*Semi-supervised Leaf Pathology Classification*

Singh, Manmeet  
Charles W. Davidson College of Engineering  
San Jose State UniversitySan Jose, California  
[manmeet.singh01@sjsu.edu](mailto:manmeet.singh01@sjsu.edu)

Vadgama, Chinmay  
Charles W. Davidson College of Engineering  
San Jose State UniversitySan Jose, California  
[chinmay.vadgama@sjsu.edu](mailto:chinmay.vadgama@sjsu.edu)

Asha Balshiram, Aher  
Charles W. Davidson College of Engineering  
San Jose State UniversitySan Jose, California[ashabalshiram.aher@sjsu.edu](mailto:ashabalshiram.aher@sjsu.edu)   
  
  
Ronak Mehta  
Charles W. Davidson College of Engineering  
San Jose State UniversitySan Jose, California[ronak.mehta@sjsu.ed](mailto:ronak.mehta@sjsu.ed)u

***Abstract*—This e**

***Keywords—semi-supervised, simclr, contrastive, learning, augmentation, wide, deep***

# Introduction

Misdiagnosis of the many diseases impacting agricultural crops can lead to misuse of chemicals leading to the emergence of resistant pathogen strains, increased input costs, and more outbreaks with significant economic loss and environmental impacts. Current disease diagnosis based on human scouting is time-consuming and expensive, and although computer-vision based models have the promise to increase efficiency, the great variance in symptoms due to age of infected tissues, genetic variations, and light conditions within trees decreases the accuracy of detection.

We explore plant disease detection by following computer vision based deep learning architectures on a large plant image dataset. We explore traditional supervised architectures such as ResNet-50, InceptionResNetV2 etc. and compare it with new self-supervised learning approaches such as SimCLR architectures. Generally there is very little labeled dataset available for plants and specific diseases. Using self-supervised techniques such as SimCLR could provide ground for using large unlabeled dataset on top of learning from small labeled dataset. New data can be collected with help of drone based cameras without any manual help and that data could be used as an unlabeled source for SimCLR network. This could get us more generalization and model adaptation can be increased even if we don’t have higher accuracy than supervised techniques.

# Related Work [Ronak]

# Data

Initially,We were exploring the medical domain and chose Alzheimer disease detection. The Alzheimer classification was carried out on the OASIS-1 Alzheimer as well as Alzheimer’s severity dataset. OASIS-1 is a cross-sectional MRI data in young,middle aged, non demented and demented older adults. OASIS 1 has 416 subjects with 434 MRI scans. We carried out data engineering and data augmentation on this dataset and implemented SimCLR. There was a scope for accuracy improvement.

We carried out our classification on Plant disease detection. Two datasets were used in order to carry out the experiment. a) Plant village b) Plant pathology.

Plant pathology dataset has images of apple leaves and those are distinguished between healthy , scab & rust.Plant pathology has a total 3642 samples of above combination leaves.Plant pathology dataset contains different plant leaves including tomato,potato and bell paper.

# Methods

For the semi-supervised learning approach, we used simclrv2 which builds on top of contrastive learning based simclr. The contrastive approaches use heavy data augmentation followed by learning of the non linear transformation between representation and the contrastive loss which substantially improves the quality of learned representations. Lastly, the contrastive learning benefits from larger batch sizes compared to supervised learning. By combining these methodologies, we are able to achieve high performance with small amounts of labeled data.

Specific to simclrv2, a large amount of unlabeled data is used for unsupervised pre-training followed by supervised fine-tuning. This paradigm uses unlabeled data in a task-agnostic way initially. The key ingredient of this approach are the deep and wide networks during pretraining and fine-tuning. The fewer the labels in this approach, the more the pre-training benefits from this approach.

In our first approach, we used simclrv2 to perform unsupervised pre training on the OASIS-1 dataset. Next, we fine-tuned using the Alzheimer's severity dataset from Kaggle. After exhausting the possibilities for this approach, we pivoted to pre-training with plant village dataset and fine-tuned on the plant pathology dataset.

We trained our models on a combination of using the P100 GPU on the HPC machine, as well as the Google Colab environment.

# Experiments

Initially, we used SimCLRv2 to pretrain on the OASIS-1 dataset. The idea was to build an embedding space that could then be used to finetune on the Alzheimer’s severity dataset. This dataset included images from 4 classes ranging from normal to high severity.

After pre-training, we did not receive great accuracy during this methodology, likely related to the fact that we need a huge pre-training dataset to gain real benefits of the contrastive approach.

Next, we tried to pre-train on plant village dataset which is a multiclass classification dataset with 38 classes. Each class contains the leaf type as well as the disease/condition being experienced by the leaf. We achieved better accuracy after fine-tuning the plant village dataset on plant pathology dataset. We followed the standard procedure for fine tuning where we removed the first projection head, attached a fine-tuning layer to the pretrained model and backpropagation of the loss for this dataset. We were about to achieve 85% accuracy with this approach.

Detailed experiment results are shown in Appendix section.

# Conclusion

In this paper, we study how to develop contrastive learning of visual representations methods to accurately detect plant disease from plant leaf images.For the ease of open research in this area, we build SimCLR model using plant pathology dataset which contains leaf images with different combination of the leaf situations which includes healthy, scab & rust. This dataset is publicly available to the date. In Deep learning models, overfitting is high risk for data-hungry deep learning models. In order to tackle this problem we develop a model which is data efficient and is able to alleviate data deficiency. In this paper we propose an approach of a simple framework for contrastive learning of visual representations that learns expressive and unbiased visual feature representations that are robust to overfitting.In order to show effectiveness of our methods, we carry out extensive experiments.

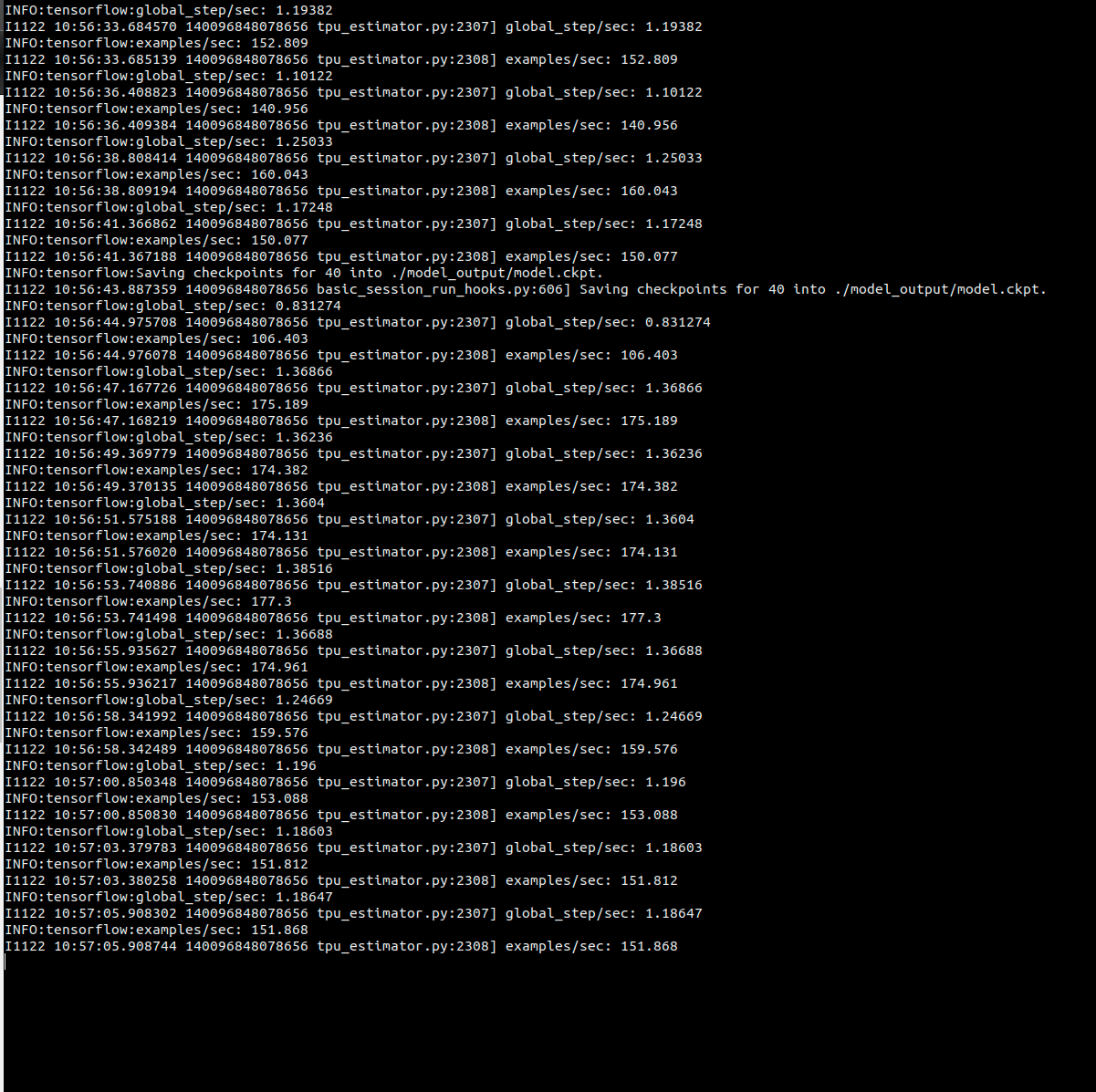
##### References

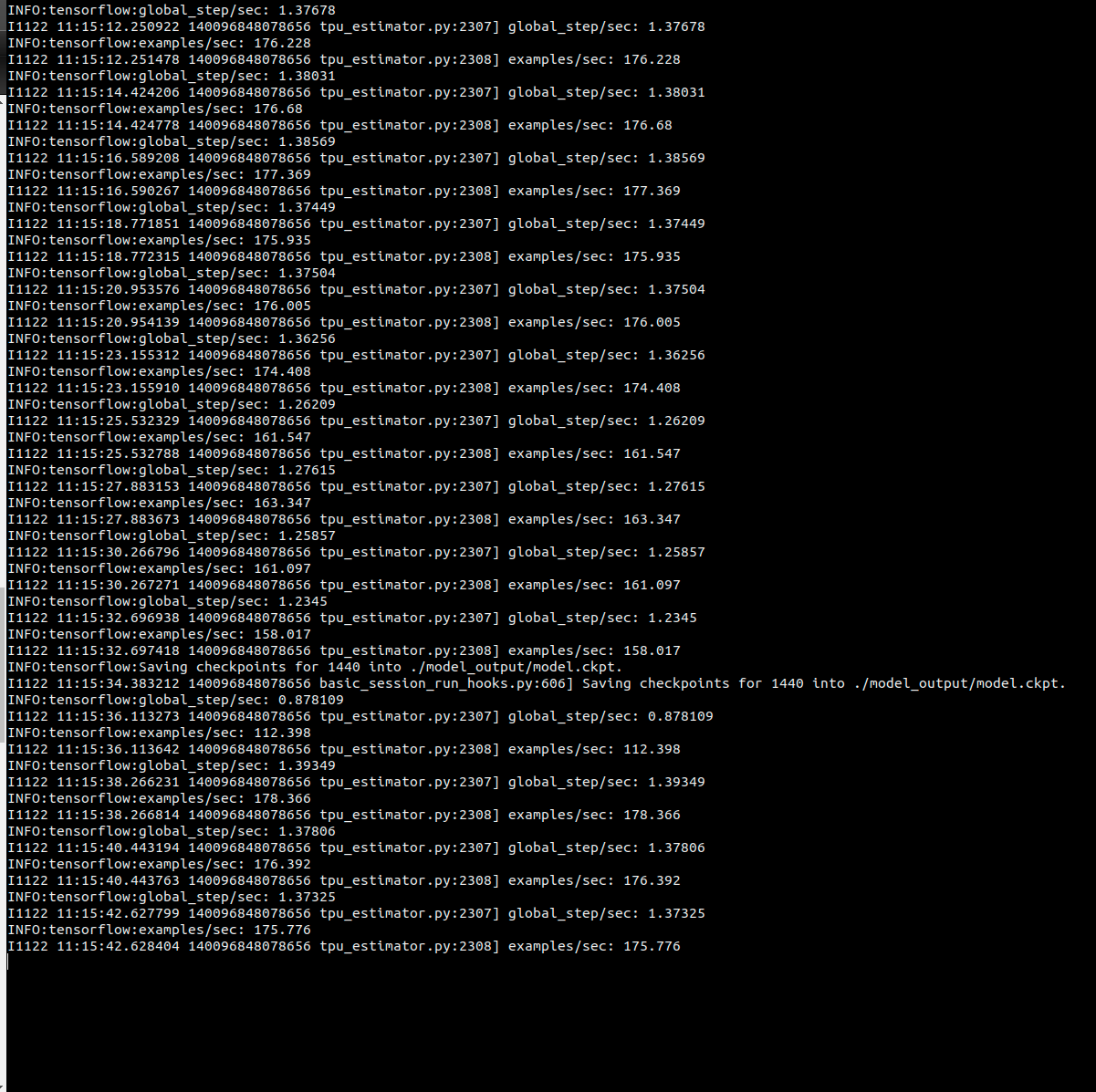
1. Ranjita Thapa1 , Noah Snavely2 , Serge Belongie2 , Awais Khan\*1,”The Plant Pathology 2020 challenge dataset to classify foliar disease of apples” arXiv:2004.11958

##### 

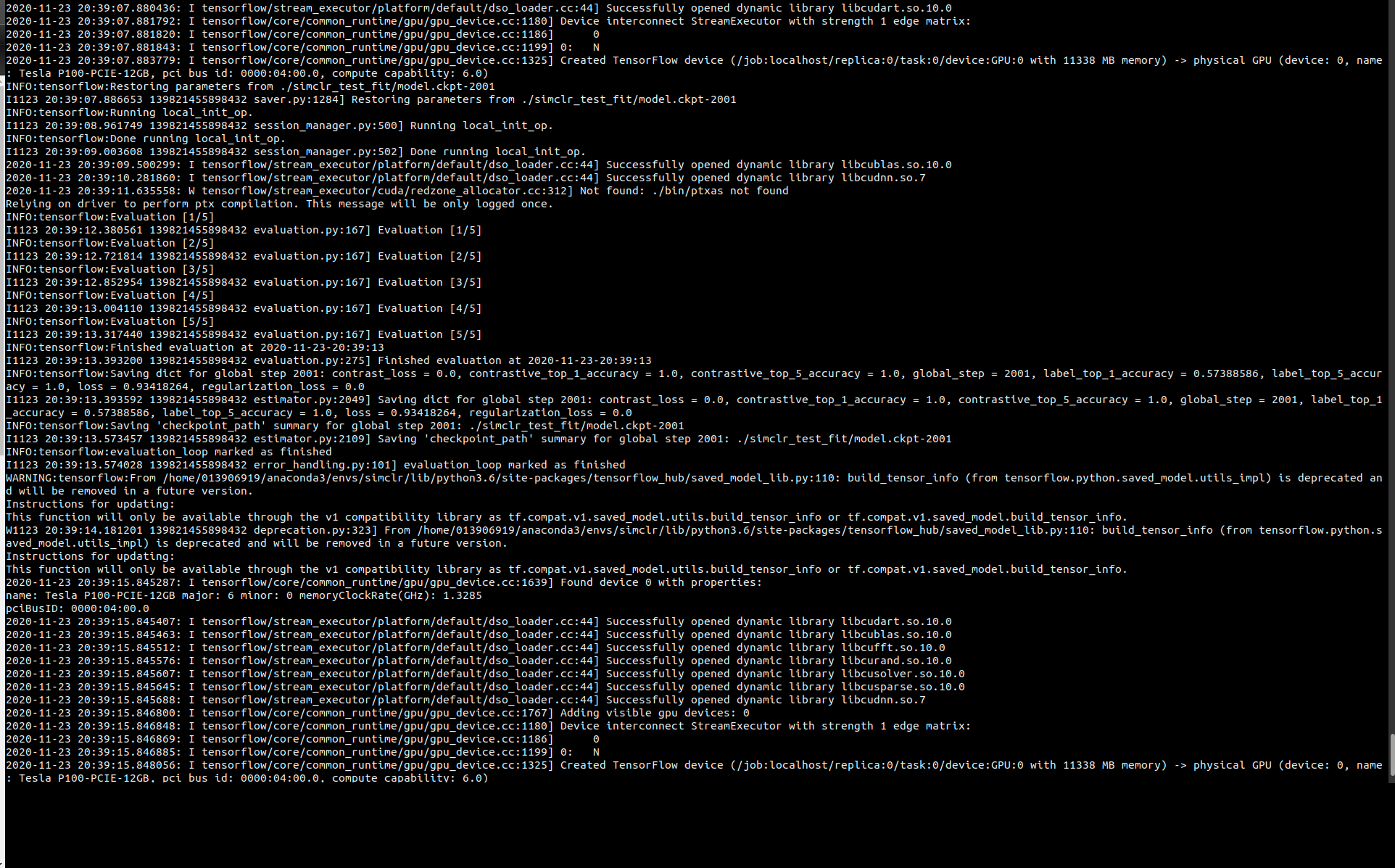
##### Appendix

**Alzheimers Dataset pretraining with OASIS-1 (on SJSU HPC)**

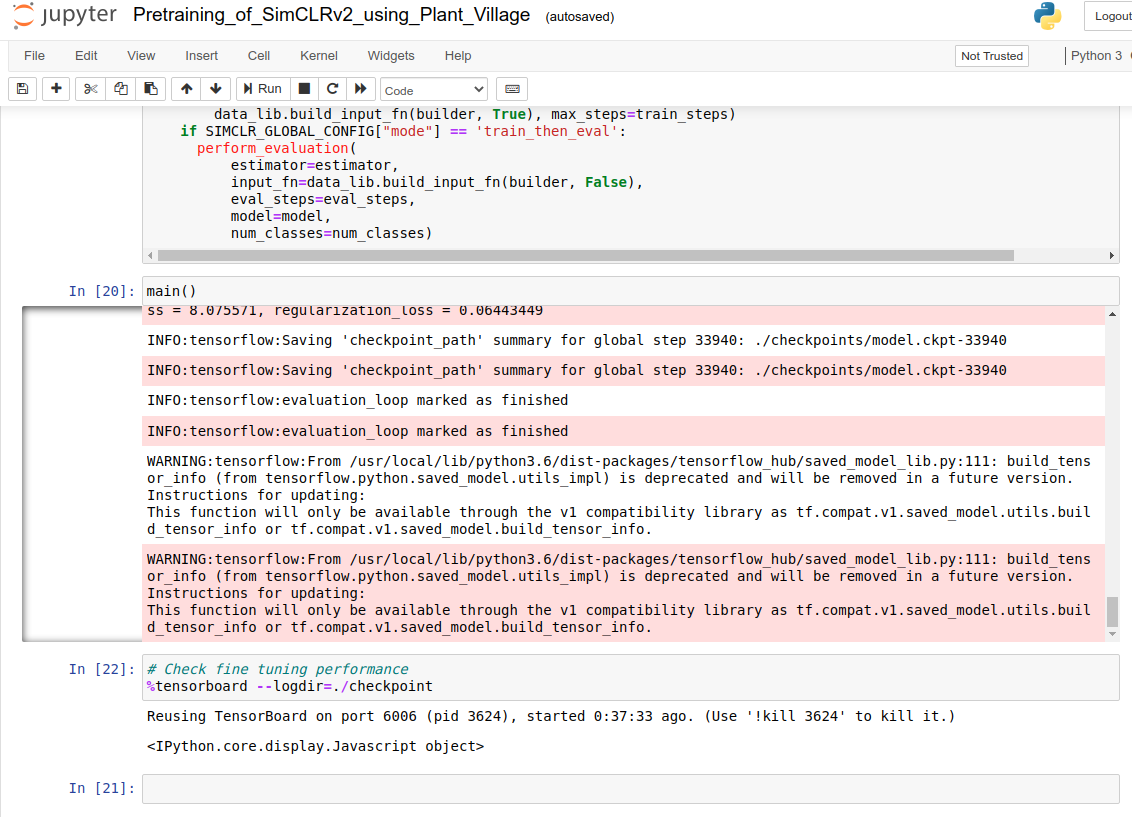




**Alzheimers Severity dataset fine-tuning**



**Plant Village Dataset pre-training**



**Plant Pathology Dataset Finetuning** (from Finuetuning\_and\_Distillation\_of\_SimCLRv2\_using\_Plant\_Pathology.ipynb)

[Iter 1] Loss: 1.7947008609771729 Top1 Accuracy: 0.328125

[Iter 2] Loss: 6.353813171386719 Top1 Accuracy: 0.3125

[Iter 3] Loss: 11.957975387573242 Top1 Accuracy: 0.359375

[Iter 4] Loss: 10.061290740966797 Top1 Accuracy: 0.34375

[Iter 5] Loss: 2.5968565940856934 Top1 Accuracy: 0.625

[Iter 6] Loss: 2.488285779953003 Top1 Accuracy: 0.640625

[Iter 7] Loss: 3.117185115814209 Top1 Accuracy: 0.46875

[Iter 8] Loss: 4.863947868347168 Top1 Accuracy: 0.546875

[Iter 9] Loss: 6.003618240356445 Top1 Accuracy: 0.578125

[Iter 10] Loss: 5.8602190017700195 Top1 Accuracy: 0.59375

[Iter 11] Loss: 1.8799350261688232 Top1 Accuracy: 0.671875

[Iter 12] Loss: 6.011013031005859 Top1 Accuracy: 0.34375

[Iter 13] Loss: 1.7753324508666992 Top1 Accuracy: 0.78125

[Iter 14] Loss: 3.5107946395874023 Top1 Accuracy: 0.640625

[Iter 15] Loss: 2.1833629608154297 Top1 Accuracy: 0.765625

[Iter 16] Loss: 2.1224470138549805 Top1 Accuracy: 0.6875

[Iter 17] Loss: 0.8193728923797607 Top1 Accuracy: 0.828125

[Iter 18] Loss: 1.5057377815246582 Top1 Accuracy: 0.796875

[Iter 19] Loss: 1.2158958911895752 Top1 Accuracy: 0.828125

[Iter 20] Loss: 1.0019350051879883 Top1 Accuracy: 0.828125

----------Evaluate the model----------

[Iter 1] test\_loss: 1.6662030220031738 test\_top1\_accuracy: 0.78125

[Iter 2] test\_loss: 1.8175444602966309 test\_top1\_accuracy: 0.75

[Iter 3] test\_loss: 0.9973149299621582 test\_top1\_accuracy: 0.828125

[Iter 4] test\_loss: 0.7481752634048462 test\_top1\_accuracy: 0.859375

[Iter 5] test\_loss: 0.8586392402648926 test\_top1\_accuracy: 0.796875

{'block\_group1': <tf.Tensor 'model/StatefulPartitionedCall:0' shape=(34, 64, 64, 256) dtype=float32>, 'block\_group2': <tf.Tensor 'model/StatefulPartitionedCall:1' shape=(34, 32, 32, 512) dtype=float32>, 'block\_group3': <tf.Tensor 'model/StatefulPartitionedCall:2' shape=(34, 16, 16, 1024) dtype=float32>, 'block\_group4': <tf.Tensor 'model/StatefulPartitionedCall:3' shape=(34, 8, 8, 2048) dtype=float32>, 'initial\_max\_pool': <tf.Tensor 'model/StatefulPartitionedCall:6' shape=(34, 64, 64, 64) dtype=float32>, 'final\_avg\_pool': <tf.Tensor 'model/StatefulPartitionedCall:4' shape=(34, 2048) dtype=float32>, 'logits\_sup': <tf.Tensor 'model/StatefulPartitionedCall:7' shape=(34, 1000) dtype=float32>, 'initial\_conv': <tf.Tensor 'model/StatefulPartitionedCall:5' shape=(34, 128, 128, 64) dtype=float32>}

Variables to train: [<tf.Variable 'model/head\_supervised\_new/kernel:0' shape=(2048, 4) dtype=float32>, <tf.Variable 'model/head\_supervised\_new/bias:0' shape=(4,) dtype=float32>]

[Iter 6] test\_loss: 0.9046571254730225 test\_top1\_accuracy: 0.8529411764705882

----------Results----------

precision recall f1-score support

healthy 0.54 0.50 0.52 363

multiple\_diseases 0.11 0.07 0.08 59

rust 0.65 0.66 0.65 429

scab 0.64 0.69 0.66 429

accuracy 0.60 1280

macro avg 0.48 0.48 0.48 1280

weighted avg 0.59 0.60 0.59 1280

