DL-Ops Assignment-1 Report

Task: Implement a neural network and utilize Tiny ImageNet and SHL-10 the dataset for the analysis

Objectives:

Question 1.

- Train ResNet18 on the Tiny ImageNet dataset.
- Use CrossEntropy as the final classification loss function
- Use Triplet Loss with hard mining as the final classification loss function
- Use Central Loss as the final classification loss function

Procedure:

CROSS ENTROPY LOSS

- Import required packages. (PyTorch, NumPy, Matplotlib, torchmetrics …etc.)
- Checking GPU availability.
- Setting up devices.
- Defining Resnet 18 model from scratch.
- Getting model instance.
- Printing model summary.

```
==========================================================================
================================================
Layer (type (var_name))              Input Shape          Output Shape          Param #
Trainable
==========================================================================
================================================
ResNet (ResNet)                    [32, 3, 224, 224]    [32, 200]          --              True
├─Sequential (conv1)                 [32, 3, 224, 224]    [32, 64, 112, 112]   --              True
│    └─Conv2d (0)                  [32, 3, 224, 224]    [32, 64, 112, 112]   9,472             True
│    └─BatchNorm2d (1)                [32, 64, 112, 112]   [32, 64, 112, 112]   128              True
│    └─ReLU (2)                  [32, 64, 112, 112]   [32, 64, 112, 112]   --              --
├─MaxPool2d (maxpool)                 [32, 64, 112, 112]   [32, 64, 56, 56]     --              --
├─Sequential (layer0)               [32, 64, 56, 56]     [32, 64, 56, 56]     --             True
│    └─ResidualBlock (0)              [32, 64, 56, 56]     [32, 64, 56, 56]     --             True
│    │    └─Sequential (conv1)          [32, 64, 56, 56]     [32, 64, 56, 56]     37,056             True
│    │    └─Sequential (conv2)          [32, 64, 56, 56]     [32, 64, 56, 56]     37,056             True
│    │    └─ReLU (relu)             [32, 64, 56, 56]     [32, 64, 56, 56]     --              --
│    └─ResidualBlock (1)              [32, 64, 56, 56]     [32, 64, 56, 56]     --             True
│    │    └─Sequential (conv1)          [32, 64, 56, 56]     [32, 64, 56, 56]     37,056             True
│    │    └─Sequential (conv2)          [32, 64, 56, 56]     [32, 64, 56, 56]     37,056             True
│    │    └─ReLU (relu)             [32, 64, 56, 56]     [32, 64, 56, 56]     --              --
├─Sequential (layer1)               [32, 64, 56, 56]     [32, 128, 28, 28]    --             True
│    └─ResidualBlock (0)              [32, 64, 56, 56]     [32, 128, 28, 28]    --             True
│    │    └─Sequential (conv1)          [32, 64, 56, 56]     [32, 128, 28, 28]    74,112             True
│    │    └─Sequential (conv2)          [32, 128, 28, 28]    [32, 128, 28, 28]    147,840            True
│    │      └─Sequential  (downsample)       [32,  64,  56,  56]       [32,  128,  28,  28]     8,576
True
│    │    └─ReLU (relu)              [32, 128, 28, 28]   [32, 128, 28, 28]    --               --
│    └─ResidualBlock (1)             [32, 128, 28, 28]   [32, 128, 28, 28]    --              True
│    │    └─Sequential (conv1)         [32, 128, 28, 28]    [32, 128, 28, 28]    147,840          True
│    │    └─Sequential (conv2)         [32, 128, 28, 28]    [32, 128, 28, 28]    147,840          True
│    │    └─ReLU (relu)              [32, 128, 28, 28]   [32, 128, 28, 28]    --               --
├─Sequential (layer2)              [32, 128, 28, 28]   [32, 256, 14, 14]    --              True
│    └─ResidualBlock (0)             [32, 128, 28, 28]   [32, 256, 14, 14]    --              True
│    │    └─Sequential (conv1)         [32, 128, 28, 28]    [32, 256, 14, 14]    295,680          True
│    │    └─Sequential (conv2)         [32, 256, 14, 14]    [32, 256, 14, 14]    590,592          True
│    │      └─Sequential  (downsample)       [32,  128,  28,  28]      [32,  256,  14,  14]     33,536
True
│    │    └─ReLU (relu)              [32, 256, 14, 14]   [32, 256, 14, 14]    --               --
│    └─ResidualBlock (1)             [32, 256, 14, 14]   [32, 256, 14, 14]    --              True
│    │    └─Sequential (conv1)         [32, 256, 14, 14]    [32, 256, 14, 14]    590,592          True
│    │    └─Sequential (conv2)         [32, 256, 14, 14]    [32, 256, 14, 14]    590,592          True
│    │    └─ReLU (relu)              [32, 256, 14, 14]   [32, 256, 14, 14]    --               --
├─Sequential (layer3)              [32, 256, 14, 14]   [32, 512, 7, 7]     --              True
│    └─ResidualBlock (0)             [32, 256, 14, 14]   [32, 512, 7, 7]     --              True
│    │    └─Sequential (conv1)         [32, 256, 14, 14]    [32, 512, 7, 7]     1,181,184          True
│    │    └─Sequential (conv2)         [32, 512, 7, 7]     [32, 512, 7, 7]     2,360,832          True
│    │      └─Sequential  (downsample)       [32,  256,  14,  14]     [32,  512,  7,  7]       132,608
True
│    │    └─ReLU (relu)              [32, 512, 7, 7]     [32, 512, 7, 7]     --               --
│    └─ResidualBlock (1)             [32, 512, 7, 7]     [32, 512, 7, 7]     --              True
│    │    └─Sequential (conv1)         [32, 512, 7, 7]     [32, 512, 7, 7]     2,360,832          True
│    │    └─Sequential (conv2)         [32, 512, 7, 7]     [32, 512, 7, 7]     2,360,832          True
│    │    └─ReLU (relu)              [32, 512, 7, 7]     [32, 512, 7, 7]     --               --
├─AvgPool2d (avgpool)               [32, 512, 7, 7]     [32, 512, 1, 1]     --               --
├─Linear (fc)                    [32, 512]         [32, 200]         102,600          True
==============================================================================
==================================================
Total params: 11,283,912
Trainable params: 11,283,912
Non-trainable params: 0
Total mult-adds (G): 58.12
==============================================================================
==================================================
Input size (MB): 19.27
Forward/backward pass size (MB): 1271.71
Params size (MB): 45.14
Estimated Total Size (MB): 1336.12
```

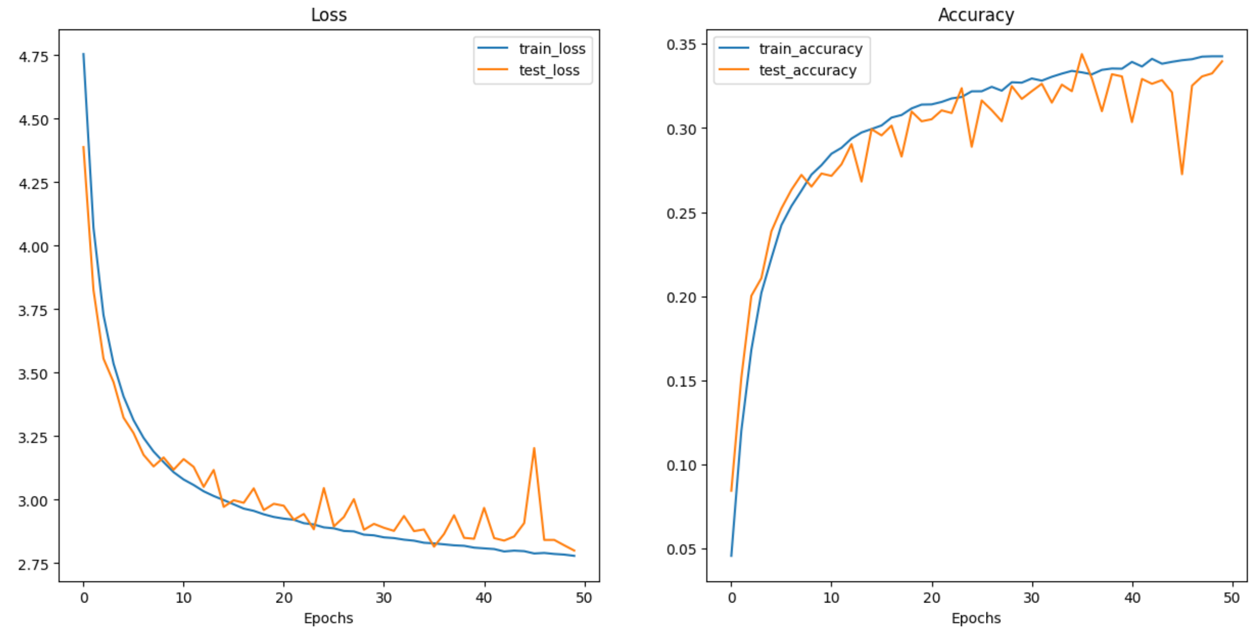
==
==

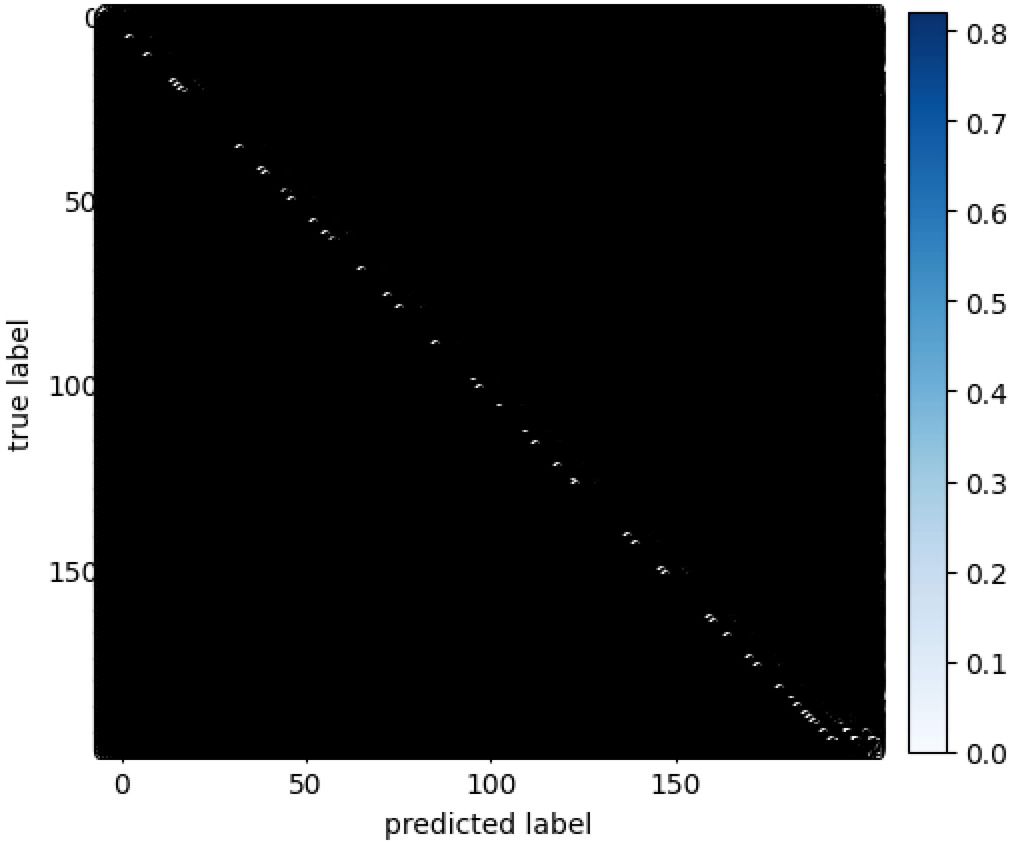
- Getting Dataset from the url with the help of wget.
- Applying Preprossing to make the val folder in the same directory structure of that of train directory.
- Applying transformation on the data set and making torch dataset object using Image Folder method.
- Visualization of the dataset random data samples.

- Converting the datasets to dataloader.
- Starting Model training.
- Defining utils functions like plot function , tain step, test step, training functions, etc.
- Writing Loss Function from Scratch for Cross Entropy loss.
- Setting up hyperparms for optimization and also setup accuracy function.
- Training model for 50 epochs.
- Results:-

```
Epoch: 1 | train_loss: 4.7549 | train_acc: 0.0458 | test_loss: 4.3887 | test_acc: 0.0845
Epoch: 2 | train_loss: 4.0711 | train_acc: 0.1201 | test_loss: 3.8258 | test_acc: 0.1516
Epoch: 3 | train_loss: 3.7269 | train_acc: 0.1681 | test_loss: 3.5554 | test_acc: 0.2003
Epoch: 4 | train_loss: 3.5361 | train_acc: 0.2021 | test_loss: 3.4631 | test_acc: 0.2108
Epoch: 5 | train_loss: 3.4069 | train_acc: 0.2226 | test_loss: 3.3233 | test_acc: 0.2386
Epoch: 6 | train_loss: 3.3134 | train_acc: 0.2423 | test_loss: 3.2625 | test_acc: 0.2522
Epoch: 7 | train_loss: 3.2438 | train_acc: 0.2536 | test_loss: 3.1768 | test_acc: 0.2633
Epoch: 8 | train_loss: 3.1898 | train_acc: 0.2628 | test_loss: 3.1311 | test_acc: 0.2722
Epoch: 9 | train_loss: 3.1482 | train_acc: 0.2723 | test_loss: 3.1666 | test_acc: 0.2652
Epoch: 10 | train_loss: 3.1085 | train_acc: 0.2779 | test_loss: 3.1184 | test_acc: 0.2730
Epoch: 11 | train_loss: 3.0796 | train_acc: 0.2847 | test_loss: 3.1599 | test_acc: 0.2716
Epoch: 12 | train_loss: 3.0580 | train_acc: 0.2882 | test_loss: 3.1293 | test_acc: 0.2785
Epoch: 13 | train_loss: 3.0332 | train_acc: 0.2937 | test_loss: 3.0507 | test_acc: 0.2904
Epoch: 14 | train_loss: 3.0147 | train_acc: 0.2974 | test_loss: 3.1173 | test_acc: 0.2682
Epoch: 15 | train_loss: 2.9987 | train_acc: 0.2994 | test_loss: 2.9719 | test_acc: 0.2993
Epoch: 16 | train_loss: 2.9820 | train_acc: 0.3015 | test_loss: 2.9981 | test_acc: 0.2956
Epoch: 17 | train_loss: 2.9651 | train_acc: 0.3062 | test_loss: 2.9879 | test_acc: 0.3014
Epoch: 18 | train_loss: 2.9562 | train_acc: 0.3077 | test_loss: 3.0449 | test_acc: 0.2830
Epoch: 19 | train_loss: 2.9426 | train_acc: 0.3117 | test_loss: 2.9593 | test_acc: 0.3098
Epoch: 20 | train_loss: 2.9322 | train_acc: 0.3139 | test_loss: 2.9840 | test_acc: 0.3040
Epoch: 21 | train_loss: 2.9256 | train_acc: 0.3140 | test_loss: 2.9763 | test_acc: 0.3052
Epoch: 22 | train_loss: 2.9212 | train_acc: 0.3155 | test_loss: 2.9210 | test_acc: 0.3105
Epoch: 23 | train_loss: 2.9076 | train_acc: 0.3176 | test_loss: 2.9441 | test_acc: 0.3089
Epoch: 24 | train_loss: 2.9026 | train_acc: 0.3184 | test_loss: 2.8836 | test_acc: 0.3237
Epoch: 25 | train_loss: 2.8912 | train_acc: 0.3218 | test_loss: 3.0460 | test_acc: 0.2888
Epoch: 26 | train_loss: 2.8874 | train_acc: 0.3218 | test_loss: 2.8951 | test_acc: 0.3164
Epoch: 27 | train_loss: 2.8772 | train_acc: 0.3245 | test_loss: 2.9320 | test_acc: 0.3106
Epoch: 28 | train_loss: 2.8753 | train_acc: 0.3222 | test_loss: 3.0024 | test_acc: 0.3040
Epoch: 29 | train_loss: 2.8625 | train_acc: 0.3272 | test_loss: 2.8818 | test_acc: 0.3249
Epoch: 30 | train_loss: 2.8598 | train_acc: 0.3270 | test_loss: 2.9050 | test_acc: 0.3173
Epoch: 31 | train_loss: 2.8519 | train_acc: 0.3295 | test_loss: 2.8895 | test_acc: 0.3219
Epoch: 32 | train_loss: 2.8490 | train_acc: 0.3281 | test_loss: 2.8774 | test_acc: 0.3263
Epoch: 33 | train_loss: 2.8428 | train_acc: 0.3305 | test_loss: 2.9361 | test_acc: 0.3151
Epoch: 34 | train_loss: 2.8388 | train_acc: 0.3323 | test_loss: 2.8764 | test_acc: 0.3258
Epoch: 35 | train_loss: 2.8307 | train_acc: 0.3340 | test_loss: 2.8830 | test_acc: 0.3219
Epoch: 36 | train_loss: 2.8281 | train_acc: 0.3331 | test_loss: 2.8153 | test_acc: 0.3439
Epoch: 37 | train_loss: 2.8244 | train_acc: 0.3319 | test_loss: 2.8652 | test_acc: 0.3295
```

```
Epoch: 38 | train_loss: 2.8205 | train_acc: 0.3346 | test_loss: 2.9392 | test_acc: 0.3100
Epoch: 39 | train_loss: 2.8186 | train_acc: 0.3354 | test_loss: 2.8500 | test_acc: 0.3320
Epoch: 40 | train_loss: 2.8114 | train_acc: 0.3352 | test_loss: 2.8463 | test_acc: 0.3307
Epoch: 41 | train_loss: 2.8085 | train_acc: 0.3393 | test_loss: 2.9680 | test_acc: 0.3035
Epoch: 42 | train_loss: 2.8059 | train_acc: 0.3366 | test_loss: 2.8490 | test_acc: 0.3292
Epoch: 43 | train_loss: 2.7963 | train_acc: 0.3411 | test_loss: 2.8392 | test_acc: 0.3263
Epoch: 44 | train_loss: 2.7996 | train_acc: 0.3382 | test_loss: 2.8554 | test_acc: 0.3285
Epoch: 45 | train_loss: 2.7975 | train_acc: 0.3393 | test_loss: 2.9081 | test_acc: 0.3212
Epoch: 46 | train_loss: 2.7884 | train_acc: 0.3403 | test_loss: 3.2035 | test_acc: 0.2726
Epoch: 47 | train_loss: 2.7905 | train_acc: 0.3409 | test_loss: 2.8417 | test_acc: 0.3252
Epoch: 48 | train_loss: 2.7865 | train_acc: 0.3424 | test_loss: 2.8418 | test_acc: 0.3307
Epoch: 49 | train_loss: 2.7840 | train_acc: 0.3426 | test_loss: 2.8205 | test_acc: 0.3325
Epoch: 50 | train_loss: 2.7791 | train_acc: 0.3426 | test_loss: 2.7999 | test_acc: 0.3397
total training time: 6284.433 sec.
LOSS & Accuracy Curves
lr: 0.001, betas: (0.9, 0.999), eps: 1e-08, weight_decay: 0.001
```




0
50
100
150
true label
0
50
100
150
predicted label
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

TRIPLET LOSS

- Import required packages. (PyTorch, NumPy, Matplotlib, torchmetrics …etc.)
- Checking GPU availability.
- Setting up devices.
- Defining Resnet 18 model from scratch.
- Getting model instance.
- Printing model summary.

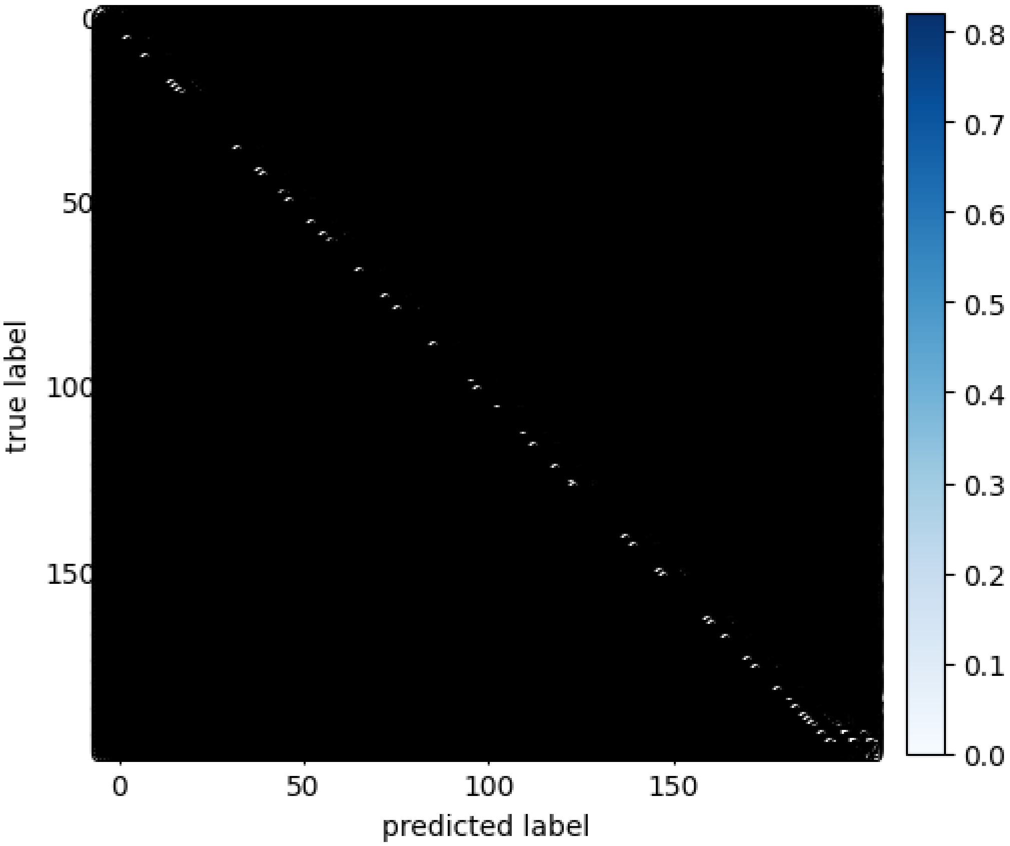
```
============================================================================
==================================================
Layer (type (var_name))                  Input Shape          Output Shape          Param #
Trainable
============================================================================
==================================================
ResNet (ResNet)                      [32, 3, 224, 224]    [32, 200]          --              True
├─Sequential (conv1)                 [32, 3, 224, 224]    [32, 64, 112, 112]   --              True
│    └─Conv2d (0)                    [32, 3, 224, 224]    [32, 64, 112, 112]   9,472            True
│    └─BatchNorm2d (1)               [32, 64, 112, 112]   [32, 64, 112, 112]   128              True
│    └─ReLU (2)                      [32, 64, 112, 112]   [32, 64, 112, 112]   --               --
├─MaxPool2d (maxpool)                [32, 64, 112, 112]   [32, 64, 56, 56]     --               --
├─Sequential (layer0)                [32, 64, 56, 56]     [32, 64, 56, 56]     --              True
│    └─ResidualBlock (0)             [32, 64, 56, 56]     [32, 64, 56, 56]     --              True
│    │    └─Sequential (conv1)       [32, 64, 56, 56]     [32, 64, 56, 56]     37,056           True
│    │    └─Sequential (conv2)       [32, 64, 56, 56]     [32, 64, 56, 56]     37,056           True
│    │    └─ReLU (relu)              [32, 64, 56, 56]     [32, 64, 56, 56]     --               --
│    └─ResidualBlock (1)             [32, 64, 56, 56]     [32, 64, 56, 56]     --              True
│    │    └─Sequential (conv1)       [32, 64, 56, 56]     [32, 64, 56, 56]     37,056           True
│    │    └─Sequential (conv2)       [32, 64, 56, 56]     [32, 64, 56, 56]     37,056           True
│    │    └─ReLU (relu)              [32, 64, 56, 56]     [32, 64, 56, 56]     --               --
├─Sequential (layer1)                [32, 64, 56, 56]     [32, 128, 28, 28]    --              True
│    └─ResidualBlock (0)             [32, 64, 56, 56]     [32, 128, 28, 28]    --              True
│    │    └─Sequential (conv1)       [32, 64, 56, 56]     [32, 128, 28, 28]    74,112           True
│    │    └─Sequential (conv2)       [32, 128, 28, 28]    [32, 128, 28, 28]    147,840          True
│    │        └─Sequential (downsample)        [32, 64, 56, 56]       [32, 128, 28, 28]     8,576
True
│    │    └─ReLU (relu)              [32, 128, 28, 28]    [32, 128, 28, 28]    --               --
│    └─ResidualBlock (1)             [32, 128, 28, 28]    [32, 128, 28, 28]    --              True
│    │    └─Sequential (conv1)       [32, 128, 28, 28]    [32, 128, 28, 28]    147,840          True
│    │    └─Sequential (conv2)       [32, 128, 28, 28]    [32, 128, 28, 28]    147,840          True
│    │    └─ReLU (relu)              [32, 128, 28, 28]    [32, 128, 28, 28]    --               --
├─Sequential (layer2)                [32, 128, 28, 28]    [32, 256, 14, 14]    --              True
│    └─ResidualBlock (0)             [32, 128, 28, 28]    [32, 256, 14, 14]    --              True
│    │    └─Sequential (conv1)       [32, 128, 28, 28]    [32, 256, 14, 14]    295,680          True
│    │    └─Sequential (conv2)       [32, 256, 14, 14]    [32, 256, 14, 14]    590,592          True
│    │     └─Sequential (downsample)       [32, 128, 28, 28]    [32, 256, 14, 14]    33,536
True
│   │   └─ReLU (relu)              [32, 256, 14, 14]   [32, 256, 14, 14]   --              --
│   └─ResidualBlock (1)             [32, 256, 14, 14]   [32, 256, 14, 14]   --              True
│   │   └─Sequential (conv1)         [32, 256, 14, 14]   [32, 256, 14, 14]   590,592          True
│   │   └─Sequential (conv2)         [32, 256, 14, 14]   [32, 256, 14, 14]   590,592          True
│   │   └─ReLU (relu)              [32, 256, 14, 14]   [32, 256, 14, 14]   --              --
├─Sequential (layer3)              [32, 256, 14, 14]   [32, 512, 7, 7]     --              True
│   └─ResidualBlock (0)             [32, 256, 14, 14]   [32, 512, 7, 7]     --              True
│   │   └─Sequential (conv1)         [32, 256, 14, 14]   [32, 512, 7, 7]     1,181,184         True
│   │   └─Sequential (conv2)         [32, 512, 7, 7]     [32, 512, 7, 7]     2,360,832         True
│    │     └─Sequential (downsample)       [32, 256, 14, 14]    [32, 512, 7, 7]      132,608
True
│   │   └─ReLU (relu)              [32, 512, 7, 7]     [32, 512, 7, 7]     --              --
│   └─ResidualBlock (1)             [32, 512, 7, 7]     [32, 512, 7, 7]     --              True
│   │   └─Sequential (conv1)         [32, 512, 7, 7]     [32, 512, 7, 7]     2,360,832         True
│   │   └─Sequential (conv2)         [32, 512, 7, 7]     [32, 512, 7, 7]     2,360,832         True
│   │   └─ReLU (relu)              [32, 512, 7, 7]     [32, 512, 7, 7]     --              --
├─AvgPool2d (avgpool)               [32, 512, 7, 7]     [32, 512, 1, 1]     --              --
├─Linear (fc)                    [32, 512]          [32, 200]          102,600          True
==========================================================================
==================================================
Total params: 11,283,912
Trainable params: 11,283,912
Non-trainable params: 0
Total mult-adds (G): 58.12
==========================================================================
==================================================
Input size (MB): 19.27
Forward/backward pass size (MB): 1271.71
Params size (MB): 45.14
Estimated Total Size (MB): 1336.12
==========================================================================
==================================================
```

- Getting Dataset from the url with the help of wget.
- Applying Preprossing to make the val folder in the same directory structure of that of train directory.
- Applying transformation on the data set and making torch dataset object using Image Folder method.
- Visualization of the dataset random data samples.

- Converting the datasets to dataloader.
- Starting Model training.
- Defining utils functions like plot function , tain step, test step, training functions, etc.
- Writing Loss Function from Scratch for Cross Entropy loss.
- Setting up hyperparms for optimization and also setup accuracy function.
- Training model for 5 Epochs.
- Results:-

```
Epochs:   0%|          | 0/5 [00:00<?, ?it/s]Training:   0%|          | 0/3125 [00:00<?, ?it/s]Epoch: 1/5
— Loss: 0.8899
Training:   0%|          | 0/3125 [00:01<?, ?it/s]Epoch: 2/5 — Loss: 0.8478
```

```
Training:   0%|          | 0/3125 [00:01<?, ?it/s]Epoch: 3/5 — Loss: 0.8146
Training:   0%|          | 0/3125 [00:01<?, ?it/s]Epoch: 4/5 — Loss: 0.7932
Training:   0%|          | 0/3125 [00:01<?, ?it/s]Epoch: 5/5 — Loss: 0.7658
```



CENTER LOSS

- Import required packages. (PyTorch, NumPy, Matplotlib, torchmetrics …etc.)
- Checking GPU availability.
- Setting up devices.
- Defining Resnet 18 model from scratch.
- Getting model instance.
- Printing model summary.

```
==============================================================================
==================================================
Layout (type (var_name))                 Input Shape          Output Shape         Param #
Trainable
==============================================================================
==================================================
ResNet (ResNet)                 [32, 3, 224, 224]   [32, 200]         --              True
├─Sequential (conv1)             [32, 3, 224, 224]   [32, 64, 112, 112]  --              True
│    └─Conv2d (0)               [32, 3, 224, 224]   [32, 64, 112, 112]  9,472            True
│    └─BatchNorm2d (1)            [32, 64, 112, 112]  [32, 64, 112, 112]  128              True
│    └─ReLU (2)                [32, 64, 112, 112]  [32, 64, 112, 112]  --               --
├─MaxPool2d (maxpool)            [32, 64, 112, 112]  [32, 64, 56, 56]    --               --
├─Sequential (layer0)            [32, 64, 56, 56]    [32, 64, 56, 56]    --              True
│    └─ResidualBlock (0)          [32, 64, 56, 56]    [32, 64, 56, 56]    --              True
│    │    └─Sequential (conv1)        [32, 64, 56, 56]    [32, 64, 56, 56]    37,056           True
│    │    └─Sequential (conv2)        [32, 64, 56, 56]    [32, 64, 56, 56]    37,056           True
│    │    └─ReLU (relu)          [32, 64, 56, 56]    [32, 64, 56, 56]    --               --
│    └─ResidualBlock (1)          [32, 64, 56, 56]    [32, 64, 56, 56]    --              True
│    │    └─Sequential (conv1)        [32, 64, 56, 56]    [32, 64, 56, 56]    37,056           True
│    │    └─Sequential (conv2)        [32, 64, 56, 56]    [32, 64, 56, 56]    37,056           True
│    │    └─ReLU (relu)          [32, 64, 56, 56]    [32, 64, 56, 56]    --               --
├─Sequential (layer1)            [32, 64, 56, 56]    [32, 128, 28, 28]   --              True
│    └─ResidualBlock (0)          [32, 64, 56, 56]    [32, 128, 28, 28]   --              True
│    │    └─Sequential (conv1)        [32, 64, 56, 56]    [32, 128, 28, 28]   74,112           True
│    │    └─Sequential (conv2)        [32, 128, 28, 28]   [32, 128, 28, 28]   147,840          True
│     │      └─Sequential (downsample)      [32, 64, 56, 56]      [32, 128, 28, 28]     8,576
True
│    │    └─ReLU (relu)          [32, 128, 28, 28]   [32, 128, 28, 28]   --               --
│    └─ResidualBlock (1)          [32, 128, 28, 28]   [32, 128, 28, 28]   --              True
│    │    └─Sequential (conv1)        [32, 128, 28, 28]   [32, 128, 28, 28]   147,840          True
│    │    └─Sequential (conv2)        [32, 128, 28, 28]   [32, 128, 28, 28]   147,840          True
│    │    └─ReLU (relu)          [32, 128, 28, 28]   [32, 128, 28, 28]   --               --
├─Sequential (layer2)            [32, 128, 28, 28]   [32, 256, 14, 14]   --              True
│    └─ResidualBlock (0)          [32, 128, 28, 28]   [32, 256, 14, 14]   --              True
│    │    └─Sequential (conv1)        [32, 128, 28, 28]   [32, 256, 14, 14]   295,680          True
│    │    └─Sequential (conv2)        [32, 256, 14, 14]   [32, 256, 14, 14]   590,592          True
│    │    └─Sequential (downsample)      [32, 128, 28, 28]    [32, 256, 14, 14]    33,536
True
│    │    └─ReLU (relu)           [32, 256, 14, 14]   [32, 256, 14, 14]   --          --
│    └─ResidualBlock (1)          [32, 256, 14, 14]   [32, 256, 14, 14]   --          True
│    │    └─Sequential (conv1)       [32, 256, 14, 14]   [32, 256, 14, 14]   590,592        True
│    │    └─Sequential (conv2)       [32, 256, 14, 14]   [32, 256, 14, 14]   590,592        True
│    │    └─ReLU (relu)           [32, 256, 14, 14]   [32, 256, 14, 14]   --          --
├─Sequential (layer3)            [32, 256, 14, 14]   [32, 512, 7, 7]    --          True
│    └─ResidualBlock (0)          [32, 256, 14, 14]   [32, 512, 7, 7]    --          True
│    │    └─Sequential (conv1)       [32, 256, 14, 14]   [32, 512, 7, 7]    1,181,184       True
│    │    └─Sequential (conv2)       [32, 512, 7, 7]    [32, 512, 7, 7]    2,360,832       True
│    │    └─Sequential (downsample)      [32, 256, 14, 14]    [32, 512, 7, 7]      132,608
True
│    │    └─ReLU (relu)           [32, 512, 7, 7]    [32, 512, 7, 7]    --          --
│    └─ResidualBlock (1)          [32, 512, 7, 7]    [32, 512, 7, 7]    --          True
│    │    └─Sequential (conv1)       [32, 512, 7, 7]    [32, 512, 7, 7]    2,360,832       True
│    │    └─Sequential (conv2)       [32, 512, 7, 7]    [32, 512, 7, 7]    2,360,832       True
│    │    └─ReLU (relu)           [32, 512, 7, 7]    [32, 512, 7, 7]    --          --
├─AvgPool2d (avgpool)            [32, 512, 7, 7]    [32, 512, 1, 1]    --          --
├─Linear (fc)               [32, 512]       [32, 200]       102,600       True
==================================================================================
==================================================
Total params: 11,283,912
Trainable params: 11,283,912
Non-trainable params: 0
Total mult-adds (G): 58.12
==================================================================================
==================================================
Input size (MB): 19.27
Forward/backward pass size (MB): 1271.71
Params size (MB): 45.14
Estimated Total Size (MB): 1336.12
==================================================================================
==================================================
```

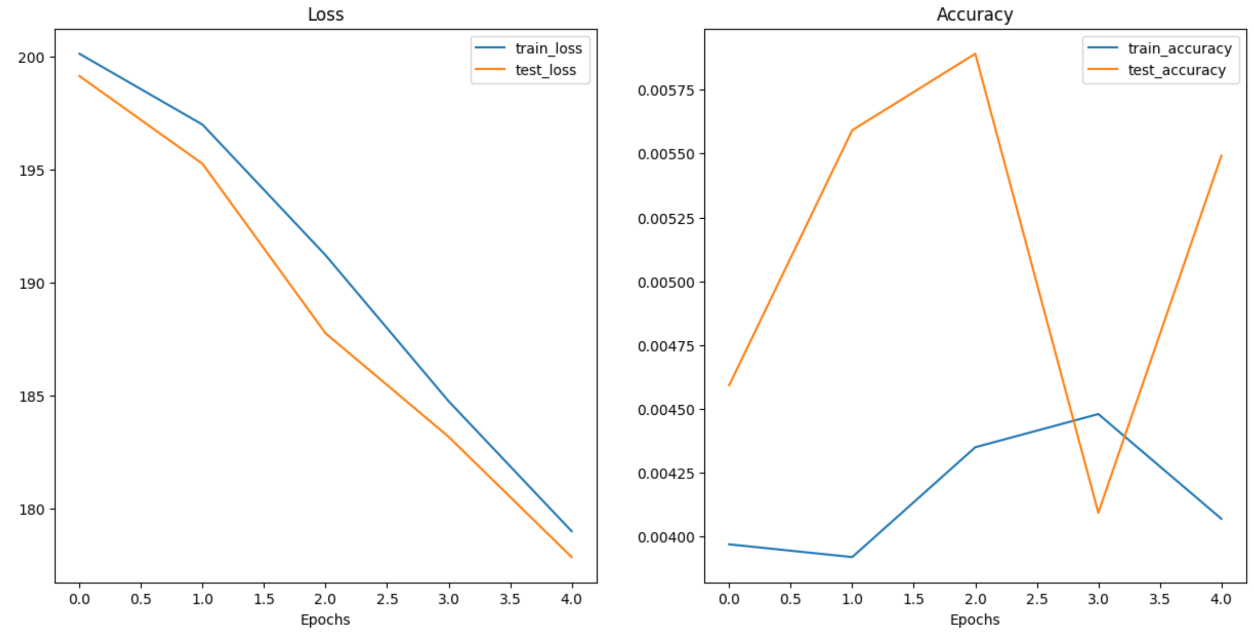
- Getting Dataset from the url with the help of wget.
- Applying Preprossing to make the val folder in the same directory structure of that of train directory.
- Applying transformation on the data set and making torch dataset object using Image Folder method.
- Visualization of the dataset random data samples.

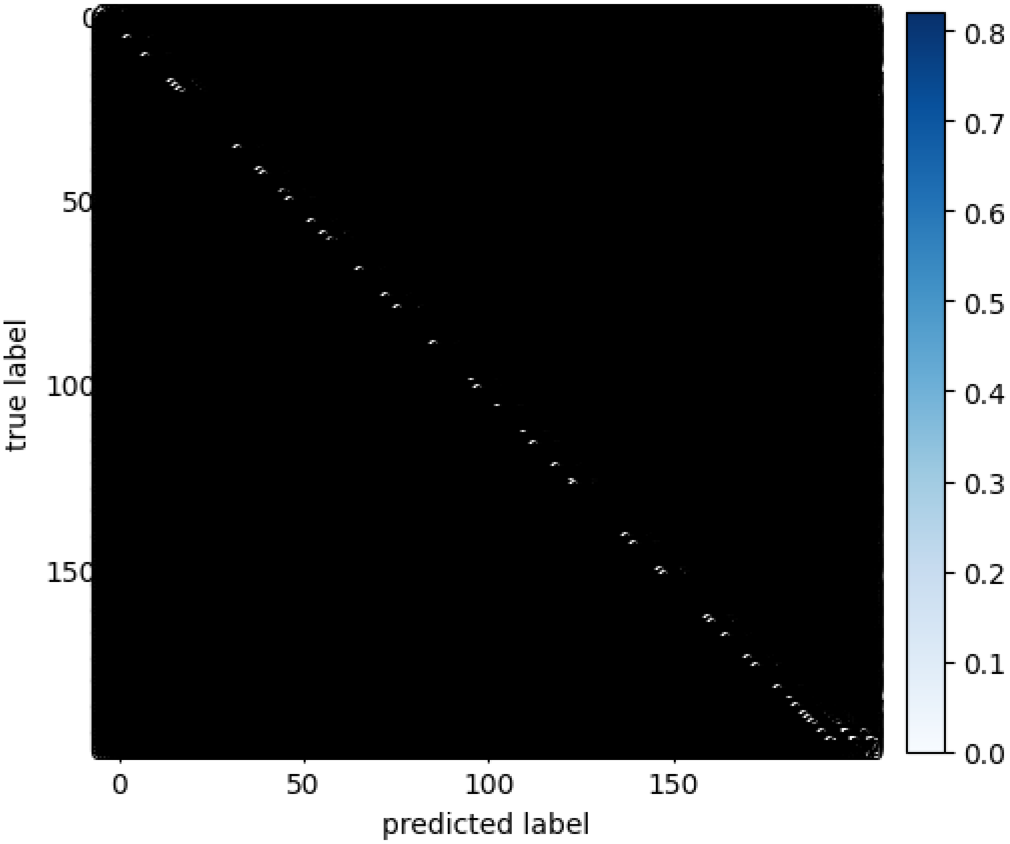
- Converting the datasets to dataloader.
- Starting Model training.
- Defining utils functions like plot function , tain step, test step, training functions, etc.
- Writing Loss Function from Scratch for Cross Entropy loss.
- Setting up hyperparms for optimization and also setup accuracy function.
- Training model for 5 Epochs.
- Results:-

current exp / total Expirements: 1 / 1

Training with: lr: 0.001, betas: (0.9, 0.999), eps: 1e-08, weight_decay: 0.001

```
  0%|             | 0/5 [00:00<?, ?it/s]Training::   0%|            | 0/3125 [00:01<?, ?it/s]Epoch: 1 |
train_loss: 200.1284 | train_acc: 0.0040 | test_loss: 199.1399 | test_acc: 0.0046
Training::   0%|          | 0/3125 [00:01<?, ?it/s]
Epoch: 2 | train_loss: 196.9904 | train_acc: 0.0039 | test_loss: 195.2539 | test_acc: 0.0056
Training::   0%|          | 0/3125 [00:01<?, ?it/s]Epoch: 3 | train_loss: 191.2061 | train_acc: 0.0043
| test_loss: 187.7480 | test_acc: 0.0059
Training::   0%|          | 0/3125 [00:01<?, ?it/s]Epoch: 4 | train_loss: 184.7402 | train_acc: 0.0045
| test_loss: 183.1657 | test_acc: 0.0041
Training::   0%|          | 0/3125 [00:01<?, ?it/s]Epoch: 5 | train_loss: 178.9814 | train_acc: 0.0041
| test_loss: 177.8362 | test_acc: 0.0055
total training time: 1352.498 sec.
LOSS & Accuracy Curves
lr: 0.001, betas: (0.9, 0.999), eps: 1e-08, weight_decay: 0.001
```





0
50
100
150
true label
0
50
100
150
predicted label
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

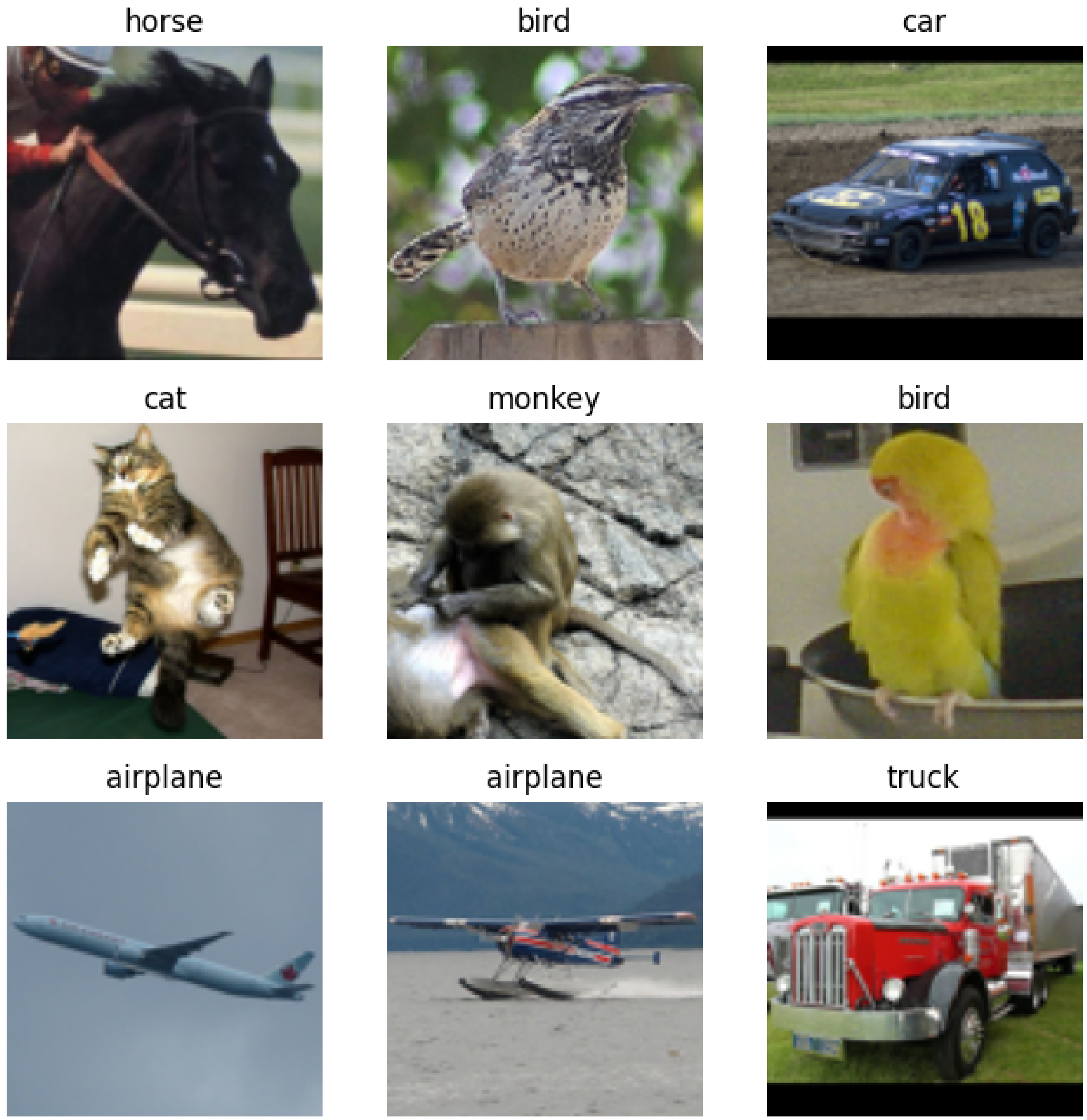
Objectives:

Question 2.

- Train layerwise Auto Encoder on the SHL-10 dataset.

Procedure:

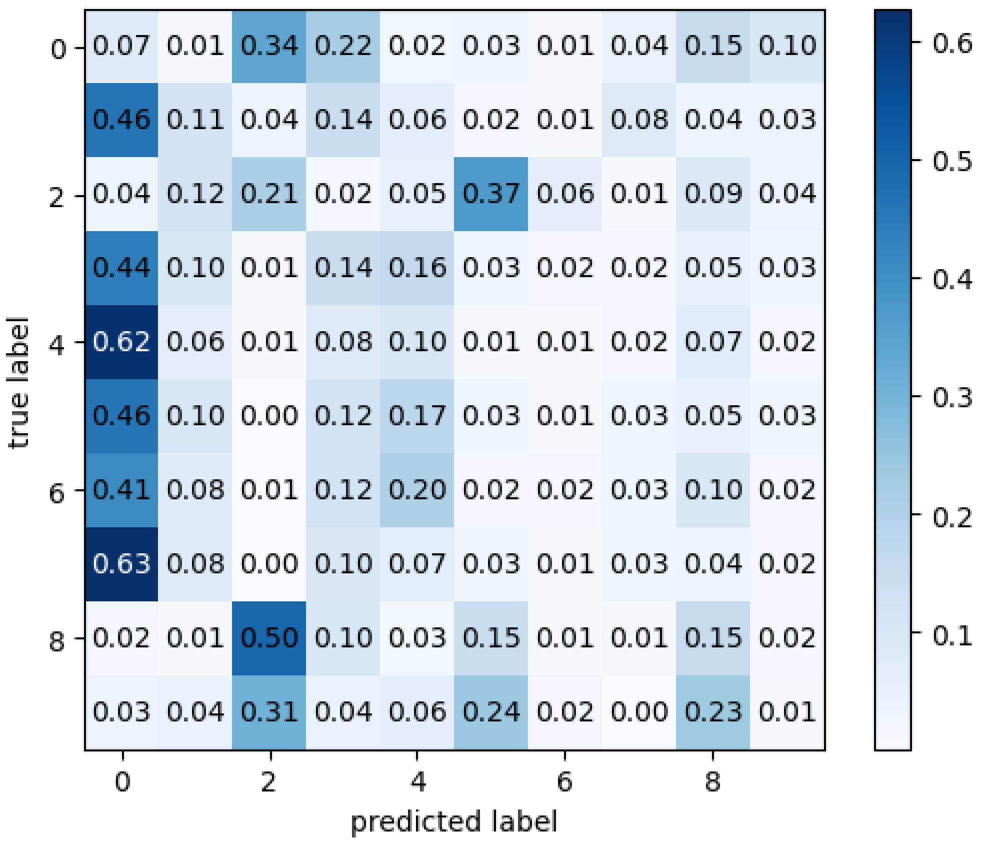
- Import required packages. (PyTorch, NumPy, Matplotlib, torchmetrics …etc.)
- Checking GPU availability.
- Setting up devices.
- Downloading dataset with torch vision.
- Visualization of the dataset random data samples.



- Converting dataset to dataloaders.
- Starting Model training.
- Defining utils functions like plot function , tain step, test step, training functions, etc.
- Writing Loss Function from Scratch for Cross Entropy loss.
- Setting up hyperparms for optimization and also setup accuracy function.
- Training model for 5 Epochs for each layers.
- Results:-

```
========================================================================
============================
>>> TRAINING THE BASE MODEL:
Files already downloaded and verified
Starting epoch 1
Starting epoch 2
Starting epoch 3
Starting epoch 4
Starting epoch 5
==================================================
Old structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=10, bias=True)
)
LOG:  Linear(in_features=1024, out_features=10, bias=True)
==================================================
New structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=10, bias=True)
)
========================================================================
============================
>>> TRAINING THE MODEL WITH 1 ADDITIONAL LAYERS:
Files already downloaded and verified
Starting epoch 1
Starting epoch 2
Starting epoch 3
Starting epoch 4
Starting epoch 5
==================================================
>>> RESULTS: Adding this layer did not improve the model loss.
=======================================================
Old structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=10, bias=True)
)
=======================================================
New structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=1200, bias=True)
  (5): Linear(in_features=1200, out_features=10, bias=True)
)
================================================================================
==============================
>>> TRAINING THE MODEL WITH 2 ADDITIONAL LAYERS:
Files already downloaded and verified
Starting epoch 1
Starting epoch 2
Starting epoch 3
Starting epoch 4
Starting epoch 5
=======================================================
>>> RESULTS: Adding this layer did not improve the model loss.
=======================================================
Old structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=1200, bias=True)
  (5): Linear(in_features=1200, out_features=10, bias=True)
)
=======================================================
New structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=1200, bias=True)
  (5): Linear(in_features=1200, out_features=728, bias=True)
  (6): Linear(in_features=728, out_features=10, bias=True)
)
===============================================================================
===============================
>>> TRAINING THE MODEL WITH 3 ADDITIONAL LAYERS:
Files already downloaded and verified
Starting epoch 1
Starting epoch 2
Starting epoch 3
Starting epoch 4
Starting epoch 5
=======================================================
>>> RESULTS: Adding this layer did not improve the model loss.
=======================================================
Old structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=1200, bias=True)
  (5): Linear(in_features=1200, out_features=728, bias=True)
  (6): Linear(in_features=728, out_features=10, bias=True)
)
=======================================================
New structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=1200, bias=True)
  (5): Linear(in_features=1200, out_features=728, bias=True)
  (6): Linear(in_features=728, out_features=512, bias=True)
  (7): Linear(in_features=512, out_features=10, bias=True)
)
===============================================================================
===============================
>>> TRAINING THE MODEL WITH 4 ADDITIONAL LAYERS:
Files already downloaded and verified
Starting epoch 1
Starting epoch 2
Starting epoch 3
Starting epoch 4
Starting epoch 5
==================================================
>>> RESULTS: Adding this layer did not improve the model loss.
==================================================
Old structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=1200, bias=True)
  (5): Linear(in_features=1200, out_features=728, bias=True)
  (6): Linear(in_features=728, out_features=512, bias=True)
  (7): Linear(in_features=512, out_features=10, bias=True)
)
==================================================
New structure:
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=27648, out_features=1024, bias=True)
  (2): ReLU()
  (3): Linear(in_features=1024, out_features=1024, bias=True)
  (4): Linear(in_features=1024, out_features=1200, bias=True)
  (5): Linear(in_features=1200, out_features=728, bias=True)
  (6): Linear(in_features=728, out_features=512, bias=True)
  (7): Linear(in_features=512, out_features=128, bias=True)
  (8): Linear(in_features=128, out_features=10, bias=True)
)
Training process has finished.

Confusion Matrix:
```



Refrences:-

- https://pytorch.org/docs/stable/
- http://rasbt.github.io/mlxtend/user_guide/plotting/plot_confusion_matrix/
- https://huggingface.co/course/chapter1/1?fw=pt
- https://github.com/christianversloot/machine-learning-articles
- https://github.com/KinWaiCheuk/pytorch-triplet-loss/blob/master/MNIST%20online%20mining%20hard.ipynb
- https://www.kaggle.com/code/hirotaka0122/triplet-loss-with-pytorch/notebook
- https://medium.com/@Skpd/triplet-loss-on-imagenet-dataset-a2b29b8c2952
- https://ydwen.github.io/papers/WenECCV16.pdf
- https://pypi.org/project/accelerate/
- https://drive.google.com/file/d/1e2tFV8ftj_171J7J8k0b32ZUdWZWSliz/view
- https://stackoverflow.com/questions/tagged/pytorch