Crop Health using Computer Vision

Vishal
Roll No: 220150029
Dept. of Data Science and AI
Indian Institute of Technology Guwahati
Email: vishal.dsai22@iitg.ac.in

***Abstract*—Plant diseases in crops are a big problem for farmers, especially in areas where getting help from agriculture experts isn't easy, which directly affects food security and farmer incomes. Many times, the signs of disease show up on leaves. This study presents a solution that uses deep learning to detect plant diseases from images of leaves. For this project, the "New Plant Disease Dataset (Augmented)" available on Kaggle was utilized. It includes images of leaves from different types of plants, some healthy and some diseased. A Convolutional Neural Network (CNN) was built using TensorFlow to classify the leaf images. The model, while not perfect, showed promising results, indicating potential for real-world deployment in low-resource agricultural environments.**

***Index Terms*—Plant Disease, Deep Learning, TensorFlow, Keras, Image Classification, Image Recognition, CNN, Smart Farming, Agriculture with AI.**

I. Introduction

I've always been curious about how farmers spot diseases in plants by looking only leaves. It's not easy task - sometime leaves look fine, but there is already something wrong inside. People just rely on guesswork or hope for the best where there are not enough agriculture experts around. That's especially when it comes to crops that feed so many families to live. Thankfully with the rise of Deep Learning, we now have a chance to automate this process. In this project, I used images of plant leaves, both healthy and sick, and tried training a machine to figure out what was wrong just by looking. Something called a CNN is really good at spotting tiny visual patterns. My goal was simple to build a model that could do a decent job recognizing plant diseases and maybe help farmers one day without needing to ask an expert every time.

II. Dataset Overview

Before starting training the model, I needed some good data to train it on. I came across a dataset that had what I was looking for — tons of pictures of plant leaves, some healthy and others showing clear signs of disease and that was the *New Plant Diseases Dataset (Augmented)* on Kaggle [1]. All the images were already sorted into different folders. So, for example, all the corn leaves with rust would be in one folder, and all the healthy ones would be in another. And they were already the same size - 128x128 pixels - and had 3 colour channels (so called standard RGB). I didn't have to resize anything manually, which saved me a lot of effort as well as time. Another thing I found really helpful was that the dataset was pre-augmented that means is that it already had versions of the same images that were flipped, rotated, or changed with some adjustments to improve the model to learn better. Normally, I'd have to do that myself, but here it was already done, which was honestly great.

III. Methodology

Once I had the leaf image data, the next thing was teaching a computer to figure out if a leaf is sick or healthy. So, I used TensorFlow since I'm pretty comfortable with it by now.

A. Data Preprocessing

First, I used TensorFlow's `image_dataset_from_directory` function to load the images. It's such a time-saver since it automatically figures out the labels based on the folder names. I went with a batch size of 32, which worked well—fast enough, but not too heavy on memory. The dataset already had some augmentation done, so I didn't bother adding any extra changes like flipping or adjusting brightness. The images were also shuffled, so the model wouldn't learn based on the order of the files.

B. CNN Model Architecture

A basic CNN was constructed using TensorFlow/Keras. It consisted of:

- Convolutional layers with ReLU activation
- MaxPooling layers for downsampling
- A Flatten layer followed by Dense layers
- Softmax output layer for multiclass classification

The architecture was kept lightweight to ensure reasonable training time on CPU, especially since I was not using any high-powered GPUs.

C. Training

I went with the Adam optimizer since it's a safe bet and tends to work well in most cases. For the loss function, I used categorical cross-entropy because I was dealing with multiple classes. To make sure I didn't waste time, I added a couple of things during training. First, early stopping—so if the model wasn't improving anymore, it would just stop. Second, model checkpointing—this saved the best version of the model during training, in case things started going downhill later.

D. Evaluation

Evaluation was done on the validation set using metrics:

- Accuracy
- Precision
- Recall
- F1-score

I also made some plots with Matplotlib and Seaborn so I could get a better idea of where the model was performing well and where it was making mistakes.

IV. Results and Analysis

The CNN model did pretty well overall on the validation dataset. But there were a few spots where it didn't do as well. The issues mostly came from categories that looked similar or didn't have enough training data to work with. When I looked at the confusion matrices, it became clear that the model was able to pick up most disease types without trouble, but it did get confused when the images were visually similar. To make it better, I think adding more variety to the data or using different augmentations could help the model separate those similar categories more clearly.

V. Conclusion

At the end, this project shows that deep learning models, especially CNNs, can work really well for image classification for plant diseases just by looking at leaf images. This could really help farmers, especially if I turn into a mobile app or a web browser tool that gives real-time guidance and awareness. Looking ahead, it would be cool to test it out in real-world conditions, make it more reliable with images directly from the field, and maybe expand it to cover even more different plant species and diseases.

Acknowledgments

Thanks to the Kaggle community for the dataset and the TensorFlow team for their amazing open-source library. Their work made this project possible by providing open access to such great tools and data.

References

[1] Vipoooool, "New Plant Diseases Dataset (Augmented)," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset