A Major Project Stage-1 Report

on

AN EXPERT SYSTEM FOR DISEASE PREDICTION AND FERTILIZER RECOMMENDATION USING DEEP LEARNING

Submitted in partial fulfillment of the requirements
for the award of degree of

BACHELOR OF TECHNOLOGY

in

Information Technology

by

K. Geethika Reddy (20WH1A1270)

G. Krishna Prathibha (20WH1A12B0)

G. Sneha (20WH1A12B5)

Under the esteemed guidance of

Mr. A. Rajashekar Reddy

Assistant Professor

Department of Information Technology

BVRIT HYDERABAD College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyd–500090

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

(NAAC 'A' Grade & NBA Accredited- ECE, EEE, CSE & IT)

December 21, 2023

BVRIT HYDERABAD

College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090
(Affiliated to Jawaharlal Nehru Technological University Hyderabad)
(NAAC 'A' Grade & NBA Accredited- ECE, EEE, CSE & IT)

CERTIFICATE

This is to certify that the Project report on **" An Expert System For Disease Prediction And Fertilizer Recommendation Using Deep Learning "** is a bonafide work carried out by **K Geethika Reddy (20WH1A1270), G Krishna Prathibha (20WH1A12B0)** and **G Sneha(20WH1A12B5)** in the partial fulfillment for the award of B.Tech degree in **Information Technology , BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad** affiliated to Jawaharlal Nehru Technological University, Hyderabad, under my guidance and supervision. The results embodied in the project work have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide
A. Rajashekar Reddy
Assistant Professor
Department of IT

Head of the Department
Dr. Aruna Rao S.L
Professor & HoD
Department of IT

External Examiner

DECLARATION

We hereby declare that the work presented in this project entitled **"An Expert System For Disease Prediction And Fertilizer Recommendation Using Deep Learning"** submitted towards completion of in IV year I sem of B.Tech IT at "BVRIT HYDERABAD College of Engineering for Women", Hyderabad is an authentic record of our original work carried out under the esteemed guidance of **Mr. A. Rajashekar Reddy, Assistant Professor**, Department of Information Technology.

K. Geethika Reddy (20WH1A1270)

G. Krishna Prathibha (20WH1A12B0)

G. Sneha (20WH1A12B5)

ACKNOWLEDGMENTS

We would like to express our profound gratitude and thanks to **Dr K.V.N. Sunitha, Principal, BVRIT HYDERABAD College of Engineering for Women** for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. Aruna Rao S.L, Professor & Head, Department of IT, BVRIT HYDERABAD College of Engineering for Women** for all the timely support, constant guidance and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Mr. A. Rajashekar Reddy, Assistant Professor, Department of IT, BVRIT HYDERABAD College of Engineering for Women** for his constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinators **Dr. J Kavitha, Associate Professor and Mr. Mukhtar Ahmad Sofi , Assistant Professor**, all the faculty and staff of Department of IT who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

K. Geethika Reddy (20WH1A1270)

G. Krishna Prathibha (20WH1A12B0)

G. Sneha (20WH1A12B5)

ABSTRACT

In the realm of agriculture, the production of tomatoes holds critical importance for global food security and economic stability. However, this vital crop is threatened by a range of diseases, and inefficient fertilizer application exacerbates these issues, leading to substantial losses for farmers. While there exist Tomato plant disease detection systems with different efficient advanced deep learning models, these disease detection systems often lack fertilizer recommendations. This project aims to bridge this gap by introducing an integrated expert system for disease detection and efficient fertilizer recommendation for tomato plants. The proposed methods for this expert system involve utilizing advanced deep learning models such as MobileNet and DenseNet for disease detection. Additionally, rule-based machine learning algorithms, including decision trees, random forests, and XGBoost, are employed for fertilizer recommendation. For further experiments and improvements, alternative algorithms such as Support Vector Machine or Naïve Bayes algorithms will be explored. The "tomato village" dataset is used for training and validation of the system. The proposed expert system for disease detection combined with fertilizer recommendation has the potential to revolutionize tomato cultivation practices, enhancing crop yield, reducing environmental impact, and ultimately bolstering agricultural sustainability. Its implementation promises a brighter and more resilient future for tomato cultivation, offering a comprehensive solution to the challenges faced by farmers and contributing to a more sustainable and efficient agricultural industry.

Keywords: Expert System, Tomato Leaf disease detection, Fertilizer recommendation, MobileNet, DenseNet, Rule-based machine learning

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The global agriculture sector, particularly concerning food security and economic stability, heavily relies on tomato production. However, the tomato crop faces significant threats from various diseases, exacerbated by inefficient fertilizer practices, leading to substantial losses for farmers. While advanced deep learning models have been successful in detecting tomato plant diseases, existing systems often lack integrated fertilizer recommendations. This project aims to bridge this gap by proposing an expert system that seamlessly combines disease detection with efficient fertilizer recommendations for tomato plants.The methodology involves utilizing cutting-edge deep learning models such as MobileNet and DenseNet for disease detection. Additionally, rule-based machine learning algorithms like decision trees, random forests, and XGBoost are employed to provide precise fertilizer recommendations. Future iterations may explore alternative algorithms like Support Vector Machine or Naïve Bayes for further enhancements. The system is trained and validated using a comprehensive dataset.The integration of disease detection and fertilizer recommendation in this expert system has the potential to transform tomato cultivation practices, optimizing crop yield, reducing environmental impact, and enhancing agricultural sustainability. Implementation of this system promises a more resilient future for tomato cultivation, providing a holistic solution to farmers' challenges and contributing to a sustainable and efficient agricultural industry.

1.1 Python Libraries

1.1.1 TensorFlow

TensorFlow, developed by Google, is a powerful open-source machine learning library. Its core strength lies in its computational graph architecture, where operations are organized into a graph for efficient execution. Tensors, multi-dimensional arrays, are the fundamental data structures in TensorFlow, representing input and output data. TensorFlow supports a variety of operations for manipulating tensors, facilitating complex mathematical computations. The library is widely used for building and training deep learning models, offering a high-level API through TensorFlow Keras. TensorFlow provides extensive support for both CPU and GPU computations, optimizing performance. It has gained popularity for its flexibility, scalability, and ease of use in implementing machine learning algorithms. TensorFlow's ecosystem includes tools like TensorBoard for visualization and TensorFlow Lite for deploying models on edge devices, making it versatile for various applications, from research to production. Overall, TensorFlow plays a pivotal role in advancing the field of machine learning and artificial intelligence.

1.1.2 OpenCV

OpenCV, or Open Source Computer Vision Library, is a versatile open-source software library widely used for image and video processing, computer vision, and machine learning. Developed initially by Intel, it supports multiple programming languages, including C++, Python, and Java. OpenCV offers a comprehensive set of functions and tools for tasks like image manipulation, object recognition, and machine learning. Its modular design facilitates easy integration into various projects, making it a popular choice among researchers, developers, and hobbyists for a wide range of computer vision applications.The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, extract 3D models of objects, find similar images from an image database, remove red eyes from images taken using flash, etc.

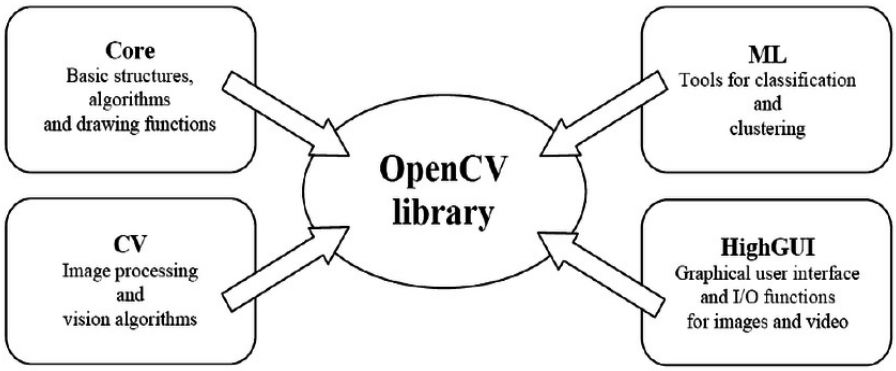


Figure 1.1: Usage of OpenCV in different domains

1.1.3 Keras

Keras is a high-level neural networks application programming interface (API) written in Python, serving as a user-friendly interface for building and training deep learning models. Known for its simplicity and modularity, Keras enables both beginners and experienced practitioners to easily construct and experiment with neural network architectures. It allows the creation of complex models by stacking or merging various building blocks, facilitating quick prototyping. Originally developed independently, Keras has been integrated seamlessly into TensorFlow, one of the most popular open-source machine learning libraries. This integration enhances the capabilities of both, combining the simplicity of Keras with the robustness of TensorFlow. With a focus on ease of use, Keras supports a wide range of neural network configurations and is widely utilized for tasks such as image classification, natural language processing, and more. Its modularity and compatibility contribute to its popularity among researchers and developers seeking a versatile and efficient tool for deep learning applications.

1.1.4 Scikit-learn

Scikit-learn is a comprehensive and user-friendly machine learning library in Python, providing tools for data analysis, modeling, and evaluation. With a consistent API and extensive documentation, it offers a straightforward approach to machine learning tasks. Scikit-learn includes a wide range of algorithms for classification, regression, clustering, and dimensionality reduction. The library supports data preprocessing and feature engineering, making it suitable for end-to-end machine learning workflows. Its built-in datasets and utilities simplify the process of experimenting with machine learning techniques. Scikit-learn empha-

sizes code readability and ease of use, making it accessible to both beginners and experienced practitioners. It integrates seamlessly with other scientific computing libraries like NumPy, SciPy, and Matplotlib. The library plays a pivotal role in the Python ecosystem, contributing to the popularity and growth of machine learning applications.

1.1.5 Seaborn

Seaborn is a powerful data visualization library built on top of Matplotlib in Python, designed for creating informative and attractive statistical graphics. In just a few lines, Seaborn allows users to generate complex visualizations with ease, enhancing the aesthetics of data representation. It simplifies the creation of various plots, including scatter plots, bar plots, and heatmaps, by providing high-level functions with informative default settings. Seaborn's integration with Pandas DataFrames makes it convenient for working with structured data. The library offers additional features like color palettes and themes, allowing customization of visualizations for better interpretation. With concise syntax, Seaborn enables quick exploration of datasets and relationships between variables. Its ability to handle categorical data and automatic estimation of confidence intervals enhances its utility in statistical analysis. Seaborn's combination of simplicity and versatility makes it a popular choice for researchers, analysts, and data scientists engaged in exploratory data analysis and visualization in Python.

1.1.6 Pandas

Pandas is a versatile and indispensable data manipulation library in Python, renowned for its efficiency in handling structured data. In just a few lines of code, Pandas empowers users to effortlessly load, clean, and analyze datasets. Its core data structures, Series and DataFrame, facilitate the representation of one-dimensional labeled arrays and two-dimensional tables, respectively. With intuitive methods such as 'head()', 'info()', and 'describe()', Pandas enables rapid exploration of datasets, providing essential insights into data types, statistics, and overall structure. Pandas supports powerful grouping and aggregation operations, making it an invaluable tool for summarizing and analyzing data based on specific criteria. Its seamless handling of time series data, efficient merging and joining capabilities, and integration with Matplotlib for easy plotting contribute to Pandas' status as an essential library for data scientists, analysts, and researchers engaged in data manipulation and analysis in Python.

1.2 Motivation

The motivation behind this project is rooted in addressing the critical challenges faced by tomato cultivation within the agricultural sector. Recognizing the pivotal importance of tomatoes for global food security and economic stability, the project is driven by the need to combat the detrimental impact of diseases and inefficient fertilizer application on tomato crops. The current reliance on labor-intensive manual inspection for disease detection and imprecise conventional fertilizer recommendation systems underscores the urgency for a more effective and streamlined approach. The project's objective is to introduce an integrated system that harnesses advanced computer vision and machine learning techniques, utilizing data-driven algorithms to offer rapid and accurate suggestions for both disease detection and precise fertilizer formulations. The motivation stems from the aspiration to significantly improve upon existing methods, enhance crop yield, minimize environmental impact, and ultimately contribute to the long-term sustainability and efficiency of tomato cultivation practices worldwide. The project's implementation holds the promise of a more resilient and brighter future for tomato growers by providing a comprehensive solution to their challenges.

1.3 Objective

The primary objective of this project is to address and improve the challenges faced in tomato cultivation by introducing an innovative integrated system. The project aims to combat the threats posed by diseases and inefficient fertilizer application, which result in significant losses for farmers. By leveraging advanced computer vision and machine learning techniques, the project seeks to provide rapid and accurate suggestions for disease detection and precise fertilizer formulations. The key focus is on enhancing the precision of fertilizer recommendations based on individual plant health and soil conditions. Through the integration of data-driven algorithms and the use of a diverse dataset for experimental validation, the goal is to achieve a validation accuracy exceeding 95%. Furthermore, the project aims to revolutionize tomato cultivation practices by incorporating soil and environmental sensors to ensure tailored nutrient application. The overarching objective is to contribute to the enhancement of crop yield, reduction of environmental impact, and the promotion of agricultural sustainability, offering a comprehensive solution to the challenges faced by growers globally.

1.4 Problem Definition

The problem at hand revolves around the agricultural challenges faced in the cultivation of tomatoes, a crop of utmost significance for global food security and economic stability. Two critical issues exacerbate the viability of tomato production: the prevalence of diseases affecting the crop and the inefficiency in fertilizer application, leading to significant losses for farmers. The current reliance on labor-intensive manual inspection for disease detection results in inaccuracies, while conventional fertilizer recommendation systems lack precision, resulting in suboptimal nutrient utilization and potential environmental harm. These challenges underscore the need for an innovative solution that can address both disease detection and fertilizer application inefficiencies. The overarching problem is to develop a comprehensive, integrated system that leverages advanced computer vision and machine learning techniques to provide rapid and accurate disease detection, coupled with data-driven algorithms for precise fertilizer recommendations.

1.5 Aim

The aim of this agricultural project is to introduce an integrated system employing advanced computer vision and machine learning to enhance disease detection and fertilizer recommendations in tomato cultivation. The focus is on surpassing the limitations of labor-intensive manual inspection for disease detection and optimizing fertilizer application with data-driven algorithms. The overarching goal is to revolutionize current practices, minimizing losses due to diseases and inefficient fertilizer use. The project targets a validation accuracy exceeding 95% through experimental validation with a diverse dataset. Ultimately, implementation of the integrated system seeks to significantly improve crop yield, reduce environmental impact, and contribute to the overall sustainability and efficiency of global tomato cultivation practices.

Chapter 2

Literature Survey

2.1 Research Papers

2.1.1 Tomato disease detection model base on densenet and transfer learning

[1] The paper "Tomato disease detection model based on densenet and transfer learning" introduces a transfer learning based model for detecting tomato leaf diseases. This study proposes a model of DenseNet201 as a transfer learning-based model and CNN classifier. A comparison study between four deep learning models (VGG16, Inception V3, ResNet152V2 and DenseNet201) done in order to determine the best accuracy in using transfer learning in plant disease detection. The used images dataset contains 22930 photos of tomato leaves in 10 different classes, 9 disorders and one healthy class.Plant diseases are a foremost risk to the safety of food. They have the potential to significantly reduce agricultural products quality and quantity. In agriculture sectors, it is the most prominent challenge to recognize plant diseases. In computer vision, the Convolutional Neural Network (CNN) produces good results when solving image classification tasks. For plant disease diagnosis, many deep learning architectures have been applied.

Table 2.1.1.1: Performance Analysis of used methods

Method	**Accuracy**
DenseNet using transfer learning	99.0%

2.1.2 Tomato Leaf Disease Detection Using Deep Learning Techniques

[2] The paper titled "Tomato Leaf Disease Detection Using Deep Learning Techniques" proposes to identify the Tomato Plant Leaf disease using image processing techniques based on Image segmentation, clustering, and open-source algorithms, thus all contributing to a reliable, safe, and accurate system of leaf disease with the specialization to Tomato Plants.Early Detection of Plant Leaf Detection is a major necessity in a growing agricultural economy like India. Not only as an agricultural economy but also with a large amount of population to feed, it is necessary that leaf diseases in plants are detected at a very early stage and predictive mechanisms to be adopted to make them safe and avoid losses to the agri-based economy.

Table 2.1.2.1: Performance Analysis of used methods

Method	**Accuracy**
Deep Neural Networks (AlexNet)	95.75%
Tomato leaf disease detection using CNN (proposed method)	98.12%

2.1.3 Less is more: Lighter and faster deep neural architecture for tomato leaf disease classification

[3] In this paper author proposes a lightweight transfer learning-based approach for detecting diseases from tomato leaves. It utilizes an effective preprocessing method to enhance the leaf images with illumination correction for improved classification. Our system extracts features using a combined model consisting of a pretrained MobileNetV2 architecture and a classifier network for effective prediction. Traditional augmentation approaches are replaced by runtime augmentation to avoid data leakage and address the class imbalance issue. Evaluation on tomato leaf images from the PlantVillage dataset shows that the proposed architecture achieves 99.30% accuracy with a model size of 9.60MB and 4.87M floating-point operations, making it a suitable choice for low-end devices..

Table 2.1.3.1: Analysis of used methods

Method	Accuracy
MobileNet v2	99.30%

2.1.4 DCNet: DenseNet-77-based CornerNet model for the tomato plant leaf disease detection and classification

[4] In this paper author presented a robust approach to tackle the existing issues of tomato plant leaf disease detection and classification by using deep learning. We have proposed a novel approach namely the DenseNet-77-based CornerNet model for the localization and classification of the tomato plant leaf abnormalities. Specifically, we have used the DenseNet-77 as the backbone network of the CornerNet which assists to compute the more nominative set of image features from the suspected samples which are later categorized into ten classes by the one-stage detector of the CornerNet model. We have evaluated the proposed solution on a standard dataset named the PlantVillage which is challenging in nature as it contains the samples with immense brightness alterations, color variations, and leaf images with different dimensions and shapes. We have conducted several experiments to assure the effectiveness of our approach for the timely recognition of the tomato plant leaf diseases that can assist the agriculturalist to replace the manual. systems.

Table 2.1.4.1: Analysis of used methods

Method	Accuracy
Cornernet DenseNet	99.7%

2.1.5 Detection of tomato leaf diseases for agro-based industries using novel PCA DeepNet

[5] The paper explains the advancement of Deep Learning and Computer Vision in the field of agriculture has been found to be an effective tool in detecting harmful plant diseases. Classification and detection of healthy and diseased crops play a very crucial role in determining the rate and quality of production. Thus the present work highlights a well-proposed novel method of detecting Tomato leaf diseases using Deep Neural Networks to strengthen agro-based industries. The present novel framework is utilized with a combination of classical Machine

Learning model Principal Component Analysis (PCA) and a customized Deep Neural Network which has been named as PCA DeepNet. The hybridized framework also consists of Generative Adversarial Network (GAN) for obtaining a good mixture of datasets. The detection is carried out using the Faster Region-Based Convolutional Neural Network (F-RCNN).

Table 2.1.5.1: Analysis of used methods

Method	**Accuracy**
Pipeline Methods include CNN, F-CNN, PCA & GAN's	99.6%

2.1.6 A real-time application-based convolutional neural network approach for tomato leaf disease classification

[6] The author proposed a lightweight custom convolutional neural network (CNN) model and utilized transfer learning (TL)-based models VGG-16 and VGG-19 to classify tomato leaf diseases. In this study, eleven classes, one of which is healthy, are used to simulate various tomato leaf diseases. In addition, an ablation study has been performed in order to find the optimal parameters for the proposed model. Furthermore, evaluation metrics have been used to analyze and compare the performance of the proposed model with the TL-based model. The proposed model, by applying data augmentation techniques, has achieved the highest accuracy and recall of 95.00% among all the models. Finally, the best-performing model has been utilized in order to construct a Web-based and Android-based end-to-end (E2E) system for tomato cultivators to classify tomato leaf disease.

Table 2.1.6.1: Performance of listed methods

Method	**Accuracy**
CNN-VGG-19	95%

2.1.7 A generic intelligent tomato classification system for practical applications using DenseNet-201 with transfer learning

[7] A generic intelligent tomato classification system based on DenseNet-201 with transfer learning was proposed and the augmented training sets obtained by data augmentation methods were employed to train the model. The trained model achieved high classification accuracy on the images of different quality, even those containing high levels of noise. Also, the trained model could accurately and efficiently identify and classify a single tomato image with only 29 ms, indicating that the proposed model has great potential value in real-world applications. The feature visualization of the trained models shows their understanding of tomato images, i.e., the learned common and high-level features. The strongest activations of the trained models show that the correct or incorrect target recognition areas by a model during the classification process will affect its final classification accuracy. Based on this, the results obtained in this study could provide guidance and new ideas to improve the development of intelligent agriculture.

Table 2.1.7.1: Performance of listed methods

Method	**Accuracy**
DenseNet201	96.16%

2.1.8 A Deep Learning-Based Approach in Classification and Validation of Tomato Leaf Disease.

[8] The author of the paper addresses the process of identification and classification of a plant leaf disease. In this paper, the Tomato plant leaf images are taken from the PlantVillage Database consisting of one healthy and eight disease classes. The disease classes are selected based on the occurrence of the disease in India. The deep learning models of AlexNet, VGG16, GoogLeNet, MobileNetv2, and SqueezeNet are used in this work for the classification of Tomato plant leaf as healthy or diseased and further which disease class it belongs to. The models used here are all the pre-trained models, so transfer learning is used to fit the total number of classes that need to be classified by the network model.

Table 2.1.8.1: Performance of listed methods

Method	Accuracy
VGG-16	99.17%

2.1.9 Image-based automatic diagnostic system for tomato plants using deep learning

[9] The paper mainly addresses the proposed an AI-based approach to detect diseases in tomato plants. Our goal is to develop an end-to-end system to diagnose essential crop problems in real-time, ensuring high accuracy. This paper employs various deep learning models to recognize and predict different diseases caused by pathogens, pests, and nutritional deficiencies. Various Convolutional Neural Networks (CNNs) are trained on a large dataset of leaves and fruits images of tomato plants. We compared the performance of ShallowNet (a shallow network trained from scratch) and the state-of-theart deep learning network (models are fine-tuned via transfer learning).

Table 2.1.9.1: Performance of listed methods

Method	Accuracy
DenseNet	95.13%

2.1.10 Plant Diseases Detection Using Image Processing and Suggest Pesticides and Managements

[10] This paper addresses studies the methods for identifying plant ailments using photos of their leaves. This work also presented unique segmentation and feature extraction techniques for plant disease identification. For feature extraction, the CNN algorithm is utilized. This research paper may be a revolutionary approach to diagnosing plant illnesses by employing a deep convolutional neural network that has been trained and fine-tuned to suit a database of a plant's leaves gathered independently for distinct plant diseases. At the end of the study we achieved an accuracy of 98 percent in detecting the plant diseases and further on implemented mobile system which can suggest pesticide accordingly.

Chapter 3

System Design

3.1 Proposed System

The proposed system for tomato disease detection leverages deep learning models like MobileNet, DenseNet various models to identify diseases in tomato crops through image classification. However, the system currently focuses solely on disease detection, offering no recommendations for fertilizer application or other agricultural practices. It utilizes a dataset of tomato plant images for model training, with the primary output being the identification of diseases. Future enhancements could involve integrating with agricultural expert systems to provide comprehensive advice, while potential expansions might include features such as pest detection or yield prediction. The system's effectiveness relies on accurate models and seamless integration into agricultural workflows.

3.2 System Architecture

The architecture for tomato crop management integrates disease detection and fertilizer recommendation. MobileNet and DenseNet models are employed for accurate disease identification in tomato plants through image analysis. Once diseases are detected, the system incorporates a fertilizer recommendation component. This dual-functionality ensures a comprehensive approach to crop health, combining advanced deep learning models for precise disease diagnosis with intelligent recommendations for optimal fertilizer application, thereby assisting farmers in making informed decisions for tomato cultivation.

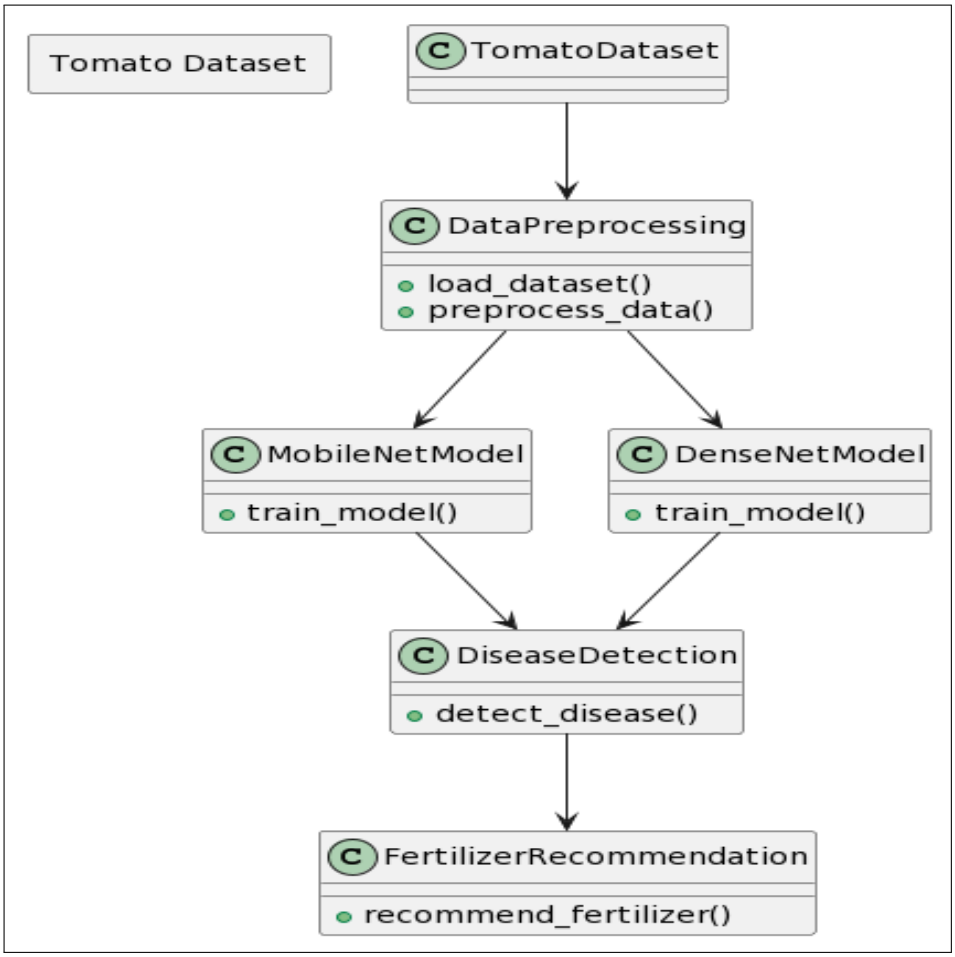


Figure 3.1: Architecture Of Tomato Crop

Chapter 4

Methodology

4.1 Data Collection

Data collection is a critical research phase involving the acquisition of raw information from various sources. Methods include surveys, interviews, observation, sensor data, web scraping, and experiments tailored to specific research needs.We developed the data set by gathering information from IBM and the Plant Village dataset.(see figure 4.1)

4.2 Data Splitting

Data splitting for tomato plant disease detection involves creating two subsets: the training set, instructing the model on patterns, and the testing set, evaluating its generalization. This strategic partitioning ensures unbiased accuracy assessment in real-world scenarios. The training set enables diverse learning, while the testing set robustly evaluates overall model performance. This streamlined approach is crucial for developing a reliable and effective tomato plant disease detection model with a focus on training and testing subsets.

4.3 Data Augmentation

Data augmentation is a pivotal technique in machine learning, particularly for image classification tasks like the Tomato Leaf Disease Detection project. Implemented using Keras' ImageDataGenerator, it involves applying transformations, such as rotation and flipping, to the original dataset. This process diversifies the training data, making the model more robust and better at generalizing to new,

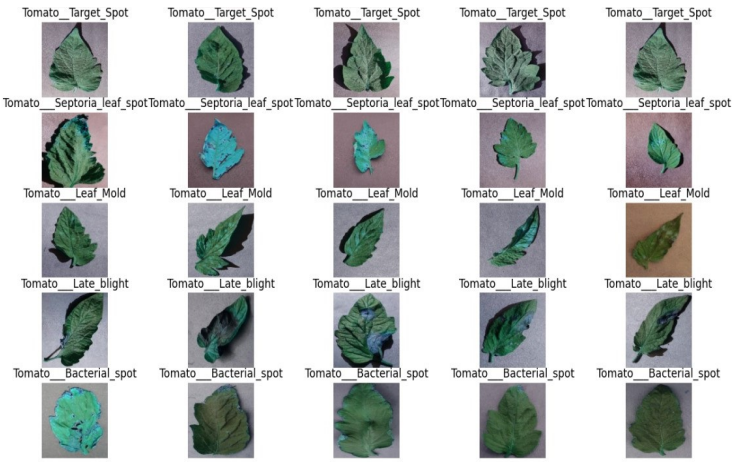


Figure 4.1: Collection of Tomato leaves Dataset

unseen instances. Data augmentation helps prevent overfitting and significantly contributes to the model's overall performance.

4.4 Model training

DenseNet and MobileNet models are trained using the designated training dataset to learn and recognize patterns in tomato plant images. The training process involves adjusting model parameters to minimize the difference between predicted and actual disease categories. Once trained, the models are tested using a separate testing dataset to assess their generalization and performance on new, unseen data. This training and testing methodology ensures the models' effectiveness in accurately detecting diseases in tomato plants and provides a reliable evaluation of their performance in real-world applications.

4.5 Model Evaluation

The model is evaluated using testing data to assess its performance on new and unseen examples with different performance metrics. In real time, users can provide a Google link containing an image of a tomato leaf, indicating whether it is infected or healthy. The model then utilizes the knowledge gained during training to predict the health status of the leaf. This real-time prediction based on the trained data enhances the model's practical utility for users seeking immediate insights into the health condition of tomato plants, showcasing the applicability of the developed model in real-world scenarios.

Chapter 5

Implemention

5.1 Importing python Libraries

The code imports a number of libraries for data preprocessing, feature selection, model evaluation, and visualization. Several neural network-related libraries are imported, including those for building, training, and evaluating deep learning models. Seaborn is employed for creating visually informative confusion matrix figures, contributing to a more intuitive understanding of model performance. Furthermore, the code suppresses warnings to enhance readability. Overall, the script is well-equipped with a diverse set of tools and libraries for a comprehensive machine learning workflow.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import cv2
import os
from tqdm import tqdm
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from keras.models import Model,Sequential, Input, load_model
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization, AveragePooling2D,
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from keras.applications import DenseNet121
from keras.applications import DenseNet201
from keras.applications import MobileNetV2
from keras.applications import ResNet50
```

Figure 5.1: Libraries imported

5.2 Data Splitting

The Plant Village dataset that consists of 9 classes of diseases of tomato leaves and 1 healthy class is pre-processed and then, will be split into training data, validation data and testing data in 70, 20, 10 respectively, the code snippet is provided below.

```
BATCH_SIZE = 64

# train 70 , val 20, test 10

# Split the train,test validation sets
X_train, X_temp, Y_train, Y_temp = train_test_split(X_Train, Y_train, test_size=0.3, random_state=SEED)
X_test, X_val, Y_test, Y_val = train_test_split(X_temp, Y_temp, test_size=1/3, random_state = SEED)
```

```
fig, ax = plt.subplots(1, 3, figsize=(15, 15))
for i in range(3):
    ax[i].set_axis_off()
    ax[i].imshow(X_train[i])
    ax[i].set_title(disease_types[np.argmax(Y_train[i])])
```

Figure 5.2: Code Snippet of Splitting and Vizualization of train data

5.3 Data Augmentation

Data augmentation is a technique employed during the training of machine learning models, particularly in computer vision tasks such as image classification. It involves applying various transformations to the existing training data to create augmented versions, thereby expanding the diversity of the dataset. Common augmentations include rotation, flipping, zooming, and changes in brightness or contrast.

```
# Generates batches of image data with data augmentation
datagen = ImageDataGenerator(rotation_range=360, # Degree range for random rotations
                        width_shift_range=0.2, # Range for random horizontal shifts
                        height_shift_range=0.2, # Range for random vertical shifts
                        zoom_range=0.2, # Range for random zoom
                        horizontal_flip=True, # Randomly flip inputs horizontally
                        vertical_flip=True) # Randomly flip inputs vertically

datagen.fit(X_train)
```

Figure 5.3: Code Snippet of Data Augmentation

In the context of the provided code snippet, data augmentation is likely implemented using the ImageDataGenerator from the Keras library. This generator applies a variety of transformations to the input images in real-time during the model training process. Data augmentation is a crucial step to prevent overfitting, where a model may memorize the training data rather than learning its underlying patterns.

5.4 Training with Different Models

The training involved four different models: ResNet50, MobileNetV2, DenseNet121, and DenseNet201. Each model was subjected to the same training procedure, including data preprocessing, data augmentation, feature extraction, and evaluation metrics. The aim was to compare their performance in an image classification task.

```
def build_densenet():
    densenet = DenseNet201(weights='imagenet', include_top=False)
    input = Input(shape=(SIZE, SIZE, N_ch))
    x = Conv2D(3, (3, 3), padding='same')(input)
    x = densenet(x)
    x = GlobalAveragePooling2D()(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)
    x = Dense(512, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)

    # multi output
    output = Dense(15,activation = 'softmax', name='root')(x)
    # model
    model = Model(input,output)
    optimizer = Adam(lr=0.002, beta_1=0.9, beta_2=0.999, epsilon=0.1, decay=0.0)
    model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    model.summary()
    return model
```

```
model = build_densenet()
annealer = ReduceLROnPlateau(monitor='val_accuracy', factor=0.5, patience=5, verbose=1, min_lr=1e-3)
checkpoint = ModelCheckpoint('model.h5', verbose=1, save_best_only=True)

# Fits the model on batches with real-time data augmentation
hist = model.fit_generator(datagen.flow(X_train, Y_train, batch_size=BATCH_SIZE),
               steps_per_epoch=X_train.shape[0] // BATCH_SIZE,
               epochs=EPOCHS,
               verbose=2,
               callbacks=[annealer, checkpoint],
               validation_data=(X_val, Y_val))
```

Figure 5.4: Code Snippet of Model building and Processing

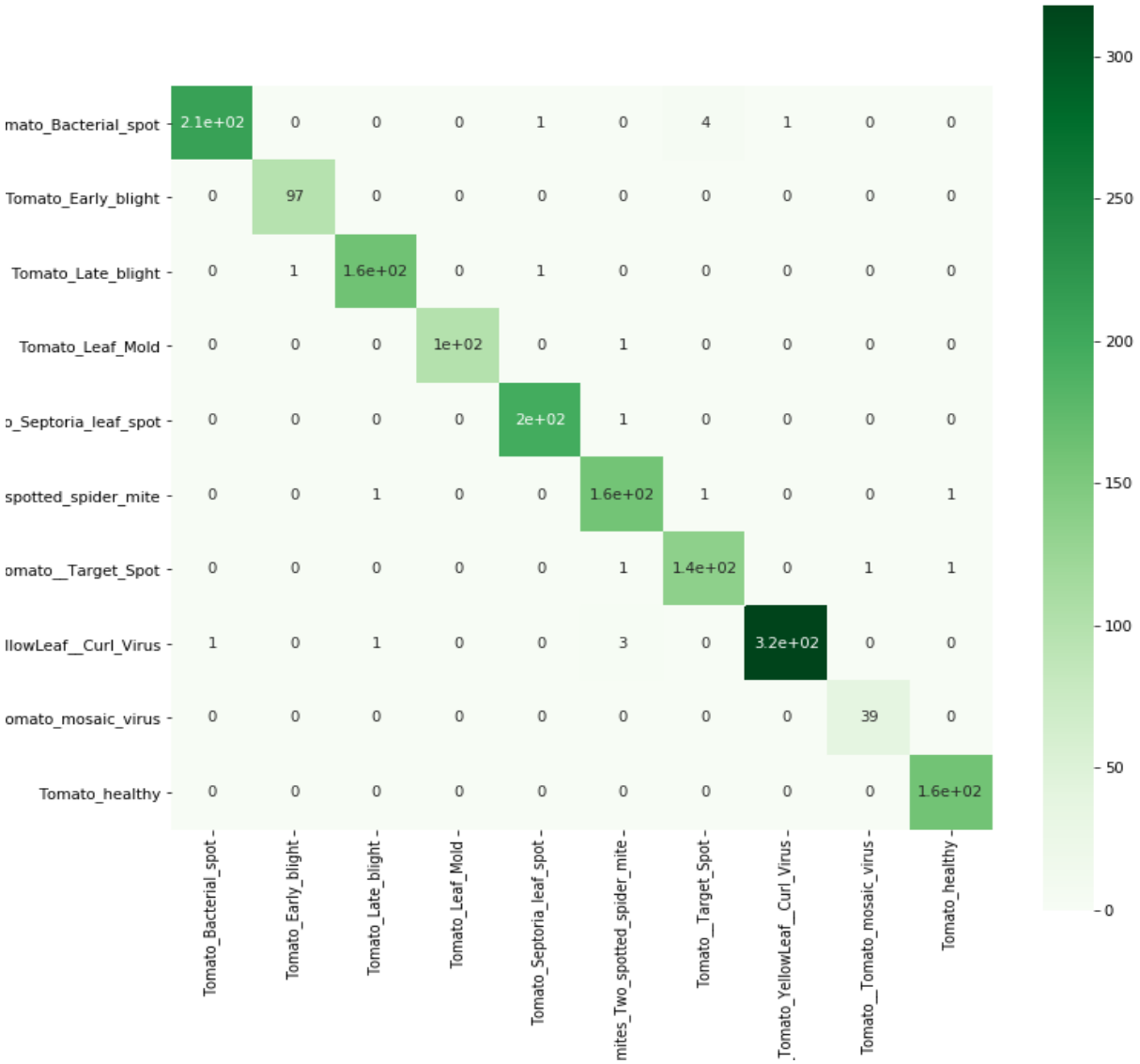


Figure 5.5: Confusion Matrix of ResNet50

ResNet50, MobileNetV2, DenseNet121, and DenseNet201 represent distinct convolutional neural network architectures with varying complexities and capabilities. ResNet50 utilizes residual connections to address vanishing gradient problems.

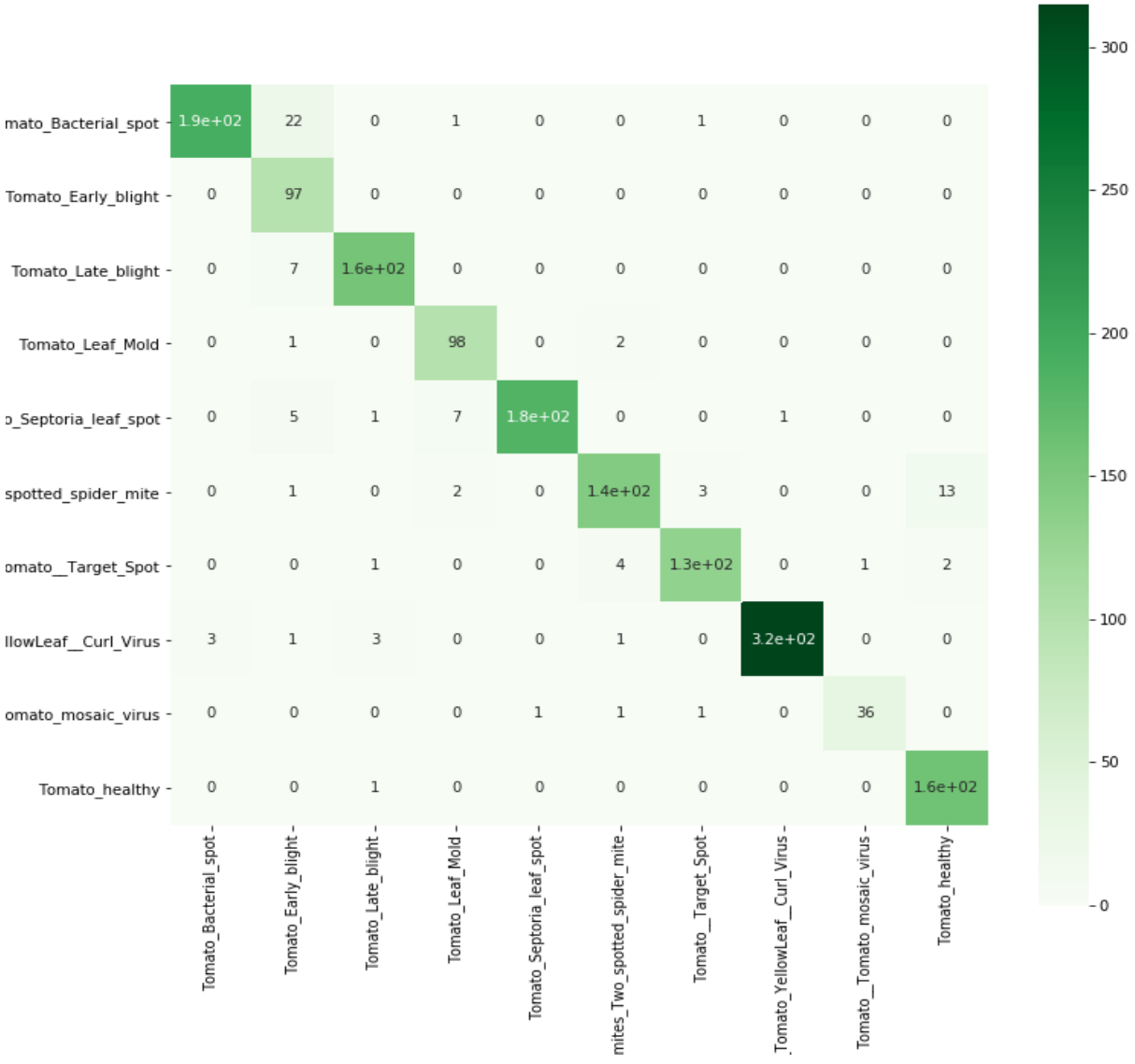


Figure 5.6: Confusion Matrix of MobileNetV2

MobileNetV2 emphasizes lightweight, efficient operations suitable for mobile devices. DenseNet121 employs densely connected blocks for enhanced feature extraction, while DenseNet201 extends this architecture for even deeper learning.

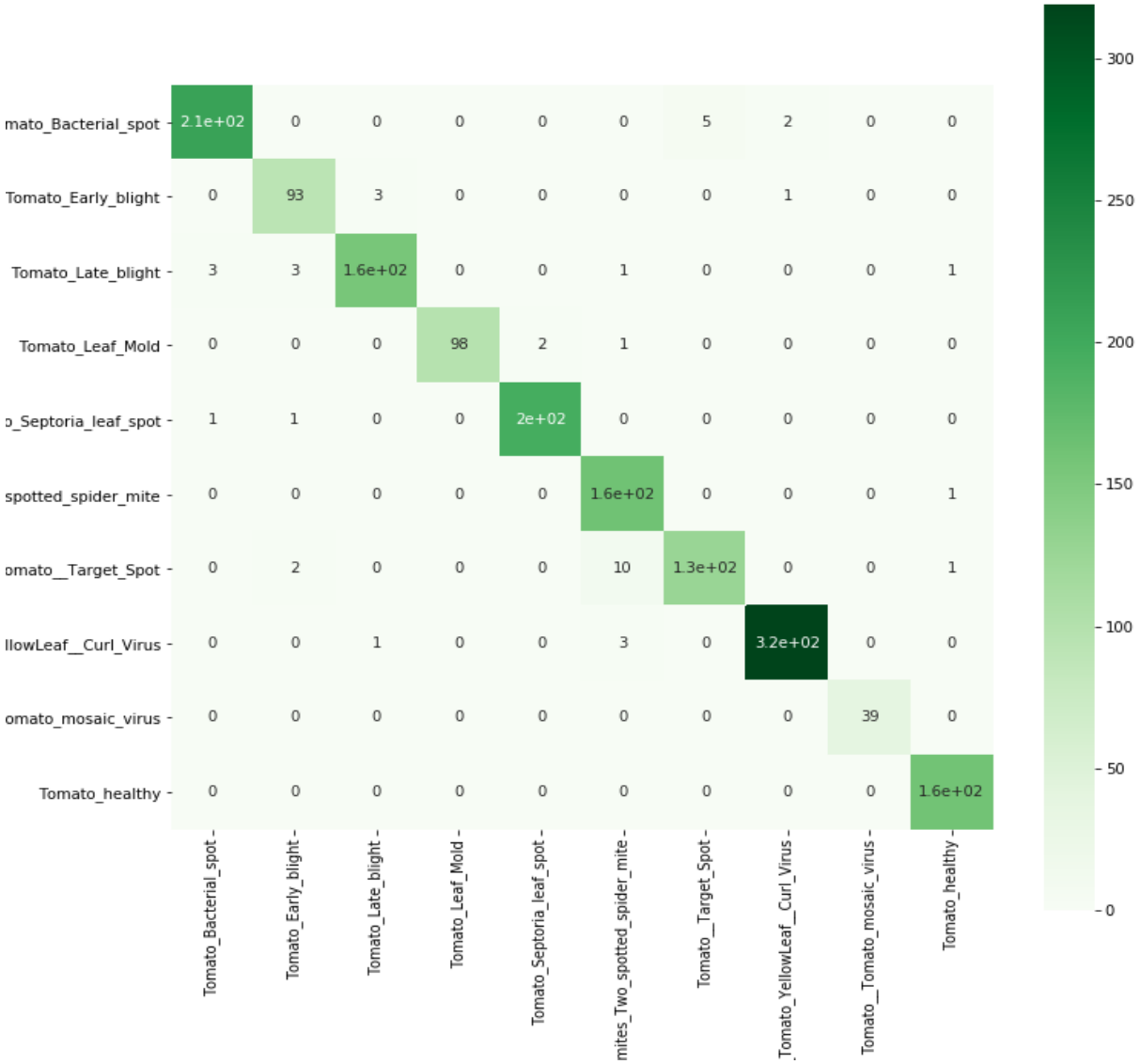


Figure 5.7: Confusion Matrix of DenseNet121

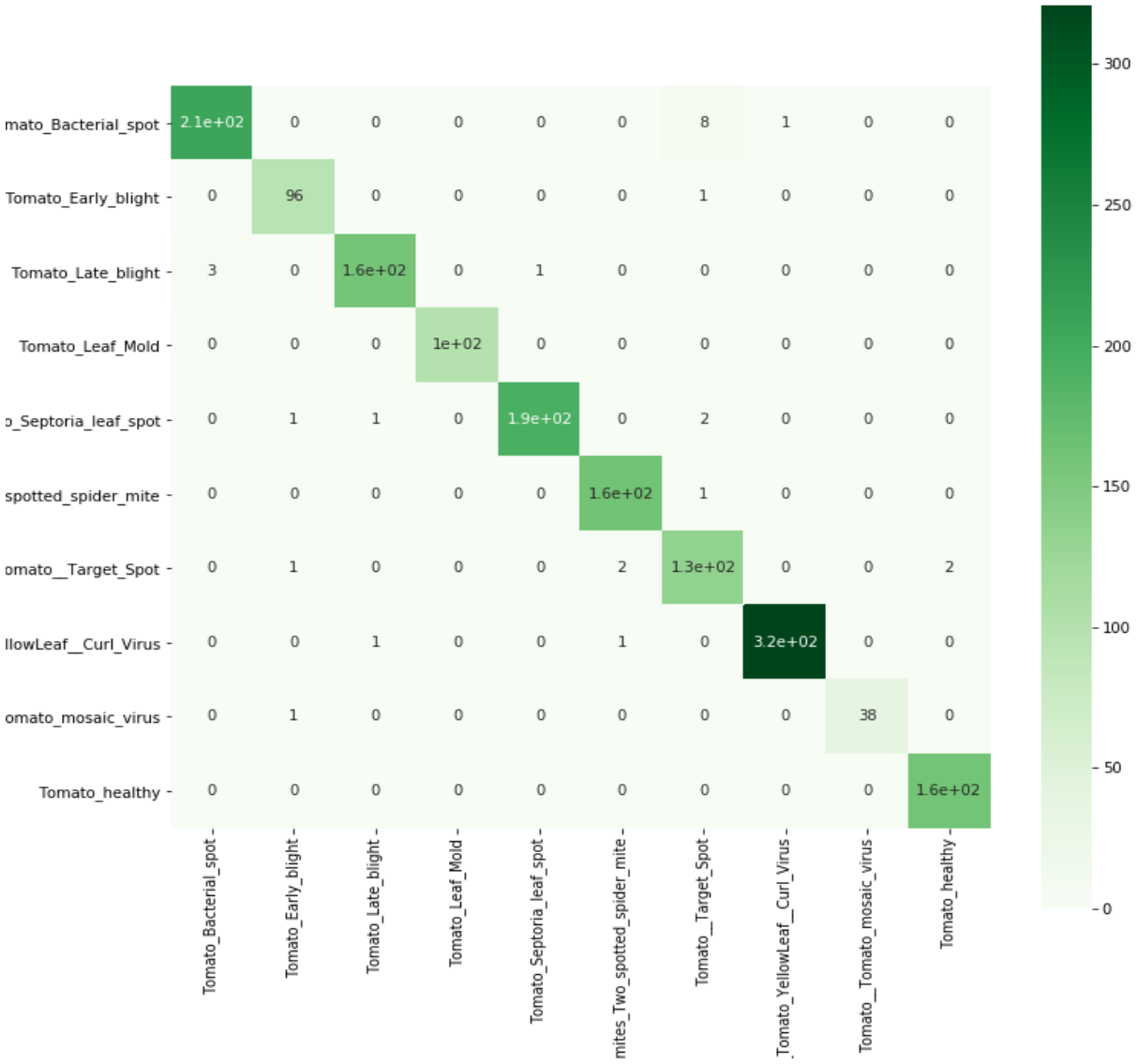


Figure 5.8: Confusion Matrix of DenseNet201

Chapter 6

Results and Discussions

6.1 Experimental Results

The data is trained with different models as shown below and obtained the following accuracies with test data ie, unseen and untrained data.

Model	**Testing Accuracy**
ResNet50	97.43%
MobileNetV2	94.47%
DenseNet121	97.24%
DenseNet201	97.50%

Table 6.1.1: Performance of different models

Below results are shown for DenseNet201 model with 97.50%. In the next module , DenseNet is used as feature Extractor and is added to CNN Classifier, that may fetch more accuarcy (expected 99.00% accuracy) than these models. If the accuarcies are not reached, data augmentation algorithms like cycle GAN (generative adversarial network) are to be used for better quality of images that improves accuracy.

Classification Report for Training Dataset:

	precision	recall	f1-score	support
Tomato_Bacterial_spot	0.99	0.96	0.97	217
Tomato_Early_blight	0.97	0.99	0.98	97
Tomato_Late_blight	0.99	0.98	0.98	164
Tomato_Leaf_Mold	1.00	1.00	1.00	101
Tomato_Septoria_leaf_spot	0.99	0.98	0.99	198
Tomato_Spider_mites_Two_spotted_spider_mite	0.98	0.99	0.99	163
Tomato__Target_Spot	0.92	0.96	0.94	139
Tomato__Tomato_YellowLeaf__Curl_Virus	1.00	0.99	1.00	323
Tomato__Tomato_mosaic_virus	1.00	0.97	0.99	39
Tomato_healthy	0.99	1.00	0.99	161
accuracy			0.98	1602
macro avg	0.98	0.98	0.98	1602
weighted avg	0.98	0.98	0.98	1602

Figure 6.1: Classification Report for Training data

Classification Report for Testing Dataset:

	precision	recall	f1-score	support
Tomato_Bacterial_spot	0.98	0.99	0.98	458
Tomato_Early_blight	0.96	0.91	0.93	206
Tomato_Late_blight	0.97	0.98	0.97	369
Tomato_Leaf_Mold	0.98	0.99	0.99	188
Tomato_Septoria_leaf_spot	0.97	1.00	0.98	326
Tomato_Spider_mites_Two_spotted_spider_mite	0.98	0.95	0.96	336
Tomato__Target_Spot	0.92	0.96	0.94	273
Tomato__Tomato_YellowLeaf__Curl_Virus	0.99	0.99	0.99	640
Tomato__Tomato_mosaic_virus	1.00	0.96	0.98	78
Tomato_healthy	1.00	0.96	0.98	328
accuracy			0.98	3202
macro avg	0.97	0.97	0.97	3202
weighted avg	0.98	0.98	0.97	3202

Figure 6.2: Classification Report for Testing data

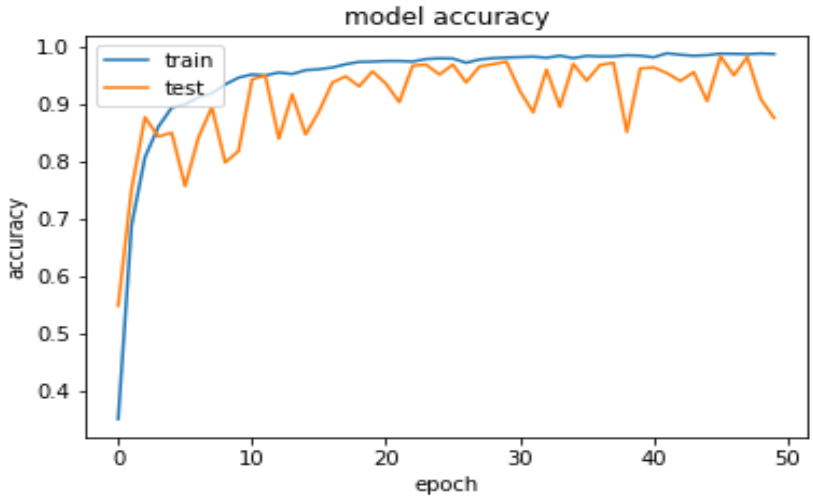


Figure 6.3: Accuracy vs Epoch

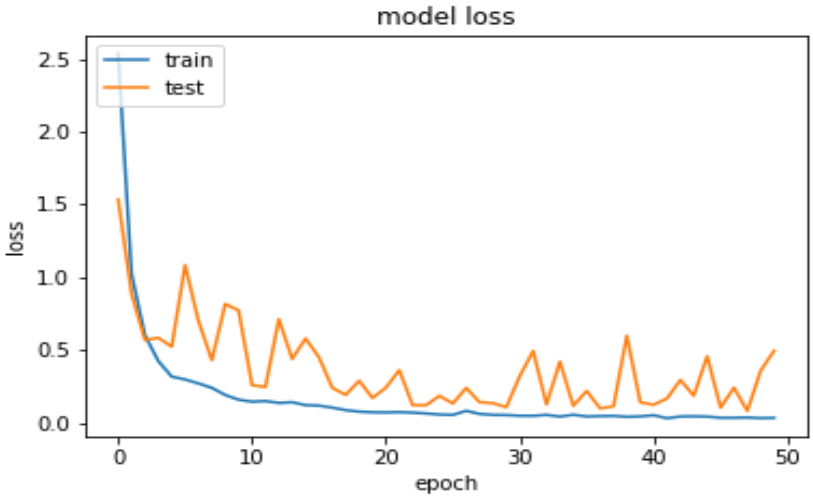


Figure 6.4: Loss vs Epoch

The DenseNet201 model achieved an impressive accuracy of 97.50%. During training, its dense connectivity pattern facilitated the extraction of complex features from images. The Adam optimizer with a defined learning rate schedule optimized parameters, and dropout layers mitigated overfitting. The model's proficiency in learning and adapting to the training data contributed to its high accuracy in image classification.

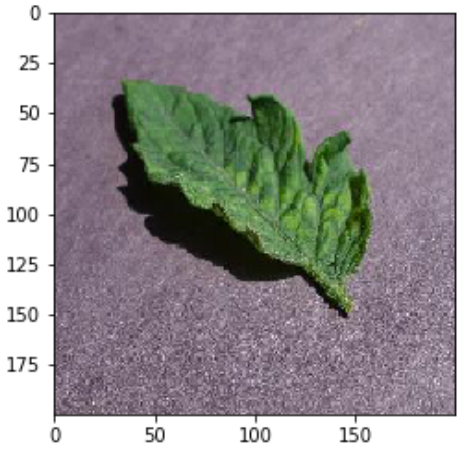
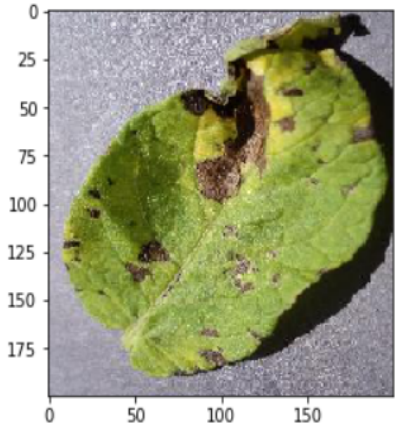


Figure 6.5: Prediction plot of tomato leaf

```
[6.7680077e-08 7.5559181e-01 2.1265849e-01 6.0369046e-03 2.3991413e-02
 1.0827882e-06 3.7004188e-06 2.6436295e-08 5.7729707e-05 8.5948058e-04
 2.7422397e-04 1.0472394e-05 1.0010289e-04 6.5775537e-05 3.4865210e-04]
```




```
Prediction: Tomato_Early_blight
```

Figure 6.6: Prediction of Tomato leaf disease

Chapter 7

Conclusions and future works

7.1 Conclusion

In conclusion, the Tomato Leaf Disease Detection project employs a sophisticated machine learning pipeline, integrating decision tree classifiers and deep learning models like DenseNet201. The code, utilizing libraries such as numpy, pandas, and Keras, showcases a meticulous approach to data preprocessing, feature selection, and model evaluation. Notably, the DenseNet201 model achieves an impressive accuracy of 97.50%. Data augmentation techniques are strategically applied to the training set, enhancing model generalization. The comparison with other models, including ResNet50 and DenseNet121, provides valuable insights into their performance trade-offs. This project exemplifies a comprehensive and effective application of machine learning for plant disease detection, with implications for advancing agricultural practices through technology.

7.2 Future Scope

Additionally, the project aims to improve the accuracy by taking DenseNet201 as feature extractor with CNN classifier. This is added to fertilizer recommendation system for recommend fertilizer after disease prediction. The project also aims to integrate a user-friendly Graphical User Interface (GUI) in the next module to enhance accessibility and usability for farmers. Through the GUI, farmers can input relevant data, receive real-time disease predictions, and obtain personalized fertilizer recommendations, fostering a user-centric approach to agricultural management. The future trajectory of the project involves broadening its applicability to multiple crops and introducing a user-friendly GUI, thereby advancing its potential impact on agricultural practices and contributing to the overarching goal of sustainable and efficient crop management.

Bibliography

[1] M. Bakr, S. Abdel-Gaber, M. Nasr, and M. Hazman, "Tomato disease detection model based on densenet and transfer learning," *Applied Computer Science*, vol. 18, no. 2, 2022.

[2] S. Ashok, G. Kishore, V. Rajesh, S. Suchitra, S. G. Sophia, and B. Pavithra, "Tomato leaf disease detection using deep learning techniques," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 979–983, IEEE, 2020.

[3] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony, and M. H. Kabir, "Less is more: Lighter and faster deep neural architecture for tomato leaf disease classification," *IEEE Access*, vol. 10, pp. 68868–68884, 2022.

[4] S. Albahli and M. Nawaz, "Dcnet: Densenet-77-based cornernet model for the tomato plant leaf disease detection and classification," *Frontiers in Plant Science*, vol. 13, p. 957961, 2022.

[5] K. Roy, S. S. Chaudhuri, J. Frnda, S. Bandopadhyay, I. J. Ray, S. Banerjee, and J. Nedoma, "Detection of tomato leaf diseases for agro-based industries using novel pca deepnet," *IEEE Access*, vol. 11, pp. 14983–15001, 2023.

[6] S. G. Paul, A. A. Biswas, A. Saha, M. S. Zulfiker, N. A. Ritu, I. Zahan, M. Rahman, and M. A. Islam, "A real-time application-based convolutional neural network approach for tomato leaf disease classification," *Array*, vol. 19, p. 100313, 2023.

[7] T. Lu, B. Han, L. Chen, F. Yu, and C. Xue, "A generic intelligent tomato classification system for practical applications using densenet-201 with transfer learning," *Scientific Reports*, vol. 11, no. 1, p. 15824, 2021.

[8] S. A. Wagle *et al.*, "A deep learning-based approach in classification and validation of tomato leaf disease.," *Traitement du signal*, vol. 38, no. 3, 2021.

[9] S. Khatoon, M. M. Hasan, A. Asif, M. Alshmari, and Y. Yap, "Image-based automatic diagnostic system for tomato plants using deep learning," *Comput. Mater. Contin*, vol. 67, no. 1, pp. 595–612, 2021.

[10] L. Sritharan, M. Anjanan, and A. Gamage, "Plant diseases detection using image processing and suggest pesticides and managements," in *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, pp. 1–8, IEEE, 2022.