ECE 269 Project Report: Learning by Teaching for Plant Disease Detection

Sanjeev Sinha
UC San Diego

Yejun Li
UC San Diego

Jiawen Zeng
UC San Diego

1 Research Problem

The research proposal suggests an exploratory approach towards plant disease detection by adapting Learning by Teaching methods. The difficulty of plant disease detection lies in segmenting the disease region and correctly labeling the disease types for different types of plants. The Learning by Teaching Algorithm introduced by Professor Xie in ECE 269, for which a teacher model improves its performance by teaching the task to others, has provided us a possible solution for plant disease where labels are limited. Because of this limitation of labeled data for practical use, we want to study the effects of the Learning by Teaching method, as we believe this method can prove to train our classifier in a much more efficient manner by having a teacher model teach the student model what it learns through the data it has seen.

2 Importance of Problem

Diagnosing the disease of the plant has been an important aspect to the agriculture since without proper identification of the disease, the disease control measures can be a waste of time and money and can lead to further plant losses. On the other hand, the need for monitoring the plant diseases manually is very costly due to the requirement of the expertise and the scale of the farm.

3 Existing Works

There have been efforts to efficiently and accurately identify plant disease detection through images. The previous work in the plant disease detection field consisting of using Machine Learning techniques such as XY-fusion, SVM, SOM, and CNN [5]. The study in XY-fusion, SVM, and SOM are focusing on a relatively small dataset(< 1k images)[6][7][8]. The dataset target is usually constructed by experts to study specific types of plant diseases. The study targeting multiple plants and corresponding plant diseases are usually studied using deep learning models such as CNN. The study we found solved the problem with 99.53 % accuracy [9]. However, several limitations exist [2]. Many images which are used for training are in a controlled environment, and thus cannot identify images that contain disease which are gathered and labeled from practical applications. Strategies such as traditional augmentation (rotations, blurring, noising) have been used to distort images so the model would be able to recognize plant disease in more realistic images. Generative Adversarial Networks (GANs) have also been used to create better data to train the plant disease classifier on. There have also been efforts for a disease recognition method using a hybrid clustering system [3], and a content-based image retrieval system was used to extract color features and means along with an SVM for classification.

4 Methods

Our proposed solution to correctly identifying the plant disease is to implement the Learning by Teaching method for image classification. Using this method to help us develop better models for

Github Link: https://github.com/garmenz/ECE269_final_proj

image classification on something with various background and hard to identify on images, such as potential plant disease.

4.1 Data Preprocessing

Since the dataset we are using isn't split into training and testing, we are planning on reorganizing the dataset into the following splits: 90/10 for training and testing, where the training data is going to have a 75/25 split between real training and validation. For our second training phase, we obtains datasets from also other backgrounds. The reasoning for this secondary training phase is because a lot of the pictures in the PlantVillage dataset are taken in a laboratory setting, which means the backgrounds for all the images are the same. Since our end goal is to be able to applicable for plant disease detection right at the field, it is important we are able to generalize our model even further to plants given any image with any background.

For data preprocessing itself, we are looking into simple transforms such as cropping, rotations by a small angle, and horizontal flipping. However, we don't want to enhance images since the whole idea is for the model to be able to generalize on real-world images. Some examples of real-world images are shown below (taken from an online search), and we can see they are certainly different in terms of resolution, background, and lighting amongst many other things.

Figure 3: Potato Leaf (Left) and Marssonia Leaf (Right) without Lab Background

4.2 Base Model

We will first start with training a base model for comparison of results. We are planning on using the Resnet-18 neural network as a starting point for our baseline results observation. The architecture is shown below: We plan on having two training phases: one using the entire PlantVillage dataset

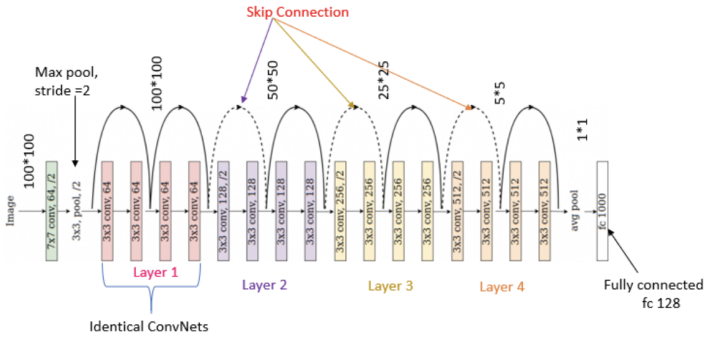

Figure 1: Resnet18 Architecture

and then one with appropriately processed images from other plant disease datasets. For our loss functions, we plan on using the following cross-entropy loss function which is calculated as:

$$H(t,p) = -\sum_{i=0}^{n} t_i * log(p_i) \tag{1}$$

where t_i is the true label and p_i is the softmax-ed estimated probability for the i-th class.

4.3 Learning by Teaching (LBT) Model

We adapt the learning model from Learning by Teaching [1] in order to attain more generalized prediction over detection of the plant disease problem. The model composes of two parts, the teacher model and the student model. The teacher model generally learns some samples and based on what it learns, it generated some psuedo-label to some unseen samples to the student. The student model learns from both a subset of samples and the generated samples from the teacher model. The student validates its learning and provided feedback to the teacher and the teacher model updates its architecture based on the feedback.

The corresponding mathematical set up would be

$$T^*(A) = min_T L(A, T, D_t^{tr}) \tag{2}$$

where $T^*(A)$ represents the optimal weight for a given architecture of the teacher model.The teacher model is trained by supervised learning and try to minimize the cross-entropy loss over the different classes.

$$S^*(T^*(A)) = min_S L(S, D_s^{tr}) + \lambda * L(S, D_{pl}(D_u, T^*(A))) \tag{3}$$

where $S^*(T^*(A))$ refers to the optimal weight for a given student architecture with student training data and the pseudo-labeled data from the teacher's model.

The final objective function would be optimized towards the architecture A of the teacher's architecture based on 2 and 3. The mathematical set-up would be the following:

$$J(A) = min_A L(T^*(A), A, D_t^{val}) + (S^*(T^*(A)), D_s^{val}) \tag{4}$$

The loss are a combination of current optimized architecture applies on teacher's validation set and student's validation set.

In order to execute the optimization, the paper proposed an optimization algorithm using one-step gradient descent update adapted from DART [12]. Approximate $T^*(A)$ using

$$T' = T - \eta_t \nabla_T L(T, a, D_t^{tr}) \tag{5}$$

which where η_t is the learning rate. In the second stage, plugged in T' to generate pseudo-label. The new student's model's parameters are updated by

$$S' = S - \eta_s \nabla_S (L(S, D_s^{tr}) + \lambda L(S, D_{pl}(D_u, T'))) \tag{6}$$

where η_s is the learning rate for student's model. In the third stage, we want to optimize the teacher's architecture. We used a differentiable architecture search proposed by Liu and etc..[12]

$$A' = A - \eta(\nabla_A L(T', A, D_t val) + \gamma * \nabla_A L(S', D_s^{val})) \tag{7}$$

where architecture update involved in both the loss of the teacher model $L(T', A, D_t val)$ and the loss of the student model $L(S', D_s^{val})$. Below is a high level overview of how LBT operates:

The algorithm set-up is given in Figure 4.

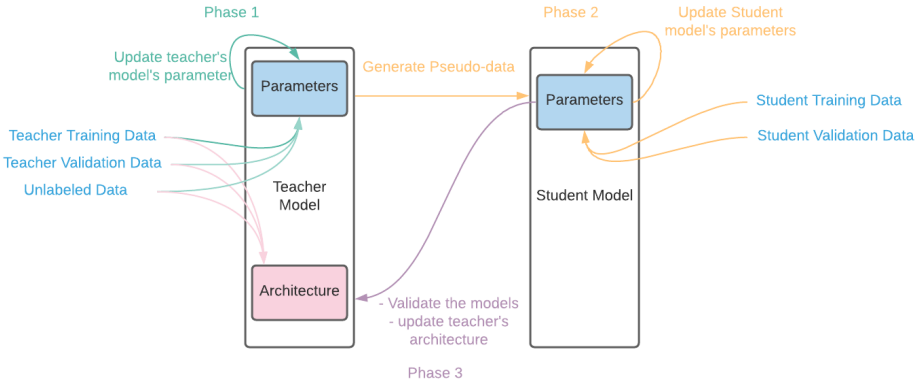

Figure 2: LBT High-Level Overview

Algorithm 1: LBT Algorithm

initialize student and teacher model;
while *not converged* **do**
    forward pass on teacher's model, update weight on teacher's model using Eq.5;
    generate pseudo-label for testing dataset by the updated teacher's model;
    forward pass on student's model, update weight on student's model using Eq. 6;
    forward pass on both student's model and teacher's model using validation data;
    update the architecture of the teacher's model using Eq.7 ;
end
take the argmax of the pseudo-label and compare it with the true testing label

5 Experiment

5.1 Datasets

In order to increase of the diversity of the samples, we used a mixture of multiple dataset for training and testing. We select 10 classes that all of the datasets have in common and have a relatively balanced amount of samples. The dataset we used include PlantVillage [14], the Leaf Type Detection dataset from Kaggle for the unlabeled dataset and PlatDoc dataset[13]. The images from PlantVillage are taken in the lab background (Fig [3]), while PlantDoc and Kaggle dataset (Fig[4]) are taken from the field background. The total number of samples we have are 17883 images and we classify them into ten classes: healthy pepper bell, pepper bell with bacterial spot, tomato with target spot, tomato with yellow leaf curl virus, tomato with bacterial spot, tomato with late blight, tomato leaf with mold, tomato with septorial leaf spot, tomato with spider mites.

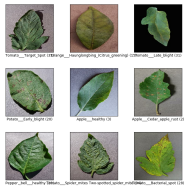

Figure 3: Example of images from PlantVillage

Figure 4: Example of image from PlantDoc

5.2 Experimental Settings

The experimental setting for baseline model is:

Hyperparameter	Value
learning rate	1e-3
pretrained	False

Table 1: Hyperparameter for baseline model

The search space for the acrchitecture search is over 3*3 maxpooling, 3*3 average pooling, 3*3 separable convolution, 5*5 separable convolution, 3*3 dilated convolution and 5*5 dilated convolution. The experimental setting for architecture search is:

Hyperparameter	Value
architecture learning rate	3e-4
architecture weight decay	1e-3
initial learning rate	0.025
minimum learning rate	0.001
total number of layers	8
cutout length	16
drop path probability	0.3
momentum	0.9
weight decay	3e-4
weight for auxiliary loss	0.4
gradient clipping	5

Table 2: Hyperparameters for architecture searching

The experimental setting for the best teacher model on the training data is:

Hyperparameter	Value
initial learning rate	0.025
total number of layers	20
momentum	0.9
weight decay	3e-4

Table 3: hyperparameters for the optimal teacher model

5.3 Results

The size of the model is listed in table [4]:

Model	number of parameters	storage size
Baseline (ResNet18)	11.1M	-
teacher model from LBT	3.9M	3.4MB

Table 4: hyperparameters for the optimal teacher model

The result of Learning by Teaching's architecture search is shown in Fig[5] for normal cell and Fig[6] for reduce cell.

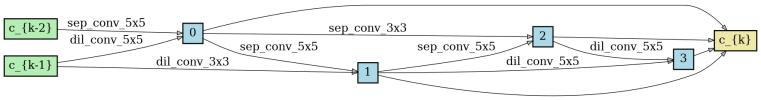

Figure 5: Resulting architecture: normal cell

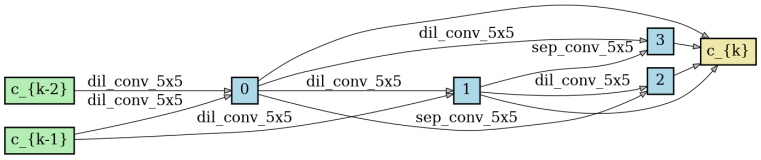

Figure 6: Resulting architecture: reduced cell

The training performance of the teacher's model is shown in Figure[13]:

The accuracy comparison of the model is shown in Table[5]

Model	train/val/test	accuracy of optimal model
Baseline (ResNet18)	train	64.4%
	validation	68.4%
	test	60.7%
teacher model from LBT	train	82.6%
	validation	84.6%
	test	82.5%

Table 5: hyperparameters for the optimal teacher model

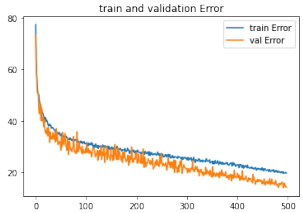

Figure 7: Training and Validation Error Curves for Teacher Model

5.4 Analysis of Results

For training the searching teacher model, the architecture development is shown in figure 5, 6, 8, and 9. As we resized the image with size 32 by 32, the pooling operation is less significant. The architecture search was able to observe the transformation and replace the pooling operation with more convolution. Comparing to the baseline model, the resulting architecture required longer training time to achieve similar result. However, the benefit for the resulting architecture is that are no sign for over-fitting. The baseline model started fluctuating in an early stage of training. That proven that the learning by teaching methods provide architecture for better generalization.

As the difficulty on classifying plant image data in the field, majority of the public dataset we found are in laboratory setting. Deep learning model would tend to achieve high accuracy on the laboratory data and low accuracy on field image data. From our baseline model experiment, we discovered that the ResNet-18 model can classify plant diseases with 93% accuracy with solely the Plant Village dataset. However, when comes to also concern samples from the field the accuracy drop drastically to around 64%. The learning by teaching model performs better in a combination of data both the lab setting and also from the field.

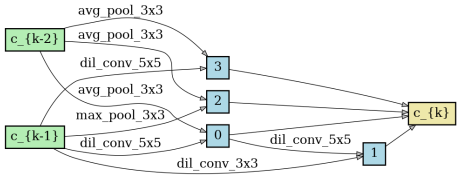

Figure 8: Initial architecture: normal cell

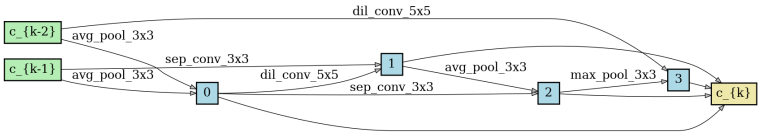

Figure 9: Initial architecture: reduced cell

5.4.1 Remarks about Base Model

We decided to use the Resnet-18 network for our student model, but not have it pretrained. We decided to have it not be pretrained and purely use it for its structure because we want the student to simply obtain information from the teacher and no one else, that way we maximize the amount of information the student learns from the teacher. We use the same transforms as those for training using LBT to be able to fully compare the two methods. Below are the training and validation curves for the base model:

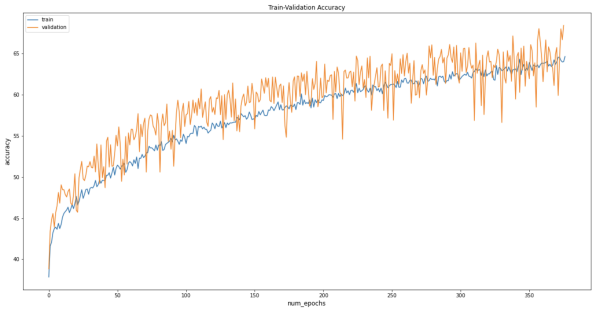

Figure 10: Training and Validation Accuracy for Baseline Model

Figure 11: Correct Classification of Tomato and Specific Disease

We see that just by using the baseline model, the model never actually converges as can be shown by the jumping in validation accuracy as epochs increase. Figures 11-13 are some examples of what the base model predicted in relation to the true class.

Here, we see that the the model is able to classify this image as the right type of plant as well as the right type of disease it has. In the figure below, we see that this isn't always the case. Perhaps due to the limitations of the dataset in terms of spread, we also encounter cases where the model is able to predict the type of plant but not the disease, or only simply the type of plant as shown in Figures 12

Figure 12: Correct Classification of only Tomato

Figure 13: Correct Classification of Diseased Tomato

and 13 respectively. Thus, we can see that simply having the base model might not be enough for correct classification at this level (plant, diseased or not, and type of disease).

6 Conclusion

In this project, we proposed a solution to identify the plant disease of different species and under different environment. We adapted the learning by teaching method to search for optimized structure in the classification task and reached an accuracy of 82.6%. This method certainly ended in having better results than the baseline model, which had an accuracy of 64.4%.

Although our new datasets includes some images taken from the field, however, these images are limited to 9 out of 10 classes of plant diseases. There also other limitations such as there is only one type disease per image, while in the reality, it might happens that a small area of leaves with different type of disease. In the future, we could explore object detection to better locate the disease area.

References

[1] Sheth, Parth & Xie, Pengtao (2020) Learning by Teaching, with Application to Neural Architecture Search.

[2] Arsenovic, Marko & Karanovic, Mirjana & Sladojevic, Srdjan & Anderla, Andras & Stefanovic, Darko (2019) Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection.

[3] S. Zhang, Z. You, and X. Wu, "Plant disease leaf image segmentation based on superpixel clustering and EM algorithm," Neural Computing and Applications, vol. 31, no. S2, pp. 1225–1232, 2019.

[4] J. K. Patil and R. Kumar, "Analysis of content based image retrieval for plant leaf diseases using color, shape and texture features," Engineering in Agriculture, Environment and Food, vol. 10, pp. 69–78, 2016. [5] Liakos, K.G. Busato, P. Moshou, D. Pearson, S. Bochtis, D. (2018) Machine Learning in Agriculture: A Review. Sensors 2018, 18, 2674. https://doi.org/10.3390/s18082674

[6]Pantazi, X.E. Tamouridou, A.A. Alexandridis, T.K. Lagopodi, A.L. Kontouris, G. Moshou, D. (2017) Detection of Silybum marianum infection with Microbotryum silybum using VNIR field spectroscopy. Comput. Electron. Agric. 2017, 137, 130–137

[7]Ebrahimi, M.A. Khoshtaghaza, M.H. Minaei, S. Jamshidi, B. (2017) Vision-based pest detection based on SVM classification method. Comput. Electron. Agric. 2017, 137, 52–58.

[8]Moshou, D. Bravo, C. Oberti, R. West, J. Bodria, L. McCartney, A. Ramon, H. (2005) Plant disease detection based on data fusion of hyper-spectral and multi-spectral fluorescence imaging using Kohonen maps. Real-Time Imaging 2005, 11, 75–83.

[9]Ferentinos, K.P. (2018) Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. 2018, 145, 311–318

[10] Kulkarni, Omkar (2018) Crop Disease Detection Using Deep Learning

[11] Cap, H.Q. & Uga, H. & Kagiwada, S. & Iyatomi, H. (2020) LeafGAN: An Effective Data Augmentation Method for Practical Plant Disease Diagnosis

[12] Liu, H. & Simonyan K. & Yang, Y. (2018) DARTS: Differentiable Architecture Search

[13] Davinder Singh & Naman Jain & Pranjali Jain & Pratik Kayal & Sudhakar Kumawat & Nipun Batra. (2019) PlantDoc: A Dataset for Visual Plant Disease Detection

[14] Sharada P Mohanty & David P Hughes & and Marcel Salathé. (2016) Using deep learning for image-based plant disease detection. Frontiers in plant science 7 (2016), 1419.