AWS Machine Learning Engineer Nano Degree Capstone Proposal

- By Prafull Parmar

Plants Disease Detection Using Deep Learning

1. Domain Background

Plant diseases are one of the major factors responsible for substantial losses in yield of plants, leading to huge economic losses[1]. According to a study by the Associated Chambers of Commerce and Industry of India, annual crop losses due to diseases and pest's amount to Rs.50,000 crore ($500 billion) in India alone, which is significant in a country where the farmers are responsible for feeding a population of close to 1.3 billion people [2]. The value of plant science is therefore huge.

Accurate identification and diagnosis of plant diseases are very important in the era of climate change and globalization for food security. Accurate and early identification of plant diseases could help in the prevention of spread of invasive pests/pathogens. In addition, for an efficient and economical management of plant diseases accurate, sensitive and specific diagnosis is necessary.

The growth of GPU's (Graphical Processing Units) has aided academics and business in the advancement of Deep Learning methods, allowing them to explore deeper and more sophisticated Neural Networks[3]. Using concepts of Image Classification and Transfer Learning we could train a Deep Learning model to categorize Plant leaf's images to predict whether the plant is healthy or has any diseases. This could help in the early detection of any diseases in plants and could help take preventive measures to prevent huge crop losses.

2. Problem Statement

We will be using Deep Learning to categorize plant leaf images to predict whether the plant is healthy or has any diseases. This project will use a plant leaf's images dataset[4] that is based on the original "PlantVillage Dataset"[5]. The original plant disease detection dataset[5] has been used in multiple Deep Learning Contests[7][8]. The task may be stated simply as categorizing the plant disease from the provided photograph of the plant's leaves.

Given the challenge is to categorize whether the plant is diseased or healthy from the plant's images, it's reasonable to conclude that we should utilize a computer to accomplish this operation. As identification of such diseases by human's naked eye is challenging and time consuming as well. Even if someone is competent at this task of plant disease categorization by looking at plant images, doing so for thousands of plant images in a matter of minutes by humans is almost impossible, or at the very least will require a large number of highly competent people for this task, to work in parallel. As a result, automating this type of activity could help in identification of the plant diseases on a large scale. This could further help build other applications like API's or mobile applications that could integrate the Machine Learning model build in this project and leverage it to predict plant disease from the plant photographs.

With the recent increase in interest in home gardening, people could also leverage the model to check the health of their home plants.

We would be focusing only on building a Machine Learning model that will be able to identify whether the plant is healthy or has any diseases based on the input plant's leaf's image, by leveraging Deep Learning concepts and Machine learning tools. The method I suggest is to build a classification model that uses a pre-trained Convolutional Neural Network to extract features from photographs, and then leveraging the concepts of Transfer Learning to build a final classification model that has been trained specifically for the task of plant disease detection.

3. Datasets And Inputs

We will be using the plant disease detection dataset[4] that is based on the original "PlantVillage Dataset"[5]. The dataset consists of 39 different classes, where 38 classes are of plant leaf categorised according to whether they are diseased or healthy and remaining one class consists of images with Background without plant leaves. The provided modified dataset[4] that we will be using has been created using six different augmentation techniques on the original dataset[5] for increasing the data-set size. The techniques are image flipping, Gamma correction, noise injection, PCA colour augmentation, rotation, and scaling. The dataset that we will be using contains 61,486 images.

Sample images of the 38 categories of the plant leaves are shown below.

Figure 1 Plant Village Dataset. Image credits[7]

All the 39 classes are :

1. Apple_scab, 2. Apple_black_rot, 3. Apple_cedar_apple_rust, 4. Apple_healthy, 5. Background_without_leaves, 6. Blueberry_healthy, 7. Cherry_powdery_mildew, 8. Cherry_healthy, 9. Corn_gray_leaf_spot, 10. Corn_common_rust, 11. Corn_northern_leaf_blight, 12. Corn_healthy, 13. Grape_black_rot, 14. Grape_black_measles, 15. Grape_leaf_blight, 16. Grape_healthy, 17. Orange_haunglongbing, 18. Peach_bacterial_spot, 19. Peach_healthy, 20. Pepper_bacterial_spot, 21. Pepper_healthy, 22. Potato_early_blight, 23. Potato_healthy, 24. Potato_late_blight, 25. Raspberry_healthy, 26. Soybean_healthy, 27. Squash_powdery_mildew, 28. Strawberry_healthy, 29. Strawberry_leaf_scorch, 30. Tomato_bacterial_spot, 31. Tomato_early_blight, 32. Tomato_healthy, 33. Tomato_late_blight, 34. Tomato_leaf_mold, 35. Tomato_septoria_leaf_spot, 36. Tomato_spider_mites_two-spotted_spider_mite, 37. Tomato_target_spot, 38. Tomato_mosaic_virus, 39. Tomato_yellow_leaf_curl_virus

Given the dataset is huge and has 39 categories, for the purpose of this project we will be using only a subset of the plant classes for our project. We will only be using the classes:

Sr No.	Plant Image Class Name	Class Image dataset size
1.	Cherry_powdery_mildew	1053 images
2.	Cherry_healthy	1001 images
3.	Pepper_bacterial_spot	1000 images
4.	Pepper_healthy	1478 images
5.	Potato_early_blight	1000 images
6.	Potato_healthy	1001 images
7.	Potato_late_blight	1000 images
8.	Strawberry_healthy	1001 images
9.	Strawberry_leaf_scorch	1110 images

Thus the total dataset of that will be used for this project will be **9644 images**. All the images vary in dimensions, they are not standardized, and they are all coloured images. So the model will be trained on the above 9 plant image classes, for our use-case.

The subset dataset for these 9 classes is already downloaded by me from the website [4], so it will be provided (as part of this project) either as a zipped version or uploaded to the github repository along with any other resources to run this project. The dataset will be split into trained set, validation set and test set. The proportion of each will be defined later during implementation.

4. Solution Statement

A Convolutional Neural Network (CNN) will be the best Deep Learning model for this project. However, training such models from scratch is time consuming as well as computationally expensive. Instead, this challenge will be solved with the help of a pre-trained model. In industry, using a pre-trained model to speed up the process is very popular.

We will be using a pre-trained CNN model to extract features from the images, and then using concepts of Transfer Learning we will finetune the above mentioned pre-trained

classification model by adding in a new Fully-Connected Neural Network to finetune the model for our specific use-case .

Success of the model trained in this project can be quantified based on the Accuracy of the model achieved during its validation/testing phase. Along with Accuracy we could calculate other evaluation metrics such as Precision, Recall and F1-score to determine the performance of the model and compare it with the known benchmark model's performance. We could also consider and analyse the trend seen in the loss functions used in the model to give us a sense of the deviation of the model predictions from reality/expected outcomes [10]. Another measure of success of the model will be to identify how well the model performs on real world test images taken and given as input to the model.

5. Benchmark Model

A lot of research has already been done on implementing Deep learning models on the Plant Village dataset[5]. So we will be using outcomes of models from two specific research papers as a benchmark for our project.

First paper titled "Using deep learning for image-based plant disease detection."[11] uses a couple of variation of two models "AlexNet" and "GoogLeNet". Here the original dataset[5] of size 54,306 images without any data augmentation was used for training and testing on all the 38 plant classes. The training was performed on colour, greyscale and Segmented plant images. As mentioned in the research paper the GoogLeNet model was able to achieve an accuracy of 99.35% on the test set for the coloured images dataset.

Second paper titled "Plant Disease Detection from Images"[12] uses a small subset of the same "PlantVillage Dataset" with data augmentation. It only used 4000 images belonging to "Tomato", "Potato" and "Pepper" plant classes for training and testing purposes. ResNet-34 and ResNet-50 pre-trained models were employed for training the model. ResNet-50 was able to outperform ResNet-34 and achieve an accuracy of 99.4% on the test set.

Given, for the current project we will be using a subset of coloured plant images dataset of only 9 classes (9644 images) from the version of the Plant Village Dataset with data augmentation[4] for our project's training and testing purposes. We will also be using a ResNet-50 pre-trained model for our project. One of the goal of the project will be to check how well the model trained only on the 9 classes performs as compared to other known model's outcomes.

Thus, we will be using the outcomes of second paper[12] as benchmark for our project's model. The comparison would take into account the Accuracy as well other evaluation metrics such as Precision, Recall and F-1 score of our model and compare it with the outcomes mentioned in the research paper. This should provide a good benchmark to assess the performance of our model. However, we will also be comparing the outcome of our model with the outcomes of the first paper[11] and mentioning all our observations in the final project report.

6. Evaluation Metric

Because the dataset that we would be using is more or less balanced, we will be using Accuracy as the final criterion to assess the outcomes of our model. For the given problem we have to classify the images into 9 categories so we will be using "categorical cross-entropy" as loss function while training our models. We will also be using other evaluation metrics like Precision, Recall and F-1 score for evaluation of the performance of our model. We will be using the ResNet50 model as a pre-trained model for our project. Further details on the model selection are mentioned in the below Project Design section.

7. Project Design

In this section, we will be discussing the approach for fine-tuning a pretrained CNN model for image classification using the plant disease image dataset.

7.1. Transfer learning

For the purpose of this project, we will be using a pre-trained ResNet-50 model to extract the bottleneck features from the images. ResNet-50[9] model is 50 layers deep and is trained on a million images of 1000 categories from the ImageNet database. Further-more the model has a lot of trainable parameters, which indicates a deep architecture that makes it better for image recognition. The input images will be resized to 244x244 and bottleneck features will be extracted. If required based on the initial performance of the model, we might need to apply some transformations like crop images or apply any other transformations in images. The classification model will be created and prepared to consume as an input, the features extracted using the pre-trained ResNet-50 CNN model. A fully-connected layer with Soft-max as activation function of the output layer will be added. Once we have a model trained, a typical and simplified structure design of an algorithm to perform the classification is shown below:

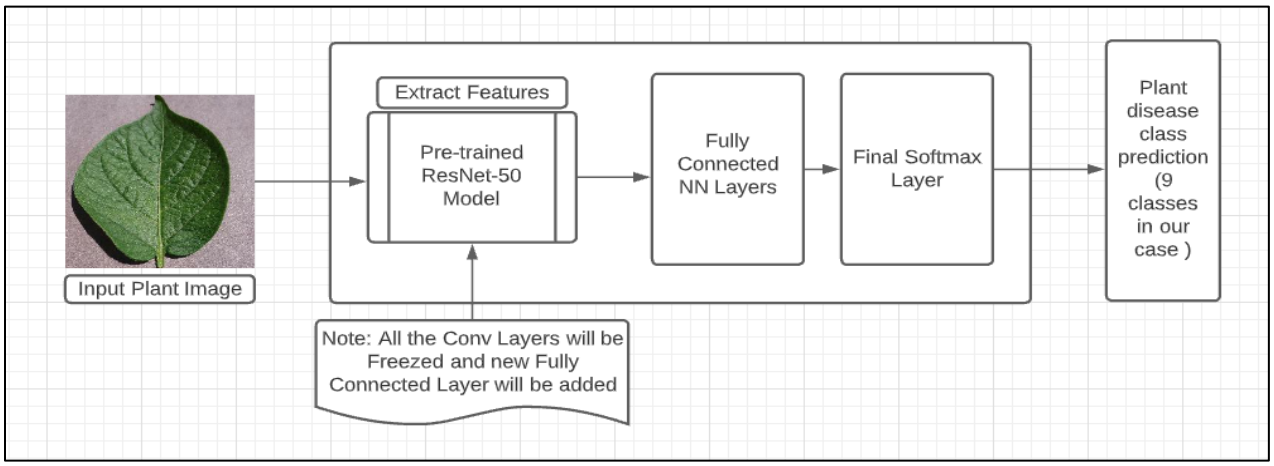

Figure 2 Simplified Design of Finetuning Pre-trained Model

7.2. Fine-tuning ML Models

For the pre-trained ML model, we could perform a couple of steps to further fine-tune the models.

One way to fine-tune the model will be the selection of the Optimizer used while training the models. For example, we could try switching between different optimizers and check the model performance, like switching from Adam to AdamW, SGD or RMSprop.

Other way would be to perform tuning of the hyper-parameters used in the optimizer and training algorithm, by checking the model performance by using different permutations and combinations of values of hyper-parameters like learning rate, batch size and momentum. We could also leverage the AWS Sagemaker's Tuner for performing hyperparameter tuning for our both the models while training phase.

8. References

[1] Balodi, Rekha & Bisht, Sunaina & Ghatak, Abhijeet & Rao, K.H.. (2017). Plant Disease Diagnosis: Technological Advancements and Challenges. Indian Phytopathology. 70. 275-281. 10.24838/ip.2017.v70.i3.72487.

[2] https://croplife.org/news/keeping-indias-pests-in-line/

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems.

[4] J, ARUN PANDIAN; GOPAL, GEETHARAMANI (2019), "Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network", Mendeley Data, V1, doi: 10.17632/tywbtsjrjv.1 (https://data.mendeley.com/datasets/tywbtsjrjv/1)

[5] https://paperswithcode.com/dataset/plantvillage

[6] https://github.com/spMohanty/PlantVillage-Dataset

[7] https://www.crowdai.org/challenges/1

[8] https://www.kaggle.com/c/plant-pathology-2020-fgvc7/overview

[9] He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90.

[10] https://dzone.com/articles/what-kpis-measure-the-success-of-an-ai-project

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

[12] Anjaneya Teja Sarma Kalvakolanu, "Plant Disease Detection from Images" (https://arxiv.org/abs/2003.05379)