Plant Disease Classification Using Contrastive Learning

Project Report

Submitted to

Prof. Vijay Eranti

CMPE-297 Special Topics

By

Bharath Gunasekaran

Tamanna Mehta

Riddhi Jain

Stuti Agarwal

Dec, 2021

Table of Contents

ABSTRACT 3

INTRODUCTION 5

RELATED WORK 7

DATASETS 8

METHODOLOGY 9

MLOPS 10

TFX Pipeline 10

- **1. Import Example Gen** 10
- **2. Statistics Gen** 11
- **3. Schema Gen** 11
- **4. Example Validator** 11
- **5. Transform** 11
- **6. Trainer** 11
- **7. Pusher** 12

EXPERIMENTS AND RESULTS 13

TENSOR BOARD VISUALIZATION: 15

DEPLOYMENT ON KUBEFLOW 16

FLASK APPLICATION 16

CONCLUSION & FUTURE WORK 19

REFERENCES 20

ABSTRACT

Deep learning is a branch of artificial intelligence. In recent years, with the advantages of automatic learning and feature extraction, it has been widely concerned by academic and industrial circles. It has been widely used in image and video processing, voice processing, and natural language processing. At the same time, it has also become a research hotspot in the field of agricultural plant protection, such as plant disease recognition and pest range assessment, etc. The application of deep learning in plant disease recognition can avoid the disadvantages caused by artificial selection of disease spot features, make plant disease feature extraction more objective, and improve the research efficiency and technology transformation speed. This review provides the research progress of deep learning technology in the field of crop leaf disease identification in recent years.

In this, we present the current trends and challenges for the detection of plant leaf disease using deep learning and advanced imaging techniques. A key ingredient of our approach is the use of big (deep and wide) networks during pretraining and fine-tuning. We find that, the fewer the labels, the more this approach (task-agnostic use of unlabeled data) benefits from a bigger network. After fine-tuning, the big network can be further improved and distilled into a much smaller one with little loss in classification accuracy by using the unlabeled examples for a second time, but in a task-specific way. The proposed semi-supervised learning algorithm can be summarized in three steps: unsupervised pretraining of a big ResNet model using SimCLRv2, supervised fine-tuning on a few labeled examples, and distillation with unlabeled examples for refining and transferring the task-specific knowledge. This procedure achieves 73.9% ImageNet top-1 accuracy with just 1% of the labels (≤13 labeled images per class) using ResNet-50, a 10×

improvement in label efficiency over the previous state-of-the-art. With 10% of labels, ResNet-50 trained with our method achieves 77.5% top-1 accuracy, outperforming standard supervised training with all the labels.

INTRODUCTION

The occurrence of plant diseases has a negative impact on agricultural production. If plant diseases are not discovered in time, food insecurity will increase. Early detection is the basis for effective prevention and control of plant diseases, and they play a vital role in the management and decision-making of agricultural production. In recent years, plant disease identification has been a crucial issue.

Disease-infected plants usually show obvious marks or lesions on leaves, stems, flowers, or fruits. Generally, each disease or pest condition presents a unique visible pattern that can be used to uniquely diagnose abnormalities. Typically, the leaves of plants are the primary source for identifying plant diseases, and most of the symptoms of diseases may begin to appear on the leaves.

Currently, agricultural and forestry experts are used to identify on-site or farmers identify fruit tree diseases and pests based on experience. This method is not only subjective, but also time-consuming, expensive, laborious, and inefficient. Farmers with less experience may misjudgment and use drugs blindly during the identification process. 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. To counter these challenges, research into the use of image processing techniques for plant disease recognition has become a hot research topic.

In this, we explore plant disease detection by using computer vision based deep learning architectures on a large plant image dataset. We compared the traditional supervised architectures such as ResNet-50, InceptionResNetV2 etc. with new self-supervised learning approaches such as SimCLR architectures. Generally the main problem is that there is very less labeled dataset

available for plants leaves and specific diseases. Using self-supervised techniques such as SimCLR would help in using large unlabeled dataset on top of learning from small labeled dataset. To collect more data, drone based cameras without any manual help are used to capture plant images. The data collected is used as an unlabeled source for the SimCLR network.

RELATED WORK

In recent years, a lot of research has been done on using deep learning and computer vision techniques in the study of Plant disease detection. There are some research papers previously presented to summarize the research about agriculture (including plant disease recognition) but they lacked some recent developments in terms of visualization techniques implemented along with the Deep Learning and modified the famous DL models, which were used for plant disease identification. Lots of research has been done using different approaches such as traditional computer vision algorithms like image segmentation, Support Vector Machines (SVM), Artificial Neural Networks (ANN) and Convolutional Neural Network (CNN). Variety of CNN models such as AlexNet, GoogleNet, VGGNet etc. have been used in Deep Learning based plant disease classification. It often comes as a challenge to find the perfect dataset size; lots of classes in multi-class classification need careful tuning of hyperparameters in order to avoid overfitting.
In our project, we have used a deep learning method called simclrv2 for plant disease recognition, which is built on top of contrastive learning based Simclr. The approach uses heavy data augmentation techniques followed by learning of the non-linear transformation between representation and the contrastive loss which substantially improves the quality of learned representations. In conclusion, contrastive learning benefits from larger batch sizes compared to supervised learning. By amalgamating these methodologies, we are able to achieve high performance with small amounts of labeled data.

DATASETS

Dataset for Pre-Training: Initially, the Simclrv2 model is trained with Plant Village dataset. The model requires data in tfds format. The Plant Village dataset consists of 54303 healthy and unhealthy leaf images divided into 38 categories by species and disease.The dataset includes images and its corresponding labels as features.

Dataset for Fine-tuning and Distillation: The pretrained model is fine-tuned and distilled with plant pathology dataset.Plant pathology dataset consist of apple leaves images which are classified as healthy, rust, scab and multi disease. The dataset includes 1821 samples.

METHODOLOGY

Contrastive Learning using Simclrv2:

As discussed earlier, We have used Contrastive learning using Simclrv2 for plant disease classification. Simclrv2 is a semi-supervised learning method for learning from few labeled examples while making the best use of a large amount of unlabeled data.The Simclrv2 is an advanced version of simclr that includes fine-tuning and distillation.The Simclrv2 uses Resnet50 model. The contrastive learning technique applies heavy data augmentation on each input image followed by learning of the non-linear transformation between representation and the contrastive loss, which substantially improves the quality of learned representations. This approach uses deep and wide networks during pretraining and fine-tuning. The fewer the labels in this approach, the more the pretraining benefits from this approach.With the use of this methodology, we are able to achieve higher accuracy for our classification task.

The Plant Pathology dataset images are first preprocessed from one hot encoding to their corresponding label, where each image belongs to one of the 4 classes. We assign label 0 for healthy leaf, 1 for rust, 2 for scab and 3 for multi disease. Heavy data augmentation such as (1) random crop and resize, (2) random color distortions, and (3) random Gaussian blur are applied to each preprocessed input image before feeding into the pretrained model. SimCLRv2 tries to maximize agreement between 2 augmented versions of the same image.The pretrained model is fine-tuned using a linear layer in order to classify plant leaf.

MLOPS

MLOPS is a compound of machine learning + information technology. It is a new discipline/focus/practice for collaboration and communication between data scientists and information technology (IT) professionals while automating and productizing machine learning algorithms.

TFX Pipeline

TFX pipeline is a sequence of components that implements a Machine Learning pipeline which is specifically designed for scalable, high-performance machine learning tasks. That includes modeling, training, serving inference, and managing deployments to online, native mobile, and web applications.

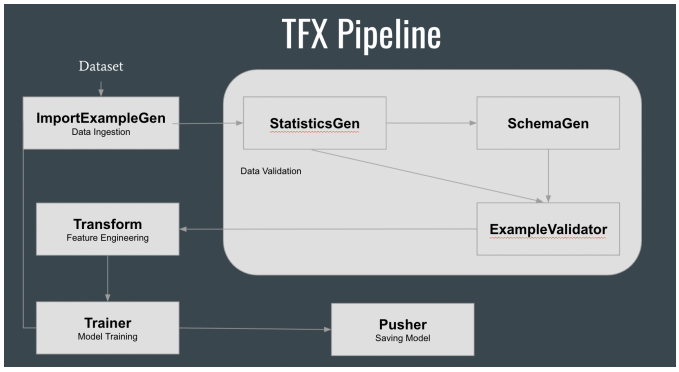

Figure 6: TFX Pipeline flow.

We have implemented following components:

1. Import Example Gen

It is an initial component of a pipeline which reads data as tfRecords and splits it into training and validation dataset and finally saves as tfRecord files for future access.

2. Statistics Gen

It is the second step of the pipeline which calculates statistics for both the train and validation data. It takes tf record files generated by import example gen as input.

3. Schema Gen

It is the third step of the pipeline which generates the schema of the tf record files generated by the import example gen component. It examines the characteristics such as data type for each column in the input dataset.

4. Example Validator

It is equivalent to the data validation step of a traditional machine learning pipeline. Output from both the Statistics Gen and Schema Gen is then passed on to the example validator to detect if there are any anomalies present in both the train and validation dataset.

5. Transform

It is equivalent to the feature engineering step of the traditional machine learning pipeline. Import example gen components generated tfRecord files are fed as an input to the transform step and data augmentation such as Rotation, ReScaling, flip etc.

6. Trainer

This component takes the model which we are planning to use for our problem and the transformed data as input. It then trains the model for the specified number of epochs. Once the model is trained on the training set, a validation set is used to calculate the accuracy of the

model. After completing the weights of the model, the model is saved as a saved_model folder for future deployment.

7. Pusher

This is the last component which is used to push the saved model and deploy it on the server url specified along with the weights learned so far in the training step. We can visualize the output of a pusher using its output method.

EXPERIMENTS AND RESULTS

The simclrv2 Resnet50 model is first pre-trained with a plant village dataset that contains images belonging to one of the 38 categories. The model requires data in tfds format. Once the model is pre-trained, the weight are used for further fine-tuning and distillation using plant disease 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 back propagated the loss for this dataset. We used LARS as an optimizer that implements the Layer-wise Adaptive Moments. With this contrastive learning approach, our model provide a good accuracy of 89 percent, as shown in figure.

```
[Iter 19] Loss: 1.4665998220443726 Top1 Accuracy: 0.765625
[Iter 20] Loss: 0.48078209161758423 Top1 Accuracy: 0.8125
****Model Evaluation****
[Iter 1] test_loss: 0.9316287040710449 test_top1_accuracy: 0.890625
[Iter 2] test_loss: 1.2279168367385864 test_top1_accuracy: 0.8125
[Iter 3] test_loss: 0.7691038250923157 test_top1_accuracy: 0.890625
[Iter 4] test_loss: 0.7224487066268921 test_top1_accuracy: 0.890625
[Iter 5] test_loss: 1.0738524198532104 test_top1_accuracy: 0.875
[Iter 6] test_loss: 0.5262660384178162 test_top1_accuracy: 0.8823529411764706
****Results****
              precision    recall  f1-score   support

           0       0.90      0.93      0.92       368
           1       0.30      0.31      0.30        61
           2       0.94      0.94      0.94       426
           3       0.91      0.88      0.89       425

    accuracy                           0.89      1280
   macro avg       0.76      0.76      0.76      1280
weighted avg       0.89      0.89      0.89      1280
```

Figure 1: Screenshot of Accuracy achieved using Pretrained Model

We also created an end to end production pipeline i.e.TFX pipeline for our model. For the TFX pipeline, first, the augmentation is applied to preprocessed images of the plant pathology dataset.

On the top of that, TF records are created which includes images in byte format and labels in integer format as the features.

Hyperparameter	**Value**
Momentum	0.9
Learning Rate	0.1
Weight_Decay	0.1
Total_Iterations	15
EETA_DEFAULT	.001
Batch_size	64

Table 1 : Hyper parameters for Fine-Tuning the model

Tfx pipeline includes everything from feature engineering to model training and predicting. Each component of the tfx pipeline (Example Gen, Schema Gen, Statistic Gen, Example validator, Trainer Module, Serving Module) is built using tfx libraries.

Figure 2: Prediction done by fine-tuned and distilled model

TENSOR BOARD VISUALIZATION:

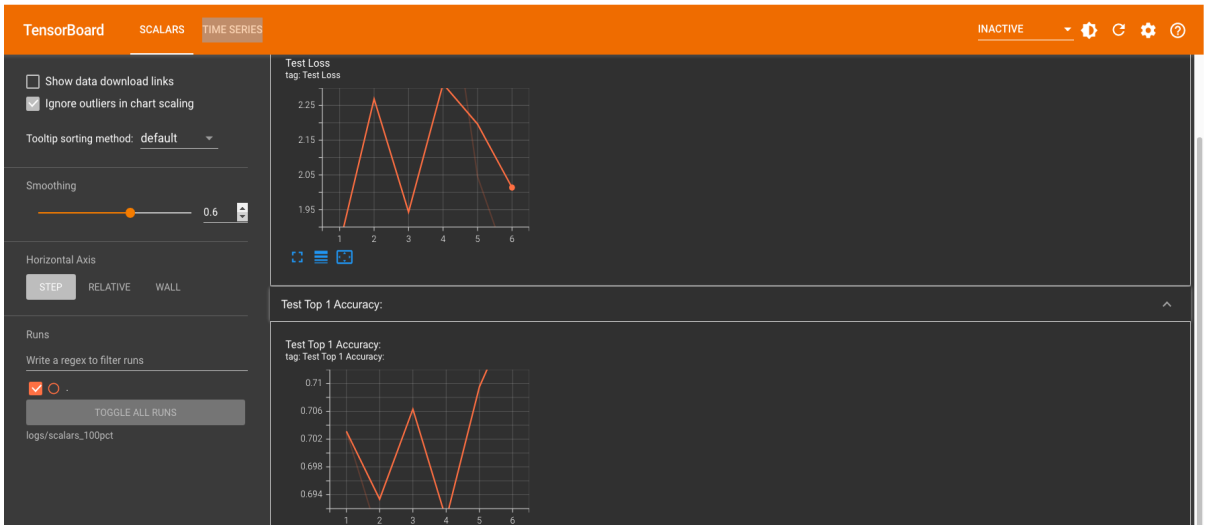

Figure 3: Tensor board graph

DEPLOYMENT ON KUBEFLOW

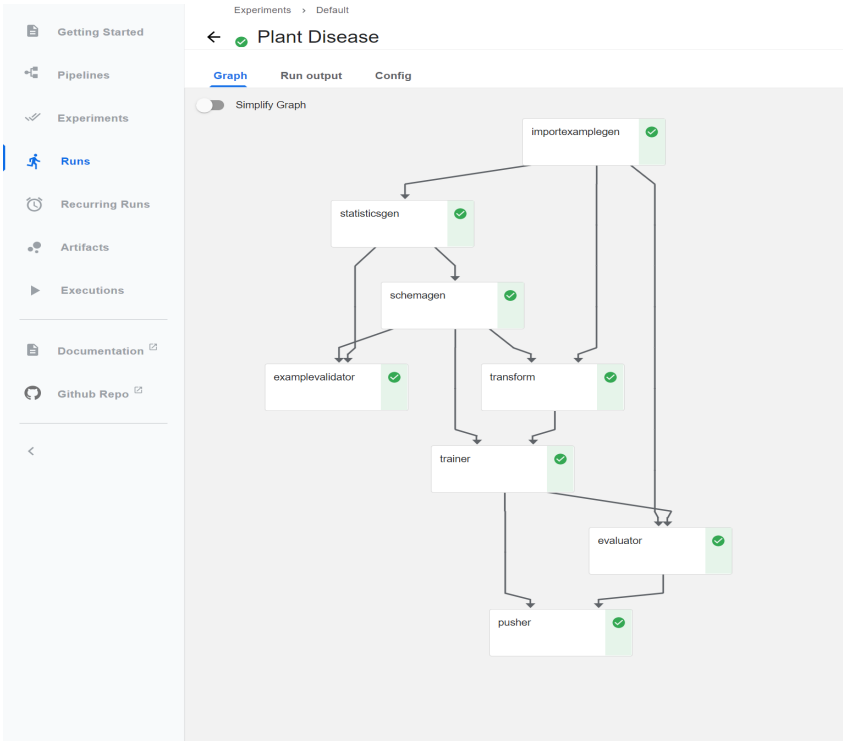

Figure 4 : Kubeflow with TFX

FLASK APPLICATION

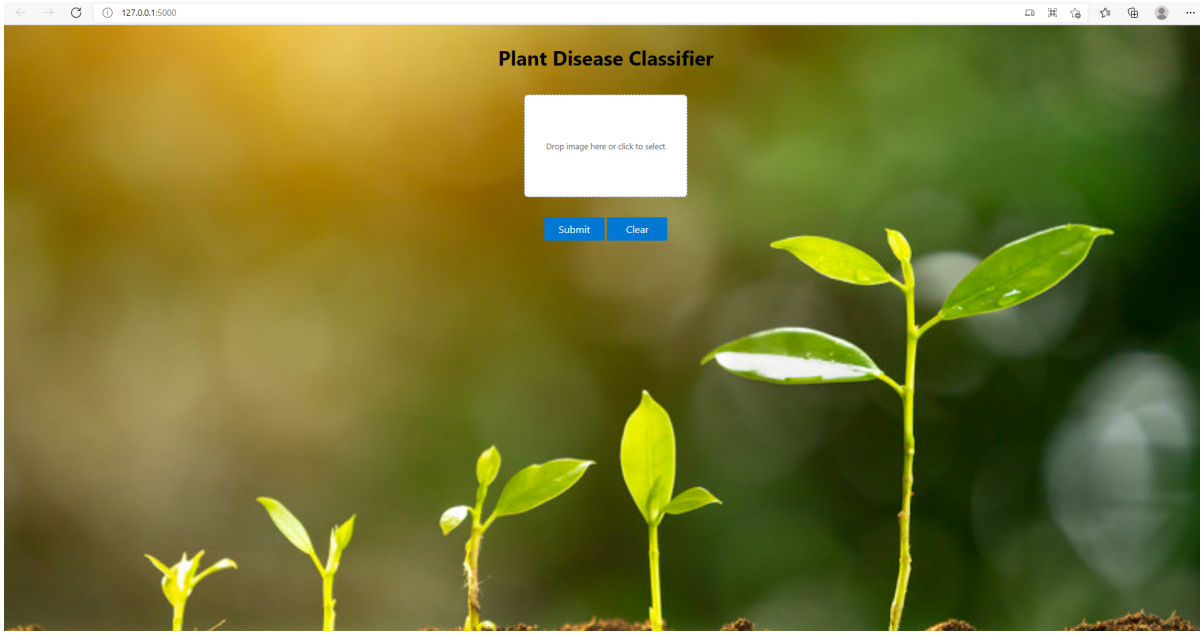

Figure 5: Flask Web application page

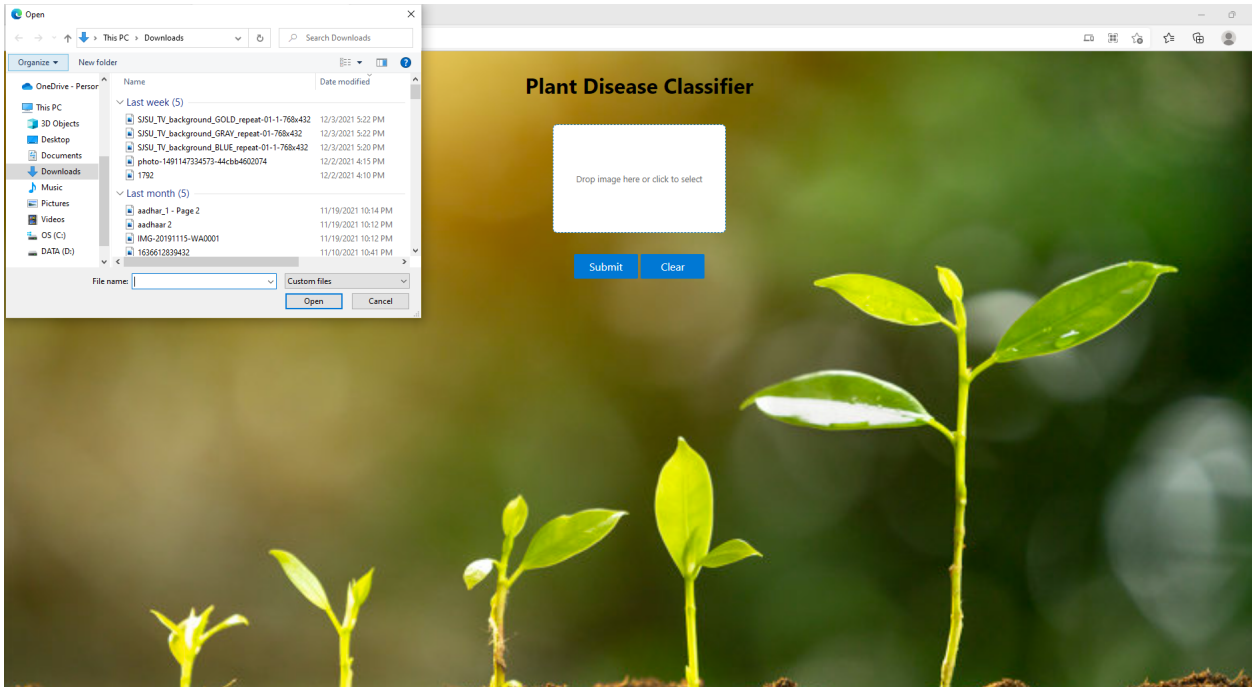

Figure 6: Application asks for a plant image we upload it from our device

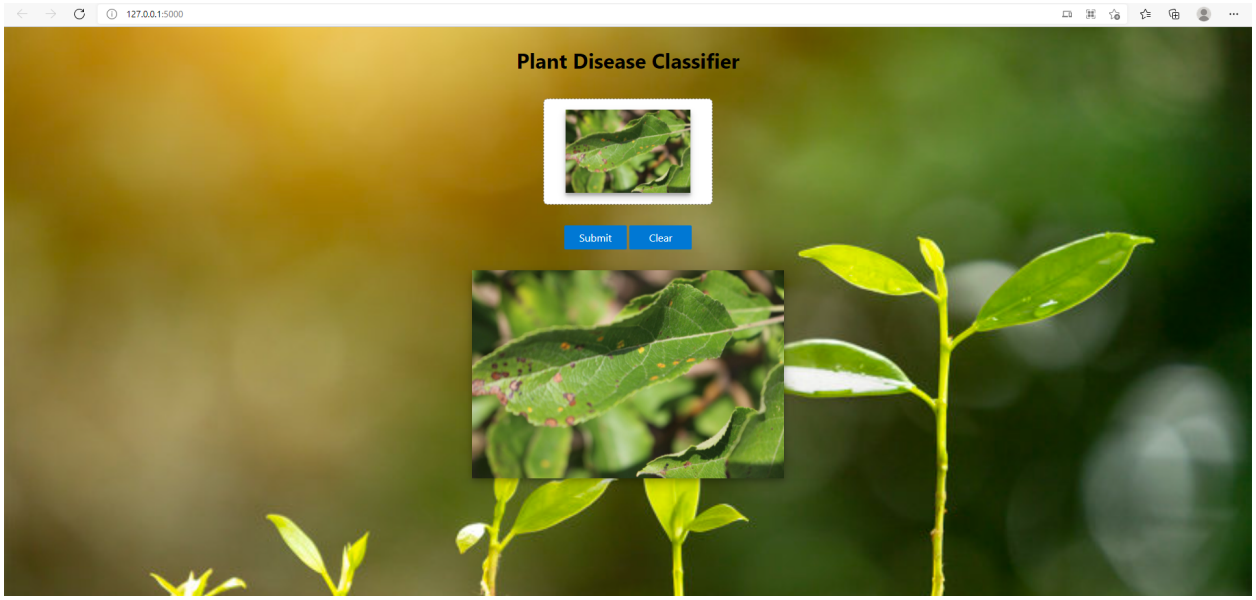

Figure 7: Image selected

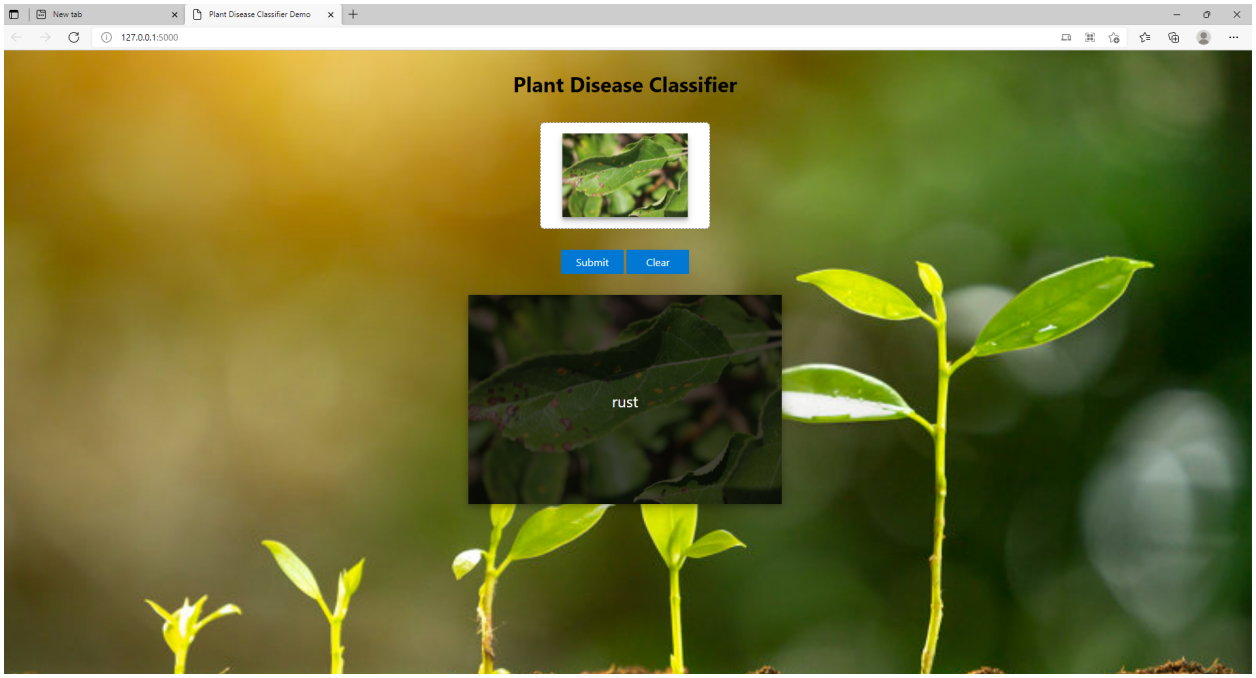

Figure 8: Image Label we get after Inference by the Model

CONCLUSION & FUTURE WORK

Generally, overfitting is a high risk for data-hungry deep learning models. In order to tackle this problem, we developed a model which is data efficient and is able to alleviate data deficiency. We try to get hands on how to use contrastive learning to accurately detect plant disease from plant leaf images. Simclrv2 model learn the general features of a plant village dataset without labels by teaching the model which data points are similar or different. With the usage of pretrained model weights, the fine-tuned model provided accuracy of around 89 percent.

For future work, we can apply the pretrained weights to detect and classify disease for other agricultural crops.

REFERENCES

[1]. https://github.com/google-research/simclr

[2]. https://arxiv.org/abs/2006.10029

[3]. https://www.tensorflow.org/datasets/catalog/plant_village

[4]. https://arxiv.org/abs/2002.05709

[5]. Saleem, Muhammad Hammad, Johan Potgieter, and Khalid Mahmood Arif. "Plant disease detection and classification by deep learning." *Plants* 8.11 (2019): 468.