# Clustering Taxonomy : Document

## Code and Model Details:

- All the plots are checked-in the code and the models are saved in the cvpr_shared directory of machines given.
- To run the code, please refer to README.md file from the code for instructions.
- For dependencies to be installed for running the code, please refer requirements.txt file from the code.

The path for checked-in code is: https://gitlab.lrz.de/ga62moj/deep-clustering
The path for code on machine is : **/usr/data/cvpr_shared/ss18/deep-clustering/**
The path for models saved is: /usr/data/cvpr_shared/ss18/deep-clustering/models/
The path for plots saved is: /usr/data/cvpr_shared/ss18/deep-clustering/plots/

The details of the models and their corresponding paths for the models and plots for the datasets I worked on are listed below in the table. The model name and its corresponding clustering plot are saved with the same name.

| S.No. | Dataset | Model Name | Latent | Acc | NMI | Plot Path | Model Path |
|---|---|---|---|---|---|---|---|
| **1** | MNIST | mnist_arch7_nz16_pretrain | 16 | 56.7 | 51.2 | mnist/arch7/nz16/ | mnist/arch7/nz16/ |
| **2** | MNIST | mnist_arch7_nz16_fine | 16 | 54.7 | 53.0 | mnist/arch7/nz16/ | mnist/arch7/nz16/ |
| **3** | MNIST | mnist_arch7_nz32_pretrain | 32 | 66.8 | 60 | mnist/arch7/nz32/ | mnist/arch7/nz32/ |
| **4** | MNIST | mnist_arch7_nz32_fine | 32 | **71.4** | **63.3** | mnist/arch7/nz32/ | mnist/arch7/nz32/ |
| **5** | MNIST | mnist_arch7_nz64_pretrain | 64 | 63.6 | 58.9 | mnist/arch7/nz64/ | mnist/arch7/nz64/ |
| **6** | MNIST | mnist_arch7_nz64_fine | 64 | 70.4 | 59 | mnist/arch7/nz64/ | mnist/arch7/nz64/ |
| **7** | MNIST | mnist_arch7_nz128_pretrain | 128 | 44.1 | 38 | mnist/arch7/nz128/ | mnist/arch7/nz128/ |
| **8** | MNIST | mnist_arch7_nz128_fine | 128 | 59.3 | 52 | mnist/arch7/nz128/ | mnist/arch7/nz128/ |
| **9** | MNIST | mnist_arch7_nz256_pretrain | 256 | 58.3 | 53.3 | mnist/arch7/nz256/ | mnist/arch7/nz256/ |
| **10** | MNIST | mnist_arch7_nz256_fine | 256 | 63 | 56 | mnist/arch7/nz256/ | mnist/arch7/nz256/ |
| **11** | FMNIST | fmnist_arch1_nz32_pretrain | 32 | 52.6 | 57.1 | fmnist/arch1/nz32 | fmnist/arch1/nz32 |
| **12** | FMNIST | fmnist_arch1_nz32_fine | 32 | **57.5** | **58.6** | fmnist/arch1/nz32 | fmnist/arch1/nz32 |
| **13** | FMNIST | fmnist_arch2_nz64_fine | 64 | 58.4 | 55.2 | fmnist/arch2/nz64 | fmnist/arch2/nz64 |
| **14** | FMNIST | fmnist_arch2_nz256_fine | 256 | 59.4 | 56.1 | fmnist/arch2/nz256 | fmnist/arch2/nz256 |
| **15** | CIFAR10 | cifar10_arch5_nz32_pretrain | 32 | 23.6 | 12.4 | cifar10/arch5/nz32/ | cifar10/arch5/nz32/ |
| **16** | CIFAR10 | cifar10_arch5_nz32_fine | 32 | **26.7** | **18.9** | cifar10/arch5/nz32/ | cifar10/arch5/nz32/ |
| **17** | STL10 | stl10_arch4_nz100_pretrain | 100 | 24.4 | 21.7 | stl10/arch4/nz100/ | stl10/arch4/nz100/ |
| **18** | STL10 | stl10_arch4_nz100_fine | 100 | **28.3** | **22.2** | stl10/arch4/nz100/ | stl10/arch4/nz100/ |

# Clustering Taxonomy : Document

## Dataset Details:

The datasets are not checked into the code. Torchvision dataloaders are used to load the data in dataloader object. So data will be downloaded automatically from 'torchvision.datasets' package when we run main.py file for the first time.
The path where datasets will be saved is: deep-clustering/data/$dataset_name$/

For detailed information how data is loaded and images are normalized, refer to deep-clustering/utils/dataset.py file from the code. Until now I have used MNIST, FMNIST, CIFAR-10, STL-10 datasets.

## Suggestions for Future Work:

The suggestions for future work are mentioned on "Future Work" slide during the presentation given.
The first important task to be done is to further improve the accuracies (for MNIST dataset first and then CIFAR,STL datasets).To improve accuracy, one thing that could be tried out is to play around more with parameters of t-sne (increasing perplexity increased Clustering Accuracy sometimes for me). Also play around with different values of alpha.

Then extend the taxonomy on bigger datasets (like ImageNet) and textual datasets (like Reuters).

## Installation Steps(If required):

1. Create Virtual Environment :
   pip3 install virtualenv

2. Create Virtual Environmet:
   virtualenv -p python3 --no-site-packages .venv

3. Activate Virtual Environment:
   source .venv/bin/activate

4. Copy Requiremnts File
   scp -P 58022 /Users/$user$/Desktop/requirements.txt
   s0152@atcremers51.informatik.tu-muenchen.de:/usr/prakt/s0152/

5. Install the dependencies:
   pip3 install -r requirements.txt

6. Check Cuda Version
   nvcc –version

7. Install PyTorch for Cuda
   pip3 install torch torchvision

# Clustering Taxonomy : Document

8. Install Tensorboard:
   pip3 install tensorboard-pytorch --no-cache-dir
   pip3 install tensorflow-gpu==1.4

   Log Path to run the server: tensorboard --logdir=$path$/logs/

## Useful Commands:

### Working with screens:

- Open a new sessions: screen
- Disconnect from session but keep it running: CRTL-A, CMD-D
- Disconnect and close screen: exit
- Re-attach to existing screen: screen -r
- Enable scrolling in screen: CRTL-A, ESC

### Get Status of Server:

- See which other users are logged in: finger
- See status of other servers: https://adm9.in.tum.de/status
- See status of GPU: nvidia-smi -l 1
- See status of CPU: top

Path to the clustering taxonomy paper:
https://sl9.informatik.tu-muenchen.de/project/59c39f7f1230f2b705566db5

Path to the old lasagne code:
https://gitlab.lrz.de/ga83fiz/dlcv-praktikum-ss17-clustering