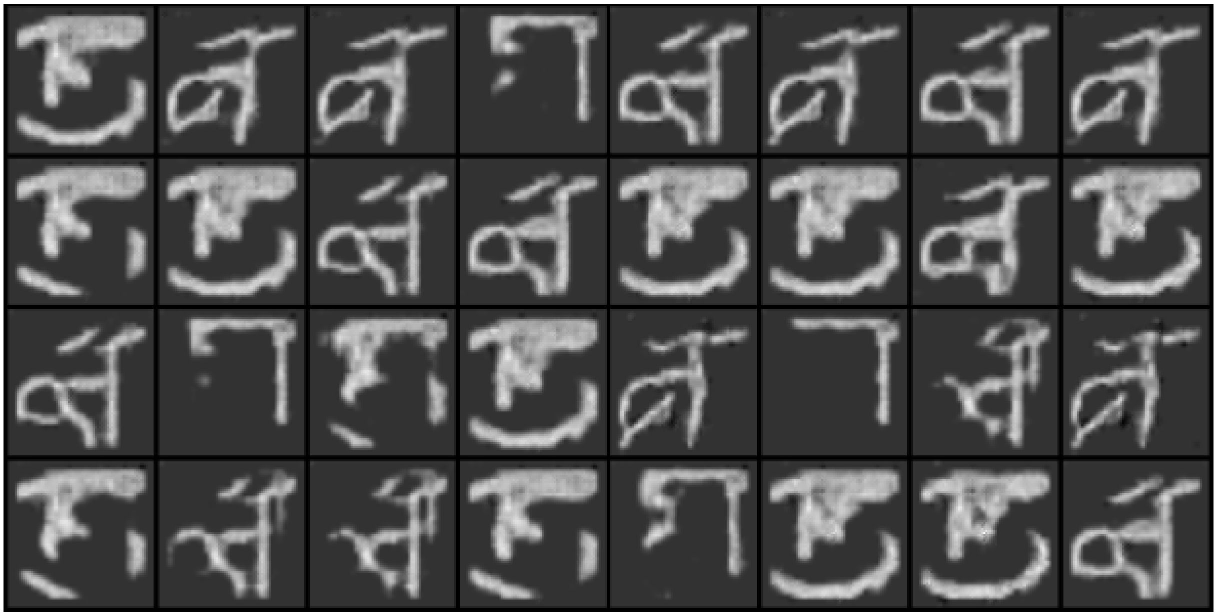
DLOps Assignment: 4

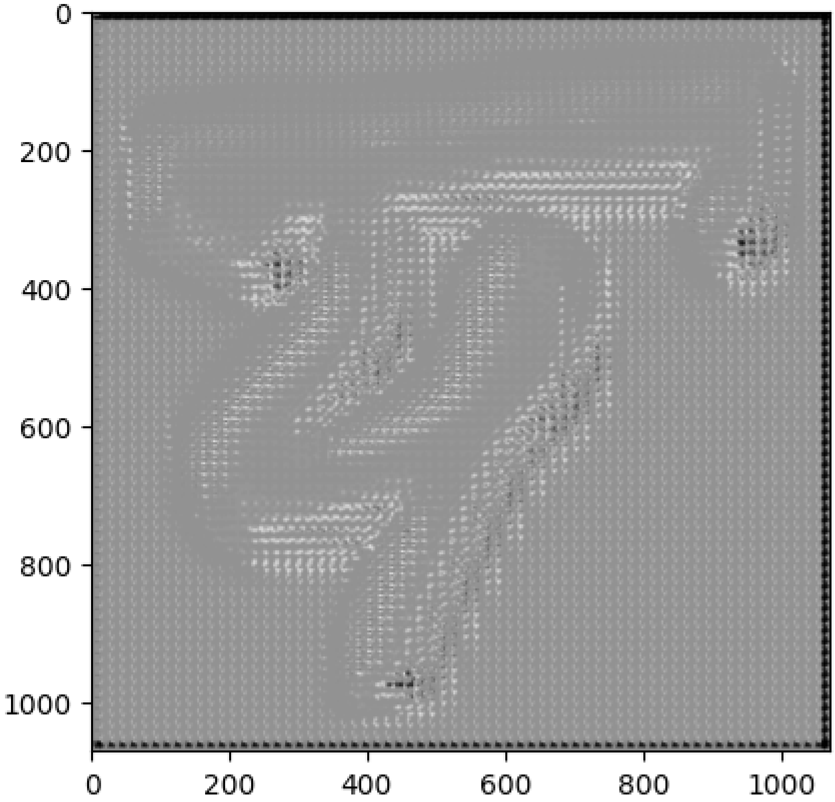
Question 1:

- Import required libraries and write device agnostic code
- Load datasets and make dataloader of 32 batch size
- Write train and test loop for resnet18
- Set hyperparameter for DCGAN and make function for weight initialization
- Now write model for generator and discriminator
- Make instance of both and train generator and discriminator
- Now train for it 10 epoch and note the result
- Now make generated image dataset and take it as new test dataset
- Now take resnet18 model and make prediction and note f1 score and confusion matrix

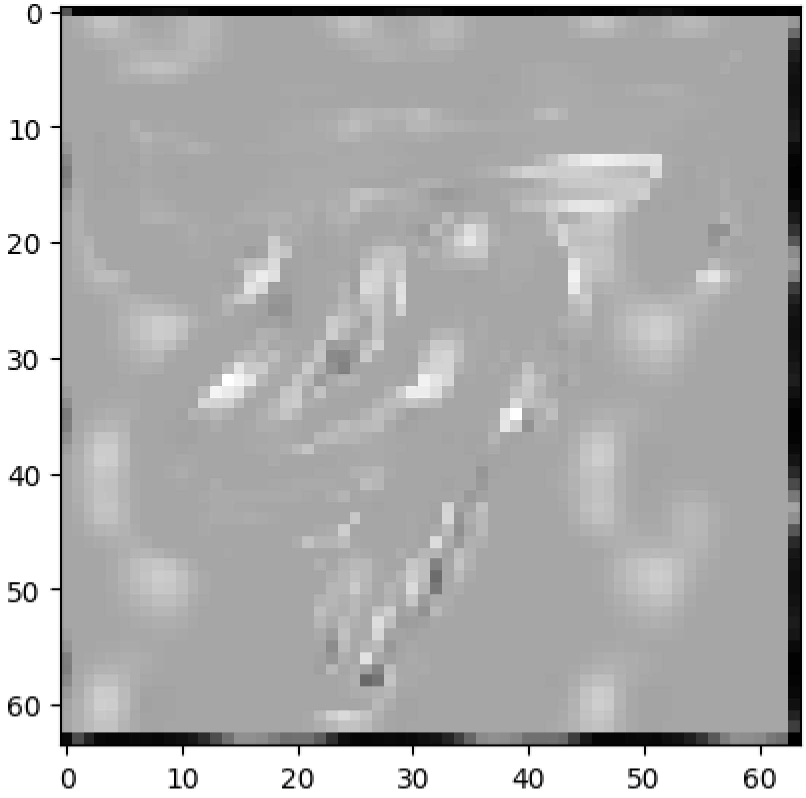
Result: For 64 epoch of DCGAN



Generated images



Solo generated image in 1172x1172 size



Solo generated image in 64x64 size

Train on Resnet18

```
Epoch: 1
Train Loss: 0.3183 | Test Loss: 5.3071 | Train Accuray: 0.4655 | Test Accuracy: 0.0005

Epoch: 2
Train Loss: 0.1198 | Test Loss: 5.2319 | Train Accuray: 0.4906 | Test Accuracy: 0.0005

Epoch: 3
Train Loss: 0.0855 | Test Loss: 9.4789 | Train Accuray: 0.4958 | Test Accuracy: 0.0005

Epoch: 4
Train Loss: 0.0714 | Test Loss: 12.0705 | Train Accuray: 0.4975 | Test Accuracy: 0.0005

Epoch: 5
Train Loss: 0.0541 | Test Loss: 9.3631 | Train Accuray: 0.4987 | Test Accuracy: 0.0005

Epoch: 6
Train Loss: 0.0440 | Test Loss: 6.9710 | Train Accuray: 0.5013 | Test Accuracy: 0.0005

Epoch: 7
Train Loss: 0.0400 | Test Loss: 12.9920 | Train Accuray: 0.5005 | Test Accuracy: 0.0005
```

Epoch: 8
Train Loss: 0.0311 | Test Loss: 10.0010 | Train Accuray: 0.5023 | Test Accuracy: 0.0005

Epoch: 9
Train Loss: 0.0290 | Test Loss: 16.1579 | Train Accuray: 0.5016 | Test Accuracy: 0.0005

Epoch: 10
Train Loss: 0.0249 | Test Loss: 6.7384 | Train Accuray: 0.5026 | Test Accuracy: 0.0005

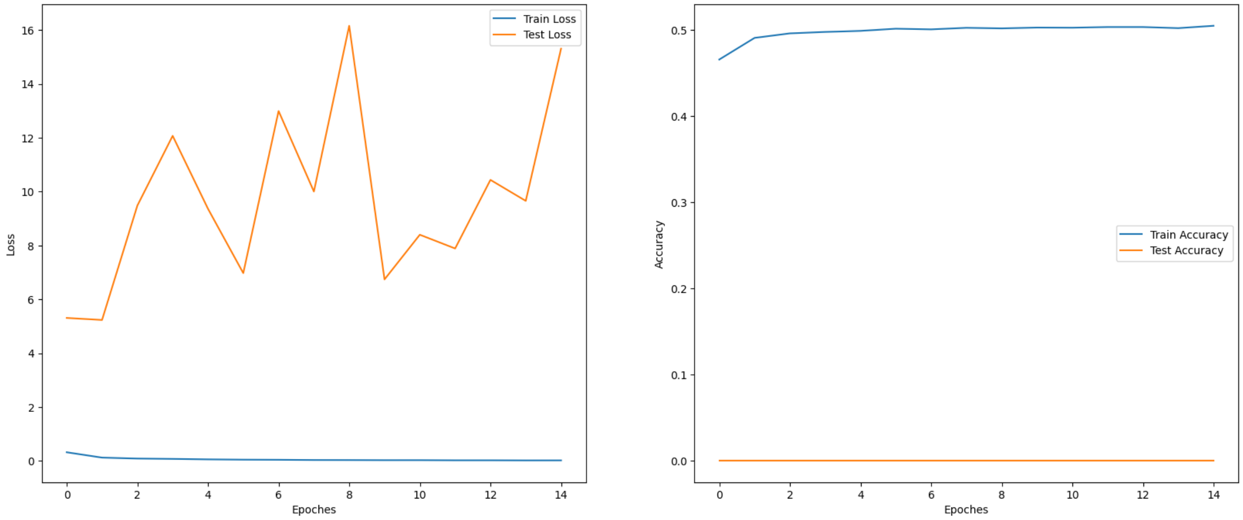
Epoch: 11
Train Loss: 0.0254 | Test Loss: 8.3974 | Train Accuray: 0.5024 | Test Accuracy: 0.0005

Epoch: 12
Train Loss: 0.0196 | Test Loss: 7.8866 | Train Accuray: 0.5032 | Test Accuracy: 0.0005

Epoch: 13
Train Loss: 0.0198 | Test Loss: 10.4347 | Train Accuray: 0.5032 | Test Accuracy: 0.0005

Epoch: 14
Train Loss: 0.0157 | Test Loss: 9.6549 | Train Accuray: 0.5019 | Test Accuracy: 0.0005

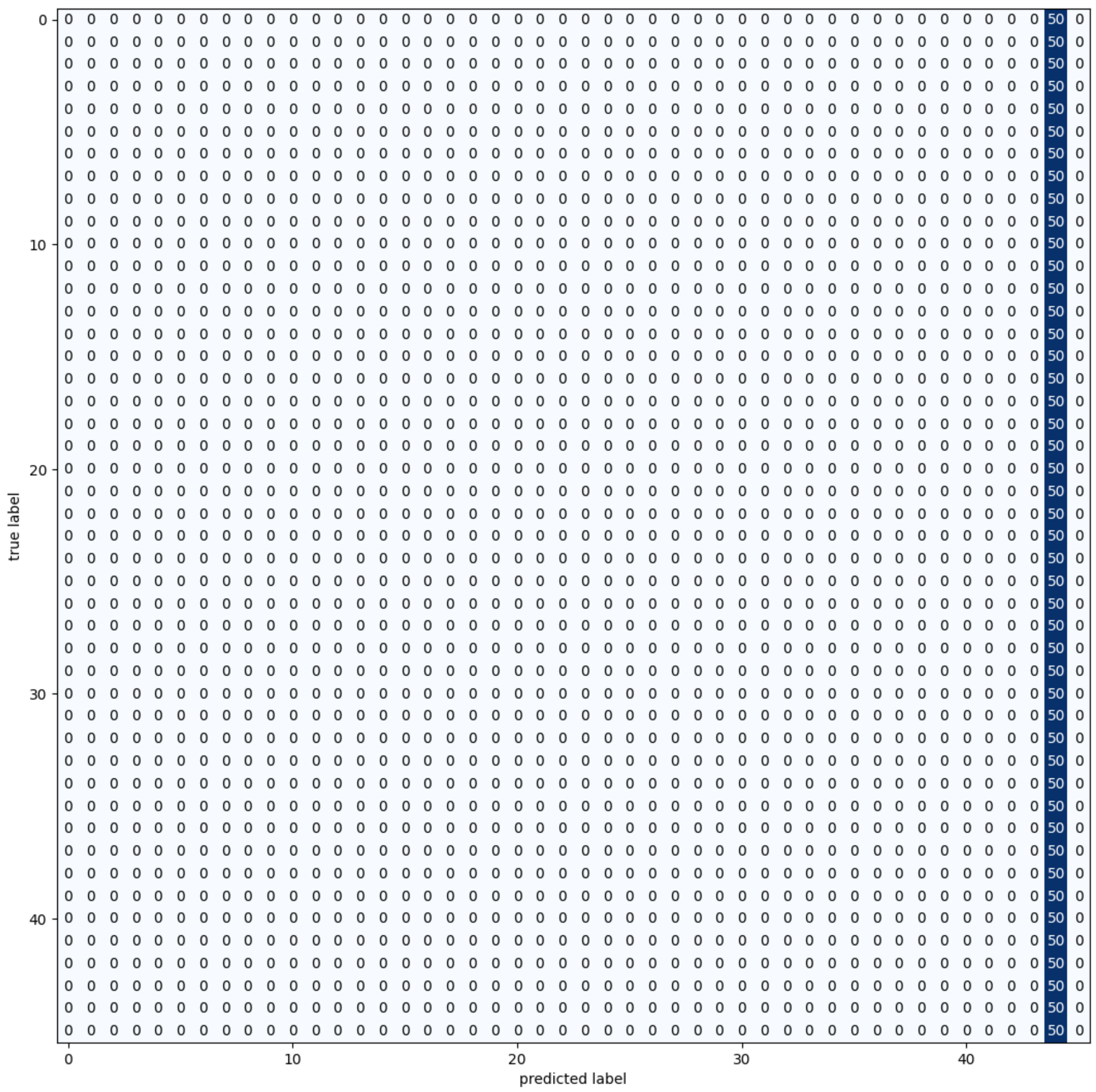
Epoch: 15
Train Loss: 0.0167 | Test Loss: 15.3086 | Train Accuray: 0.5047 | Test Accuracy: 0.0005



Execution time: 770.1250406219988 Seconds.

F1 Score: `tensor(0.0009, device='cuda:0')`

Confusion matrix:



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0

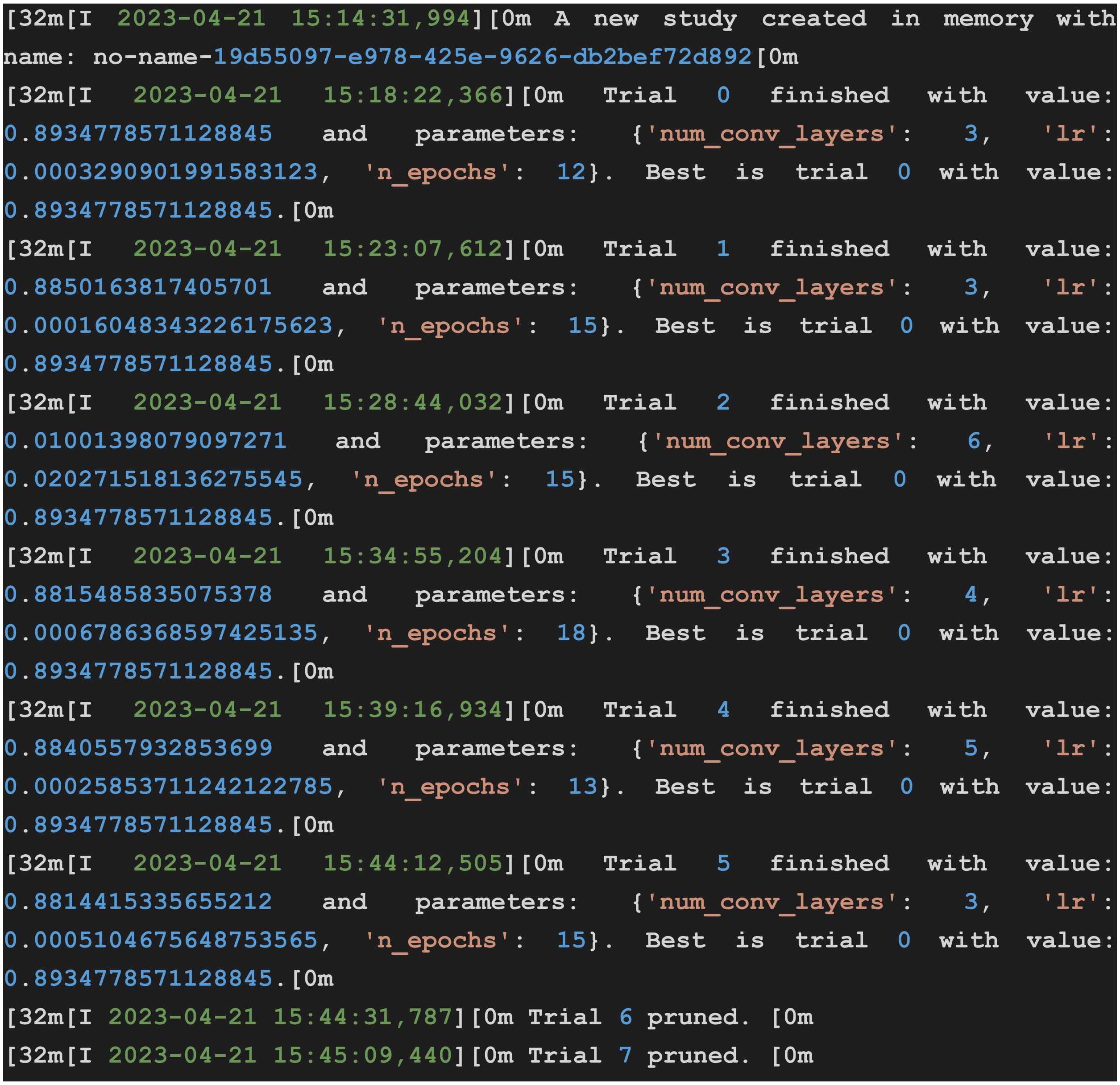
Observation:

- It gives poor performance on generated images.
- But I train it on 10 epochs and 64 epochs respectively on DCGAN. However 64 epochs DCGAN generates give good performance compared to 10 epochs DCGAN.
- Also model which train on more datasets give good performance compared to less train datasets.
- Therefore forbetter Generated image we need to train DCGAN for mode number of epochs as possible.

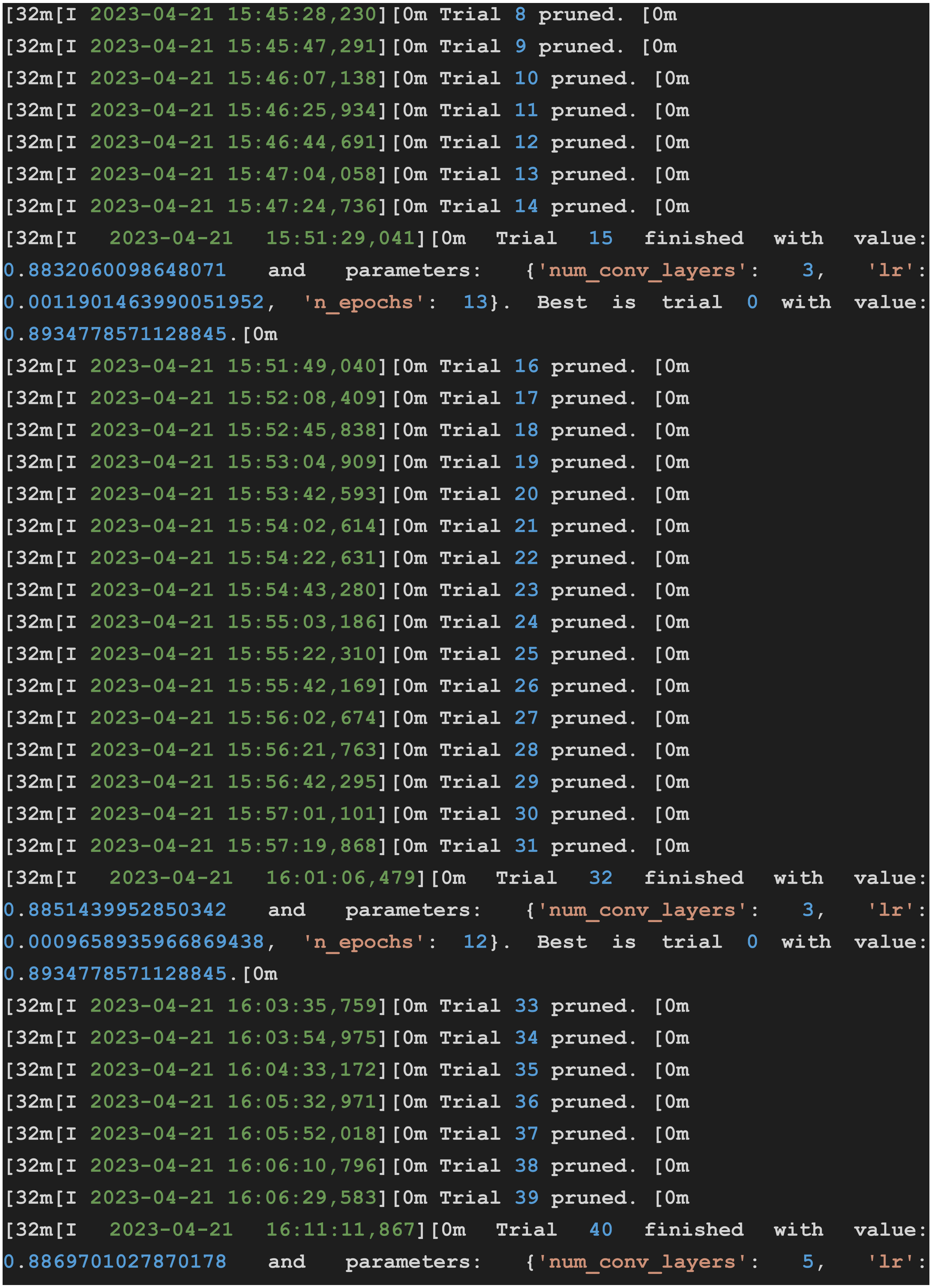
Question 2:

- Import required libraries and write device agnostic code
- Load datasets and make dataloader of 32 batch size
- Write train and test loop for custom model
- Make custom model of cnn and make objective function using optuna
- Hyperparameters
    - Take 0.0001 to 0.1 as learning rate
    - Take 3 to 6 CNN layer as architecture
    - Take 10 to 20 epochs (20 because on slurm and colab gpu time expire on 50 and 25 epochs)
- Now tune the hyperparameters using optuna and mention result

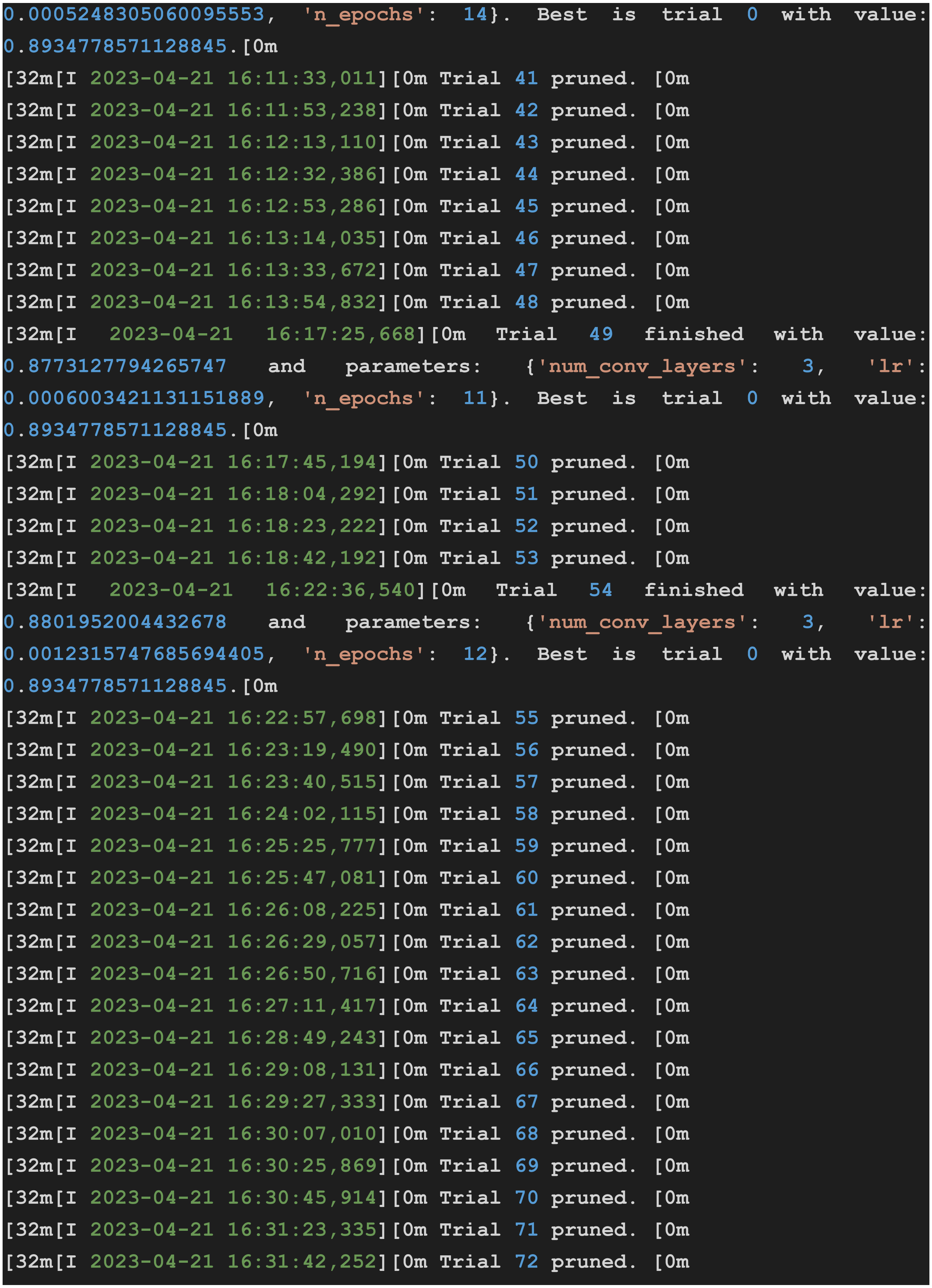
Result:



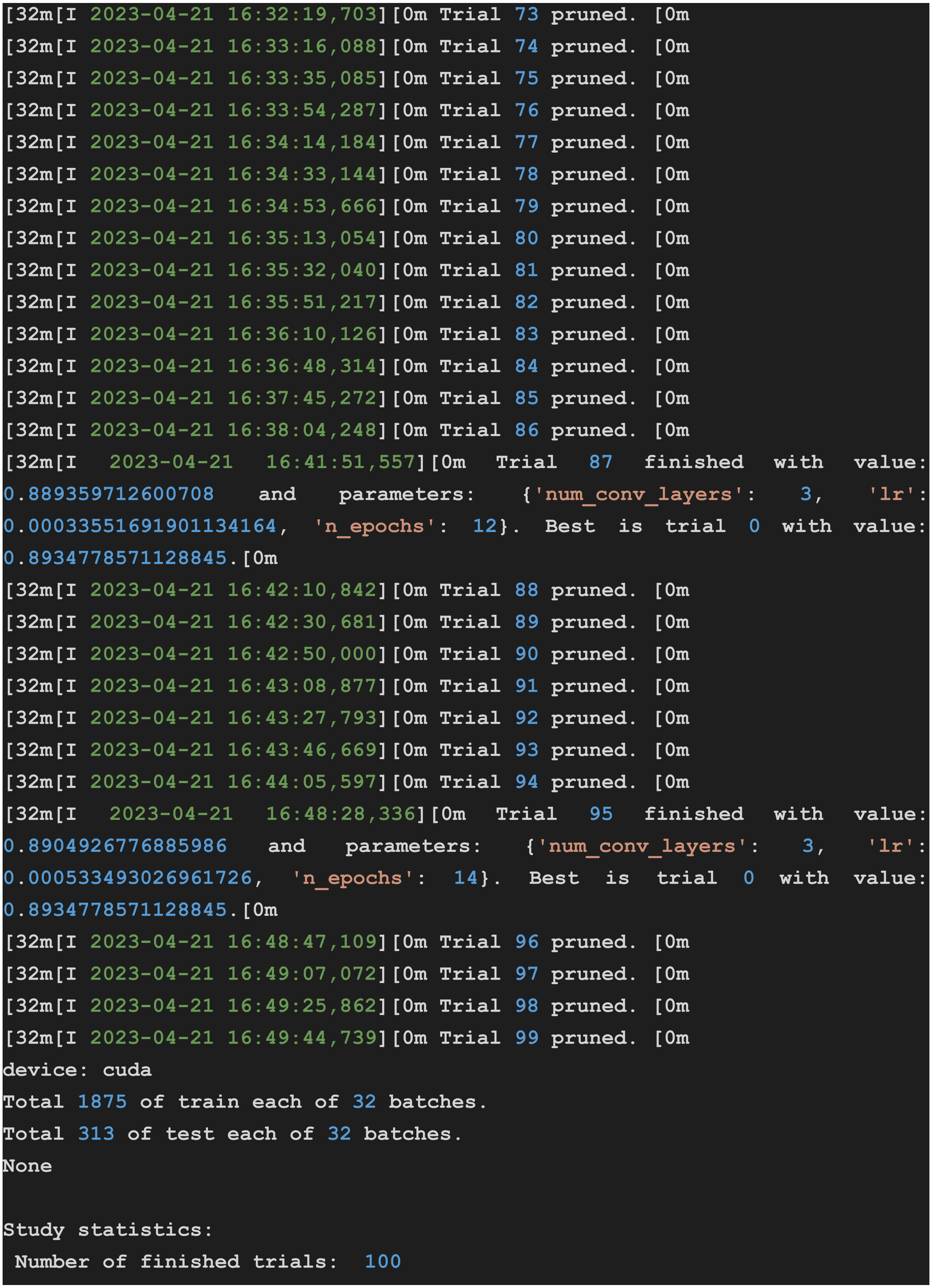
```
[32m[I 2023-04-21 15:14:31,994][0m A new study created in memory with
name: no-name-19d55097-e978-425e-9626-db2bef72d892[0m
[32m[I 2023-04-21 15:18:22,366][0m Trial 0 finished with value:
0.8934778571128845 and parameters: {'num_conv_layers': 3, 'lr':
0.0003290901991583123, 'n_epochs': 12}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 15:23:07,612][0m Trial 1 finished with value:
0.8850163817405701 and parameters: {'num_conv_layers': 3, 'lr':
0.00016048343226175623, 'n_epochs': 15}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 15:28:44,032][0m Trial 2 finished with value:
0.01001398079097271 and parameters: {'num_conv_layers': 6, 'lr':
0.020271518136275545, 'n_epochs': 15}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 15:34:55,204][0m Trial 3 finished with value:
0.8815485835075378 and parameters: {'num_conv_layers': 4, 'lr':
0.0006786368597425135, 'n_epochs': 18}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 15:39:16,934][0m Trial 4 finished with value:
0.8840557932853699 and parameters: {'num_conv_layers': 5, 'lr':
0.00025853711242122785, 'n_epochs': 13}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 15:44:12,505][0m Trial 5 finished with value:
0.8814415335655212 and parameters: {'num_conv_layers': 3, 'lr':
0.0005104675648753565, 'n_epochs': 15}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 15:44:31,787][0m Trial 6 pruned. [0m
[32m[I 2023-04-21 15:45:09,440][0m Trial 7 pruned. [0m
```



```
[32m[I 2023-04-21 15:45:28,230][0m Trial 8 pruned. [0m
[32m[I 2023-04-21 15:45:47,291][0m Trial 9 pruned. [0m
[32m[I 2023-04-21 15:46:07,138][0m Trial 10 pruned. [0m
[32m[I 2023-04-21 15:46:25,934][0m Trial 11 pruned. [0m
[32m[I 2023-04-21 15:46:44,691][0m Trial 12 pruned. [0m
[32m[I 2023-04-21 15:47:04,058][0m Trial 13 pruned. [0m
[32m[I 2023-04-21 15:47:24,736][0m Trial 14 pruned. [0m
[32m[I 2023-04-21 15:51:29,041][0m Trial 15 finished with value:
0.8832060098648071 and parameters: {'num_conv_layers': 3, 'lr':
0.0011901463990051952, 'n_epochs': 13}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 15:51:49,040][0m Trial 16 pruned. [0m
[32m[I 2023-04-21 15:52:08,409][0m Trial 17 pruned. [0m
[32m[I 2023-04-21 15:52:45,838][0m Trial 18 pruned. [0m
[32m[I 2023-04-21 15:53:04,909][0m Trial 19 pruned. [0m
[32m[I 2023-04-21 15:53:42,593][0m Trial 20 pruned. [0m
[32m[I 2023-04-21 15:54:02,614][0m Trial 21 pruned. [0m
[32m[I 2023-04-21 15:54:22,631][0m Trial 22 pruned. [0m
[32m[I 2023-04-21 15:54:43,280][0m Trial 23 pruned. [0m
[32m[I 2023-04-21 15:55:03,186][0m Trial 24 pruned. [0m
[32m[I 2023-04-21 15:55:22,310][0m Trial 25 pruned. [0m
[32m[I 2023-04-21 15:55:42,169][0m Trial 26 pruned. [0m
[32m[I 2023-04-21 15:56:02,674][0m Trial 27 pruned. [0m
[32m[I 2023-04-21 15:56:21,763][0m Trial 28 pruned. [0m
[32m[I 2023-04-21 15:56:42,295][0m Trial 29 pruned. [0m
[32m[I 2023-04-21 15:57:01,101][0m Trial 30 pruned. [0m
[32m[I 2023-04-21 15:57:19,868][0m Trial 31 pruned. [0m
[32m[I 2023-04-21 16:01:06,479][0m Trial 32 finished with value:
0.8851439952850342 and parameters: {'num_conv_layers': 3, 'lr':
0.0009658935966869438, 'n_epochs': 12}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 16:03:35,759][0m Trial 33 pruned. [0m
[32m[I 2023-04-21 16:03:54,975][0m Trial 34 pruned. [0m
[32m[I 2023-04-21 16:04:33,172][0m Trial 35 pruned. [0m
[32m[I 2023-04-21 16:05:32,971][0m Trial 36 pruned. [0m
[32m[I 2023-04-21 16:05:52,018][0m Trial 37 pruned. [0m
[32m[I 2023-04-21 16:06:10,796][0m Trial 38 pruned. [0m
[32m[I 2023-04-21 16:06:29,583][0m Trial 39 pruned. [0m
[32m[I 2023-04-21 16:11:11,867][0m Trial 40 finished with value:
0.8869701027870178 and parameters: {'num_conv_layers': 5, 'lr':
```

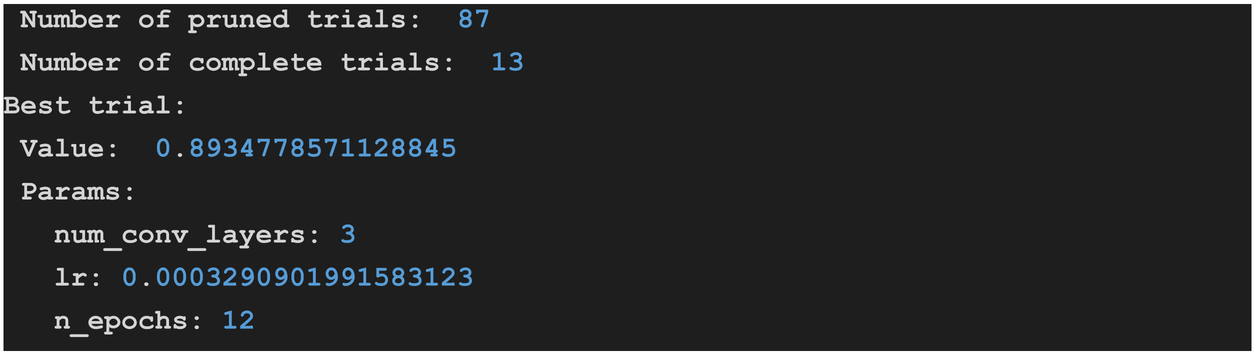


```
0.0005248305060095553, 'n_epochs': 14}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 16:11:33,011][0m Trial 41 pruned. [0m
[32m[I 2023-04-21 16:11:53,238][0m Trial 42 pruned. [0m
[32m[I 2023-04-21 16:12:13,110][0m Trial 43 pruned. [0m
[32m[I 2023-04-21 16:12:32,386][0m Trial 44 pruned. [0m
[32m[I 2023-04-21 16:12:53,286][0m Trial 45 pruned. [0m
[32m[I 2023-04-21 16:13:14,035][0m Trial 46 pruned. [0m
[32m[I 2023-04-21 16:13:33,672][0m Trial 47 pruned. [0m
[32m[I 2023-04-21 16:13:54,832][0m Trial 48 pruned. [0m
[32m[I 2023-04-21 16:17:25,668][0m Trial 49 finished with value:
0.8773127794265747 and parameters: {'num_conv_layers': 3, 'lr':
0.0006003421131151889, 'n_epochs': 11}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 16:17:45,194][0m Trial 50 pruned. [0m
[32m[I 2023-04-21 16:18:04,292][0m Trial 51 pruned. [0m
[32m[I 2023-04-21 16:18:23,222][0m Trial 52 pruned. [0m
[32m[I 2023-04-21 16:18:42,192][0m Trial 53 pruned. [0m
[32m[I 2023-04-21 16:22:36,540][0m Trial 54 finished with value:
0.8801952004432678 and parameters: {'num_conv_layers': 3, 'lr':
0.0012315747685694405, 'n_epochs': 12}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 16:22:57,698][0m Trial 55 pruned. [0m
[32m[I 2023-04-21 16:23:19,490][0m Trial 56 pruned. [0m
[32m[I 2023-04-21 16:23:40,515][0m Trial 57 pruned. [0m
[32m[I 2023-04-21 16:24:02,115][0m Trial 58 pruned. [0m
[32m[I 2023-04-21 16:25:25,777][0m Trial 59 pruned. [0m
[32m[I 2023-04-21 16:25:47,081][0m Trial 60 pruned. [0m
[32m[I 2023-04-21 16:26:08,225][0m Trial 61 pruned. [0m
[32m[I 2023-04-21 16:26:29,057][0m Trial 62 pruned. [0m
[32m[I 2023-04-21 16:26:50,716][0m Trial 63 pruned. [0m
[32m[I 2023-04-21 16:27:11,417][0m Trial 64 pruned. [0m
[32m[I 2023-04-21 16:28:49,243][0m Trial 65 pruned. [0m
[32m[I 2023-04-21 16:29:08,131][0m Trial 66 pruned. [0m
[32m[I 2023-04-21 16:29:27,333][0m Trial 67 pruned. [0m
[32m[I 2023-04-21 16:30:07,010][0m Trial 68 pruned. [0m
[32m[I 2023-04-21 16:30:25,869][0m Trial 69 pruned. [0m
[32m[I 2023-04-21 16:30:45,914][0m Trial 70 pruned. [0m
[32m[I 2023-04-21 16:31:23,335][0m Trial 71 pruned. [0m
[32m[I 2023-04-21 16:31:42,252][0m Trial 72 pruned. [0m
```



```
[32m[I 2023-04-21 16:32:19,703][0m Trial 73 pruned. [0m
[32m[I 2023-04-21 16:33:16,088][0m Trial 74 pruned. [0m
[32m[I 2023-04-21 16:33:35,085][0m Trial 75 pruned. [0m
[32m[I 2023-04-21 16:33:54,287][0m Trial 76 pruned. [0m
[32m[I 2023-04-21 16:34:14,184][0m Trial 77 pruned. [0m
[32m[I 2023-04-21 16:34:33,144][0m Trial 78 pruned. [0m
[32m[I 2023-04-21 16:34:53,666][0m Trial 79 pruned. [0m
[32m[I 2023-04-21 16:35:13,054][0m Trial 80 pruned. [0m
[32m[I 2023-04-21 16:35:32,040][0m Trial 81 pruned. [0m
[32m[I 2023-04-21 16:35:51,217][0m Trial 82 pruned. [0m
[32m[I 2023-04-21 16:36:10,126][0m Trial 83 pruned. [0m
[32m[I 2023-04-21 16:36:48,314][0m Trial 84 pruned. [0m
[32m[I 2023-04-21 16:37:45,272][0m Trial 85 pruned. [0m
[32m[I 2023-04-21 16:38:04,248][0m Trial 86 pruned. [0m
[32m[I 2023-04-21 16:41:51,557][0m Trial 87 finished with value:
0.889359712600708 and parameters: {'num_conv_layers': 3, 'lr':
0.00033551691901134164, 'n_epochs': 12}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 16:42:10,842][0m Trial 88 pruned. [0m
[32m[I 2023-04-21 16:42:30,681][0m Trial 89 pruned. [0m
[32m[I 2023-04-21 16:42:50,000][0m Trial 90 pruned. [0m
[32m[I 2023-04-21 16:43:08,877][0m Trial 91 pruned. [0m
[32m[I 2023-04-21 16:43:27,793][0m Trial 92 pruned. [0m
[32m[I 2023-04-21 16:43:46,669][0m Trial 93 pruned. [0m
[32m[I 2023-04-21 16:44:05,597][0m Trial 94 pruned. [0m
[32m[I 2023-04-21 16:48:28,336][0m Trial 95 finished with value:
0.8904926776885986 and parameters: {'num_conv_layers': 3, 'lr':
0.000533493026961726, 'n_epochs': 14}. Best is trial 0 with value:
0.8934778571128845.[0m
[32m[I 2023-04-21 16:48:47,109][0m Trial 96 pruned. [0m
[32m[I 2023-04-21 16:49:07,072][0m Trial 97 pruned. [0m
[32m[I 2023-04-21 16:49:25,862][0m Trial 98 pruned. [0m
[32m[I 2023-04-21 16:49:44,739][0m Trial 99 pruned. [0m
device: cuda
Total 1875 of train each of 32 batches.
Total 313 of test each of 32 batches.
None

Study statistics:
 Number of finished trials:  100
```



```
 Number of pruned trials:  87
 Number of complete trials:  13
Best trial:
 Value:  0.8934778571128845
 Params:
   num_conv_layers: 3
   lr: 0.0003290901991583123
   n_epochs: 12
```

Observation:

- It take less time than traditional methods because it pruned many of epochs with result are not improve
- For 100 hyperparameter combination it take just 2 hours but traditional method take more than 4 hours
- Also give best outcomes compared to the old school method.

REFERENCE:

https://www.kaggle.com/code/just4jcgeorge/dcgan-fashion-mnist-pytorch/notebook
https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html
https://github.com/elena-ecn/optuna-optimization-for-PyTorch-CNN
https://www.analyticsvidhya.com/blog/2020/11/hyperparameter-tuning-using-optuna/
Also from DR. Anush CoLab File