_acc: 0.5935
Epoch: 25 | train_loss: 1.0353 | train_acc: 0.6262 | test_loss: 1.1393 | test_acc: 0.5918
```

Observations (for 6 attention heads)

- The model execution took 24.3 minutes for 25 epochs.
- Model performance is slightly better than with 4 attention heads. ~63 % vs ~60 % accuracies (for training) and ~60 % vs ~58 % accuracies (for testing)

Results for attention heads=8

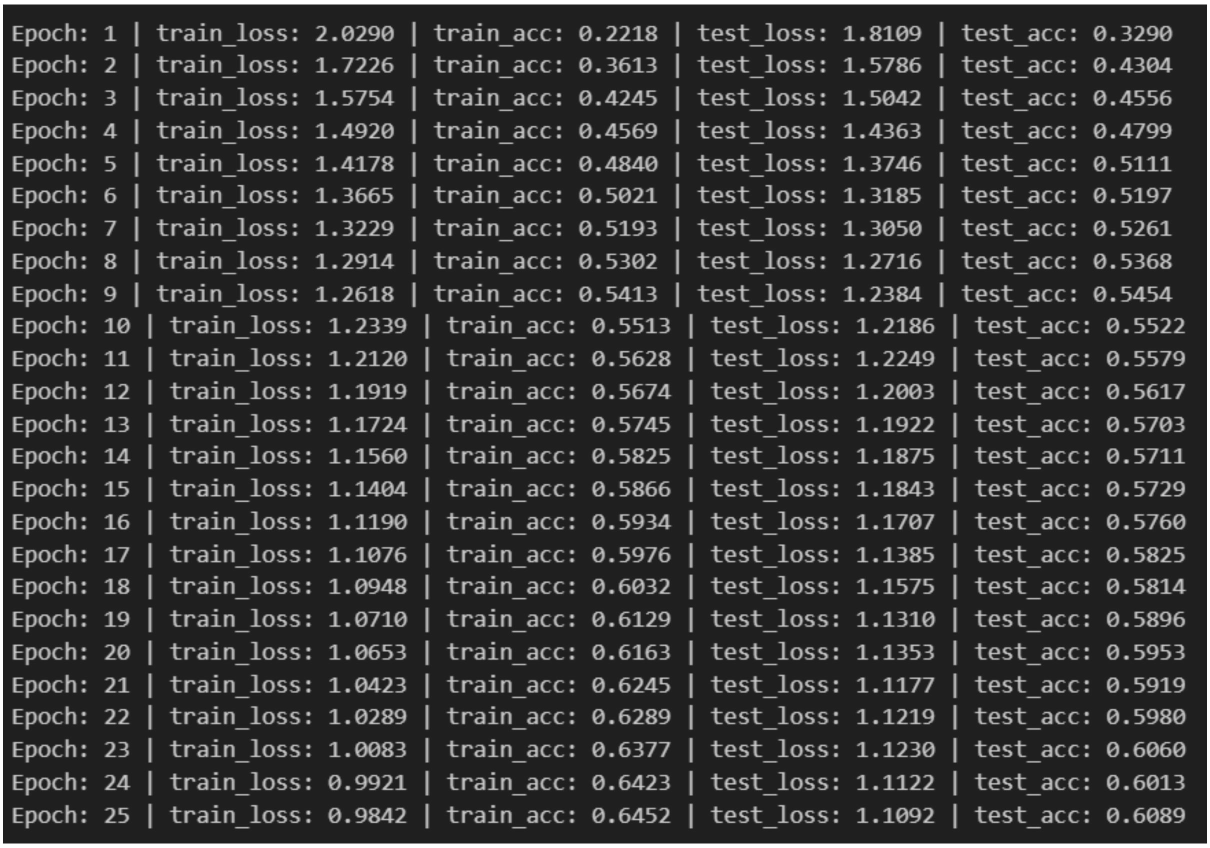


```
Epoch: 1 | train_loss: 2.0392 | train_acc: 0.2190 | test_loss: 1.8190 | test_acc: 0.3246
Epoch: 2 | train_loss: 1.7333 | train_acc: 0.3566 | test_loss: 1.6165 | test_acc: 0.4123
Epoch: 3 | train_loss: 1.5793 | train_acc: 0.4206 | test_loss: 1.5538 | test_acc: 0.4343
Epoch: 4 | train_loss: 1.4926 | train_acc: 0.4554 | test_loss: 1.4195 | test_acc: 0.4823
Epoch: 5 | train_loss: 1.4370 | train_acc: 0.4734 | test_loss: 1.3759 | test_acc: 0.5024
Epoch: 6 | train_loss: 1.3885 | train_acc: 0.4916 | test_loss: 1.3687 | test_acc: 0.4936
Epoch: 7 | train_loss: 1.3557 | train_acc: 0.5047 | test_loss: 1.3348 | test_acc: 0.5171
Epoch: 8 | train_loss: 1.3148 | train_acc: 0.5229 | test_loss: 1.2887 | test_acc: 0.5371
Epoch: 9 | train_loss: 1.2904 | train_acc: 0.5289 | test_loss: 1.2734 | test_acc: 0.5364
Epoch: 10 | train_loss: 1.2601 | train_acc: 0.5412 | test_loss: 1.2496 | test_acc: 0.5472
Epoch: 11 | train_loss: 1.2361 | train_acc: 0.5479 | test_loss: 1.2375 | test_acc: 0.5491
Epoch: 12 | train_loss: 1.2145 | train_acc: 0.5579 | test_loss: 1.2164 | test_acc: 0.5504
Epoch: 13 | train_loss: 1.1959 | train_acc: 0.5680 | test_loss: 1.2149 | test_acc: 0.5640
Epoch: 14 | train_loss: 1.1693 | train_acc: 0.5750 | test_loss: 1.1917 | test_acc: 0.5704
Epoch: 15 | train_loss: 1.1560 | train_acc: 0.5820 | test_loss: 1.1927 | test_acc: 0.5702
Epoch: 16 | train_loss: 1.1356 | train_acc: 0.5884 | test_loss: 1.2111 | test_acc: 0.5623
Epoch: 17 | train_loss: 1.1272 | train_acc: 0.5932 | test_loss: 1.1733 | test_acc: 0.5746
Epoch: 18 | train_loss: 1.1078 | train_acc: 0.6011 | test_loss: 1.1611 | test_acc: 0.5802
Epoch: 19 | train_loss: 1.1004 | train_acc: 0.6009 | test_loss: 1.1663 | test_acc: 0.5758
Epoch: 20 | train_loss: 1.0758 | train_acc: 0.6130 | test_loss: 1.1502 | test_acc: 0.5863
Epoch: 21 | train_loss: 1.0615 | train_acc: 0.6187 | test_loss: 1.1571 | test_acc: 0.5842
Epoch: 22 | train_loss: 1.0496 | train_acc: 0.6230 | test_loss: 1.1342 | test_acc: 0.5918
Epoch: 23 | train_loss: 1.0327 | train_acc: 0.6275 | test_loss: 1.1198 | test_acc: 0.5984
Epoch: 24 | train_loss: 1.0180 | train_acc: 0.6339 | test_loss: 1.1508 | test_acc: 0.5895
Epoch: 25 | train_loss: 1.0116 | train_acc: 0.6364 | test_loss: 1.1345 | test_acc: 0.5944
```

Observations (for 8 attention heads)

- The model execution took 21.3 minutes for 25 epochs.
- Model performance is slightly better than with 6 attention heads and hence better than 4 attention heads. ~64 % vs ~63 % accuracies (for training) and marginally better accuracy for testing.
- The model performance is seen to be better against those with models with 6 and 4 attention heads from the first epoch itself

<u>Results for attention heads=10</u>



```
Epoch: 1 | train_loss: 2.0290 | train_acc: 0.2218 | test_loss: 1.8109 | test_acc: 0.3290
Epoch: 2 | train_loss: 1.7226 | train_acc: 0.3613 | test_loss: 1.5786 | test_acc: 0.4304
Epoch: 3 | train_loss: 1.5754 | train_acc: 0.4245 | test_loss: 1.5042 | test_acc: 0.4556
Epoch: 4 | train_loss: 1.4920 | train_acc: 0.4569 | test_loss: 1.4363 | test_acc: 0.4799
Epoch: 5 | train_loss: 1.4178 | train_acc: 0.4840 | test_loss: 1.3746 | test_acc: 0.5111
Epoch: 6 | train_loss: 1.3665 | train_acc: 0.5021 | test_loss: 1.3185 | test_acc: 0.5197
Epoch: 7 | train_loss: 1.3229 | train_acc: 0.5193 | test_loss: 1.3050 | test_acc: 0.5261
Epoch: 8 | train_loss: 1.2914 | train_acc: 0.5302 | test_loss: 1.2716 | test_acc: 0.5368
Epoch: 9 | train_loss: 1.2618 | train_acc: 0.5413 | test_loss: 1.2384 | test_acc: 0.5454
Epoch: 10 | train_loss: 1.2339 | train_acc: 0.5513 | test_loss: 1.2186 | test_acc: 0.5522
Epoch: 11 | train_loss: 1.2120 | train_acc: 0.5628 | test_loss: 1.2249 | test_acc: 0.5579
Epoch: 12 | train_loss: 1.1919 | train_acc: 0.5674 | test_loss: 1.2003 | test_acc: 0.5617
Epoch: 13 | train_loss: 1.1724 | train_acc: 0.5745 | test_loss: 1.1922 | test_acc: 0.5703
Epoch: 14 | train_loss: 1.1560 | train_acc: 0.5825 | test_loss: 1.1875 | test_acc: 0.5711
Epoch: 15 | train_loss: 1.1404 | train_acc: 0.5866 | test_loss: 1.1843 | test_acc: 0.5729
Epoch: 16 | train_loss: 1.1190 | train_acc: 0.5934 | test_loss: 1.1707 | test_acc: 0.5760
Epoch: 17 | train_loss: 1.1076 | train_acc: 0.5976 | test_loss: 1.1385 | test_acc: 0.5825
Epoch: 18 | train_loss: 1.0948 | train_acc: 0.6032 | test_loss: 1.1575 | test_acc: 0.5814
Epoch: 19 | train_loss: 1.0710 | train_acc: 0.6129 | test_loss: 1.1310 | test_acc: 0.5896
Epoch: 20 | train_loss: 1.0653 | train_acc: 0.6163 | test_loss: 1.1353 | test_acc: 0.5953
Epoch: 21 | train_loss: 1.0423 | train_acc: 0.6245 | test_loss: 1.1177 | test_acc: 0.5919
Epoch: 22 | train_loss: 1.0289 | train_acc: 0.6289 | test_loss: 1.1219 | test_acc: 0.5980
Epoch: 23 | train_loss: 1.0083 | train_acc: 0.6377 | test_loss: 1.1230 | test_acc: 0.6060
Epoch: 24 | train_loss: 0.9921 | train_acc: 0.6423 | test_loss: 1.1122 | test_acc: 0.6013
Epoch: 25 | train_loss: 0.9842 | train_acc: 0.6452 | test_loss: 1.1092 | test_acc: 0.6089
```

Observations (for 10 attention heads)

- The model execution took 35 minutes for 25 epochs.
- Model performance is slightly better than with 8 attention heads and hence better than 6 and 4 attention heads. ~64.5 % vs ~64 % accuracies (for training) and ~61 v/s ~60% marginally better accuracy for testing.

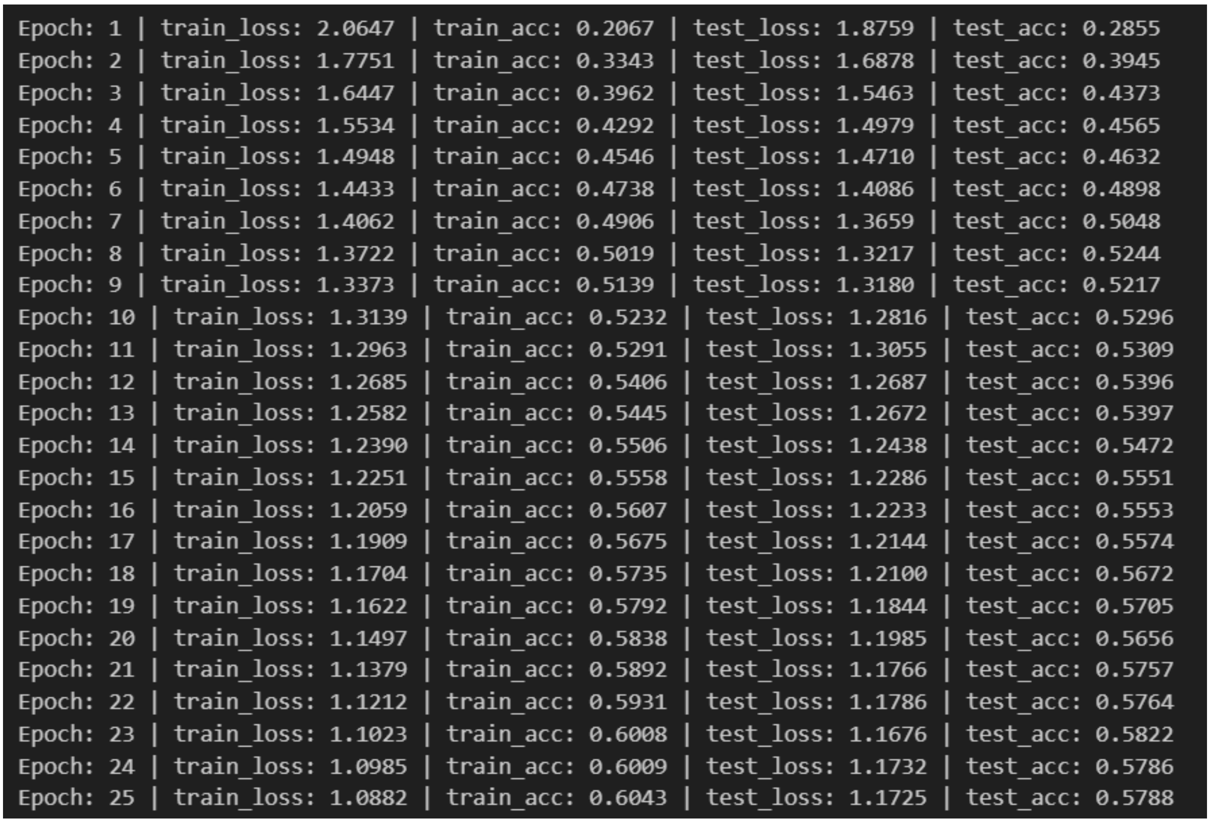
Final Remarks for experiment 3: With increasing number of attention heads, the model performance increasingly gets better for both training as well as testing.

4. Perform classification by using the CLS token from different layers of the model.
   a. Sub-experiment I: CLS token is used after patch embedding layer and the before the multi-layer attention and multi-layer perceptron layer
   b. Sub-experiment II: CLS token is used after the multi-layer attention and multilayer perceptron layer.

Experiment 4 Hyperparameters	**Values Used**

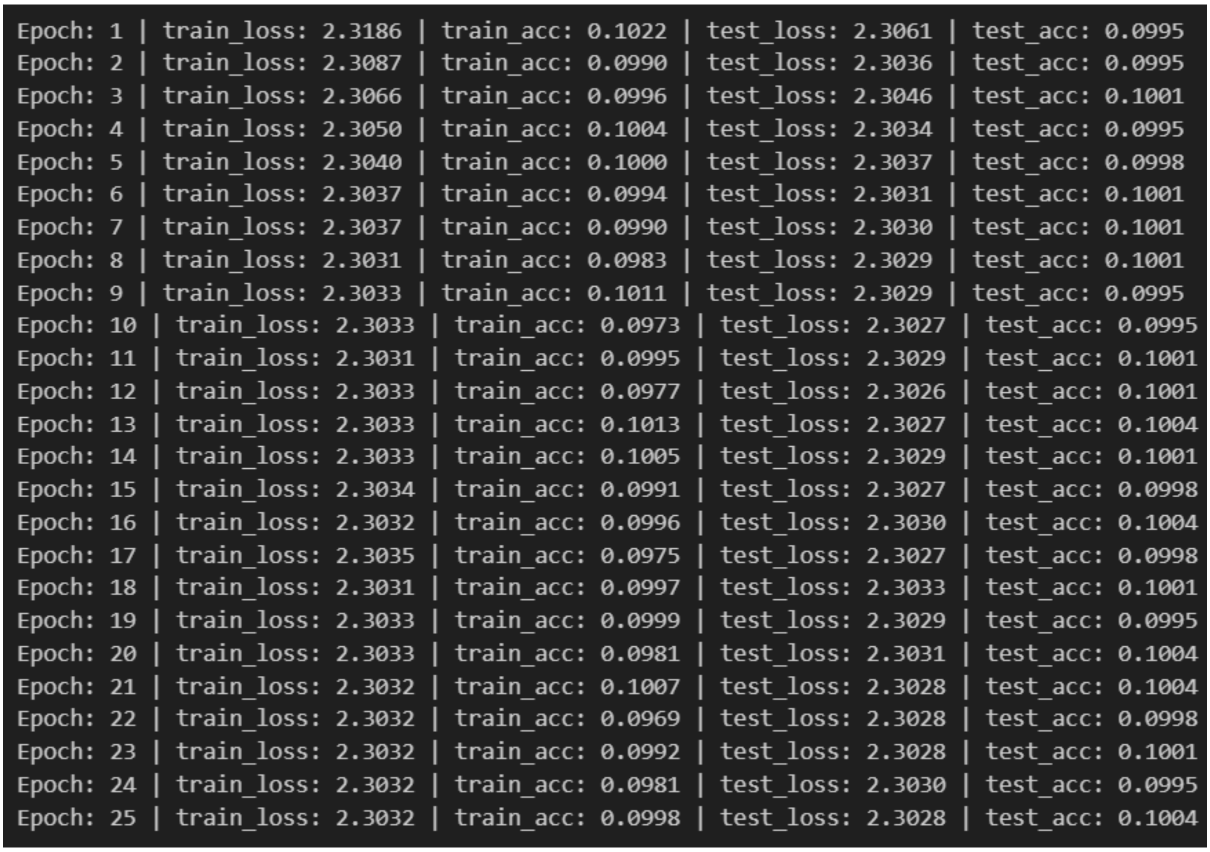
No. of Training Epochs	25
Patch size	4 X 4
Number of attention heads	4
Overlapping Used	No
CLS Token	Used at different layers

Results for Sub-experiment I



```
Epoch: 1 | train_loss: 2.0647 | train_acc: 0.2067 | test_loss: 1.8759 | test_acc: 0.2855
Epoch: 2 | train_loss: 1.7751 | train_acc: 0.3343 | test_loss: 1.6878 | test_acc: 0.3945
Epoch: 3 | train_loss: 1.6447 | train_acc: 0.3962 | test_loss: 1.5463 | test_acc: 0.4373
Epoch: 4 | train_loss: 1.5534 | train_acc: 0.4292 | test_loss: 1.4979 | test_acc: 0.4565
Epoch: 5 | train_loss: 1.4948 | train_acc: 0.4546 | test_loss: 1.4710 | test_acc: 0.4632
Epoch: 6 | train_loss: 1.4433 | train_acc: 0.4738 | test_loss: 1.4086 | test_acc: 0.4898
Epoch: 7 | train_loss: 1.4062 | train_acc: 0.4906 | test_loss: 1.3659 | test_acc: 0.5048
Epoch: 8 | train_loss: 1.3722 | train_acc: 0.5019 | test_loss: 1.3217 | test_acc: 0.5244
Epoch: 9 | train_loss: 1.3373 | train_acc: 0.5139 | test_loss: 1.3180 | test_acc: 0.5217
Epoch: 10 | train_loss: 1.3139 | train_acc: 0.5232 | test_loss: 1.2816 | test_acc: 0.5296
Epoch: 11 | train_loss: 1.2963 | train_acc: 0.5291 | test_loss: 1.3055 | test_acc: 0.5309
Epoch: 12 | train_loss: 1.2685 | train_acc: 0.5406 | test_loss: 1.2687 | test_acc: 0.5396
Epoch: 13 | train_loss: 1.2582 | train_acc: 0.5445 | test_loss: 1.2672 | test_acc: 0.5397
Epoch: 14 | train_loss: 1.2390 | train_acc: 0.5506 | test_loss: 1.2438 | test_acc: 0.5472
Epoch: 15 | train_loss: 1.2251 | train_acc: 0.5558 | test_loss: 1.2286 | test_acc: 0.5551
Epoch: 16 | train_loss: 1.2059 | train_acc: 0.5607 | test_loss: 1.2233 | test_acc: 0.5553
Epoch: 17 | train_loss: 1.1909 | train_acc: 0.5675 | test_loss: 1.2144 | test_acc: 0.5574
Epoch: 18 | train_loss: 1.1704 | train_acc: 0.5735 | test_loss: 1.2100 | test_acc: 0.5672
Epoch: 19 | train_loss: 1.1622 | train_acc: 0.5792 | test_loss: 1.1844 | test_acc: 0.5705
Epoch: 20 | train_loss: 1.1497 | train_acc: 0.5838 | test_loss: 1.1985 | test_acc: 0.5656
Epoch: 21 | train_loss: 1.1379 | train_acc: 0.5892 | test_loss: 1.1766 | test_acc: 0.5757
Epoch: 22 | train_loss: 1.1212 | train_acc: 0.5931 | test_loss: 1.1786 | test_acc: 0.5764
Epoch: 23 | train_loss: 1.1023 | train_acc: 0.6008 | test_loss: 1.1676 | test_acc: 0.5822
Epoch: 24 | train_loss: 1.0985 | train_acc: 0.6009 | test_loss: 1.1732 | test_acc: 0.5786
Epoch: 25 | train_loss: 1.0882 | train_acc: 0.6043 | test_loss: 1.1725 | test_acc: 0.5788
```

Results for Sub-experiment II



```
Epoch: 1 | train_loss: 2.3186 | train_acc: 0.1022 | test_loss: 2.3061 | test_acc: 0.0995
Epoch: 2 | train_loss: 2.3087 | train_acc: 0.0990 | test_loss: 2.3036 | test_acc: 0.0995
Epoch: 3 | train_loss: 2.3066 | train_acc: 0.0996 | test_loss: 2.3046 | test_acc: 0.1001
Epoch: 4 | train_loss: 2.3050 | train_acc: 0.1004 | test_loss: 2.3034 | test_acc: 0.0995
Epoch: 5 | train_loss: 2.3040 | train_acc: 0.1000 | test_loss: 2.3037 | test_acc: 0.0998
Epoch: 6 | train_loss: 2.3037 | train_acc: 0.0994 | test_loss: 2.3031 | test_acc: 0.1001
Epoch: 7 | train_loss: 2.3037 | train_acc: 0.0990 | test_loss: 2.3030 | test_acc: 0.1001
Epoch: 8 | train_loss: 2.3031 | train_acc: 0.0983 | test_loss: 2.3029 | test_acc: 0.1001
Epoch: 9 | train_loss: 2.3033 | train_acc: 0.1011 | test_loss: 2.3029 | test_acc: 0.0995
Epoch: 10 | train_loss: 2.3033 | train_acc: 0.0973 | test_loss: 2.3027 | test_acc: 0.0995
Epoch: 11 | train_loss: 2.3031 | train_acc: 0.0995 | test_loss: 2.3029 | test_acc: 0.1001
Epoch: 12 | train_loss: 2.3033 | train_acc: 0.0977 | test_loss: 2.3026 | test_acc: 0.1001
Epoch: 13 | train_loss: 2.3033 | train_acc: 0.1013 | test_loss: 2.3027 | test_acc: 0.1004
Epoch: 14 | train_loss: 2.3033 | train_acc: 0.1005 | test_loss: 2.3029 | test_acc: 0.1001
Epoch: 15 | train_loss: 2.3034 | train_acc: 0.0991 | test_loss: 2.3027 | test_acc: 0.0998
Epoch: 16 | train_loss: 2.3032 | train_acc: 0.0996 | test_loss: 2.3030 | test_acc: 0.1004
Epoch: 17 | train_loss: 2.3035 | train_acc: 0.0975 | test_loss: 2.3027 | test_acc: 0.0998
Epoch: 18 | train_loss: 2.3031 | train_acc: 0.0997 | test_loss: 2.3033 | test_acc: 0.1001
Epoch: 19 | train_loss: 2.3033 | train_acc: 0.0999 | test_loss: 2.3029 | test_acc: 0.0995
Epoch: 20 | train_loss: 2.3033 | train_acc: 0.0981 | test_loss: 2.3031 | test_acc: 0.1004
Epoch: 21 | train_loss: 2.3032 | train_acc: 0.1007 | test_loss: 2.3028 | test_acc: 0.1004
Epoch: 22 | train_loss: 2.3032 | train_acc: 0.0969 | test_loss: 2.3028 | test_acc: 0.0998
Epoch: 23 | train_loss: 2.3032 | train_acc: 0.0992 | test_loss: 2.3028 | test_acc: 0.1001
Epoch: 24 | train_loss: 2.3032 | train_acc: 0.0981 | test_loss: 2.3030 | test_acc: 0.0995
Epoch: 25 | train_loss: 2.3032 | train_acc: 0.0998 | test_loss: 2.3028 | test_acc: 0.1004
```

Combined Observations (for Sub-experiments I & 2)

- Sub-Experiment I took ~20 mins while Sub-Experiment II took ~25 minutes for execution of 25 epochs.
- Sub-Experiment I is distinguishably producing far better results than II. This is applicable to both training as well as testing accuracies. This can bet attributed to the fact that when CLS token is added after the multihead and MLP layers, this form of architecture does not take into consideration the class labels till the actual classification is performed and hence which results in very poor classification since the CLS token is effectively not trained by the transformer architecture.