Report for Major II

Submitted by

500069734	R134218123	Pratham Pandey
500068183	R134218125	Pulkit Mittal
500067543	R134218152	Shashwat Chitransh
500068730	R134218206	Karmanya Dadhich

Under the mentorship of

Ms. Shahina Anwarul

Department of Systemics

Plant Disease Detection

INDEX

Abstract

The project aims at designing a system that can perform plant disease diagnosis, on leaves of multiple plants. Food losses due to crop infections from pathogens such as bacteria, viruses, and fungi are persistent issues. Thus, we plan on tackling this issue. Various approaches for the same will be taken into consideration to build a highly accurate deep learning model for the same. A web application will be developed providing the user a platform to easily perform his task with ease.

Keywords: Deep learning, OpenCV, Convolutional Neural Network

Introduction

Human society now has the ability to produce enough food to feed more than 7 billion people thanks to modern technologies. However, a number of variables, such as climate change, pollinator decrease, plant diseases, and others, continue to pose a danger to food security. Plant diseases are a global threat to food security and may be devastating for smallholder farmers whose livelihoods are dependent on healthy crops.

To prevent crop loss due to disease, a variety of strategies have been devised. Agricultural extension groups or other institutions, such as local plant clinics are usually involved in disease identification. More recently, similar attempts have been aided by making disease diagnosis information available online, taking advantage of the world's growing Internet access.

In the last few years, computer vision, and specifically object identification, has made great progress. As an example of end-to-end learning, deep neural networks have lately been successfully deployed in a variety of disciplines. Neural networks convert an input—for example, a picture of a diseased plant—to an output—for example, a crop-disease pair. A neural network's nodes are mathematical functions that receive numerical inputs from the incoming edges and output numerical values as an outgoing edge.

We implemented deep learning to build a model capable of identifying diseases in multiple plants.

Related Works

[1] Random forest, artificial neural network (ANN), k-nearest neighbour (KNN), and support vector machine (SVM) are some of the popular classification algorithms used for disease detection in plants.

[2] To recognise cucumber leaf disease, Kawasaki et al. presented CNN designs and achieved 94.9 percent accuracy. In both small and large-scale datasets, CNN is the most useful classifier for image recognition.

[3] Using GoogleNet and AlexNet architecture, Mohanty et al. trained a deep learning model to recognise 14 crop species and 26 crop illnesses with 99.35 percent accuracy.

[4] Srdjan et al. suggested a CNN-based plant disease detection system for classifying healthy leaves and 13 illnesses. The findings show that CNN is a good choice for disease recognition because of its robust computational infrastructure.

[5] Nigam et al. used deep learning to construct a Convolutional neural network model for plant disease detection using wheat crop photos of healthy and yellow rust affected leaves.

Problem Statement

Crop diseases are a huge danger to food security, but due to a lack of infrastructure in many regions of the world, timely detection is challenging. Food losses caused by pathogens such as bacteria, viruses, and fungus infecting crops are a continuous problem. The situation has worsened by the fact that diseases are now more easily transmitted globally than ever before.

Objectives

The project aims at designing a system that can perform plant disease diagnosis, on leaves of multiple plants. Various approaches were taken into consideration to build a highly accurate deep learning model for the same.

Sub-objectives

- Comparative analysis of different classifiers for Plant Disease Detection.
- Collection and cleaning of dataset of plant leaves with several diseases for plant disease classification.
- Design a classification model for predicting plant disease accurately.
- Evaluation of the proposed model.
- Building an interactive web application for the user.

Methodology

The project involves all the steps involved in a typical ML project. They are:

- Loading the dataset
- Exploratory data analysis
- Data pre-processing
- Building the model
- Compiling the model
- Training the model
- Using our model to make predictions

We took a dataset having images of leaves of 3 plant namely tomato, potato and bell peppers with a folder for healthy yield and some folders for various diseases found in them, from Kaggle. The dataset consists of fifteen folders comprising of images with healthy and diseased leaves of potato, tomato and bell peppers.

Tomato -> Healthy, Bacterial spots, Early Blight, Late Blight, Leaf Mold, Septoria leaf spot, Spider mites, Target spot, Yellow leaf curl virus, Mosaic Virus

Potato -> Healthy, Early Blight, Late Blight

Bell Pepper -> Healthy, Bacterial Spot

Classes	Number Of Images
Pepper__bell___Bacterial_spot	797
Pepper__bell___healthy	1182
Potato___Early_blight	800
Potato___healthy	121
Potato___Late_blight	800
Tomato_Bacterial_spot	1701
Tomato_Early_blight	800
Tomato_healthy	1272
Tomato_Late_blight	1527
Tomato_Leaf_Mold	761
Tomato_Septoria_leaf_spot	1416
Tomato_Spider_mites_Two_spotted_spider_mite	1340
Tomato__Target_Spot	1123
Tomato__Tomato_mosaic_virus	298
Tomato__Tomato_YellowLeaf__Curl_Virus	2567

Once the dataset was cleaned and processed, we loaded it and split it into three different folders for training, validation and testing. We used tensorflow.keras module to build our model and train it. Once the model was trained, we passed the testing dataset through it to predict the plant disease and to obtain the accuracy.

We built a web-application that a user will be able to access the feature for enhanced convenience.

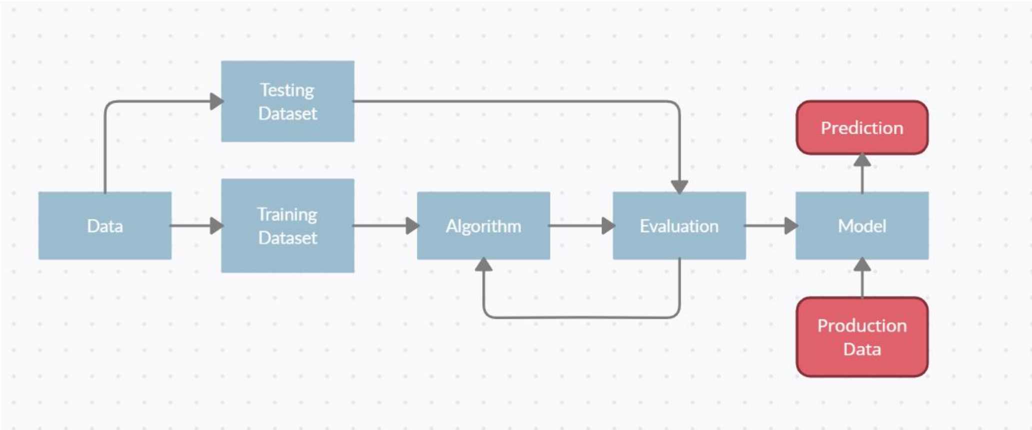


Fig 1. Workflow

Requirements

Hardware:

RAM -> 4GB

ROM-> 2GB

Recommended Processor -> intel Core i3

Software: Python, Web Browser

Technical Diagrams

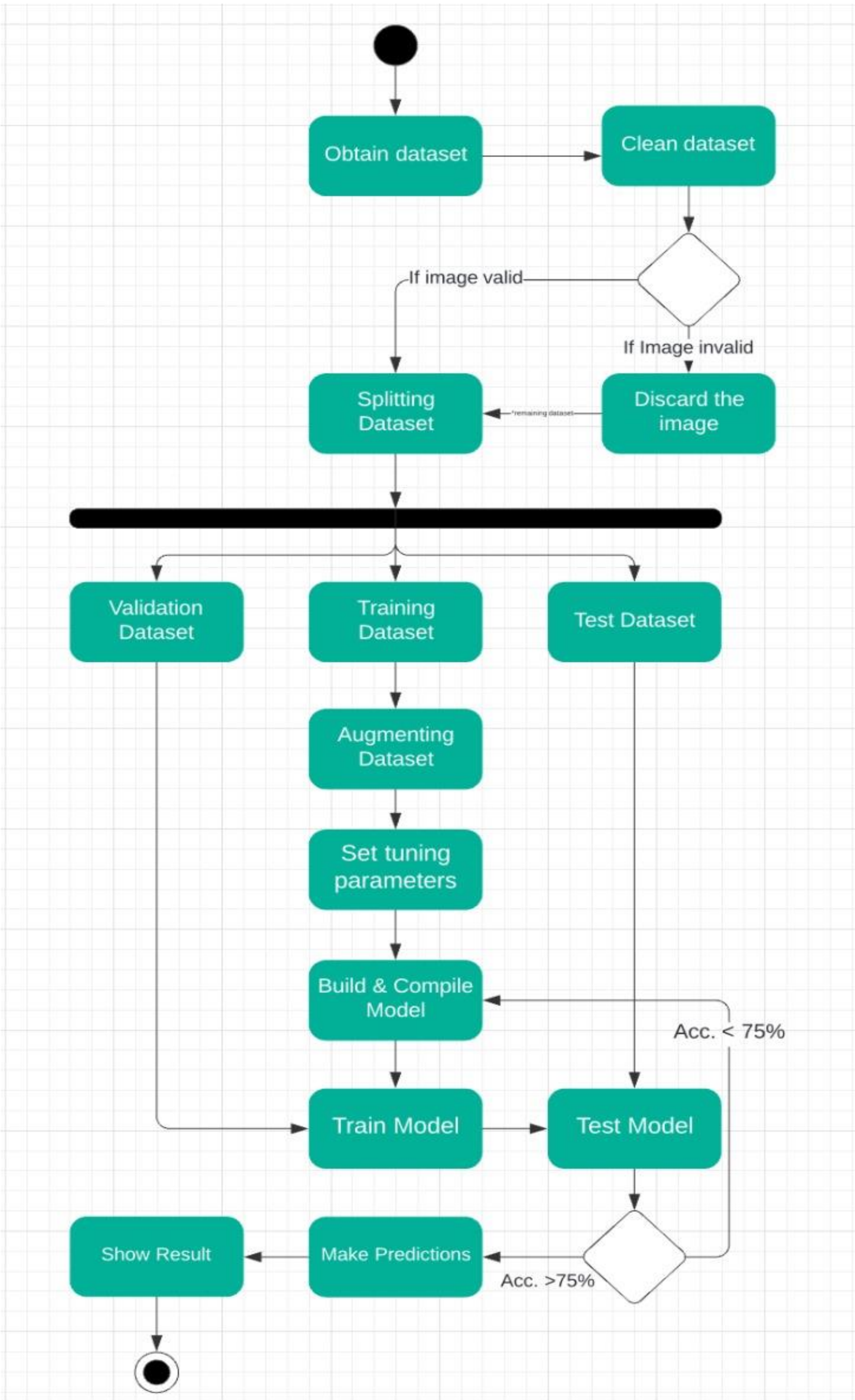


Fig 2. Activity Diagram

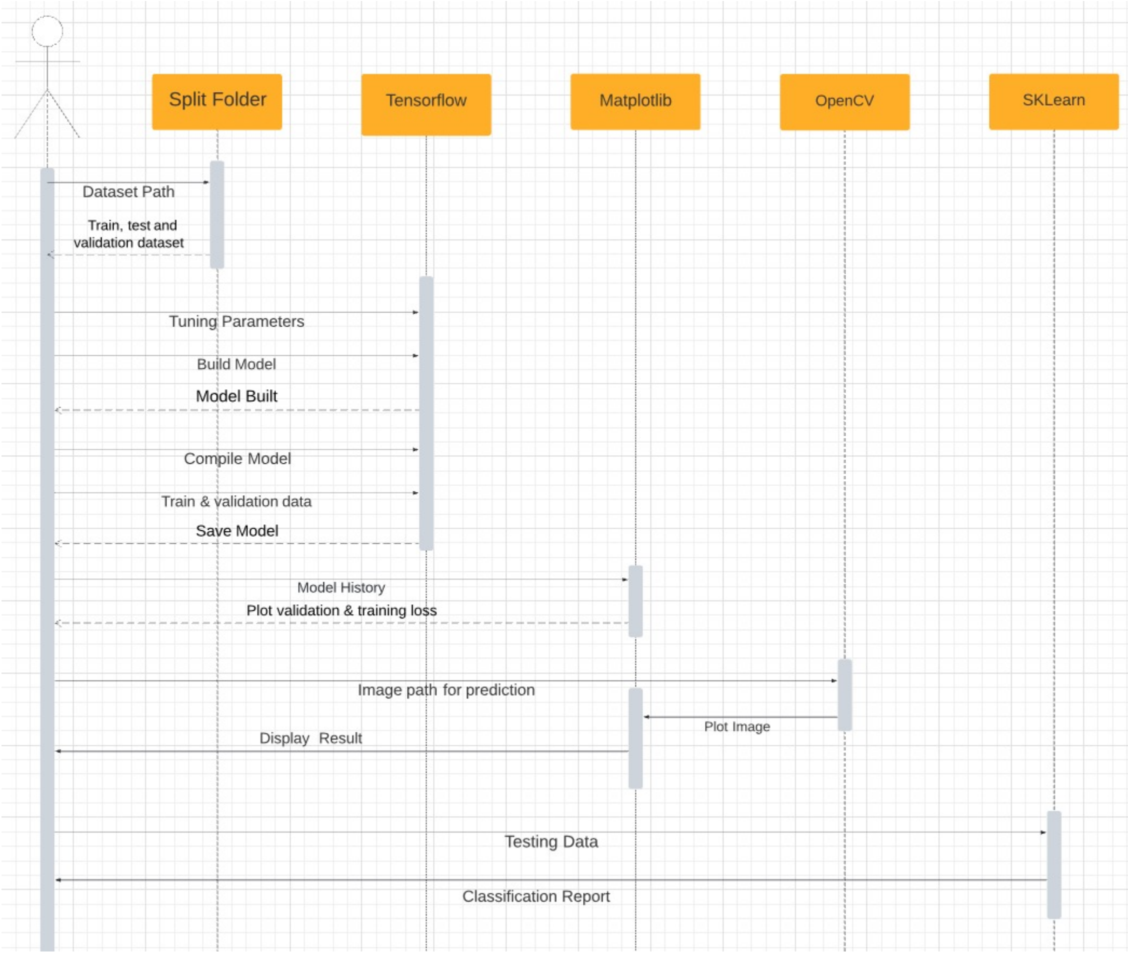


Fig 3. Sequence Diagram

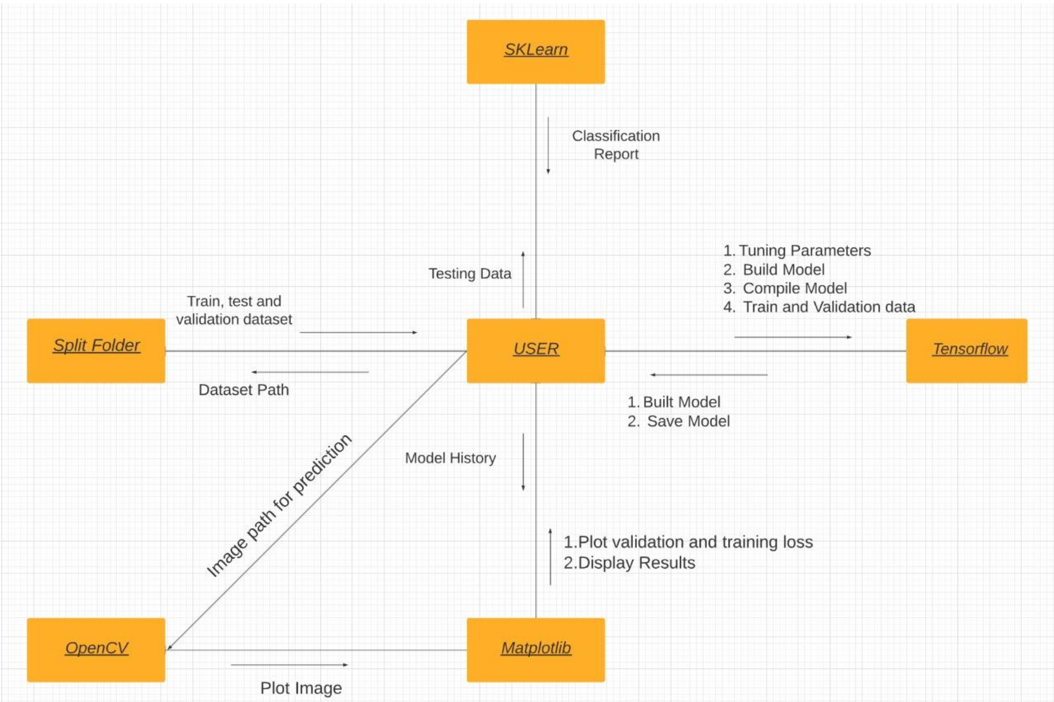


Fig 4. Collaboration Diagram

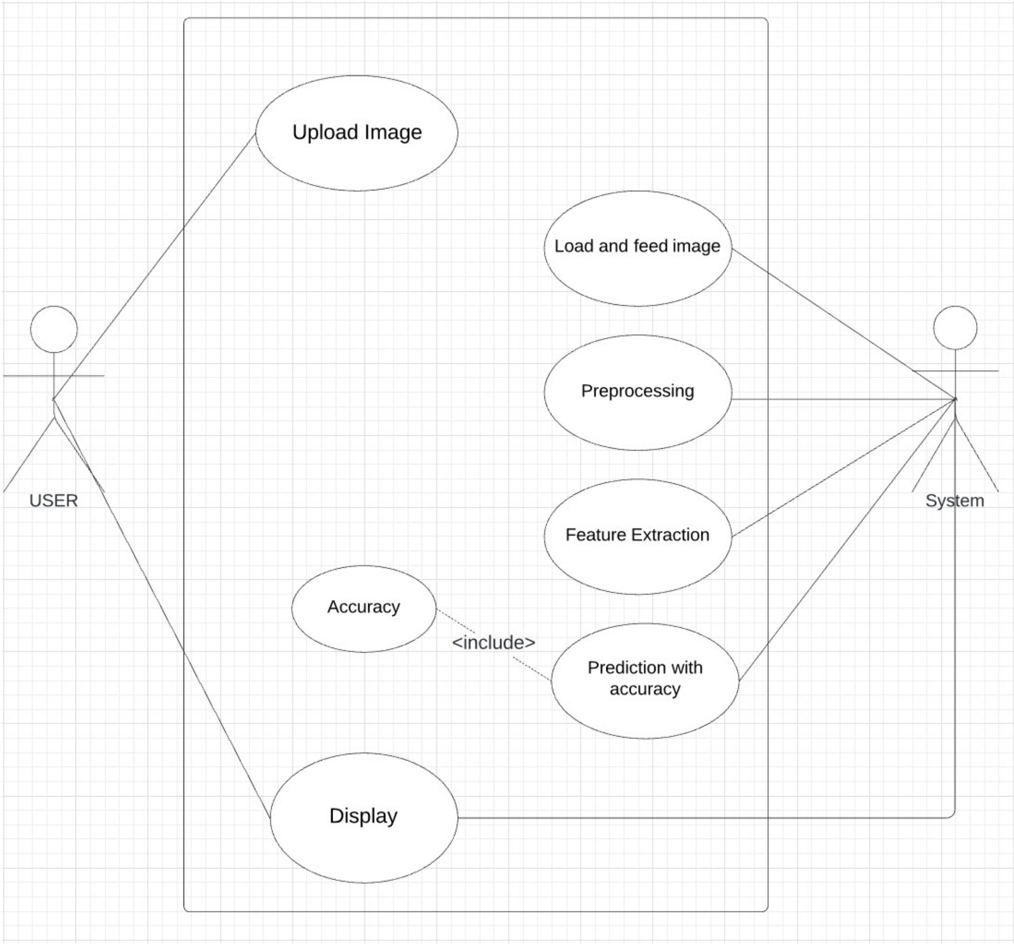


Fig 5. Use Case Diagram

Design & Architecture

The project is comprised of multiple components which have been integrated into one to work effectively. The components are as follows:

- The dataset is split into training, validation and testing datasets in 8:1:1 ratio using **splitfolders**.
- A piece of code is used to train a model to perform plant disease detection. It includes setting the tuning parameters for training the model, loading the dataset, pre-processing the dataset, applying various augmentations to the training dataset, building, compiling, training and assessing the model. Following modules are used for the same:
  - **os**: For joining and handling paths.
  - **tensorflow**: For building, compiling and training a sequential model which will be used for making predictions.
  - **OpenCV**: For reading images for detection.
  - **matplotlib**: For plotting images and graphs.
  - **Scikit-learn**: To build a confusion matrix to assess the model.
  - **Numpy**: To perform various mathematical operations.
- Now that we've trained the model, we need to use the model to perform predictions. This part is done using python in app.py, which includes different functions to take image as input via upload. The images are handled using opencv. Flask is used to work with the frontend, which is then integrated with HTML pages, styled using CSS and Javascript. Different components of functions have different endpoints which render a certain component according to functionality needed.
- The HTML component includes two files namely base.html and index.html. base.html contains a basic template for the frontend which is inherited by index.html. Index.html is rendered upon starting the server and opening the page for the project in a web browser.

- CSS is used to beautify the webpage by adding design elements and a beautiful background.
- The image request and response are handled using javascript.
- A weather api is implemented in javascript to display the current weather of the place.
- The website has also been deployed on cloud using herokuapp.

Model

The model was built using a CNN architecture. A Convolutional Neural Network (CNN) is a type of neural network that specialises in processing data with a grid-like architecture, such as an image. A binary representation of visual data is a digital image. It consists of a grid-like arrangement of pixels with pixel values indicating how bright and what colour each pixel should be.

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

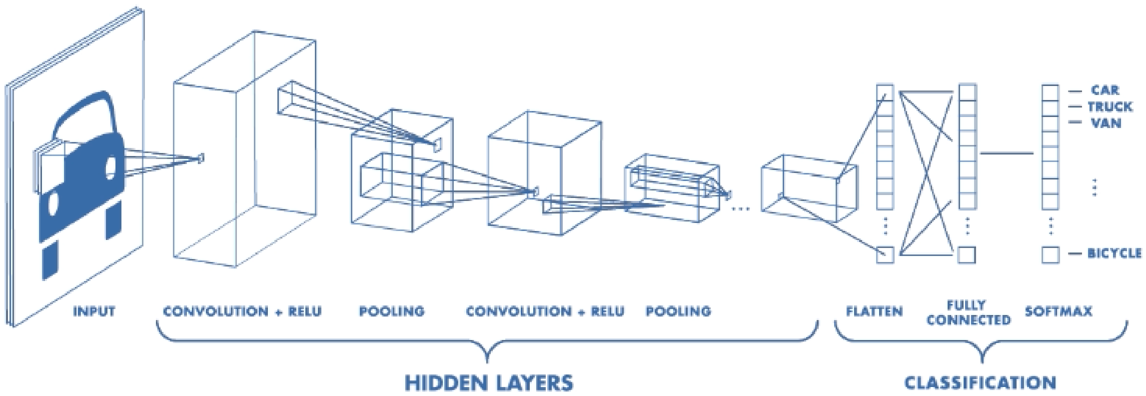


Fig 5. CNN Architecture

Our model is built using the following architecture:

[INPUT]

→[CONV] → [BATCH NORM] → [ReLU] → [POOL] → [Dropout]

→[CONV] → [BATCH NORM] → [ReLU]

→[CONV] → [BATCH NORM] → [ReLU] → [POOL] → [Dropout]

→[CONV] → [BATCH NORM] → [ReLU]

→[CONV] → [BATCH NORM] → [ReLU] → [POOL] → [Dropout]

→[CONV] → [BATCH NORM] → [ReLU]

→[CONV] → [BATCH NORM] → [ReLU] → [POOL] → [Dropout]

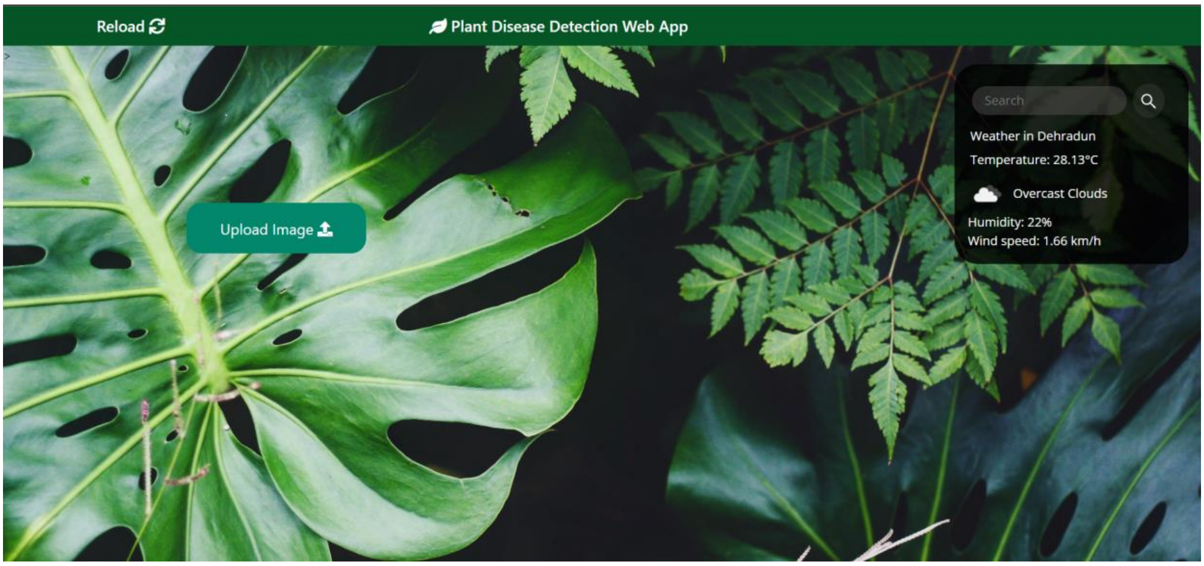
→[FLATTEN] → [BATCH NORM] → [ReLU] → [POOL] → [Dropout]

→ [FC LAYER] →[SOFTMAX] → [RESULT]

The model was compiled using Adam Classifier. The model's assessment was done on the basis of accuracy it provided

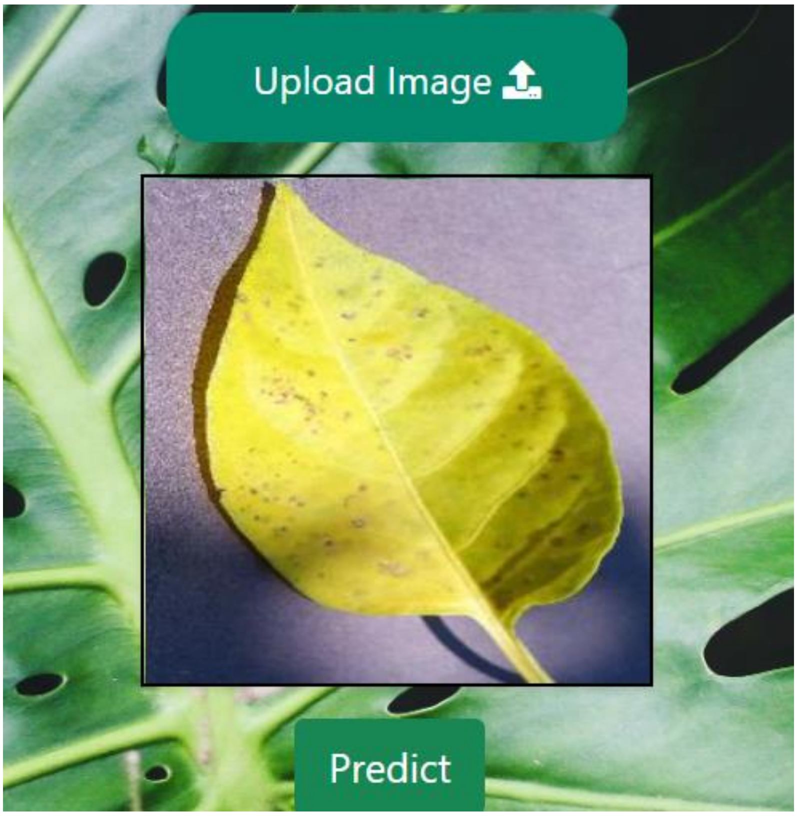
Result

The web application:

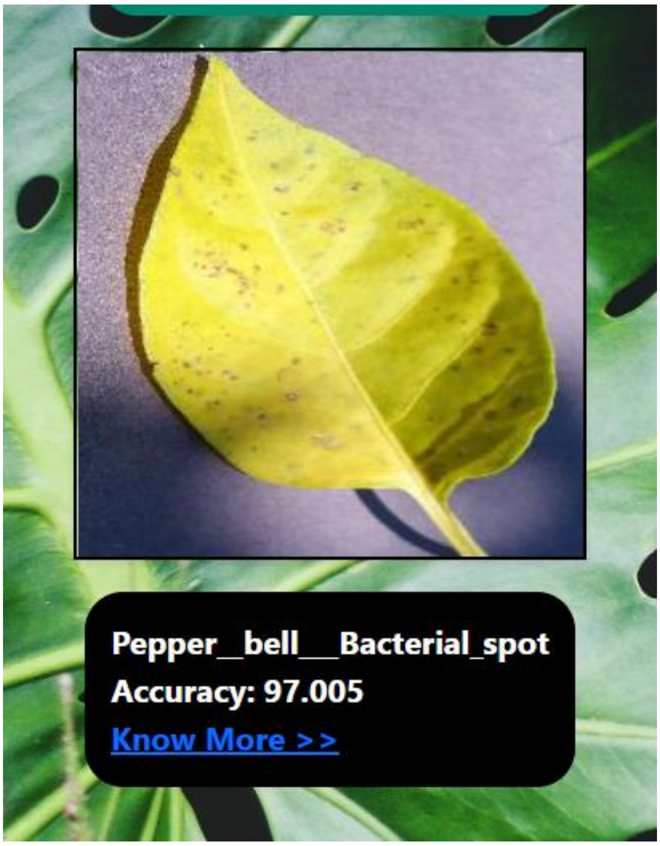


The GUI was built using Flask, JavaScript, HTML and CSS.
The welcome window has the option to upload the image.

The upload image option opens a file explorer window, allowing the user to find the image he/she wants to upload.



Once the user has selected the picture, he/she can click on predict and wait for results.



Along with the predicted result and accuracy, an additional link is provided which redirects the user to websites containing more information about the specific disease.

Conclusion

Failure of crops has always been an issue for farmers around the globe, reports of more than 50% of yield lost are frequent. Even if high measures are taken to protect crops some of them are still bound to fail. The threat to food security has only grown in the past 3 decades due to various factors like climate change and use of harsh chemicals on plantation. This problem can be mitigated by creating a system that intelligently detects plant diseases thus ensuring high yield percentage.

In our project we use concepts of Deep learning to build a model that can accurately detect diseases in a number of plants. We were able to obtain an accuracy of around **93.78%.**

The dataset contained over 16000 images belonging to 15 classes for three plants.

Dataset used: https://www.kaggle.com/datasets/emmarex/plantdisease

Web App Link: http://detect-plant-disease.herokuapp.com/

Github Link: https://github.com/Pulkit3108/Plant-Disease-Detection

Future Scope

Dataset used by us provides images with clear or no background which doesn't simulate the real-world environment. Adding more real-world images in the dataset will increase its accuracy.

Deep learning model can be trained and improved such that it can detect even the stage and severity of a particular disease.

Developments can be made to add this technology in a surveillance system to further automate the process

Adding a whole new design language to the project by introducing UI elements is also possible. Developing a web application or a native app will also help in the distribution of this idea.

References

[1] Plant diseases and pests detection based on deep learning: a review | Plant Methods | Full Text (biomedcentral.com)

[2] Richard N Strange and Peter R Scott. Plant disease: a threat to global food security. Annual review of phytopathology, 43, 2005.

[3] Yusuke Kawasaki, Hiroyuki Uga, Satoshi Kagiwada, and Hitoshi Iyatomi. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In International Symposium on Visual Computing, pages 638–645. Springer, 2015.

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

[5] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016, 2016.

[6] Nigam S, Jain R, Marwaha S, Arora A, Singh K V. 2019. Deep learning for plant disease identification. (In) Proceeding of International Conference on Agricultural Statistics, New Delhi, November 18-21.

[7] Frontiers | Using Deep Learning for Image-Based Plant Disease Detection | Plant Science (frontiersin.org)

Mentor Name **Mentor Signature**

Ms. Shahina Anwarul