Curriculum Learning for CNN

Jakub Sawicki, Katsiaryna Bokhan, Milanna Pahasian, Rafał Pyzowski

01

Training CNN and Transformers model on the DTD from scratch

02

Getting pre-trained on the ImageNet dataset CNN and Transformers models

03

Using transfer learning to adapt four models to the RSSCN7 dataset

04

Comparing results using metrics and XAI

Project Results Review

Research Proposal: Motivation

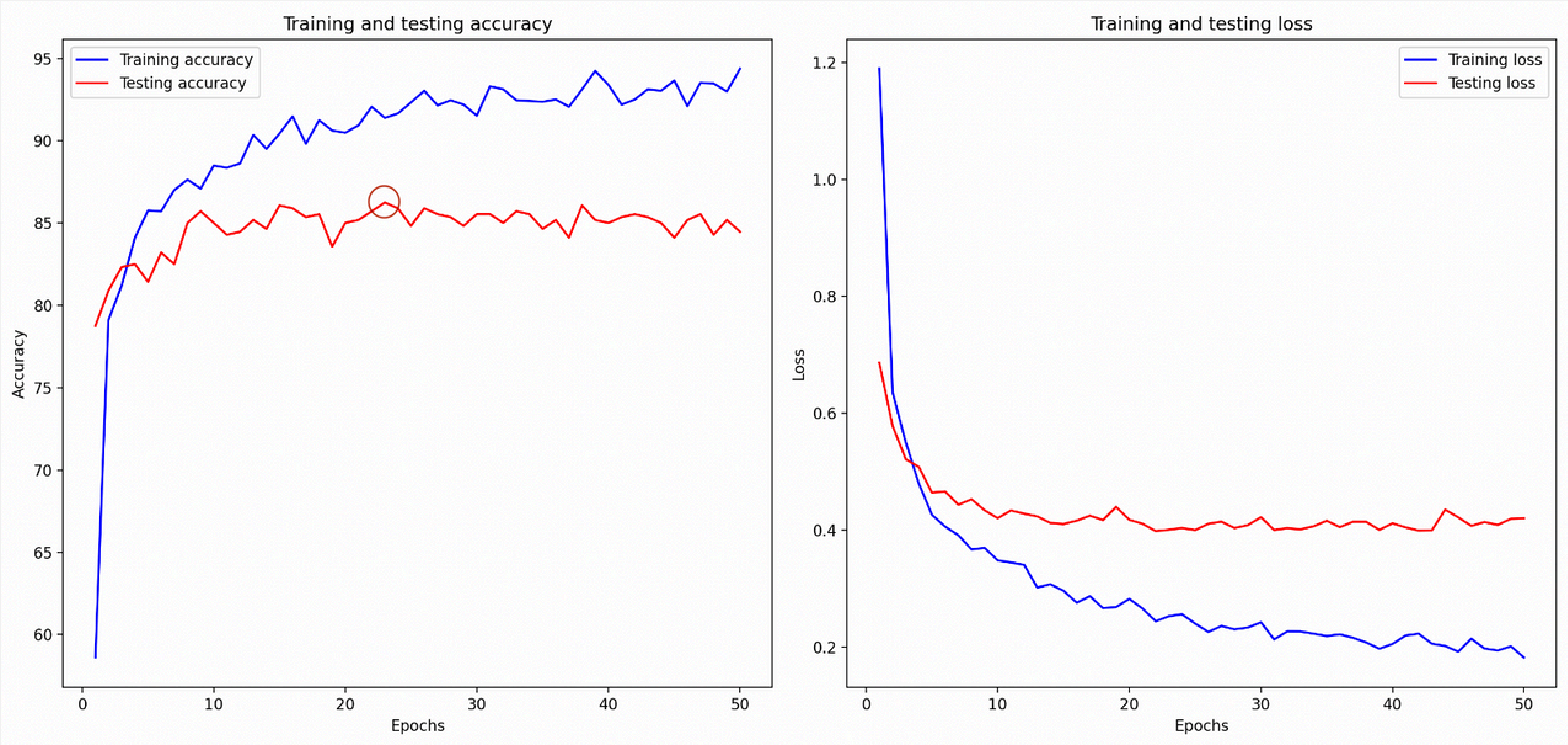
01 CNN models are easier to train

02 They have fewer hyperparameters

03 CNN models gave better results in our case

04 The best model so far was achieved using transfer learning on a pre-trained ImageNet model

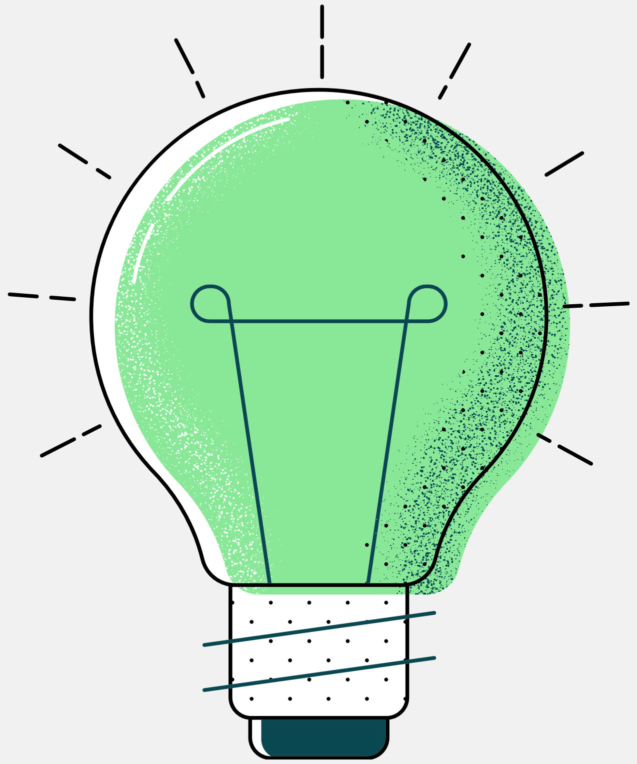
05 We wanted to test how to improve performance further



Research Proposal

Curriculum Learning

Self-Paced Learning **Manual Division**



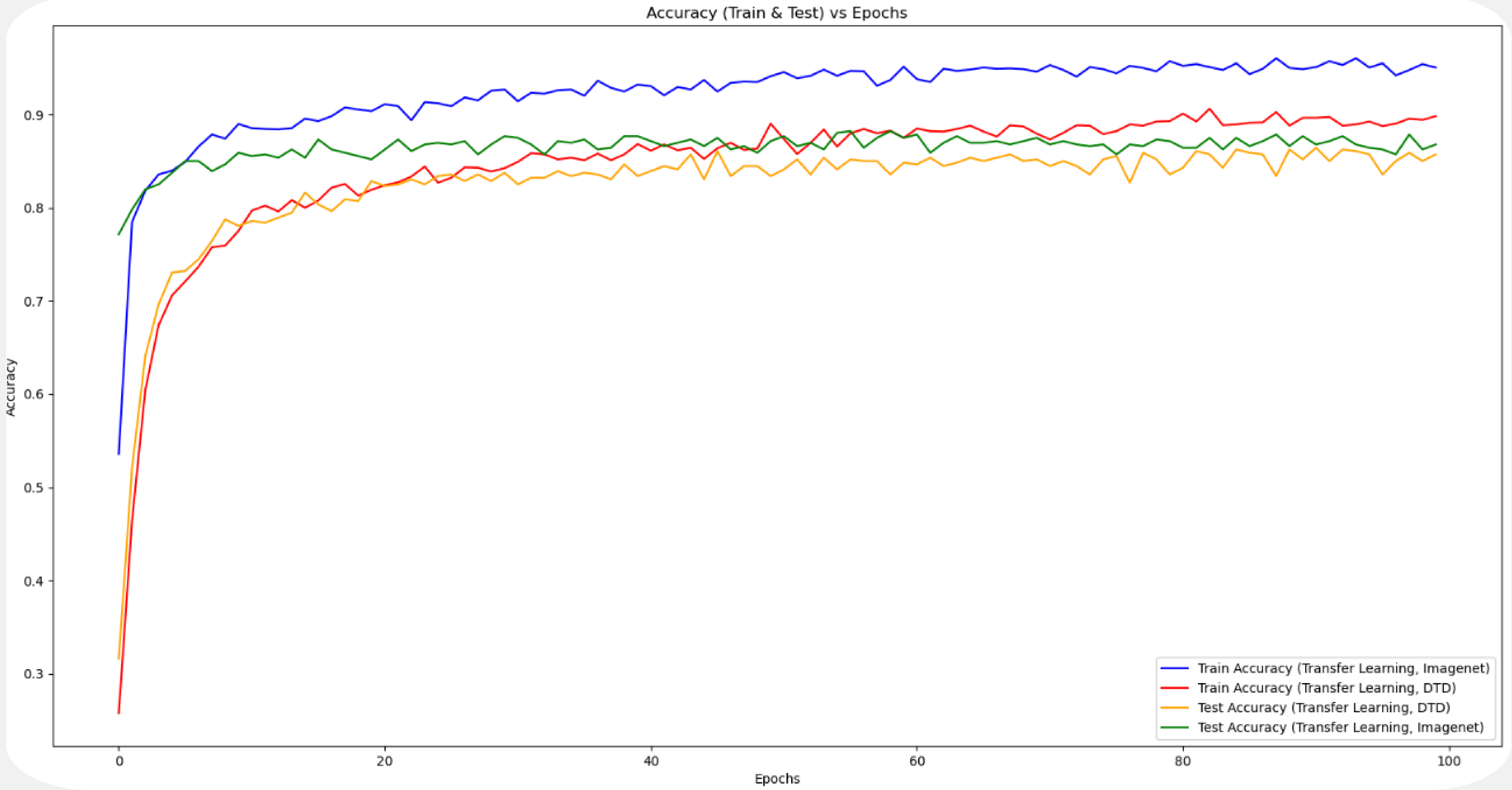
KNN: adding a new metric for more efficient analysis

The hypothesis: curriculum learning will improve the performance of CNN

Transfer Learning
after
learning rate tuning

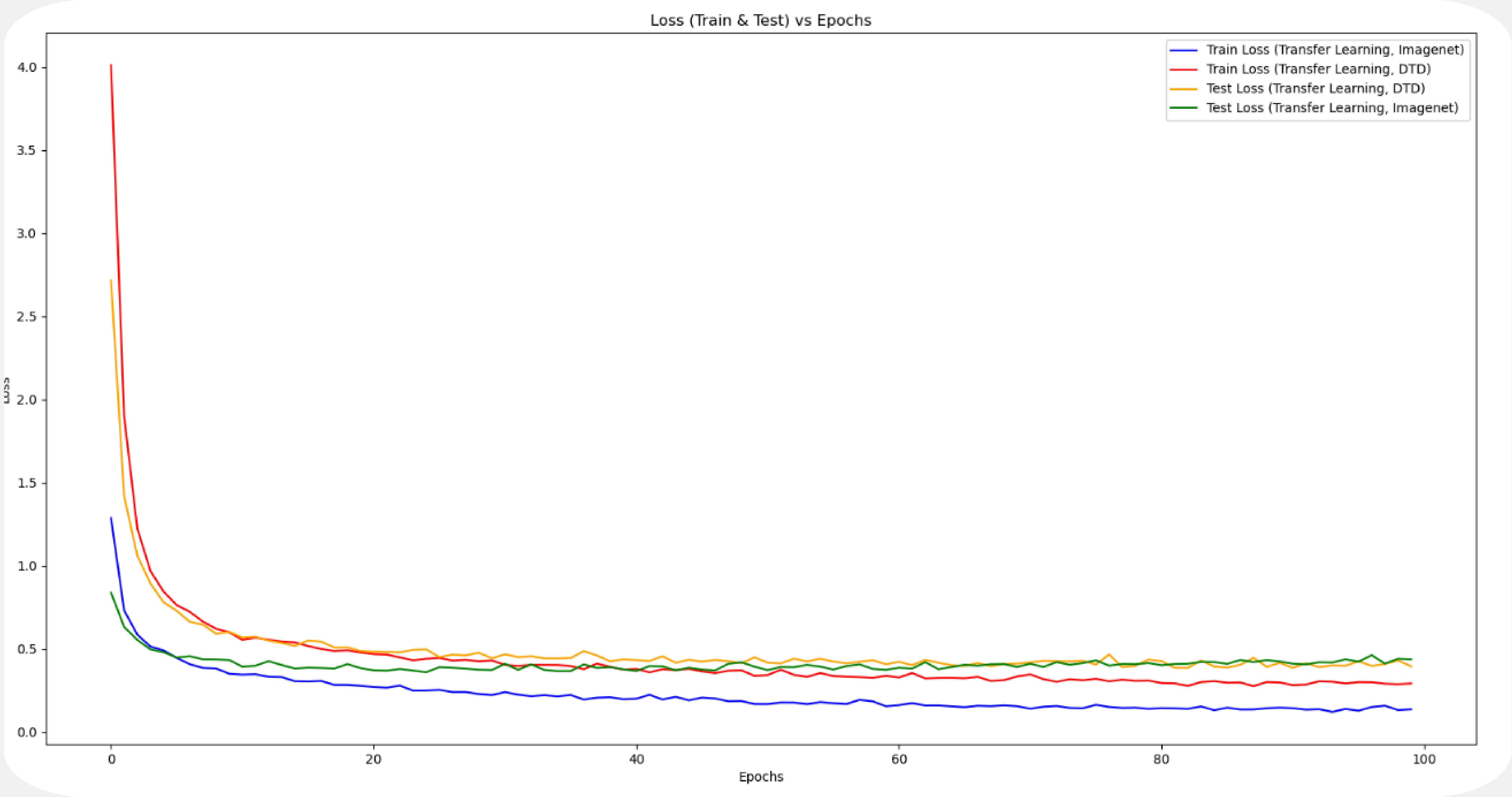
Transfer Learning

Accuracy



Transfer Learning (after learning rate tuning)

Loss



Curriculum Learning
Manual Division

Details in implementation and separation of images

Number of classes: 7; 400 images per class

Easy: ~135 images per class

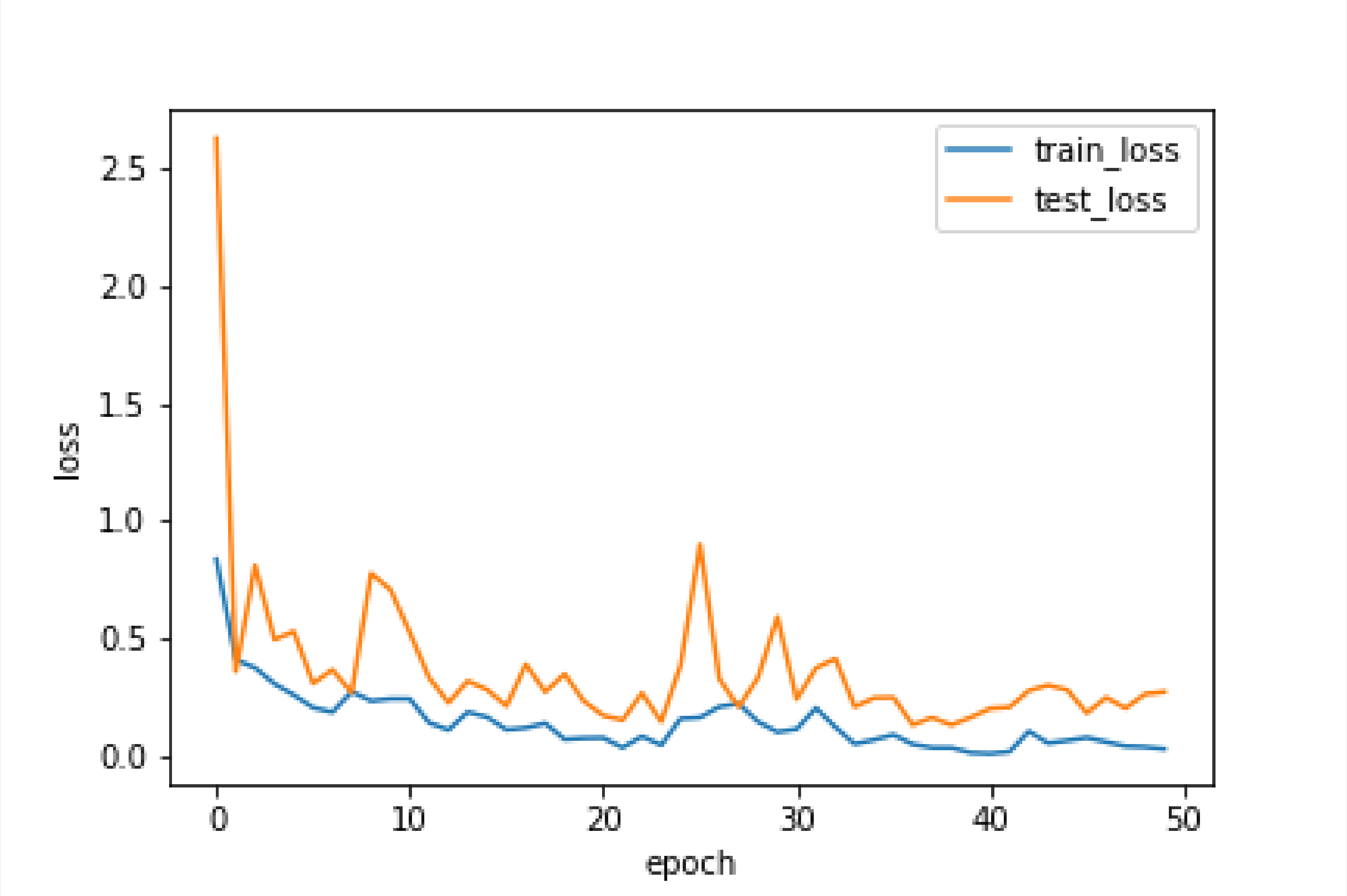
Medium: ~160-170 images per class

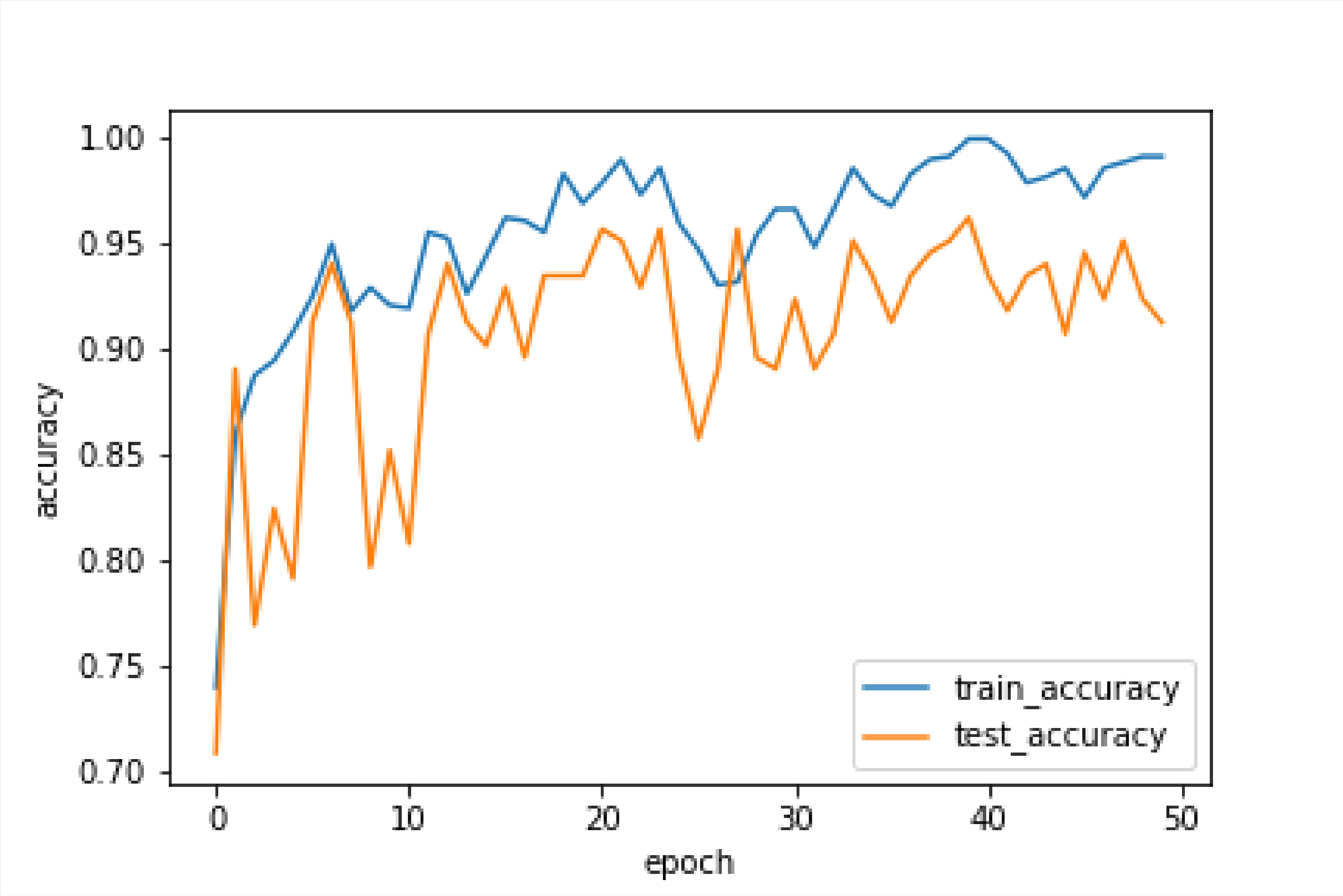
Hard: ~90-100 images per class

Training Stages: Easy → Medium → Hard

Results on easy

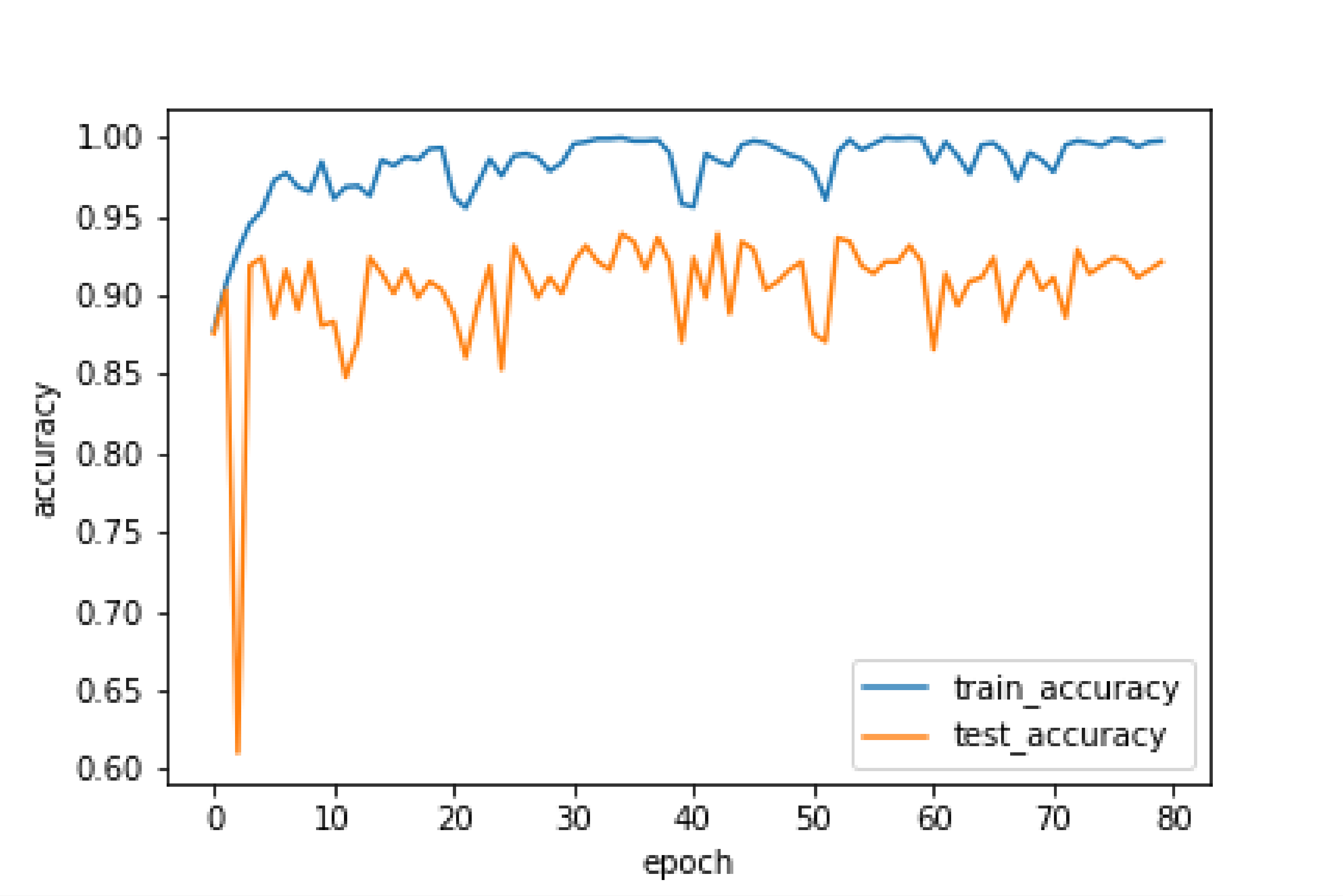
Loss



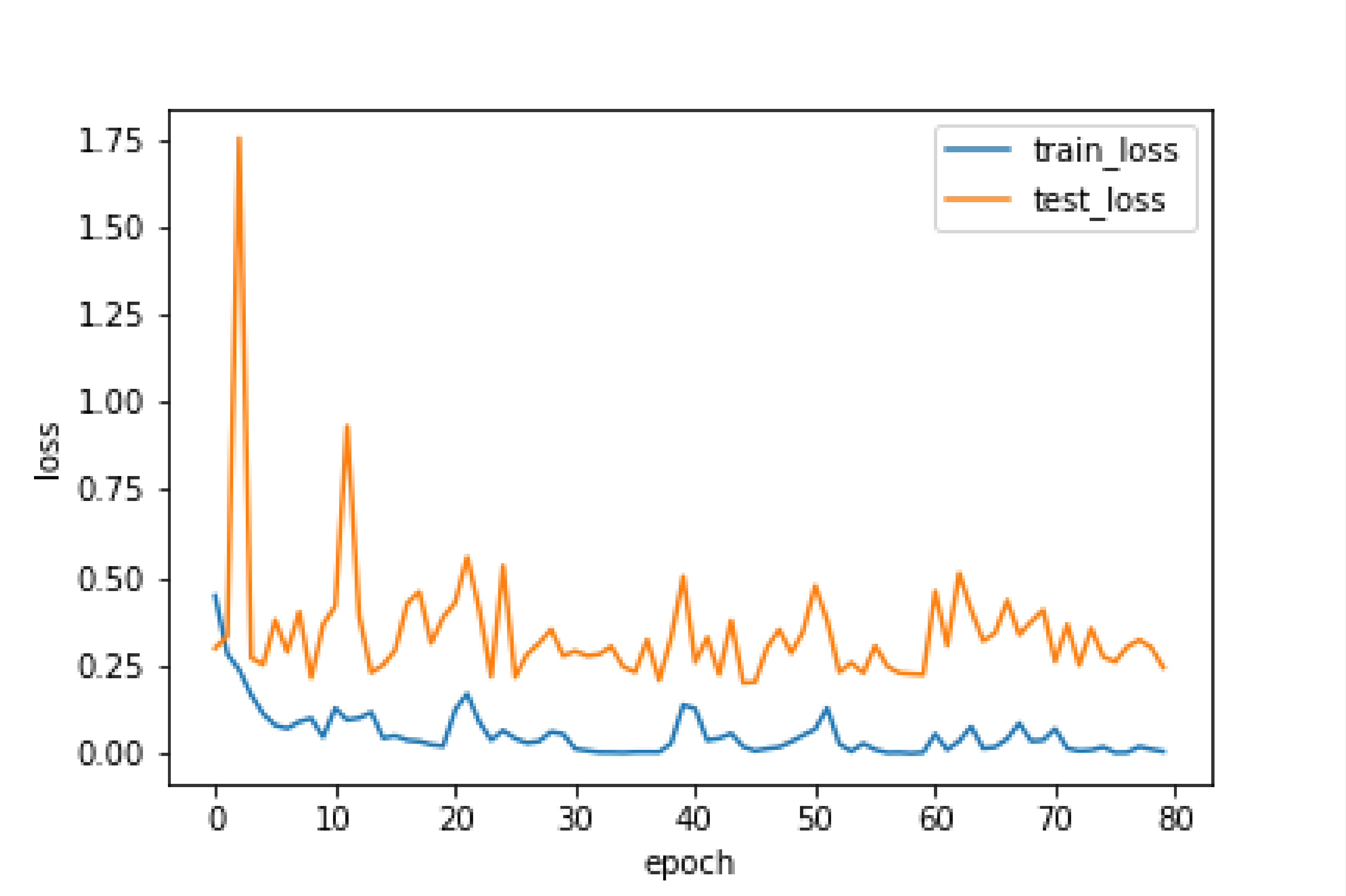


Accuracy

Results on easy+medium



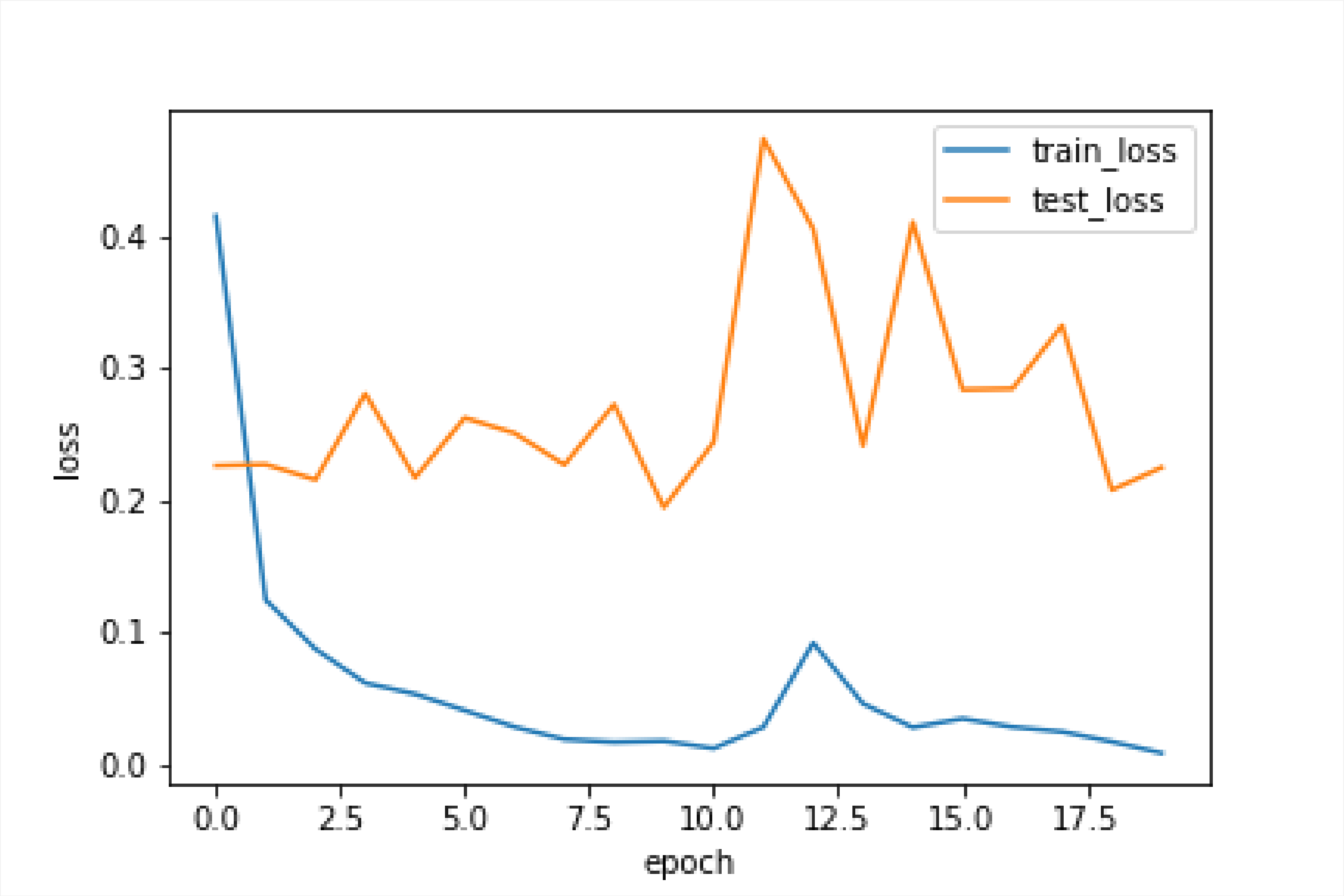
Accuracy

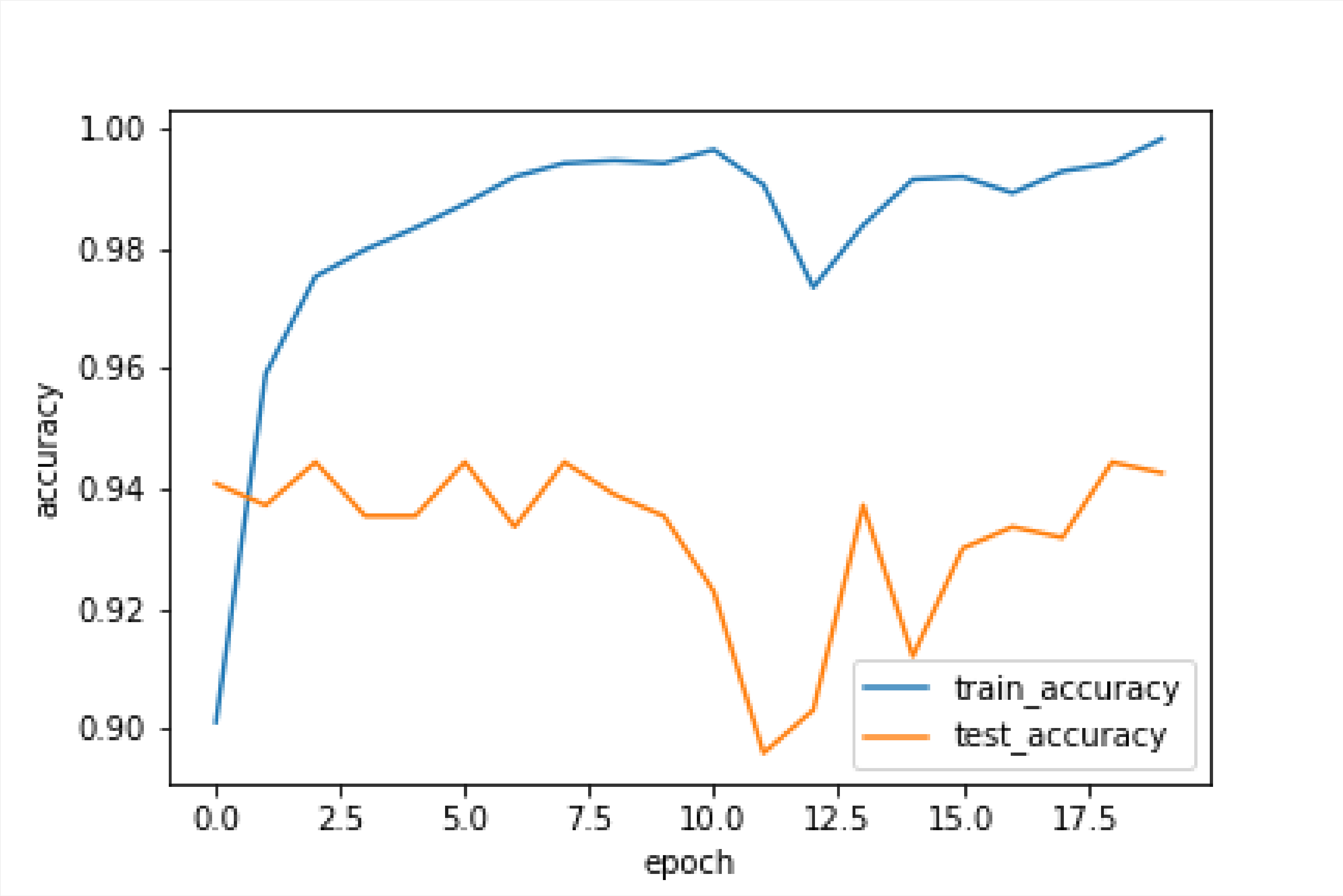


Loss

Final results

Loss





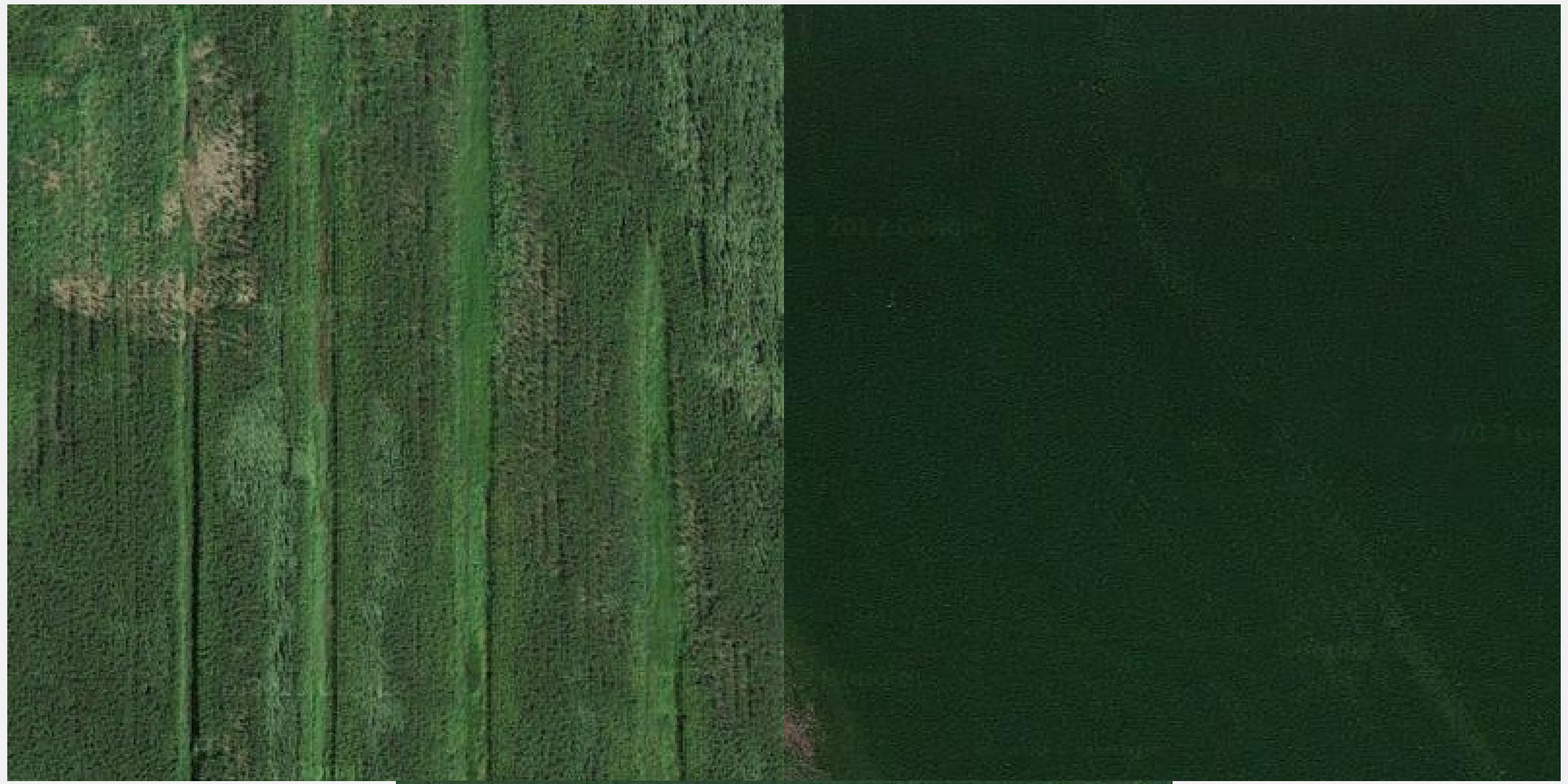
Accuracy

Question:

Was there any bias in separating the pictures due to human factor ?

class grass

class field



Self-Paced Learning

General idea

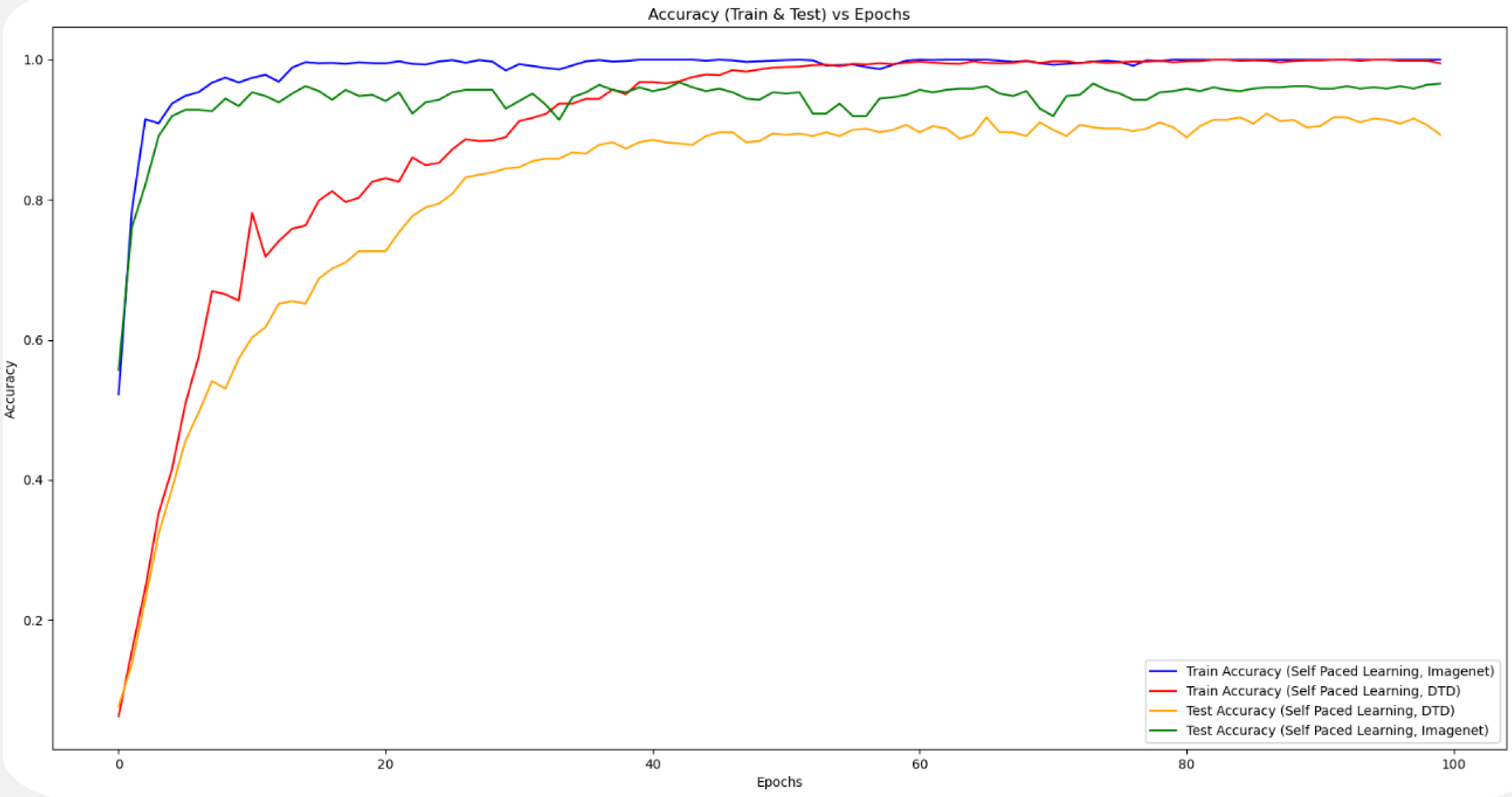
- use parameter λ which will be 0.1 in the beginning of training, later increase it. In other words λ sets the maximum complexity of the images on which we train the network, and increases the complexity within epochs
- at the beginning of each epoch we sort images (model is eval mode) by loss value and select λ% of the easiest ones

Details in implementation

- in the beginning - only 10% of images (224), then portions of 5% (112)
- the model had to maintain train_accuracy at a certain level (0.8) before receiving a new portion of pictures, later (after training on 60% of all images from training set) had to see all images twice before getting new portion
- learning rates - from 0.0001 to 0.0000001

Results

Accuracy

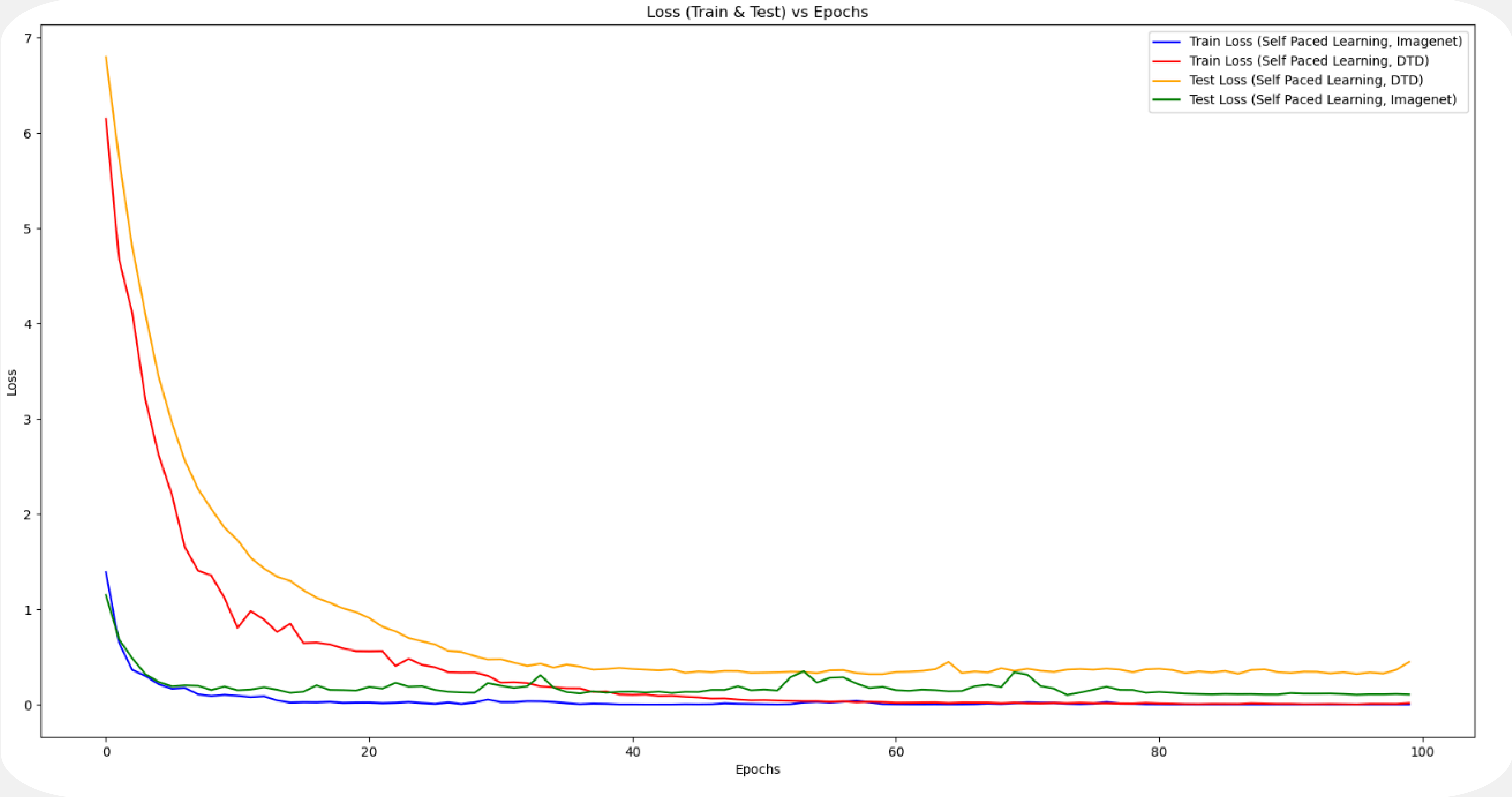


Imagenet:

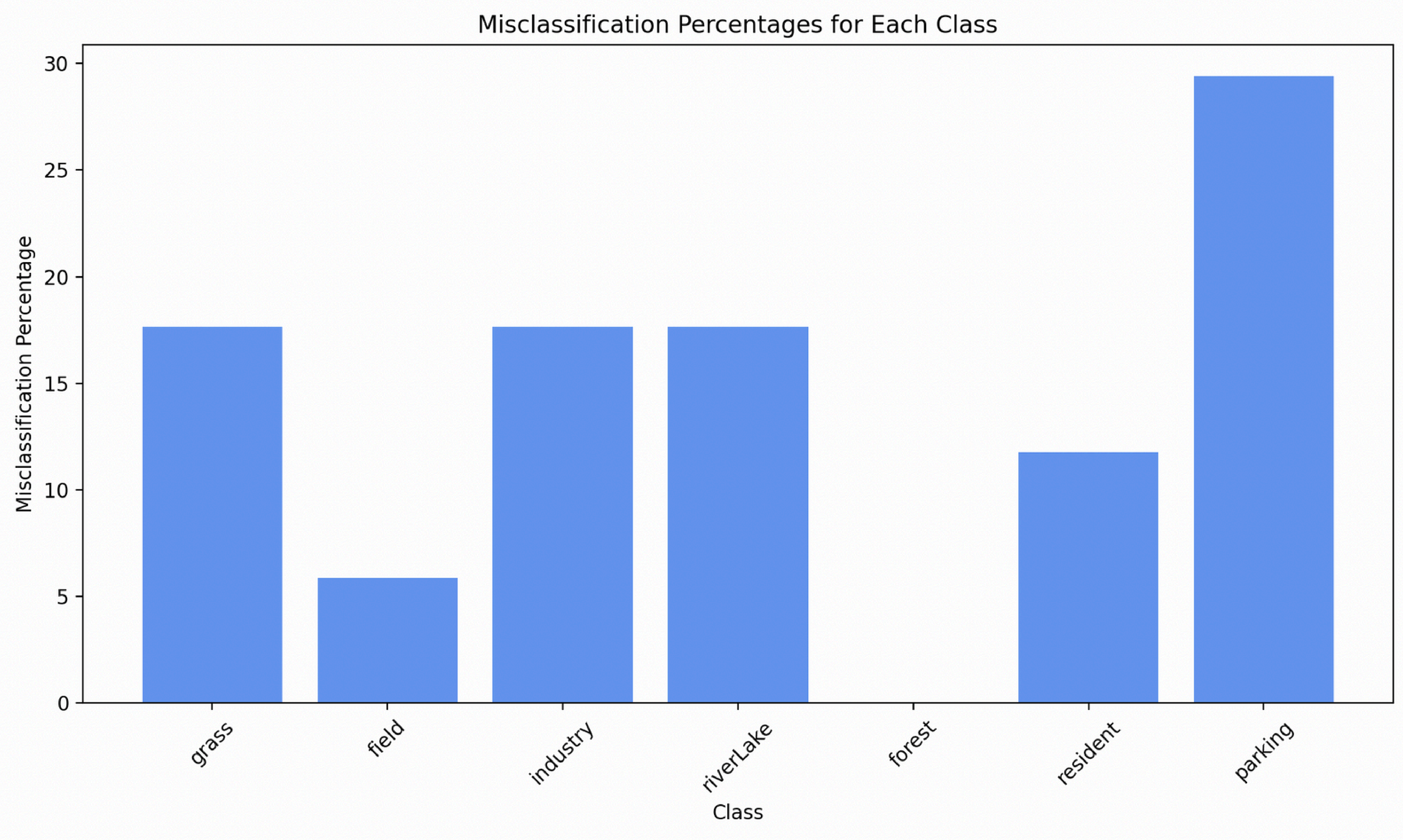
- 28 epoch - all training dataset
- 4 epoch - 89% accuracy on test set (91 seconds, 20% of dataset)
- 43 epochs to achieve the best result (96.78% accuracy on test)

Results

Loss



Misclassification of Self-Paced (ImageNet)

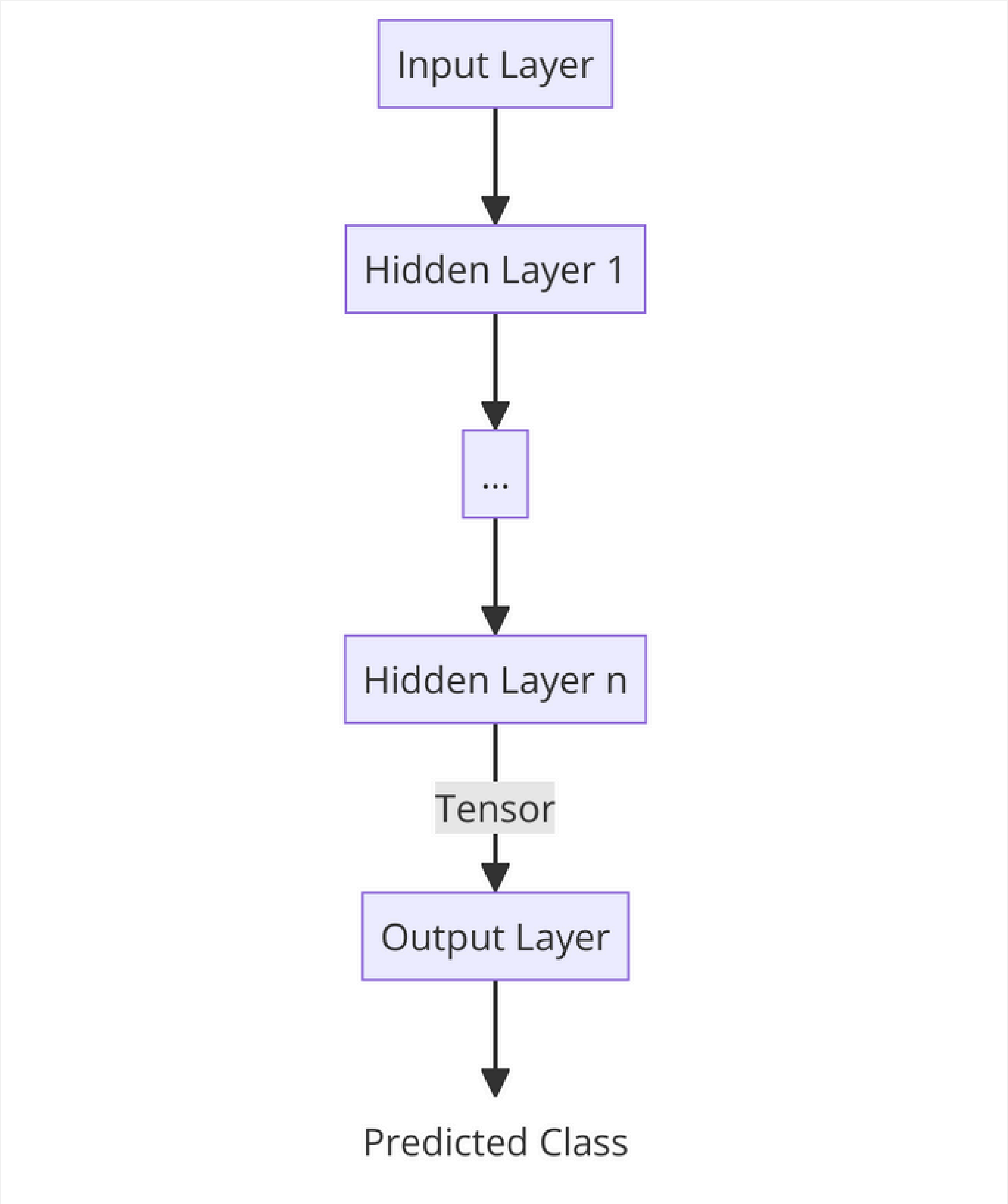



Field
Industry
RiverLake
Grass
Parking
Resident

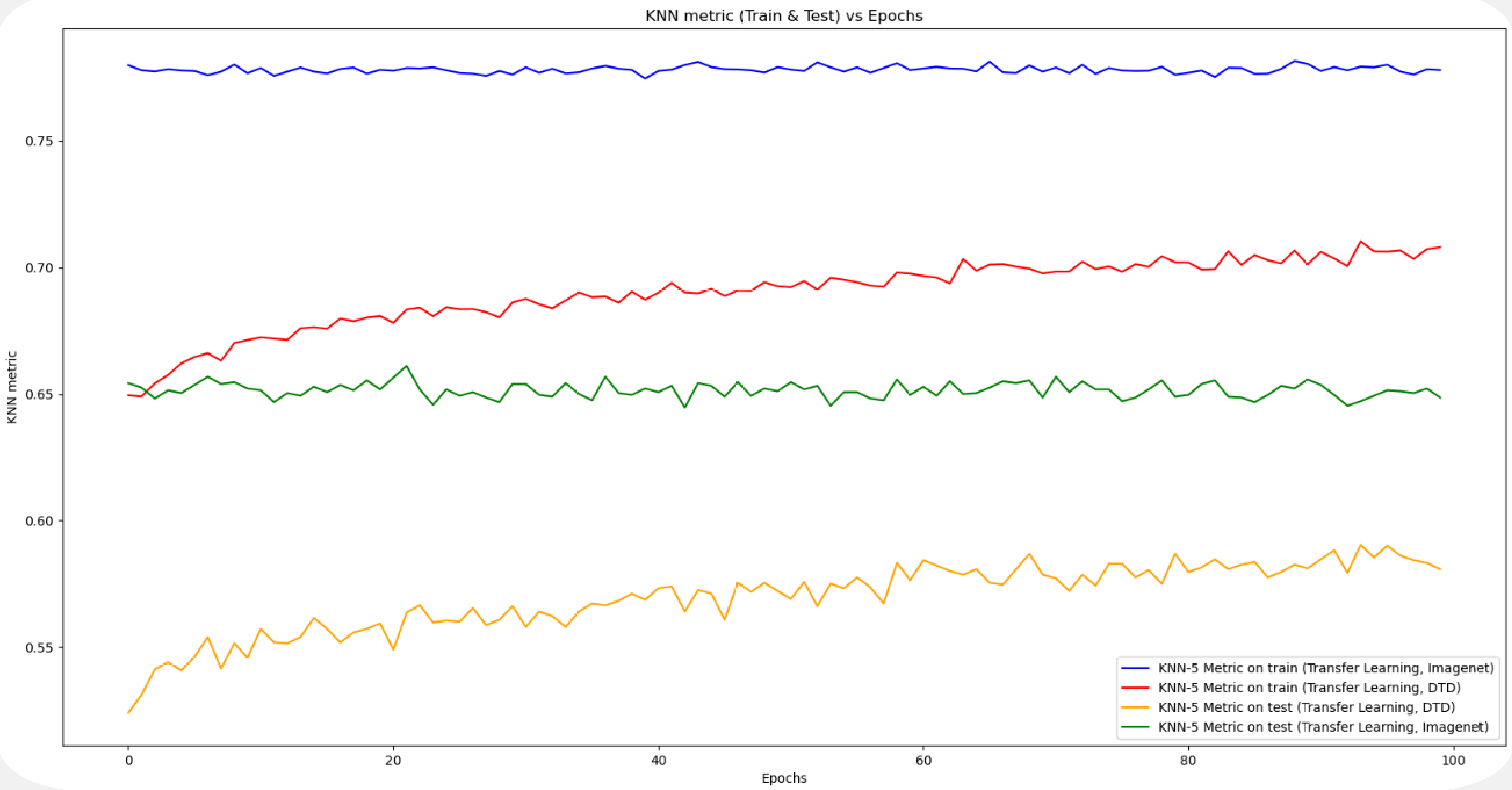
K Nearest Neighbours Metric (KNN metric)

Details in implementation and how it works

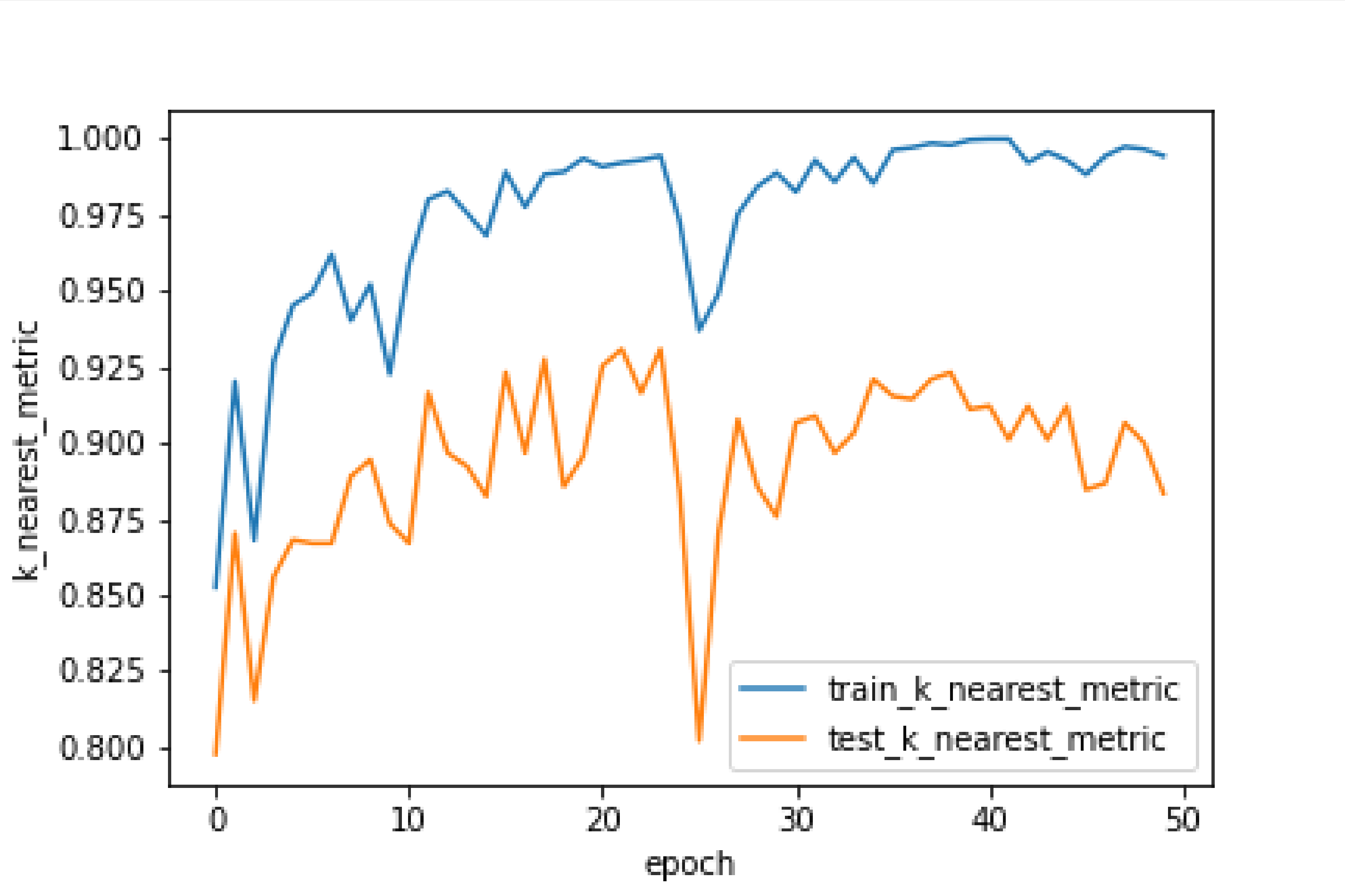
- Extracting tensors
- Counting distances
- Selecting k nearest tensors
- Counting classes
- Aggregating results



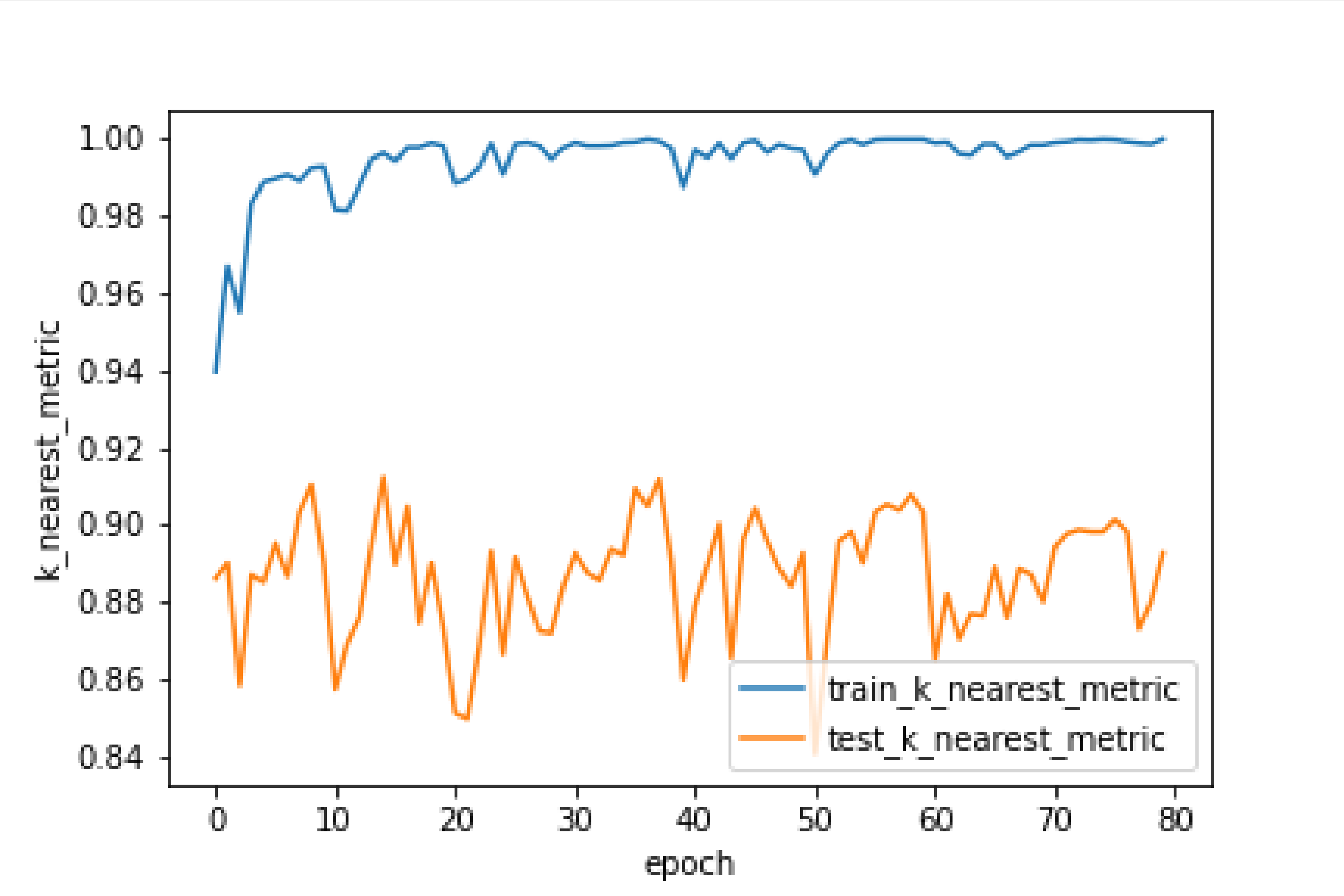
KNN (k=5) on Transfer Learning



KNN on Curriculum Learning (Manual division)

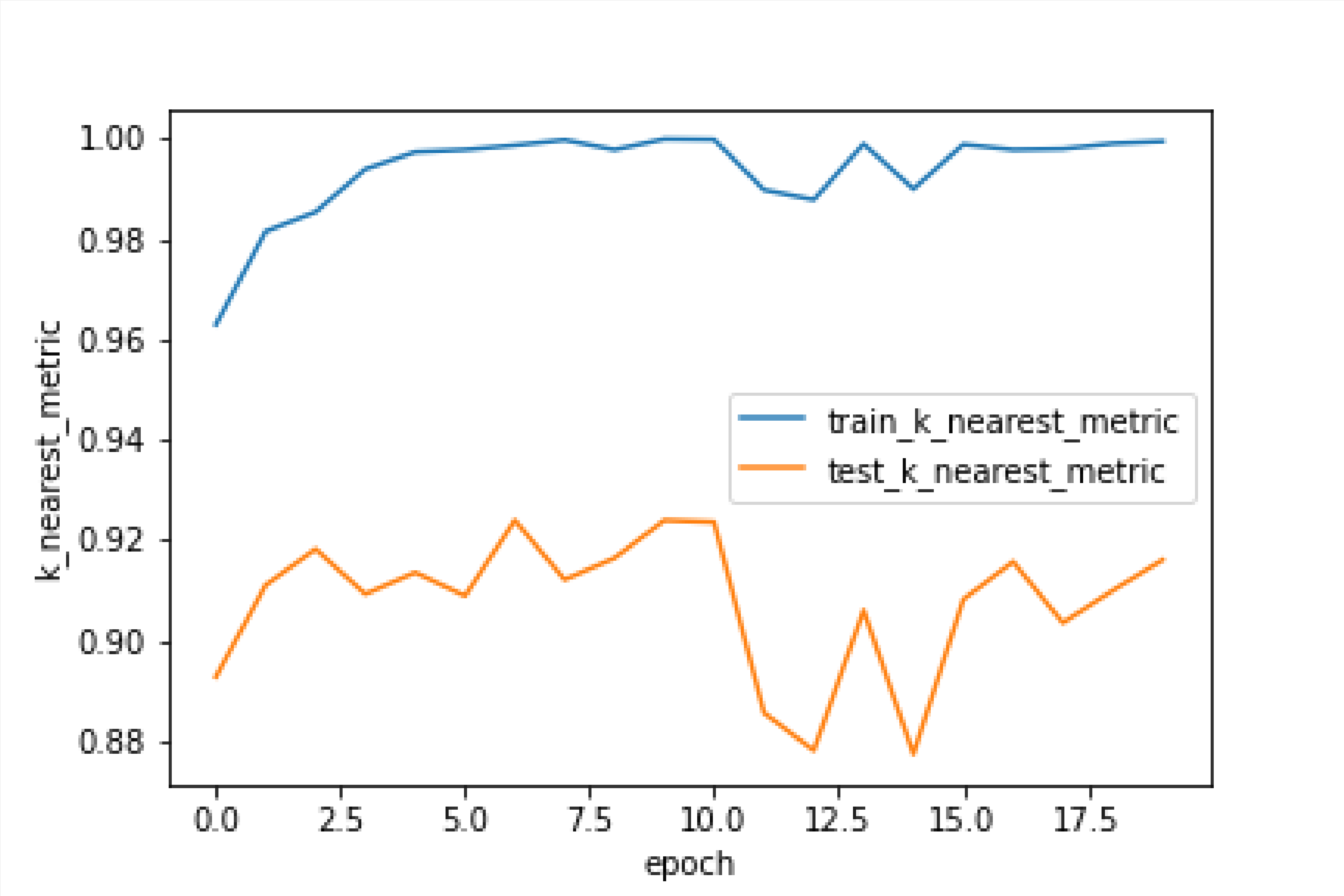


easy

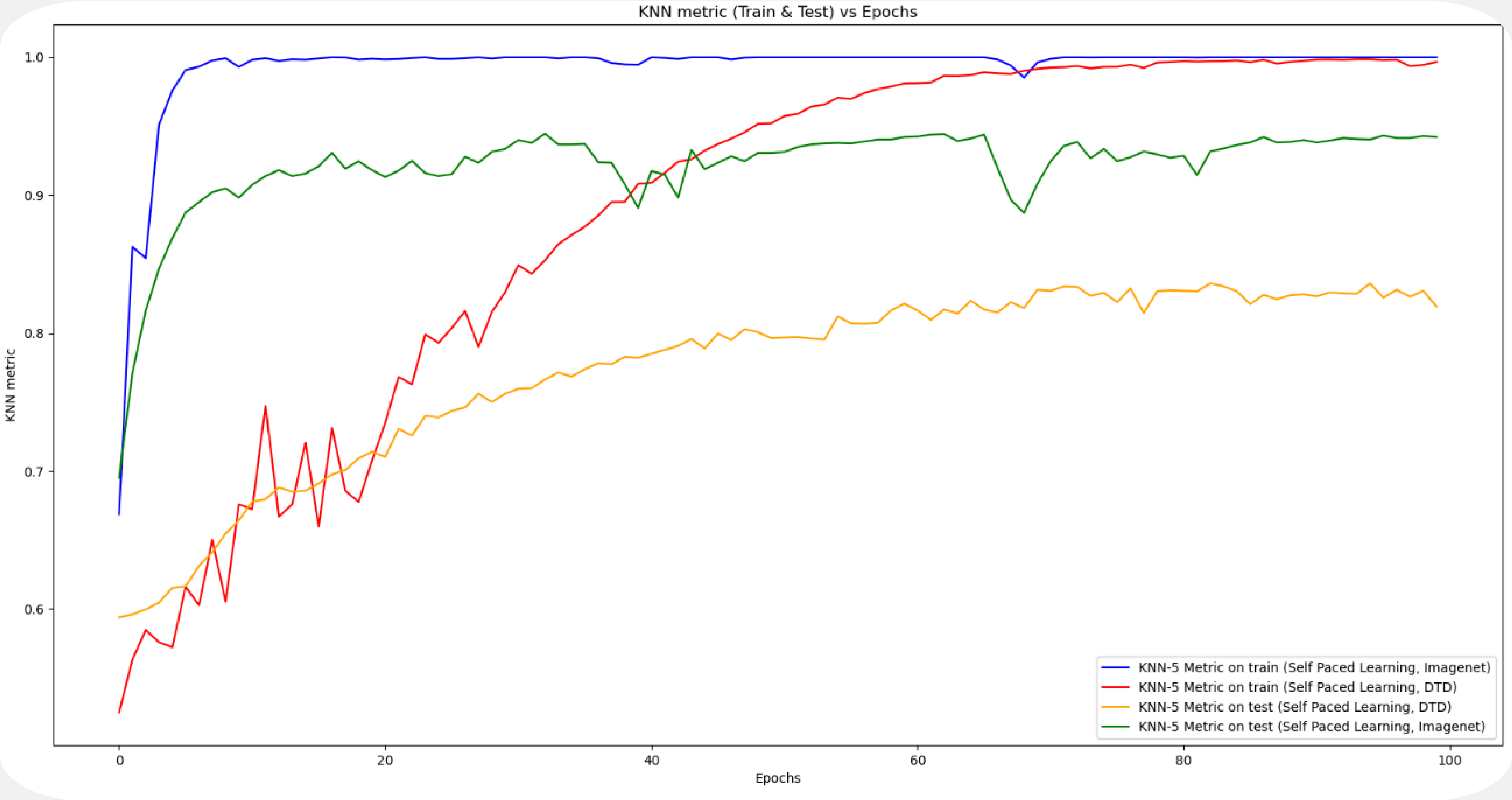


easy+medium

Final KNN on Curriculum Learning (Manual division)



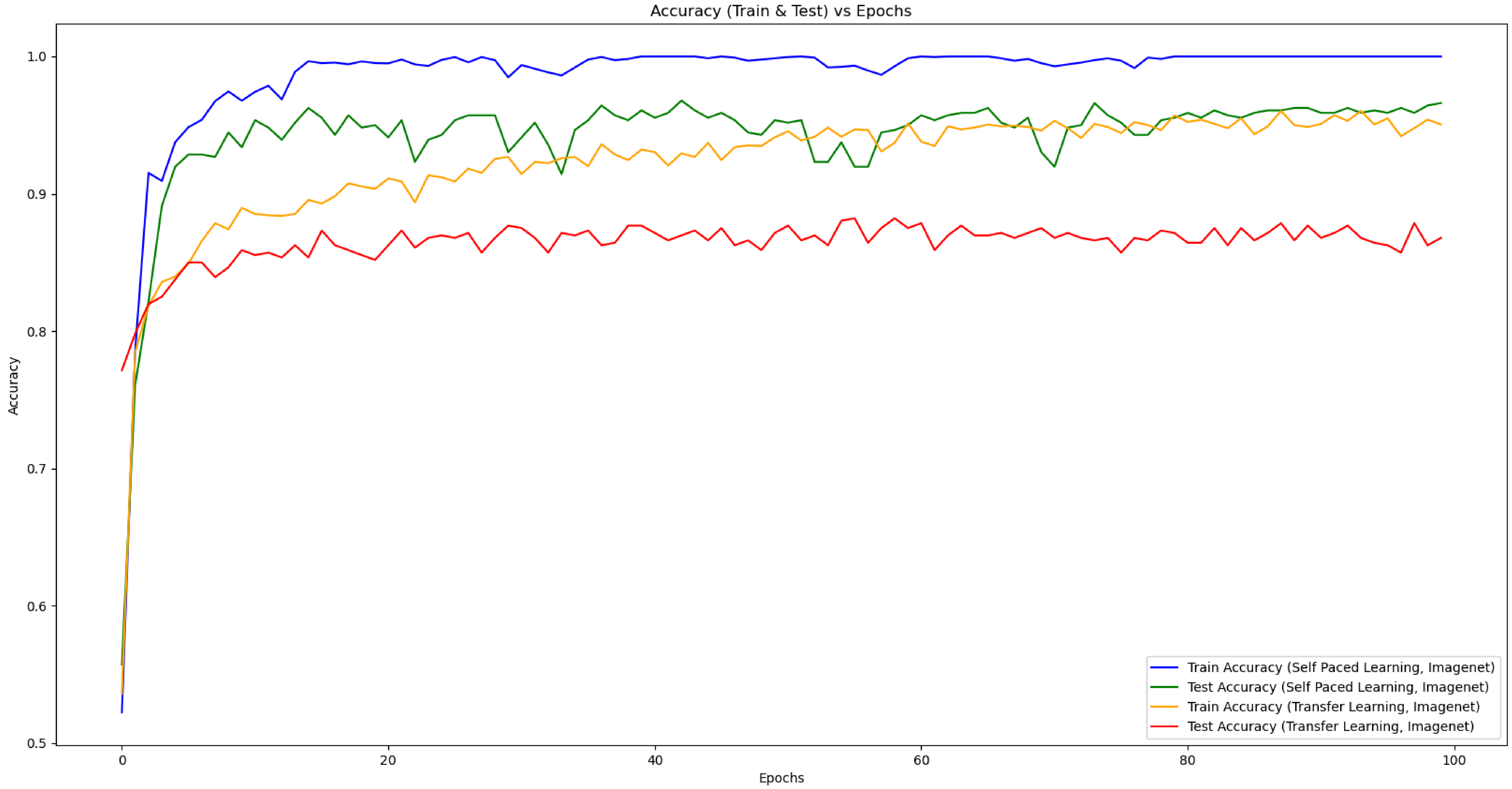
KNN on Self-Paced Learning



An evidence supporting the hypothesis

	Transfer learning (ImageNet)	Transfer Learning (DTD)	Manual Division (ImageNet)	Self-Paced Learning (ImageNet)	Self-Paced Learning (DTD)
Accuracy	0.88	0.86	0.94	0.9679	0.9232
Loss	0.38	0.39	0.22	0.1347	0.32
KNN	0.66	0.59	0.92	0.9446	0.84

Best Self-Paced Learning vs best Transfer Learning



Conclusion

01 Our hypothesis was confirmed

02 The best results were achieved by ResNet18 (ImageNet) + Curriculum Learning

03 KNN Metric is an interesting metric: grew perfectly across epochs in Curriculum Learning and almost did not grow in Transfer Learning

04 Manual division of images is not the best option. Automatic choosing of the easiest images works much better

Thank You!