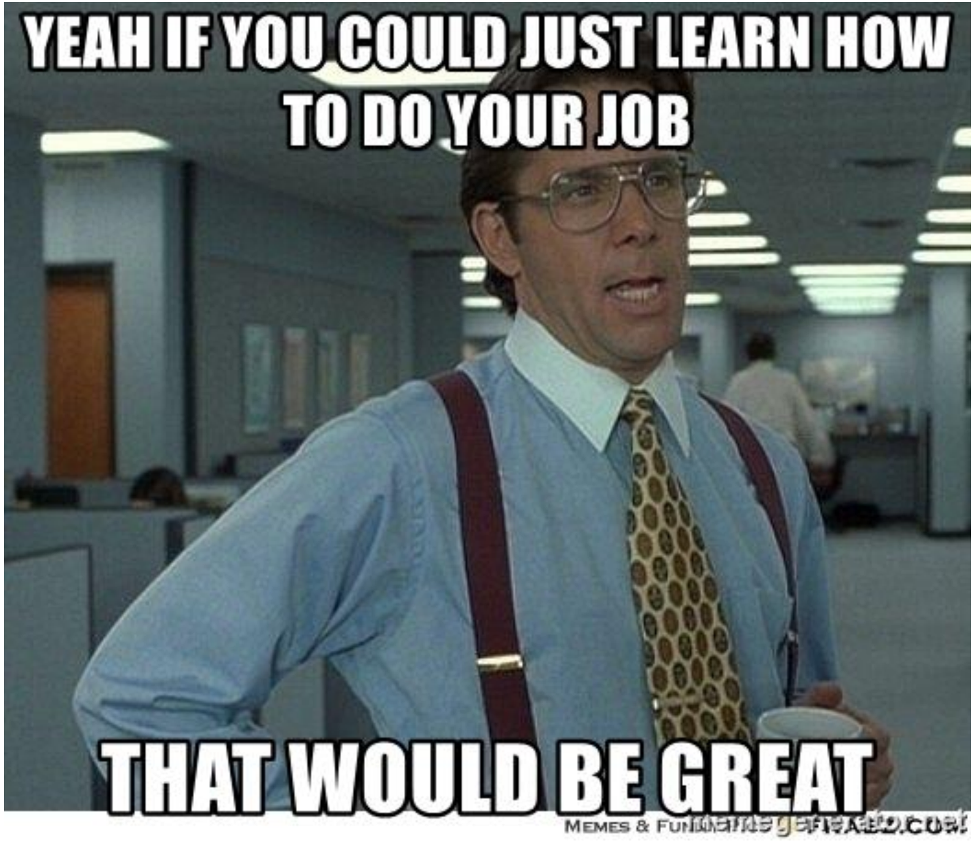
Studying the Performance of Self-Supervised Models

Vishwesh Kirthivasan (vk2470)
Hitesh Agarwal (HA2598)
Prof. Parijat Dube

Introduction



Self-Supervised vs Semi-Supervised

1. Semi-Supervised
    a. Train a teacher network from small amount of labelled data
    b. Create more labelled data by classifying lots of unlabelled data using this teacher network
    c. Train a student network with all of the data.
2. Self-Supervised
    a. Use lots of unlabelled data to capture underlying distribution as a pretrainer.
    b. Use small amount of labelled data to train a classifier.

Motivation

Motivation (Continued)

1. There is a lot of information in data, with or without labels
2. Labelled data is:
    a. Costly to get
    b. Time consuming
    c. Prone to human error
3. Unlabelled data is:
    a. Highly abundant in the world of internet
    b. Cheap to get
    c. Could be noisy though!

Motivation (Continued)

1. Massive pretrained models in Natural Language Processing (BERT, GPT, ELMo)
2. Unsupervised, task agnostic training to give better “initial” representation for downstream tasks as opposed to starting randomly in supervised learning

Why not Existing Base Nets?

1. Trained on unrelated data
2. Trained on unrelated labels
3. Features are not easily transferable

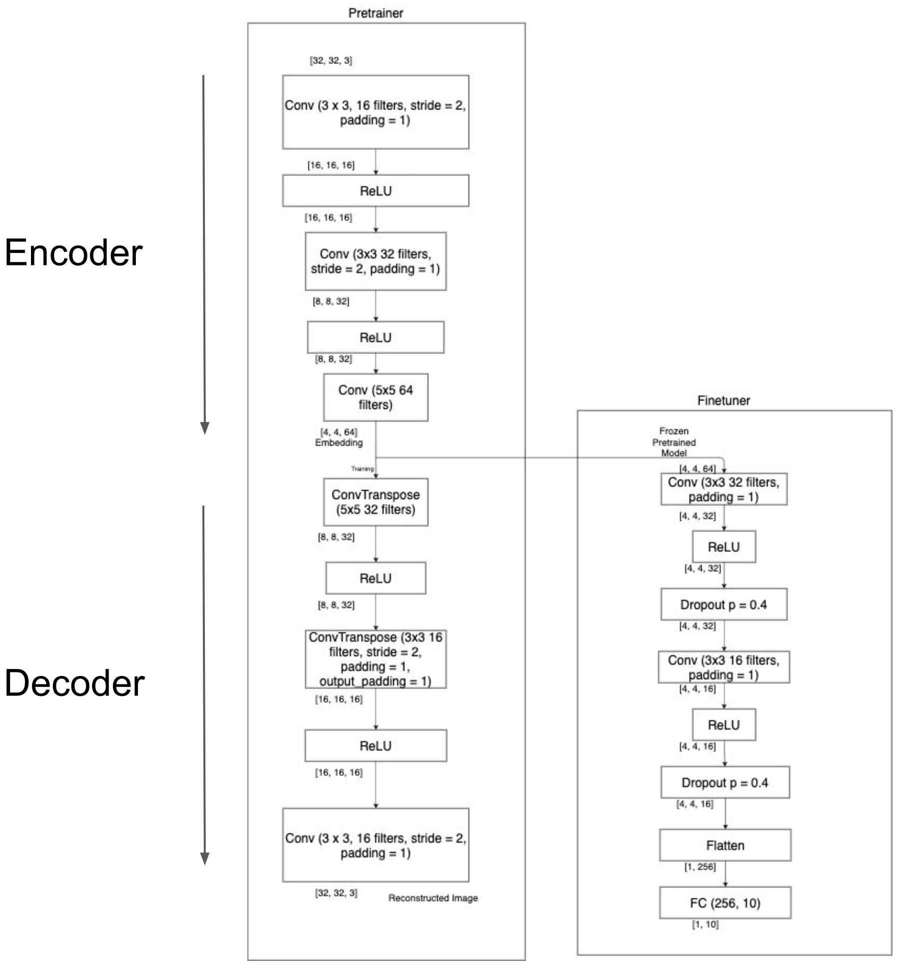
Our Hypothesis

1. If we keep the amount of labelled data fixed, and change the amount of unlabelled data that is used to train our model (coming up), then we can study how performance varies with unlabelled data.
2. We can compare this with the same model trained with the same amount of labelled data end to end (giving a much more specific model).

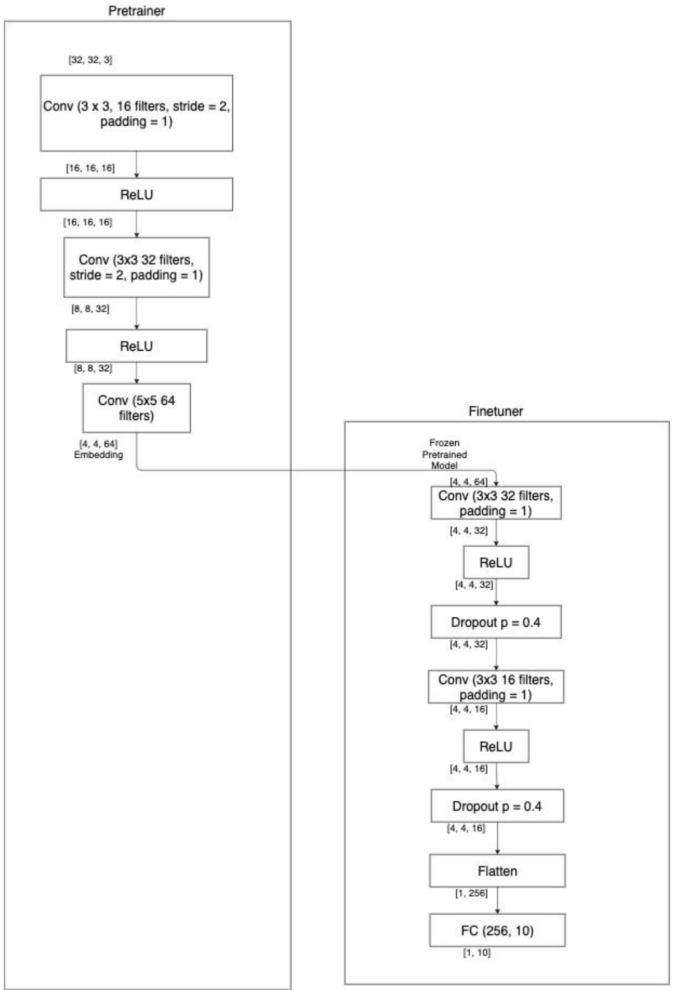
Metrics

1. F1-Score (accuracy)
2. Time to Accuracy
3. Wall Clock Time

Model

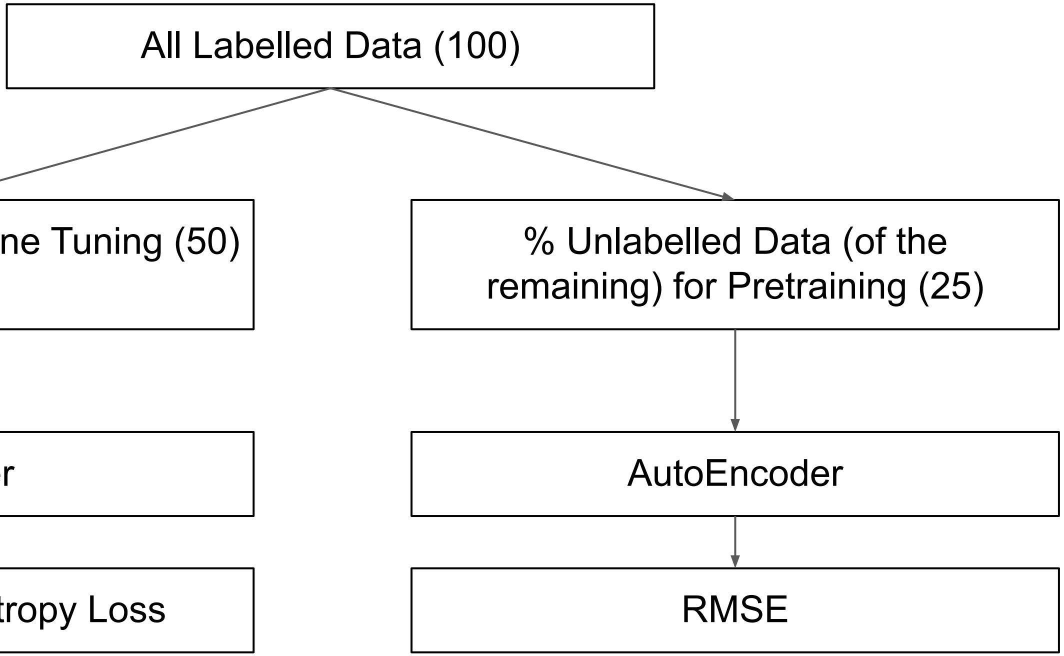

Encoder
Decoder
Pretrainer
[32, 32, 3]
Conv (3 x 3, 16 filters, stride = 2, padding = 1)
[16, 16, 16]
ReLU
[16, 16, 16]
Conv (3x3 32 filters, stride = 2, padding = 1)
[8, 8, 32]
ReLU
[8, 8, 32]
Conv (5x5 64 filters)
[4, 4, 64] Embedding
Training
ConvTranspose (5x5 32 filters)
[8, 8, 32]
ReLU
[8, 8, 32]
ConvTranspose (3x3 16 filters, stride = 2, padding = 1, output_padding = 1)
[16, 16, 16]
ReLU
[16, 16, 16]
Conv (3 x 3, 16 filters, stride = 2, padding = 1)
[32, 32, 3]
Reconstructed Image
Finetuner
Frozen Pretrained Model
[4, 4, 64]
Conv (3x3 32 filters, padding = 1)
[4, 4, 32]
ReLU
[4, 4, 32]
Dropout p = 0.4
[4, 4, 32]
Conv (3x3 16 filters, padding = 1)
[4, 4, 16]
ReLU
[4, 4, 16]
Dropout p = 0.4
[4, 4, 16]
Flatten
[1, 256]
FC (256, 10)
[1, 10]

Baseline Model

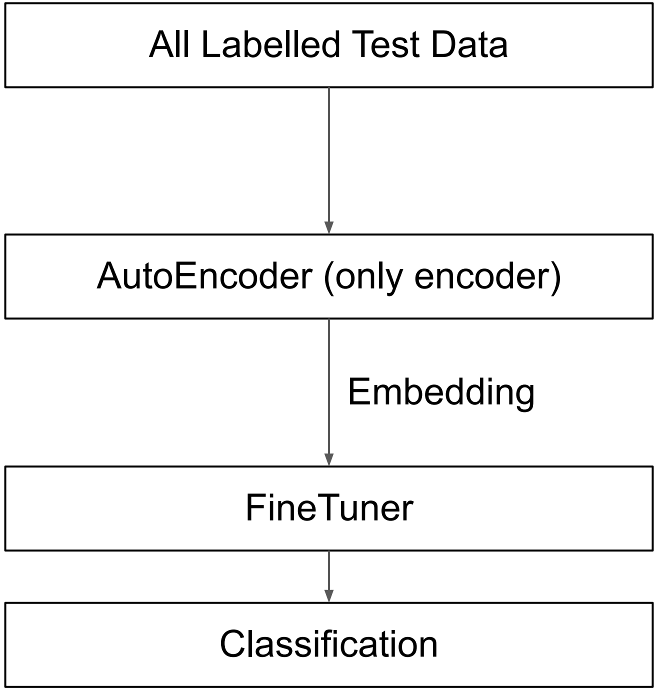


Our Approach (Training)

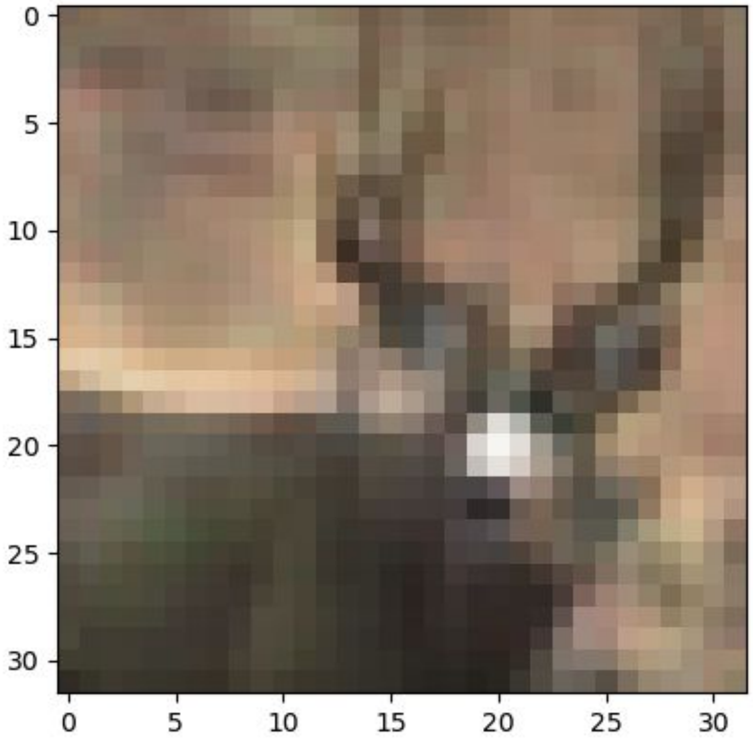
Example - 50% labelled data + 50% unlabelled data

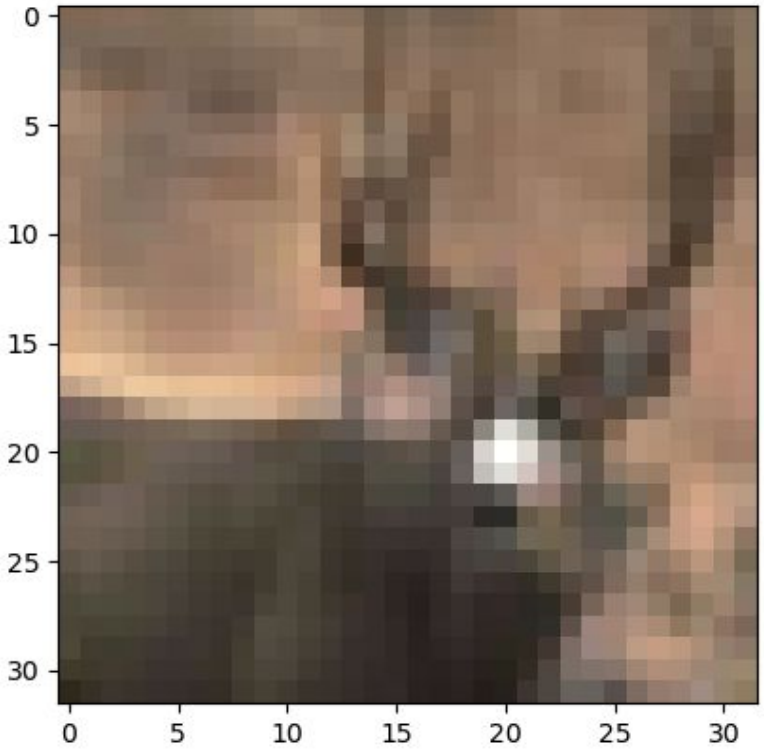


Our Approach (Testing)

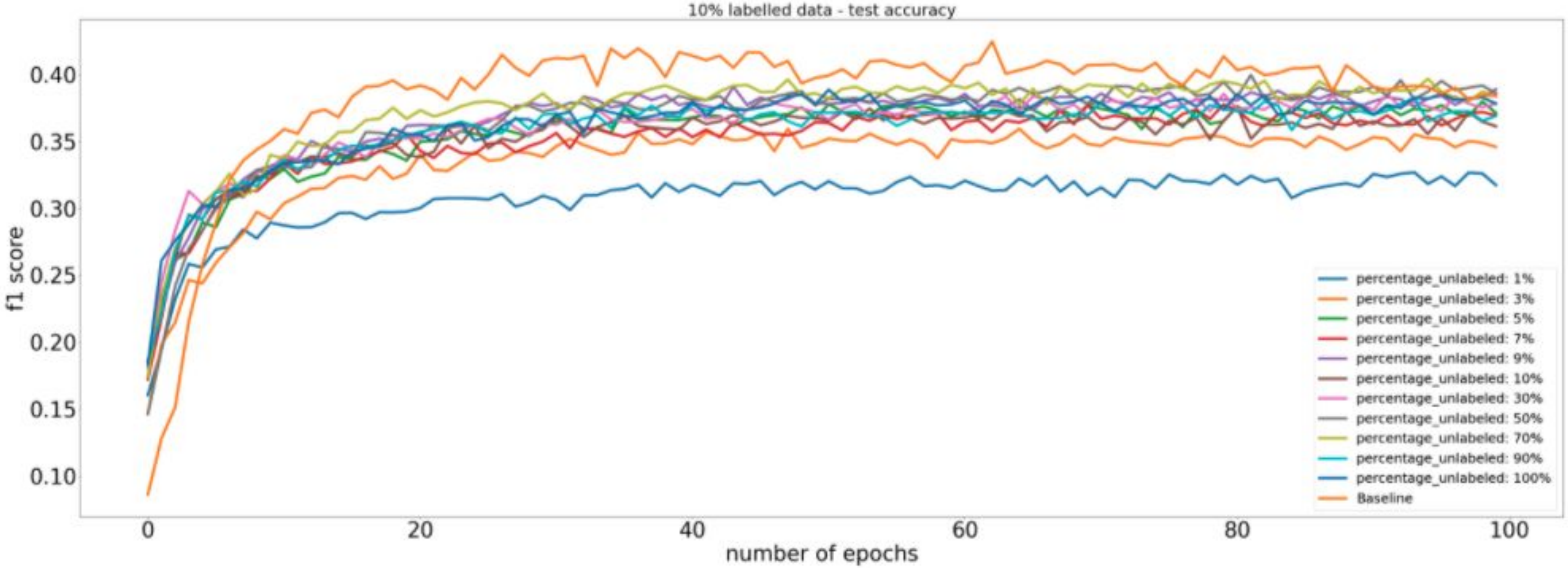


Results (AutoEncoder with 10% unlabeled data)

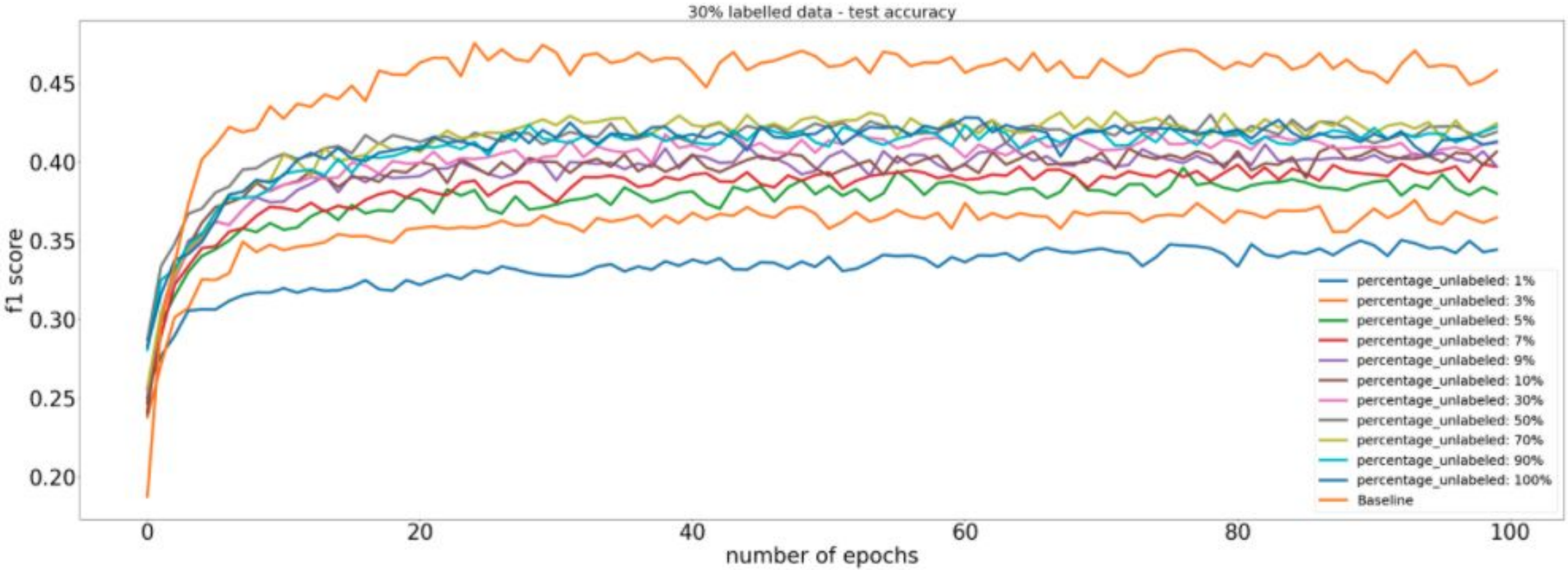




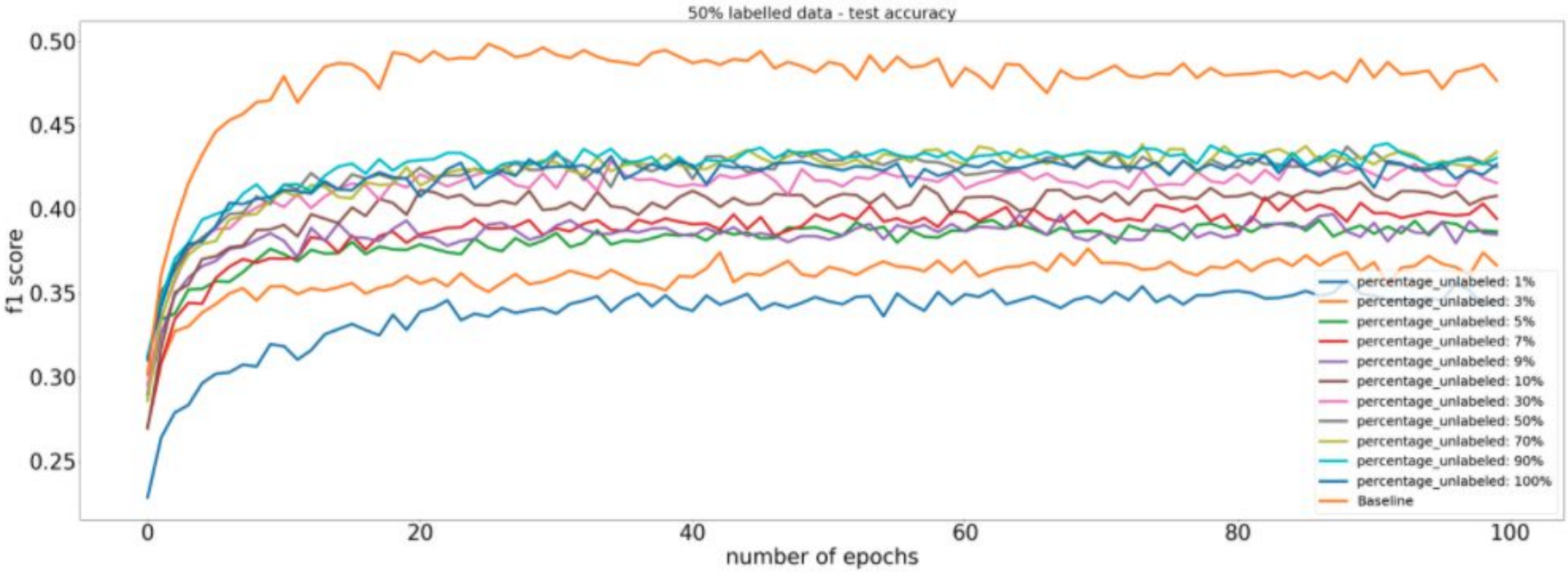
Results (F1-Score - 10% labeled data)



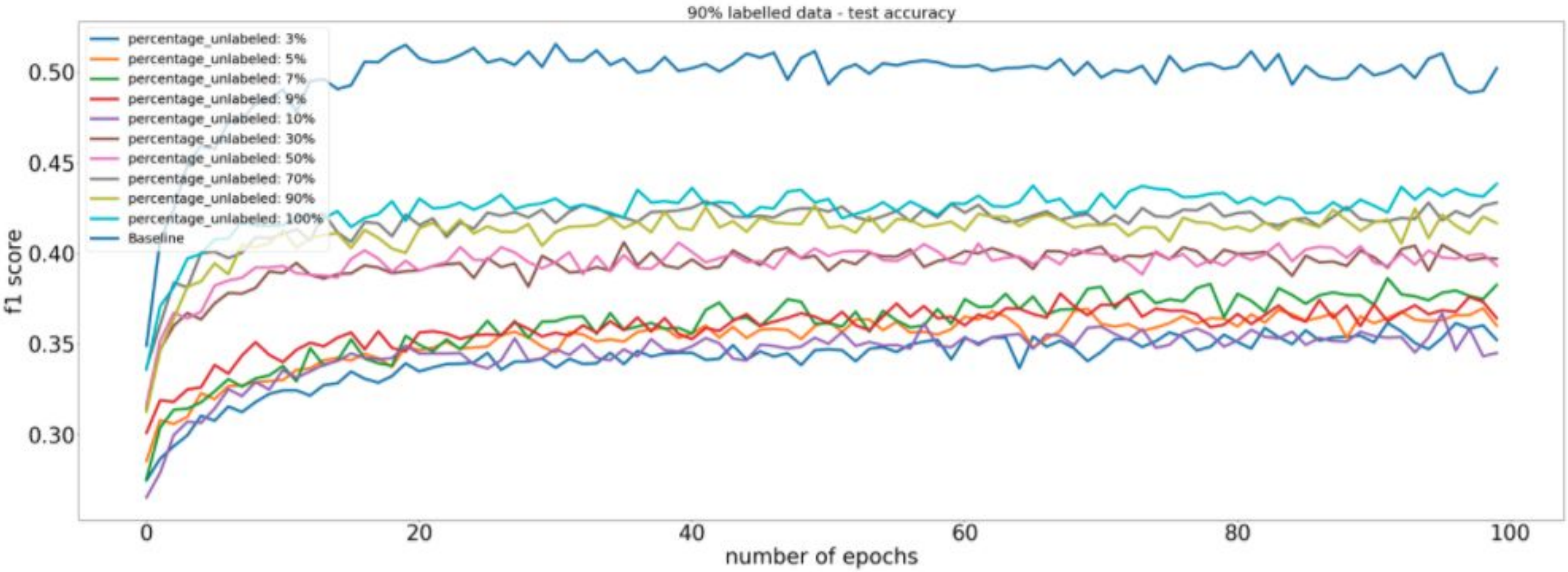
Results (F1-Score - 30% labeled data)



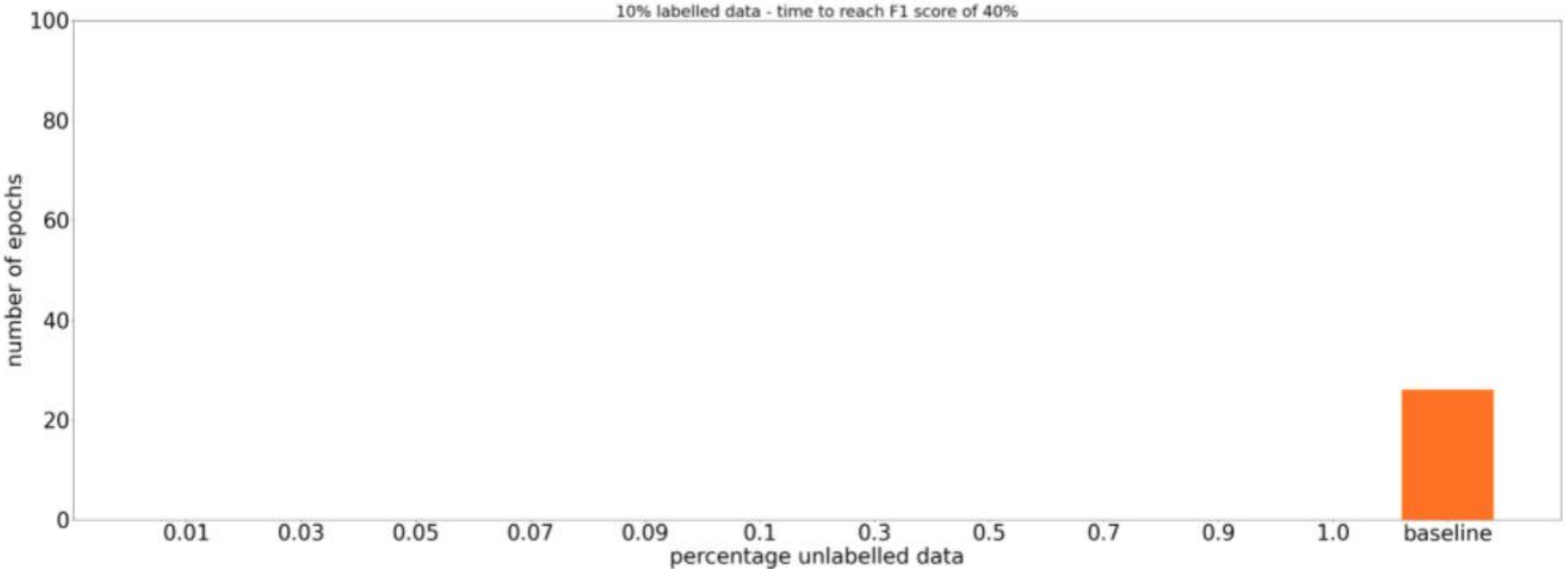
Results (F1-Score - 50% labeled data)



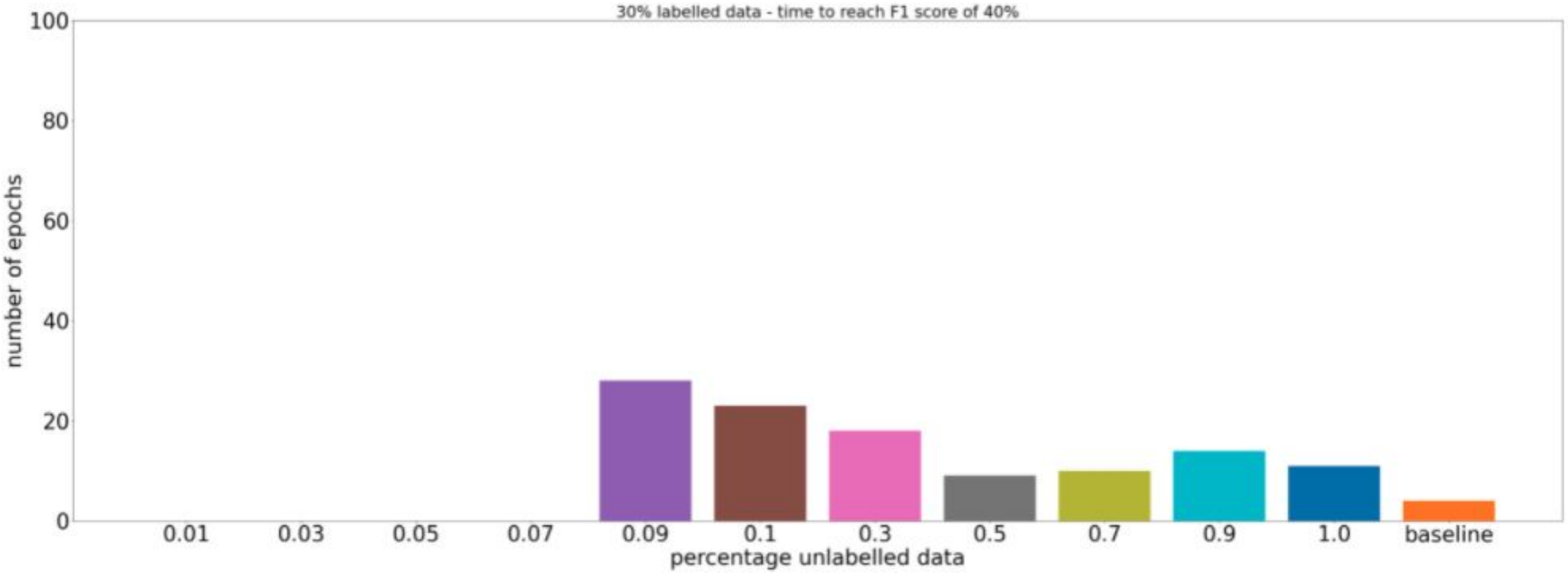
Results (F1-Score - 90% labeled data)



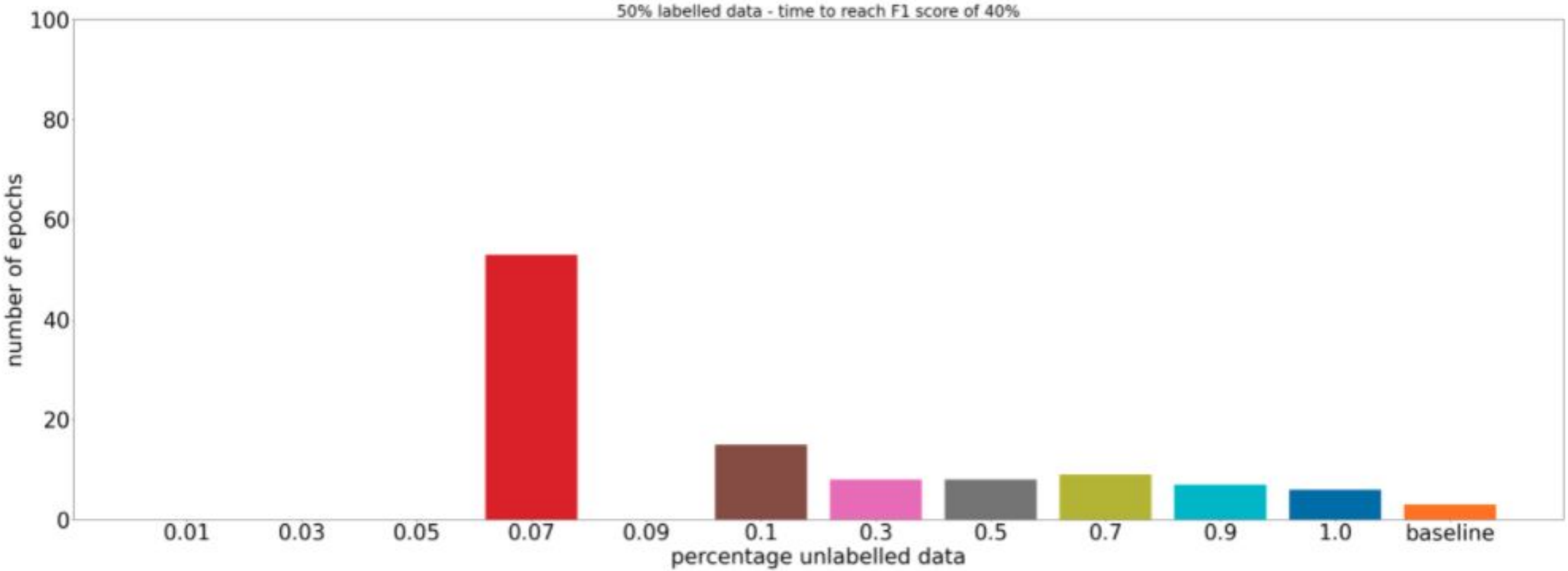
Results (Time To Accuracy - 10 % labeled data)



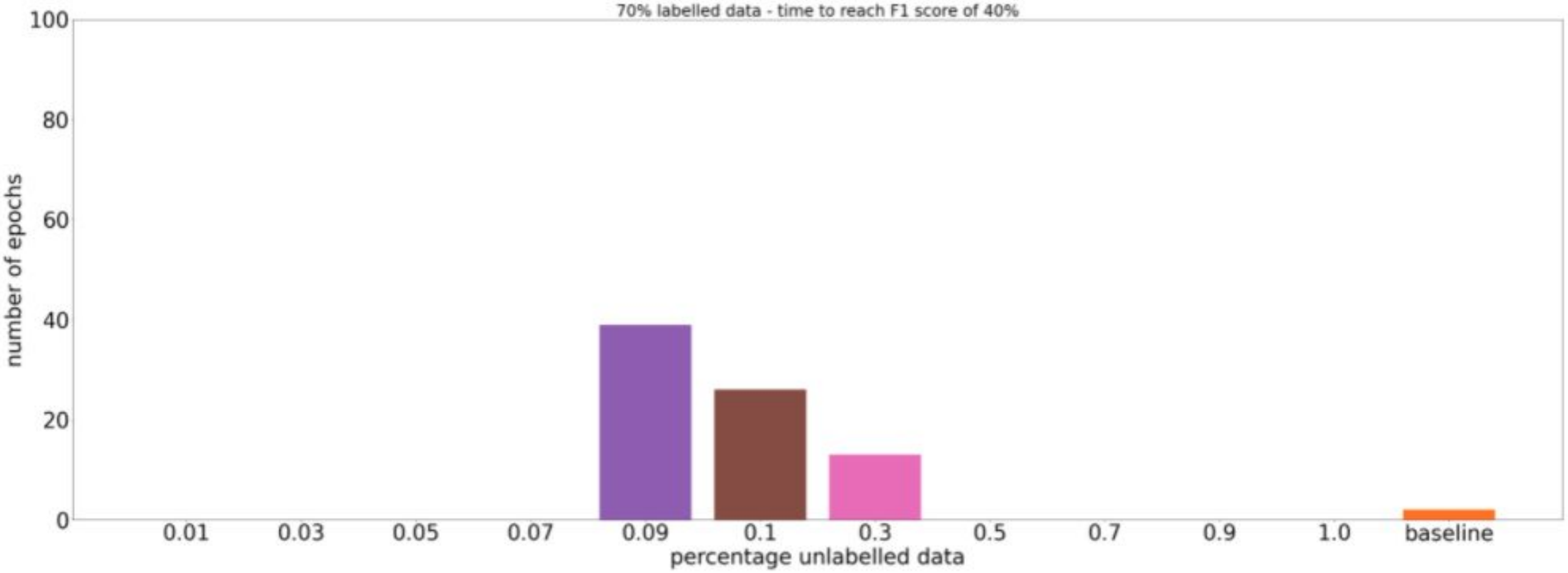
Results (Time To Accuracy - 30 % labeled data)



Results (Time To Accuracy - 50 % labeled data)



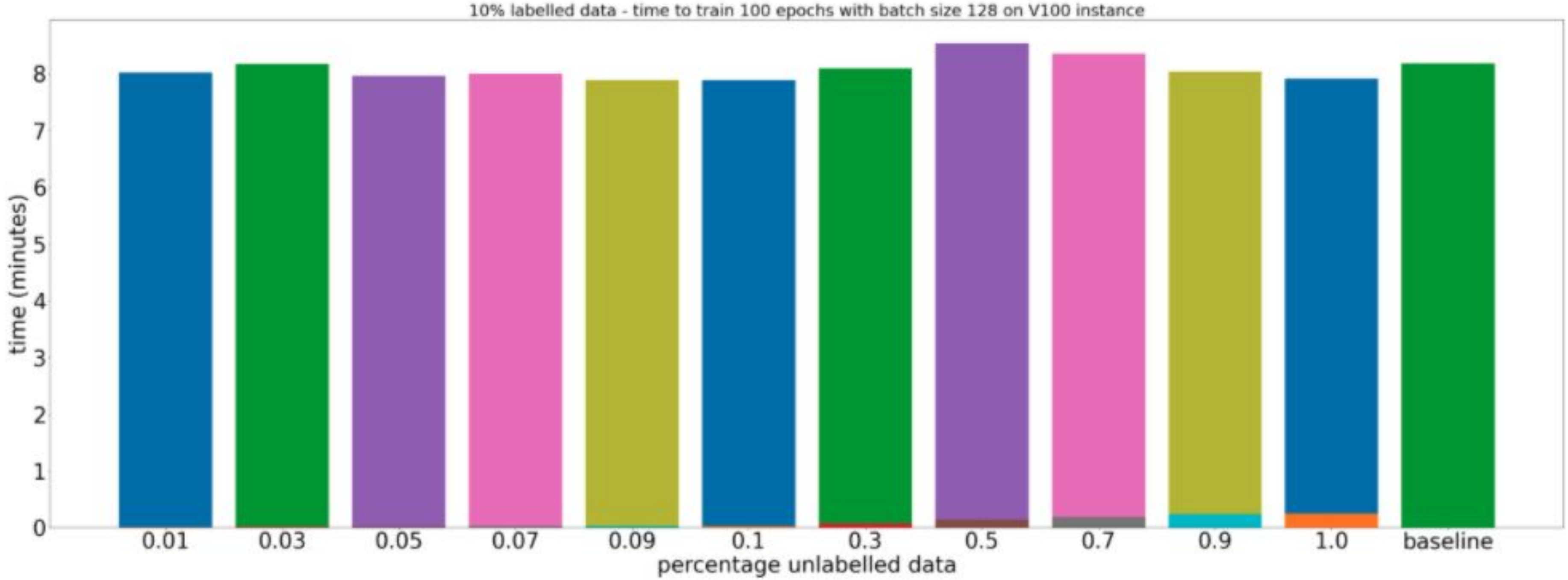
Results (Time To Accuracy - 70 % labeled data)



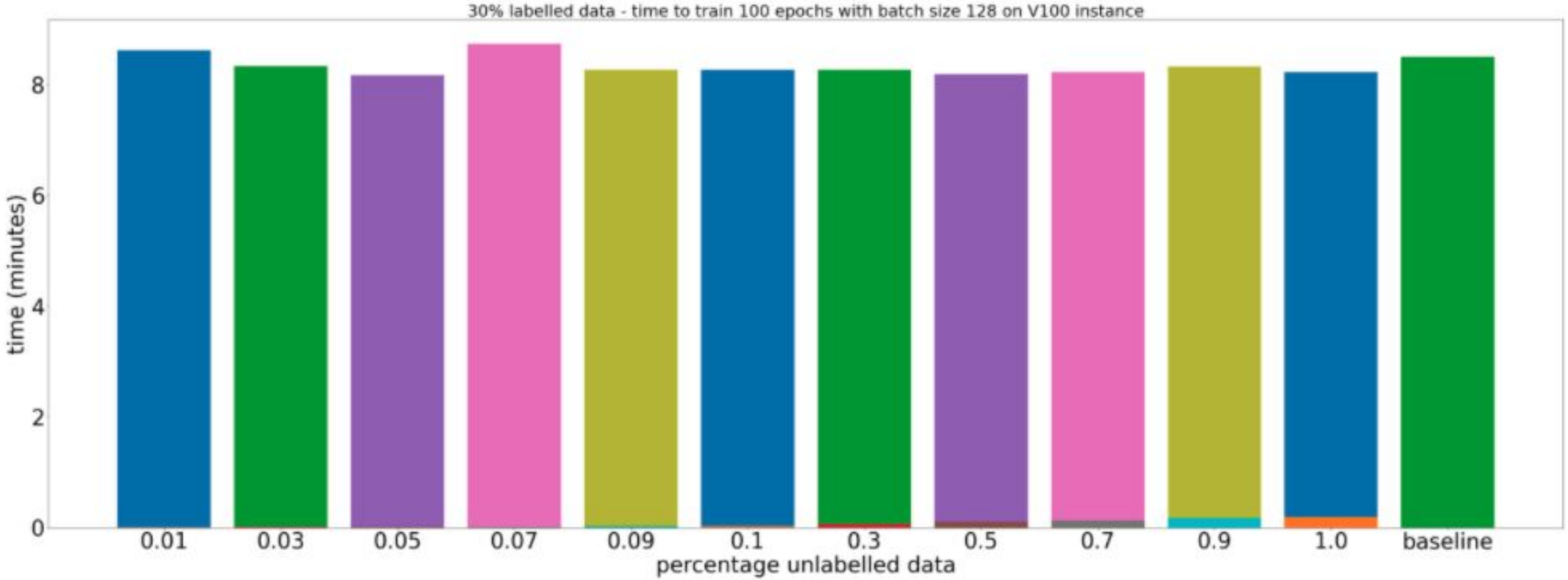
Observations

1. Increase in performance in the classification task if we increase the amount of data to the pretraining model
2. But increasing the amount of unlabeled data does not always mean that the performance improves.
3. Useful in practice because -
    a. For a fixed amount of labelled data we can easily get more unlabelled data
    b. When we find that the performance plateaus after some level of unlabelled, it is an indication that we may need to focus on the model complexity before collecting more data.

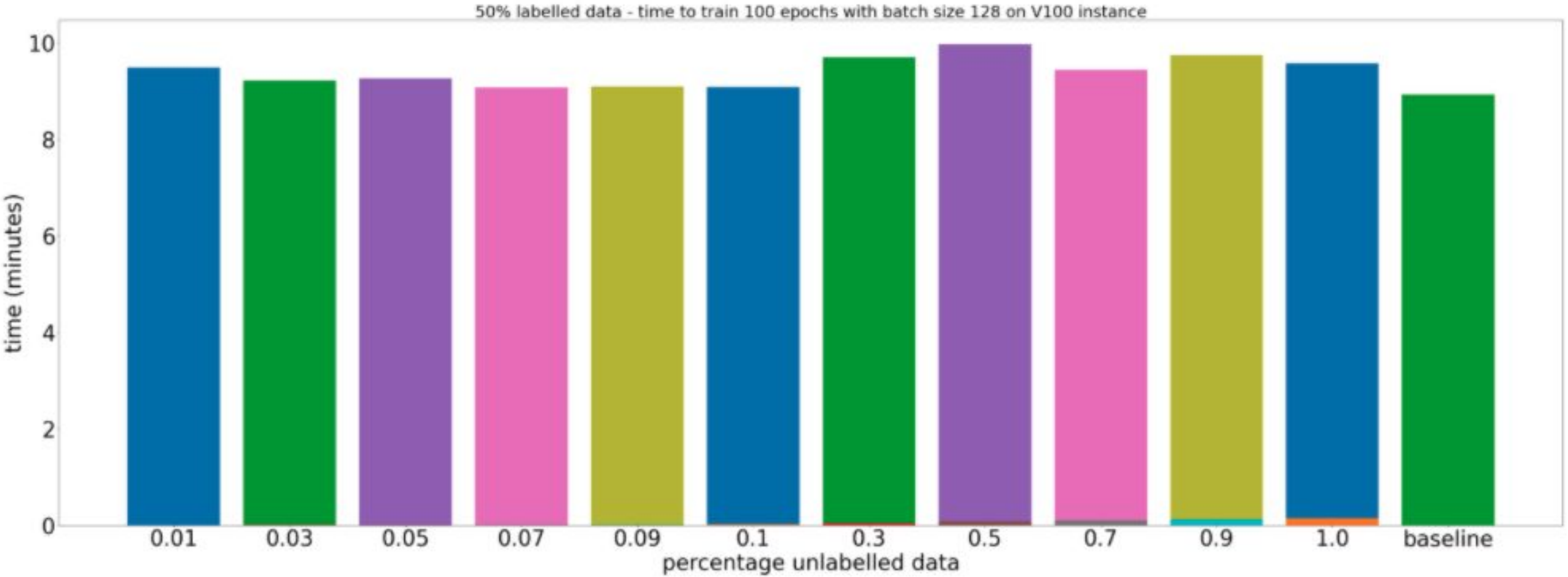
Time to train 100 epochs with 10% labeled data



Time to train 100 epochs with 30% labeled data



Time to train 100 epochs with 50% labeled data



Challenges we Faced

1. The biggest challenge that we faced was in getting the model to generalize well - solved this by data augmentation (Gaussian noise, vertical flip, and horizontal flip) and adding dropout.
2. Further, we faced the problem of pretrainer completely getting overridden during fine-tuning if we trained both networks (even if we had a less learning rate for the base net)
    a. It also made sense in this context, because we want to analyze the behavior with increasing unlabeled data. To override the model that uses unlabeled data defeats that purpose here.
    b. We hence froze the base net while finetuning.
3. Accuracy is still not great :(

Future Work

1. Use more powerful models
2. Test on multiple datasets
3. Use other methods for pretraining
    a. Contrastive Loss
    b. GANs

References

Papers (some of these papers were not directly used in implementing, but more for the purposes of learning for bettering our model’s performance):

1. VAE: https://openreview.net/pdf?id=Sy2fzU9gl
2. Understanding the Behaviour of Contrastive Loss: https://arxiv.org/pdf/2012.09740.pdf
3. Unsupervised Learning Using Generative Adversarial Training and Clustering: https://openreview.net/pdf/16c049b9a075af86f550b81707b62d5e520daed1.pdf
4. Transfusion: Understanding Transfer Learning for Medical Imaging: https://arxiv.org/pdf/1902.07208.pd

Blogs:

1. Self Supervised Learning
   https://towardsdatascience.com/self-supervised-learning-methods-for-computer-vision-c25ec10a91bd
2. Contrastive Loss:
   https://towardsdatascience.com/contrastive-loss-explaned-159f2d4a87ec

Thank You

Vishwesh Kirthivasan (vk2470)
Hitesh Agarwal (HA2598)
Prof. Parijat Dube