LeafShield: Digital Solutions for Plant Health

A Project Report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

K.Yashwanth (2010030364)

A.Sai Pranay(2010030467)

C.Sathwik (2010030468)

S.Padmashree(2010030508)

(DEEMED TO BE UNIVERSITY)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
K L DEEMED TO BE UNIVERSITY
AZIZNAGAR, MOINABAD , HYDERABAD-500 075

MARCH 2024

BONAFIDE CERTIFICATE

This is to certify that the project titled **LeafShield:Digital Solutions for Plant Health** is a bonafide record of the work done by

K.Yashwanth (2010030364)

A.Sai Pranay(2010030467)

C.Sathwik (2010030468)

S.Padmashree(2010030508

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** of the **K L DEEMED TO BE UNIVERSITY, AZIZNAGAR, MOINABAD , HYDERABAD-500 075**, during the year 2023-2024.

Mrs. B.Uma Rani	**Dr. Arpita Gupta**
Project Guide	Head of the Department

Project Viva-voce held on ________________________

Internal Examiner	**External Examiner**

ABSTRACT

Our domain is Computer Vision and sub domain is Image Classification. our problem statement is to detect plant disease by digital image processing. By this project of plant detection, we can detect which plant it is, what are the requirements, any disease just by clicking a pic of that plant, we can come to know about all the details of that plant. This will be much fast detecting process, so that every person can know easily about a plant. One of the important tasks in agricultural practices is detection of disease on crops. It requires huge time as well as skilled labor. This paper proposes a smart and efficient technique for detection of crop disease which uses computer vision and machine learning techniques. The proposed system is able to detect 20 different diseases of 5 common plants with 93 percent accuracy.

Plant disease automation in agriculture science is the primary concern for every country, as the food demand is increasing at a fast rate due to an increase in population. Moreover, the increased use of technology today has increased the efficiency and accuracy of detecting diseases in plants and animals. The detection process marks the beginning of a series of activities to fight the diseases and reduce their spread. Some diseases are also transmitted between animals and human beings, making it hard to fight them.

For many years, scientists have researched how to deal with the common diseases that affect humans and plants. However, there are still many parts of the detection and discovery process that have not been completed. The technology used in medical procedures has not been adequate to detect all diseases on time, and that is why some diseases turn out to become pandemics because they are hard to detect on time. Our focus is to clarify the details about the diseases and how to detect them promptly with artificial intelligence. Identification of the plant diseases is the key to preventing the losses in the yield and quantity of the agricultural product. The studies of the plant diseases mean the studies of visually observable patterns seen on the plant.

Health monitoring and disease detection on plant is very critical for sustainable agriculture. It is very difficult to monitor the plant diseases manually. It requires tremendous amount of work, expertise in the plant diseases, and also require the excessive processing time. Hence, image processing is used for the detection of plant diseases. Disease detection involves the steps like image acquisition, image pre-processing, image segmentation, feature extraction and classification.

ACKNOWLEDGEMENT

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

Mrs. B.Uma Rani, our project guide, for helping us and guiding us in the course of this project .

Dr.Arpita Gupta, the Head of the Department, Department of DEPARTMENT NAME.

Our internal reviewers, **Dr. Suma Lakshmi** , **Dr. Madhukar Rao G** , **Dr Shahin Fathima** , **Mr. Pandu Raju** for their insight and advice provided during the review sessions.

We would also like to thank our individual parents and friends for their constant support.

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 Background of the Project

In our daily life we used to see a lot of plants and flowers. But in most case we have no knowledge about the specific plants and flowers. Even we are not sure about it's name.Our application recognizes the plant in real time by using pictures of the plant. This project is an attempt at using the concepts of neural networks to create an image classifier by TensorFlow. Convolutional neural networks are popular realm of machine learning, and are widely used in image classification.

The agricultural land mass is more than just being a feeding sourcing in today's world. Indian economy is highly dependent of agricultural productivity. Therefore in field of agriculture, detection of disease in plants plays an important role. To detect a plant disease in very initial stage, use of automatic disease detection technique is beneficial.

Plant disease identification by visual way is more laborious task and at the same time, less accurate and can be done only in limited areas. Whereas if automatic detection technique is used it will take less efforts, less time and become more accurate. In plants, some general diseases seen are brown and yellow spots, early and late scorch, and others are fungal, viral and bacterial diseases. Image processing is used for measuring affected area of disease and to determine the difference in the color of the affected area

1.2 Problem Statement

Our domain is Computer Vision and sub domain is Image Classification.our problem statement is to detect plant by digital image processing . By this project of plant detection, we can detect which plant it is, what are the requirements, any disease. By clicking a pic of that plant, we can come to know about all the details of that plant. This will be a faster detecting process, so that every person can know easily know about a plant .

1.3 Objectives

The objective of LeafShield is to revolutionize plant health management through digital solutions. Our goal is to empower growers, farmers, and agricultural professionals with innovative technology that enhances crop monitoring, disease detection, and pest management. By leveraging cutting-edge digital tools such as AI-driven analytics, remote sensing, and data-driven insights, LeafShield aims to:

Improve Crop Monitoring: Implement real-time monitoring systems that track plant health parameters such as moisture levels, nutrient status, and growth patterns. This enables growers to proactively address potential issues and optimize cultivation practices.

Enhance Disease Detection: Develop advanced algorithms capable of early disease detection through image recognition and spectral analysis. By identifying signs of disease onset at an early stage, growers can take prompt action to mitigate its spread and minimize crop losses.

Optimize Pest Management: Utilize predictive modeling and geospatial analysis to forecast pest outbreaks and recommend targeted interventions. By integrating pest population dynamics with environmental factors, LeafShield helps growers implement precise and eco-friendly pest management strategies.

Provide Actionable Insights: Generate actionable insights from aggregated data collected across multiple farms and regions. By analyzing trends and patterns, Leaf Shield offers personalized recommendations to optimize crop productivity, resource utilization, and sustainability.

1.4 Scope of the Project

The scope of the Leaf Shield project encompasses a comprehensive range of digital solutions tailored to address various aspects of plant health management. This includes but is not limited to:

Technology Development: Designing and developing innovative digital tools and platforms leveraging technologies such as artificial intelligence (AI), machine learning (ML), Internet of Things (IoT), and remote sensing.

Data Acquisition and Integration: Collecting and aggregating data from diverse sources including field sensors, satellite imagery, weather stations, and historical records. Integrating this data into a centralized platform for analysis and decision-making.

Crop Monitoring Systems: Implementing real-time monitoring systems to track key parameters affecting plant health such as soil moisture, nutrient levels, temperature, and humidity. Deploying sensors and IoT devices in the field to collect continuous data streams.

Disease Detection Algorithms: Developing advanced algorithms for early detection of plant diseases using techniques such as image recognition, spectral analysis, and pattern recognition. Integrating these algorithms into monitoring systems for timely intervention.

Chapter 2

Literature Review

2.1 Literature Review

Title: Image Processing Technique for the Detection of Alberseem Leaves Diseases Based on Soft Computing

Detecting plant diseases using the traditional method such as the naked eye can sometimes lead to incorrect identification and classification of the diseases. Consequently, this traditional method can strongly contribute to the losses of the crop. Image processing techniques have been used as an approach to detect and classify plant diseases.

This study aims to focus on the diseases affecting the leaves of al-berseem and how to use image processing techniques to detect al-berseem diseases. Early detection of diseases important for finding appropriate treatment quickly and avoid economic losses. Detect the plant disease is based on the symptoms and signs that appear on the leaves. The detection steps include image preprocessing, segmentation, and identification. The image noise is removed in the preprocessing stage by using the MATLAB features energy, mean, homogeneity, and others.

The k-mean-clustering is used to detect the affected area in leaves. Finally, KNN will be used to recognize unhealthy leaves and determines disease types (fungal diseases, pest diseases (shall), leaf minor (red spider), and deficiency of nutrient (yellow leaf)); these four types of diseases will detect in this thesis. Identification is the last step in which the disease will identify and classified.

Title: A Hybrid Approach for Plant Leaf Disease Detection and Classification Using Digital Image Processing Methods.

This paper presents a novel hybrid approach for the detection and classification of plant leaf diseases utilizing advanced digital image processing techniques. Plant diseases pose significant threats to agricultural productivity and food security worldwide, highlighting the urgent need for efficient disease detection methods. Traditional manual inspection methods are time-consuming and often prone to human error, underscoring the importance of automated systems for early disease diagnosis.

In this study, They propose a hybrid methodology that combines the strengths of multiple image processing algorithms to achieve accurate and robust disease detection and classification. The proposed approach involves several stages, including image acquisition, preprocessing, feature extraction, feature selection, classification, and post-processing. We employ various image processing techniques such as thresholding, morphological operations, texture analysis, and machine learning algorithms for feature extraction and classification.

Additionally, They introduce a novel feature selection method to enhance classification accuracy and reduce computational complexity. Experimental results demonstrate the effectiveness of the proposed hybrid approach in accurately detecting and classifying plant leaf diseases across different crop species and disease types. The proposed methodology offers a promising solution for early disease diagnosis and timely intervention, thereby aiding in the efficient management of plant health in agricultural systems.

Title: Image Processing Methods in Agricultural Observation Systems.

Image processing is an essential part of the agricultural observation system. This chapter is the first attempt to provide an overview of the image processing methods, technologies, and tools from the perspective of agro-geoinformatics. First, They introduce the origins, definitions, and basic steps of digital image processing. Along with the traditional image processing hardware and software, the state-of-the-art technologies for agricultural image processing, such as mobile device-based image processing and cloud computing-based image processing, are covered.

Image data could be acquired by different sensors in different ways. We discuss three common approaches to collect agricultural image data, in situ, airborne-based, and space-borne-based data collection, as well as the big data challenge in agro-geoinformatics. As the core image processing operation in the agricultural observation system, information extraction aims to understand agro-geoinformation from the raw image data.

This chapter also illustrates several image information extraction methods that are widely employed in agro-geoinformatics, such as knowledge-based expert system, machine learning-based decision tree, and artificial neural network. Furthermore, a case study of the production of Cropland Data Layer (CDL) data, a comprehensive, raster-formatted, geo-referenced, annual crop-specific land cover map produced by the U.S. Department of Agriculture (USDA) National Agricultural Statistics Service (NASS), is demonstrated.

Title	Author	Published-On	Pro’s	Con’s
Image Processing Technique for the Detection of Alberseem Leaves Diseases Based on Soft Computing.	Eimad -Abdu Abusham	April 2021	The SVM- Support Vector Machine Classifier techniques is used in this Prediction accuracy is high and having robust working when training example have error in them.	This techniques Require long training time
A Hybrid Approach for Plant Leaf Disease Detection and Classification Using Digital Image Processing Methods	Anusha Rao , S.B. Kulkarni	October 2020	It is easy to implement and gives quick result.	Slow learner and fully not robust to the noise data in large training
Image Processing Methods in Agricultural Observation Systems	Chen - Zhang, Li Lin	April 2021	It will help to sustaining the needs of today world by image processing	It requires good online datasets to get access to it
Plant Leaf disease identification using image processing and SVM, ANN classifier method	S.Siva Sakthi	May 2020	The ANN techniques have Good potential with ability to detect plant leaf disease.	It Require more time
Plant diseases and pests detection based on deep learning	Jun Liu, Xuewei Wang	February 2021	It is an upcoming method which have good potential	It is an upcoming method which have good potential

Title: Plant Leaf disease identification using image processing and SVM, ANN classifier method

Main methodology of our system is to identify diseases on the plant leaf. At first preprocessing is completed which contain two steps HSV transformation and Second phase is k-means based Image segmentation which ultimately does image analysis. Third phase is feature extraction that includes color features and texture features. And after that classification of diseases is accomplished victimization our projected formula

The objective of this analysis work is to develop Improvement automatic data processing system which will regulate the illness affected a part of a leaf spot by victimization the image analysis method. Estimate of the diseases and discuss recommendation is finished. The producers will adjust the Yield and measure back the loss. Through this predictable system the farmer's problem has been reduced and protects their life. Accuracy of recognition can be increased when using SVM classifier with advance number of features included to it. The color constructed segmentation using kmeans classifier is achieved to separate the different area with classify in the image. The SVM classifier is used to calculate the accuracy of diseased leaf region.

In the future the same system can be recycled to test all kind of plant leaves and catch out whether the plant is being affected by any disease or not .If it is affected then it furthermore displays the name of the disease.

The color based segmentation using kmeans classifier is performed to separate the different region with classify in the image. The multiclass svm classifier is used to calculate the accuracy of infected leaf region.

Title:Plant diseases and pests detection based on deep learning

Plant diseases and pests are important factors determining the yield and quality of plants. Plant diseases and pests identification can be carried out by means of digital image processing. In recent years, deep learning has made breakthroughs in the field of digital image processing, far superior to traditional methods. How to use deep learning technology to study plant diseases and pests identification has become a research issue of great concern to researchers. This review provides a definition of plant diseases and pests detection problem, puts forward a comparison with traditional plant diseases and pests detection methods.

According to the difference of network structure, this study outlines the research on plant diseases and pests detection based on deep learning in recent years from three aspects of classification network, detection network and segmentation network, and the advantages and disadvantages of each method are summarized. Common datasets are introduced, and the performance of existing studies is compared.

On this basis, this study discusses possible challenges in practical applications of plant diseases and pests detection based on deep learning. In addition, possible solutions and research ideas are proposed for the challenges, and several suggestions are given. Finally, this study gives the analysis and prospect of the future trend of plant diseases and pests detection based on deep learning.

Compared with traditional image processing methods, which deal with plant diseases and pests detection tasks in several steps and links, plant diseases and pests detection methods based on deep learning unify them into end-to-end feature extraction, which has a broad development prospects and great potential. Although plant diseases and pests detection technology is developing rapidly, it has been moving from academic research to agricultural application, there is still a certain distance from the mature application in the real natural environment, and there are still some problems to be solved.

2.2 Advantages and Limitations of existing systems

Advantages:

- **Early Detection**:-Automated systems can detect diseases at early stages, allowing farmers to take timely preventive measures and minimize crop losses.

- **High Accuracy:**:- Machine learning algorithms trained on large datasets can achieve high accuracy in disease classification, often outperforming human experts in identifying subtle symptoms.

- **Scalability**:- -Automated systems can be deployed on a large scale, enabling rapid screening of crops in vast agricultural fields, thereby increasing efficiency and productivity.

Limitations:

- **Limited Accessibility**:- Advanced technologies may require specialized equipment or technical expertise, limiting their accessibility to resource-constrained farmers, especially in developing countries.
- **Dependency on Data Quality**:-The accuracy of machine learning models depends on the quality and representativeness of the training data. Poor-quality or biased datasets can lead to inaccurate predictions and unreliable results

Chapter 3

Proposed System

3.1 Algorithms and Techniques used

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. It was developed by the Google Brain team for Google's internal use in research and production.

It invovlves the following steps:

- **Image Acquisition**:- In this images of plant leaves are gathered using digital media like camera, mobile phones etc. with desired resolution and size. The images can also be taken from web. The formation of database of images is completely dependent on the application system developer. The image database is responsible for better efficiency of the classifier in the last phase of the detection system.

- **Feature Extraction**:- In this the features from this area of interest need to be extracted. These features are needed to determine the meaning of a sample image. Features can be based on colour, shape, and texture. Recently, most of the researchers are intending to use texture features for detection of plant diseases. There are various methods of feature extraction that can be employed for developing the system such as gray-level co-occurrence matrix (GLCM), color cooccurrence method, spatial grey- level dependence matrix, and histogram based feature extraction. The GLCM method is a statistical method for texture classification.

- **Classification**:- The classification phase implies to determine if the input image is healthy or diseased. If the image is found to be diseased, some existing works have further classified it into a number of diseases. For classification, a software routine is required to be written in VS Code, also referred to as classifier. A number of classifiers have been used in the past few years by researchers such as k-nearest neighbour (KNN), support vector machines (SVM), artificial neural network(ANN), back propagation neural network (BPNN), Negatives Bayes and Decision tree classifiers. The most commonly used classifier is found to be SVM. Every classifier has its advantages and disadvantages, SVM is simple to use and robust technique.

Feature Detection and Matching using SVM (Support Vector Machine)

- Feature Extraction.
- Feature Matching.
- Generating Training Data.
- Feature Representation.
- Training the SVM Classifier.
- Post-Processing
- Display Output

3.2 System Requirements

Hardware:

- 4GB RAM
- computer system with Intel Core 2Duo E7300
- faster graphic card

Software:

- **VS Code**:- Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE

- **Python**:-Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

- **TensorFlow**:-TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. It was developed by the Google Brain team for Google's internal use in research and production.

- **Numpy**:-Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

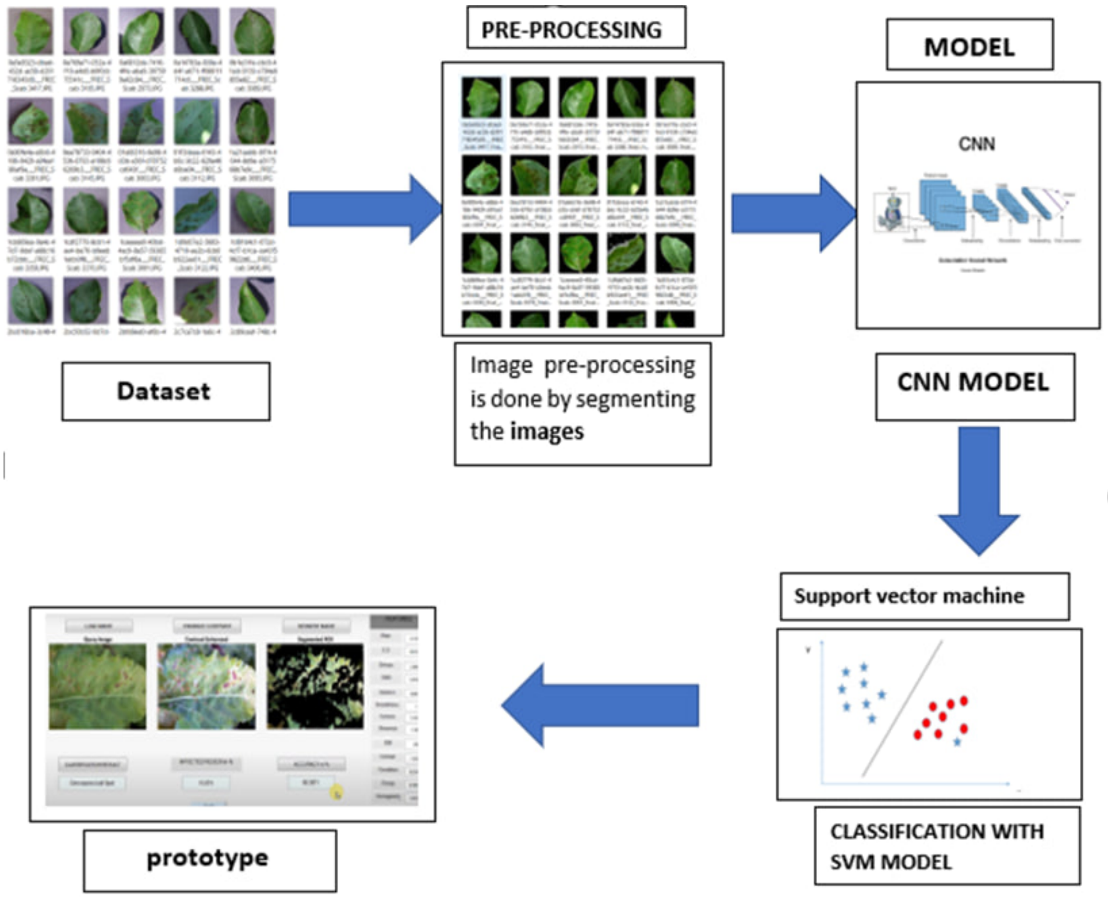

Figure 3.1: **Flowchart**

Chapter 4

Implementation

4.1 Tools and Technologies used

- Dataset
- Image Processing
- Feature Extraction using CNN Model and SVM Model
- Python
- Machine Vision and python libraries

DATASETS	CHRACTERISTIC	REFERENCE LINK	TECHNIQUES And Models
Plant Village	It is a dataset that contains roughly 54,000 plant leaf picture	https://www.kaggle.com/abdallahalidev/plantvillage-dataset	1.Data Augmentation like:- rotation, scaling, flipping 2. image segmentation.
PlantDoc	This dataset contains the details about plant diseases and different image	https://github.com/pratikkayal/PlantDoc-Dataset	1.ResnetV2 2.MobileNet
Edible wild plant	This dataset contain 62 edible wild plant Images	https://www.kaggle.com/gverzea/edible-wild-plants	1.Resnet50 3.CNN Model
Ornamental plants	This dataset contain 500 Images of Flower	https://www.kaggle.com/abdalnassir/ornamental-plants	1.Image classification 2.Instance segmentation

Figure 4.1: **Dataset**

4.2 Modules and their descriptions

CNN MODEL:- It is a type of neural network model which allows us to extract higher representations for the image content.

CNN- It is a convolutional neural network it is a deep neural networks, most commonly it is applied to analyze visual imagery. It is one of the main categories to do images recognition, classification, Segmentation. object detection, recognition faces. In this it takes an input image , process it and classify it. Computer takes Input under pixels and based on image resolution it process the image and classify them.

SVM:-Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine

4.3 TECHNIQUES:-

IMAGE SEGMENTATION:-It is the process by which a digital image is partitioned into various subgroups (of pixels) called Image Objects, which can reduce the complexity of the image, and thus analyzing the image becomes simpler.

DATA AUGMENTATION:-It generate different versions of a real dataset artificially to increase its size. Computer vision and natural language processing (NLP) models use data augmentation strategy to handle with data scarcity and insufficient data diversity.

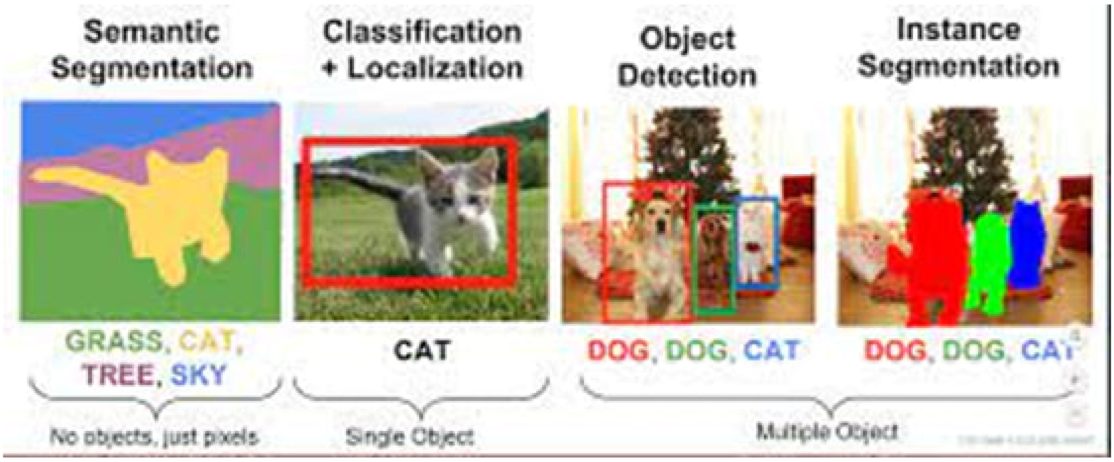

Figure 4.2: **Segmentation**

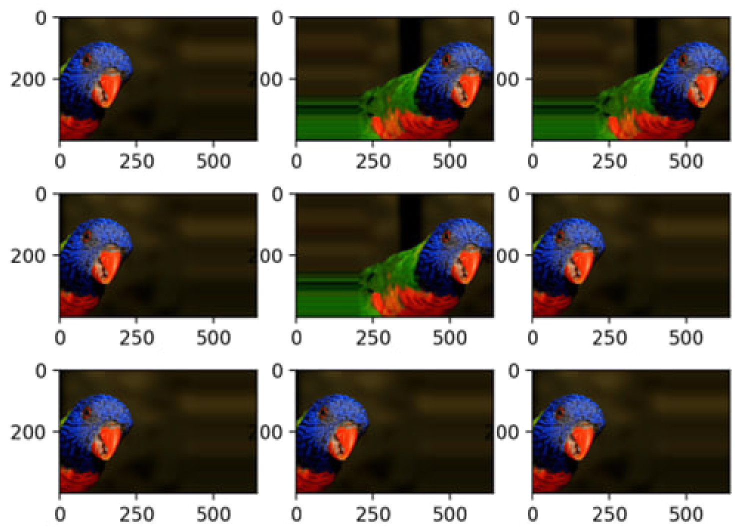

Figure 4.3: **Agumentation**

Chapter 5

Results and Analysis

5.1 Performance Evaluation

- Scalability
- Accuracy
- Comparative Analysis
- Visual Interpretation

5.2 Comparison with existing systems

- Blur Images can be Accepted
- Portable and Reliable
- Uses Machine Visison for better results
- Well Trained Datasets

5.3 Limitations and future scope

Limitations:

- Data Availability
- Disease Progression
- Hardware Constraints

Future Scope:

- Explainable AI
- Large-scale Data Collection
- Automated Phenotyping

Chapter 6

Conclusion and Recommendations

6.1 Summary of the Project

Our project is all about identification of plant Disease. Our developed Web application take the image of a plant and identify the disease it is suffering from. So that it will help the farmers to predict the disease very fast comparing today's techniques where they need many skilled labours and required much time. So, this web application will reduce both time and also detect accurate disease which helps the farmer to take perfect steps to cure the plant without any help of skilled labour. We have gone through some image processing techniques and used CNN model to complete this project.

6.2 Contributions and achievements

A project description is a high-level overview of why you're doing a project. The document explains a project's objectives and its essential qualities. Think of it as the elevator pitch that focuses on what and why without delving into how. the proposed LeafShield: Digital Solutions for Plant Health model with its multi-denomination support, feature extraction, algorithmic analysis, and user-friendly interface is a valuable contribution to detect the disease of a plant as early as possible. It offers a practical solution to address the ongoing challenge of disease detection for the farmer in cultivation of crop in circulation and can be a useful tool for individuals, farmer and for organic companies. By offering a solution in plant disease detection have revolutionized agriculture by enabling early disease detection, improving crop health management, enhancing disease surveillance, supporting precision agriculture applications, advancing crop breeding programs, empowering farmers, fostering interdisciplinary collaboration, and contributing to global food security efforts. Continued innovation and investment in disease detection technologies are essential for addressing current and emerging challenges in agricultural sustainability and food production.

Early Disease Detection: One of the key objectives of this project is to detect the disease as early as possible.Plant disease detection systems have enabled early and accurate identification of diseases in crops. This early detection allows farmers to implement timely management strategies, such as targeted pesticide application or disease-resistant crop planting, to minimize crop losses and maximize yields

Enhanced Disease Surveillance: The project's feature Enhanced the detection systems contribute to enhanced disease surveillance efforts at local, regional, and global scales. By monitoring disease incidence and prevalence in real-time, these systems help identify emerging diseases, track disease spread, and inform disease control measures and policy decisions.

Improved Crop Health Management: By providing farmers with timely information about disease outbreaks and pest infestations, disease detection systems support proactive crop health management. Farmers can implement integrated pest management (IPM) practices, including cultural, biological, and chemical control methods, more effectively to prevent and manage diseases.

6.3 Recommendations for future work

Many different adaptations, tests and innovations have been kept for the future due to the lack of time. As future work concerns deeper analysis of particular mechanisms, new proposals to try different methods or simple curiosity.

1. Explainable AI and Interpretability:

As a starting point, the current project focuses on detecting disease in plants. However, the principles and techniques applied in this project can be extended with a explainable AI techniques to enhance the interpretability of disease detection models, allowing users to understand the basis of model predictions and build trust in automated decision-making processes. Interpretability is crucial for fostering adoption and acceptance of disease detection technologies by farmers and stakeholders.

Challenges and Opportunities:

- **Complexity of Models:** Many AI models, particularly deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are inherently complex and opaque. Understanding the inner workings of these models and explaining their decisions in human-interpretable terms can be challenging.

- **Black-box Nature:** Some AI models operate as black boxes, making it difficult to discern how they arrive at specific decisions or predictions. This lack of transparency can hinder trust and acceptance, particularly in critical domains such as healthcare, finance, and criminal justice.

- **Insight Generation:**Interpretability techniques can provide valuable insights into the underlying patterns and relationships in data, helping users better understand complex phenomena and uncover actionable insights.

- **Regulatory Compliance:** Regulatory bodies and standards organizations are increasingly emphasizing the importance of transparency and accountability in AI systems. XAI techniques can help organizations comply with regulatory requirements by providing transparent explanations for AI-driven decisions.

2. Real-time Monitoring and Decision Support:

In today's digital age, Designing real-time monitoring systems equipped with wireless sensor networks, IoT devices, and mobile applications to enable continuous surveillance of plant health and disease status. Develop decision support systems that provide actionable insights and recommendations to farmers for timely intervention and disease management.

- **Data Quality and Reliability:** Ensuring the quality, reliability, and integrity of real-time data streams is crucial for making accurate and timely decisions. Challenges such as data noise, missing values, sensor errors, and network latency can affect the reliability of real-time monitoring systems and decision support algorithms.

- **Human-System Interaction:** Effectively integrating human expertise and judgment into real-time decision support systems is essential for ensuring user acceptance and trust.

- **User Engagement:** Integrating with a mobile app could encourage users to actively participate in counterfeit currency prevention and reporting. Users could easily report counterfeit notes they come across.

- **Enhanced Decision Support:** Real-time decision support systems provide stakeholders with actionable insights, recommendations, and decision alternatives based on real-time data analysis and modeling.

Bibliography

[1] Peng Jiang , Yuehan Chen , Bin Liu , Dongjian He , Chunquan Liang ,' Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks', (Volume: 7), pp. 06 May 2019.

[2] Zhou, R., Kaneko, S., Tanaka, F., Kayamori, M., Shimizu, M., 'Disease detection of Cercospora Leaf Spot in sugar beet by robust template matching', Computers and Electronics in Agriculture, Volume 108, pp.58-70, 2014.

[3] Barbedo, J.G.A., Godoy, C.V., 'Automatic Classification of Soybean Diseases. Based on Digital Images of Leaf Symptoms', SBI AGRO,2015.

[4] Barbedo, J.G.A., 'A review on the main challenges in automatic plant disease identification based on visible range images', 2016 , Biosystems Engineering,Volume 144, pp. 52-60

[5] Bashish, D.A., Braik, M., Ahmad, S.B., 'A Framework for Detection and Classification of Plant Leaf and Stem Diseases', International Conference on Signal and Image Processing, pp. 113-118, 2010.

Appendices

Appendix A

Source code

app.py:

```
import os
import tensorflow as tf
import numpy as np
from tensorflow import keras
from skimage import io
from tensorflow.keras.preprocessing import image
import skimage as ski
os.environ['TF_CPP_MIN_LOG_LEVEL']='2'


# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer


app = Flask(name)

model =tf.keras.models.load_model('PlantDNet.h5',compile=False)
print('Model loaded. Check http://127.0.0.1:5000/')


def model_predict(img_path, model):
    img = image.load_img(img_path, grayscale=False, target_size=(64,
    64))
    show_img = image.load_img(img_path, grayscale=False, target_size
    =(64, 64))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = np.array(x, 'float32')
    x /= 255
    preds = model.predict(x)
    return preds


@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')

```

```

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':

        f = request.files['file']


        basepath = os.path.dirname(file)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)

        # Make prediction
        preds = model_predict(file_path, model)
        print(preds[0])

        # x = x.reshape([64, 64]);
        disease_class = ['Pepper__bell___Bacterial_spot', 'Pepper__bell___healthy', 'Potato___Early_blight',
                         'Potato___Late_blight', 'Potato___healthy', 'Tomato_Bacterial_spot', 'Tomato_Early_blight',
                         'Tomato_Late_blight', 'Tomato_Leaf_Mold', 'Tomato_Septoria_leaf_spot',
                         'Tomato_Spider_mites_Two_spotted_spider_mite', 'Tomato__Target_Spot',
                         'Tomato__Tomato_YellowLeaf__Curl_Virus', 'Tomato__Tomato_mosaic_virus', 'Tomato_healthy']
        a = preds[0]
        ind=np.argmax(a)
        print('Prediction:', disease_class[ind])
        result=disease_class[ind]
        return result
    return None


if name == 'main':

    http_server = WSGIServer(('', 5000), app)
    http_server.serve_forever()
    app.run()
```

import.html:

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Plant Disease Diagnosis</title>
    <link href="{{ url_for('static', filename='css/bootstrap.min.css') }}" rel="stylesheet">
    <script src="{{ url_for('static', filename='js/jquery.min.js') }}"></script>
    <script src="{{ url_for('static', filename='js/bootstrap.min.js') }}"></script>
    <link href="{{ url_for('static', filename='css/test.css') }}" rel="stylesheet">
    <script src="{{ url_for('static', filename='js/newjs.js') }}" type="text/javascript"></script>
</head>

<body>
    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">Plant Disease Diagnosis</a>

        </div>
    </nav>

    {% block content %} {% endblock %}

</body>

<footer>
    <div class="foot">

    </div>

</footer>

</html>
```

index.html:

```
{% extends "import.html" %}
{% block content %}

<center>
<br><h2>Plant Disease Diagnosis</h2><br>

    <form id="upload-file" method="post" enctype="multipart/form-data
    ">
         <input type="file" name="file" class="btn btn-success" id="
    imageUpload" accept=".png, .jpg, .jpeg">
    </form>


    <div class="image-section" style="display:none;">
         <img id="imagePreview" class="img-responsive" src="#" style
    ="width:300px;height:300px;"/><br><br>
         <div>
            <button type="button" class="btn btn-info btn-lg " id="
    btn-predict">Predict!</button>
         </div>
    </div>



    <div class="loader" style="display:none;"></div>

    <h3 id="result">
        <span> </span>
    </h3>


</center><br><br>
{% endblock %}
```

Appendix B

Screen shots

B.1 Results

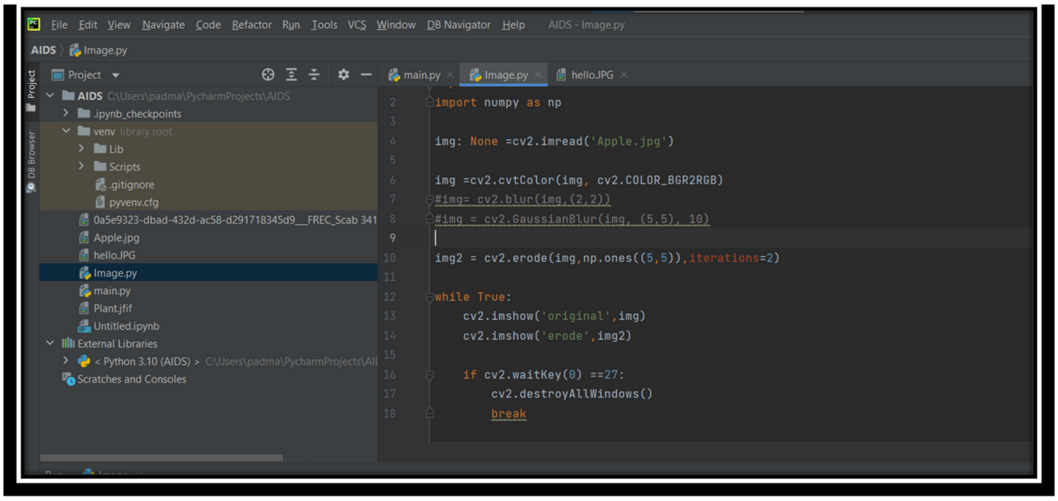

Figure B.1: RGB Image Processing

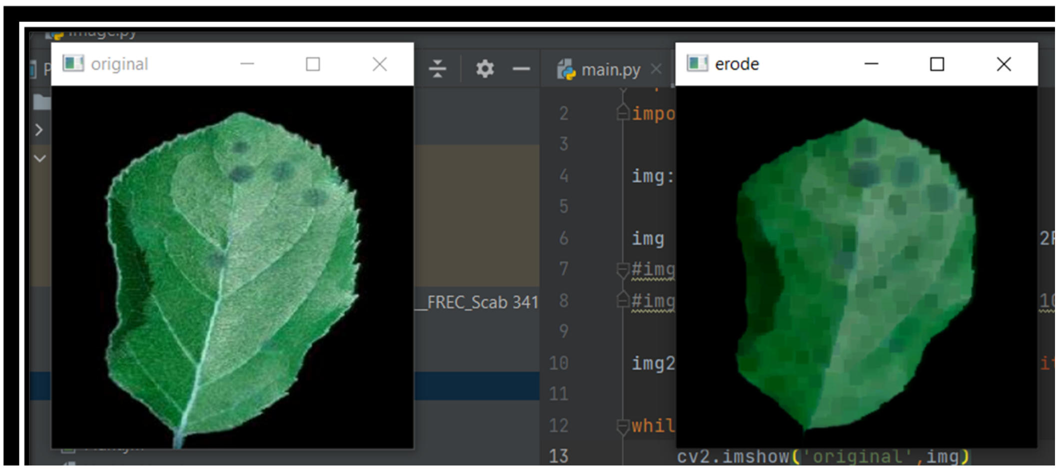

Figure B.2: RGB Image Processing

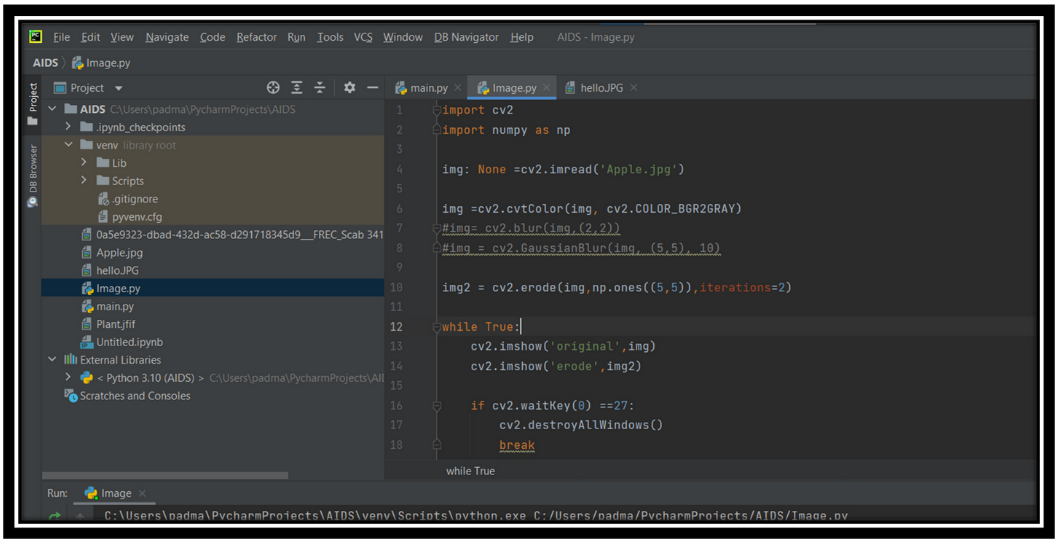

Figure B.3: Gray scale image

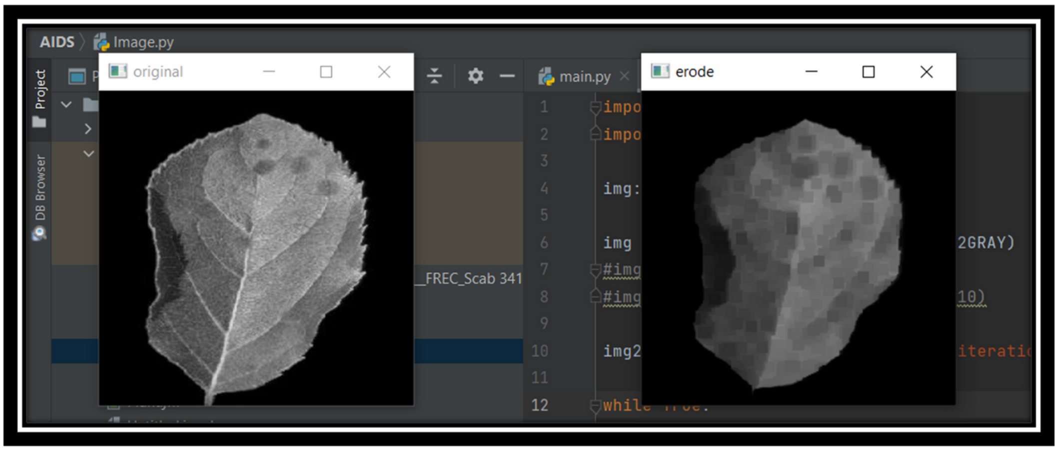

Figure B.4: Gray scale image

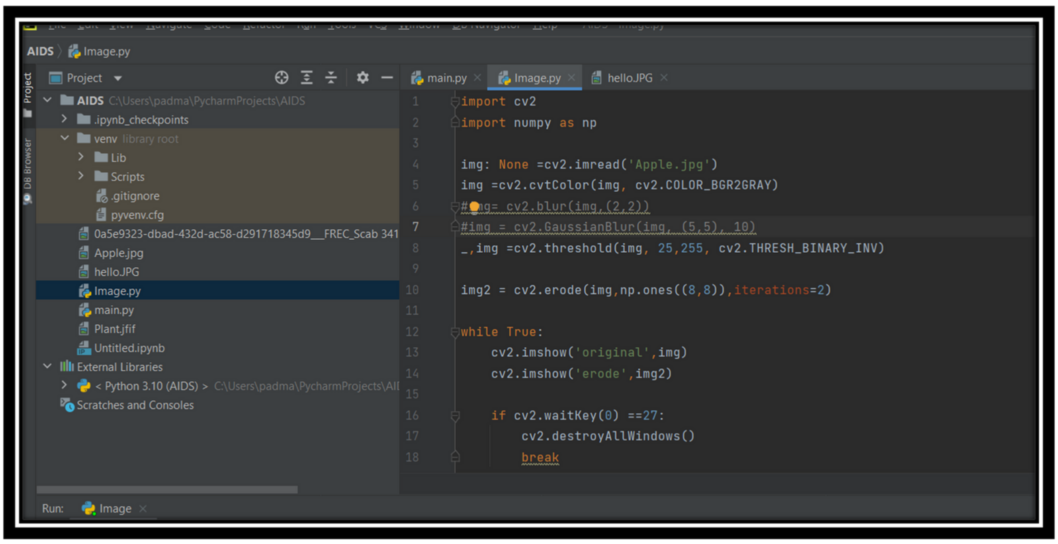

Figure B.5: Thresholding Image

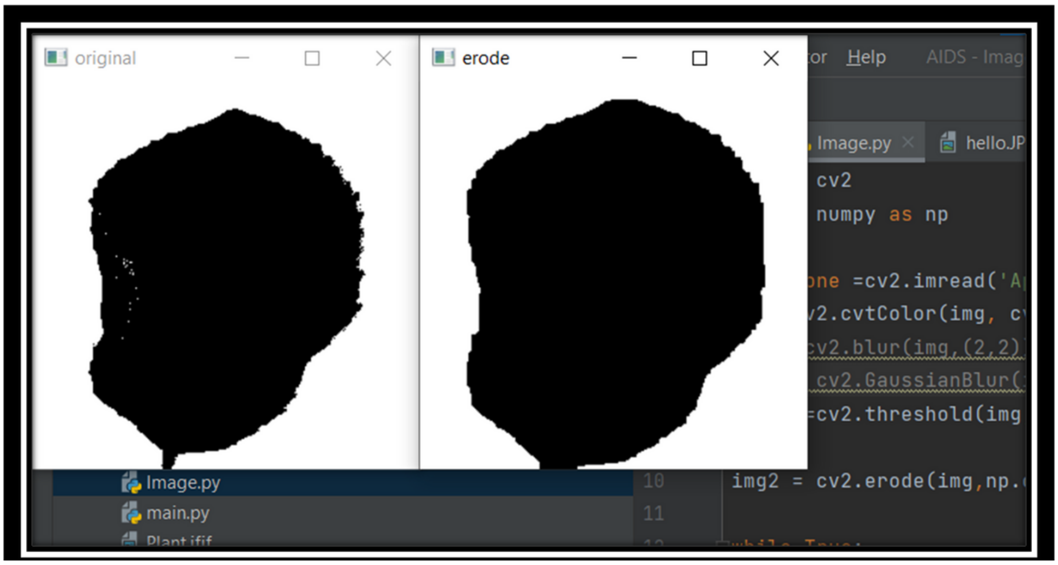

Figure B.6: Thresholding Image

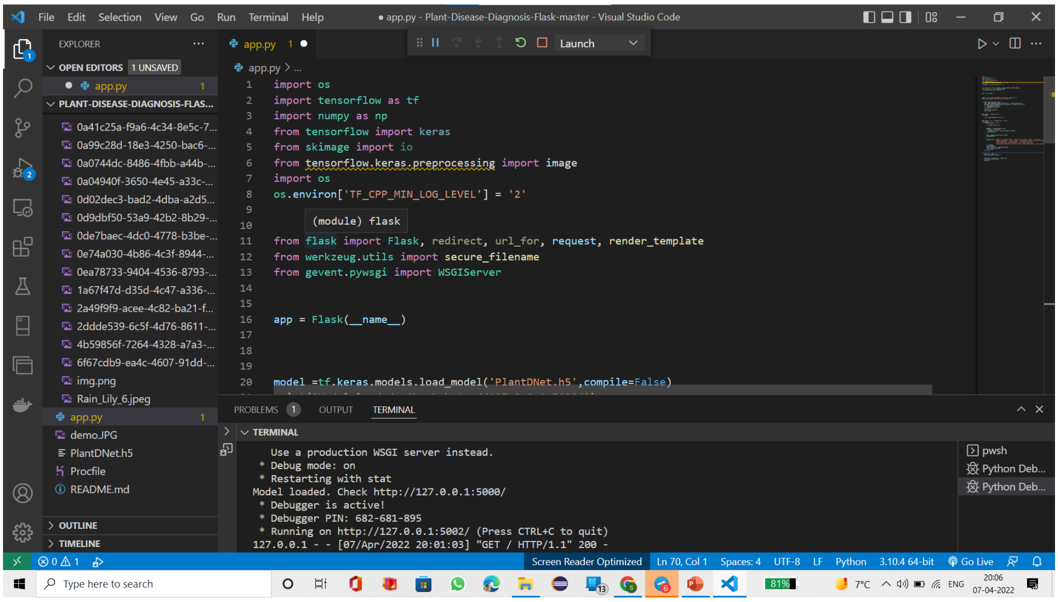

Figure B.7: app.py

```python
import os
import tensorflow as tf
import numpy as np
from tensorflow import keras
from skimage import io
from tensorflow.keras.preprocessing import image
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'


from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer


app = Flask(__name__)



model =tf.keras.models.load_model('PlantDNet.h5',compile=False)
print('Model loaded. Check http://127.0.0.1:5000/')


def model_predict(img_path, model):
    img = image.load_img(img_path, grayscale=False, target_size=(64, 64))
    show_img = image.load_img(img_path, grayscale=False, target_size=(64, 64))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = np.array(x, 'float32')
    x /= 255
```

Figure B.8: app.py

```python
    preds = model.predict(x)
    return preds


@app.route('/', methods=['GET'])
def index():

    return render_template('index.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':

        f = request.files['file']


        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)


        preds = model_predict(file_path, model)
        print(preds[0])


        disease_class = ['Pepper__bell___Bacterial_spot', 'Pepper__bell___healthy', 'Potato___Early_blight',
                         'Potato___Late_blight', 'Potato___healthy', 'Tomato_Bacterial_spot', 'Tomato_Early_blight',
                         'Tomato_Late_blight', 'Tomato_Leaf_Mold', 'Tomato_Septoria_leaf_spot',
```

Figure B.9: app.py

app.py > model_predict

```python
                                                'Tomato_Late_blight', 'Tomato_Leaf_Mold', 'Tomato_Septoria_leaf_spot',
                                                'Tomato_Spider_mites_Two_spotted_spider_mite', 'Tomato__Target_Spot',
                                                'Tomato__Tomato_YellowLeaf__Curl_Virus', 'Tomato__Tomato_mosaic_virus', 'Tomato_healthy']
            a = preds[0]
            ind=np.argmax(a)
            print('Prediction:', disease_class[ind])
            result=disease_class[ind]
            return result
        return None


if __name__ == '__main__':
    app.run(port=5002, debug=True)


    http_server = WSGIServer(('', 5000), app)
    http_server.serve_forever()
    app.run()
```

Figure B.10: app.py

```
2022-05-06 12:14:28.303727: I tensorflow/stream
_executor/cuda/cuda_diagnostics.cc:176] hostnam
e: LAPTOP-1T9SDL78
2022-05-06 12:14:28.308244: I tensorflow/core/p
latform/cpu_feature_guard.cc:151] This TensorFl
ow binary is optimized with oneAPI Deep Neural
Network Library (oneDNN) to use the following C
PU instructions in performance-critical operati
ons:  AVX AVX2
To enable them in other operations, rebuild Ten
sorFlow with the appropriate compiler flags.
Model loaded. Check http://127.0.0.1:5000/
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do no
t use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
Model loaded. Check http://127.0.0.1:5000/
 * Debugger is active!
 * Debugger PIN: 762-973-976
 * Running on http://127.0.0.1:5002/ (Press CTR
L+C to quit)
```

Figure B.11: output 1

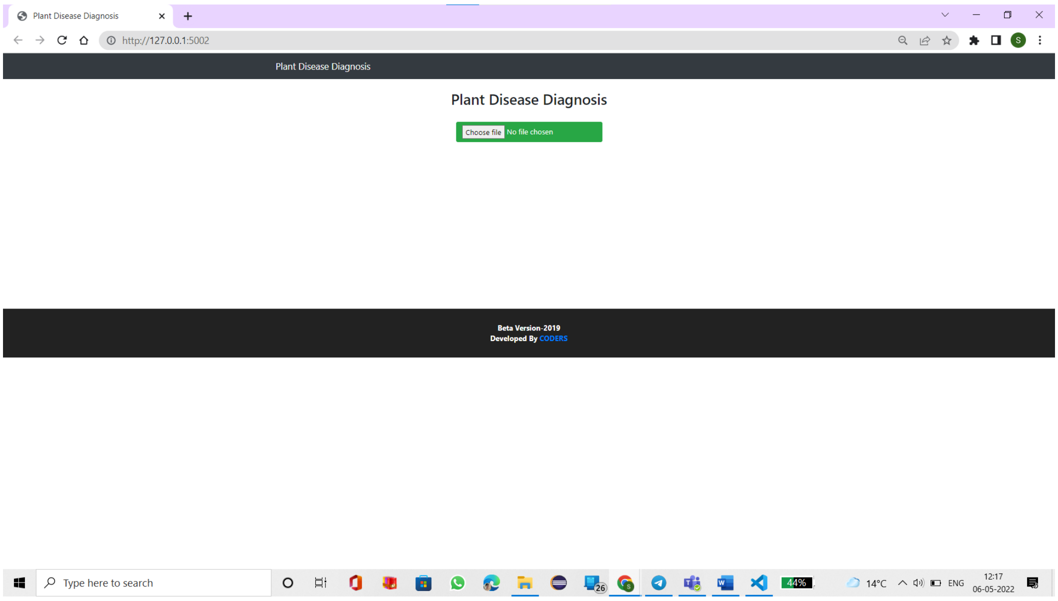

Figure B.12: output 2

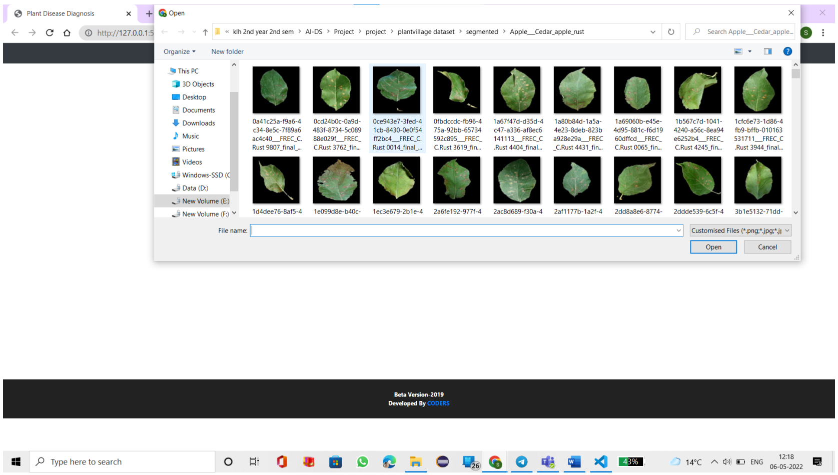

Figure B.13: output 3

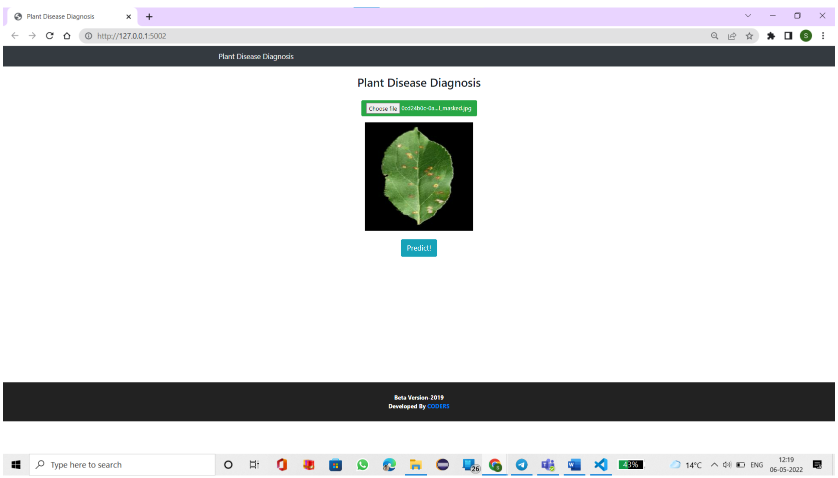

Figure B.14: output 4

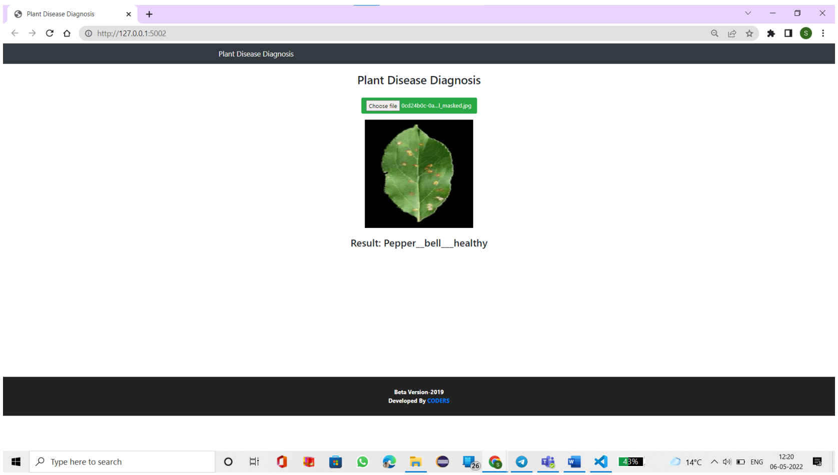

Figure B.15: output 5

Appendix C

Data sets used in the project

Creating a dataset for leaf classification typically involves collecting images or data that represent both various types of leaves. Below, I'll provide you with a sample dataset structure that you can use as a template. This dataset can be used for training and evaluating machine learning models for leaf classification. In practice, you should collect a larger dataset with a variety of leaf types, different lighting conditions, and various angles to create an effective model. Remember that creating a comprehensive and effective dataset for leaf classification may require a substantial amount of effort and careful curation to ensure the dataset is both diverse and accurate. Additionally, you may want to consider obtaining the necessary permissions or complying with relevant laws when working with images of leaves.

Figure C.1: color

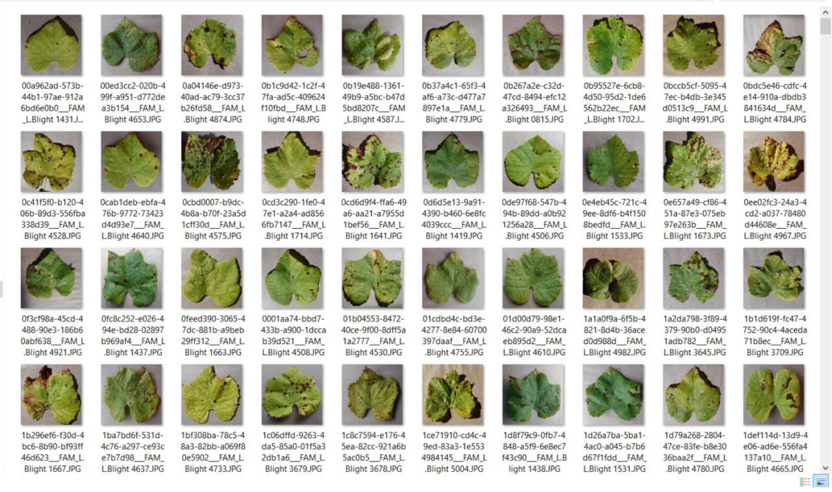

Figure C.2: color

Figure C.3: gray scale

Figure C.4: gray scale

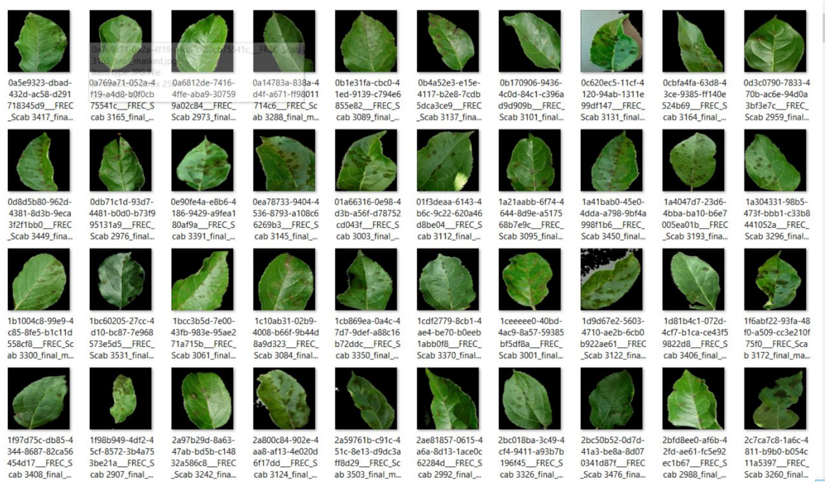

Figure C.5: segmented

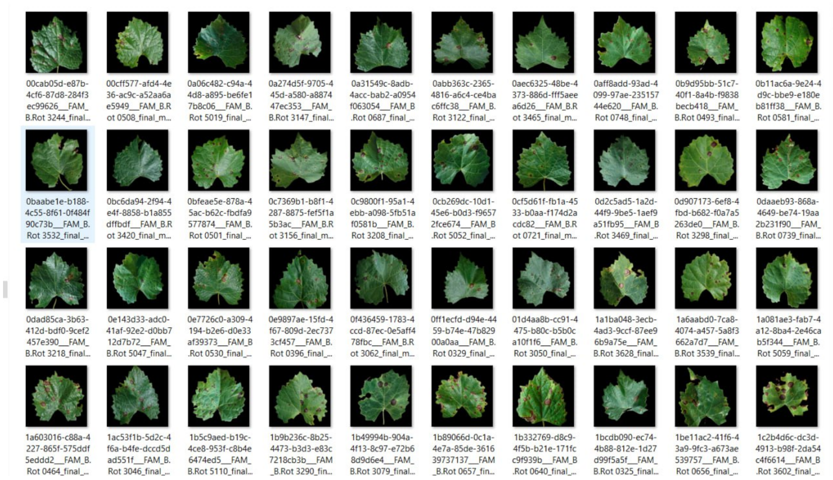

Figure C.6: segmented