A REAL-TIME MONITORING SYSTEM FOR ACCURATE PLANT LEAVES DISEASE DETECTION USING DEEP LEARNING

By
Kazi Naimur Rahman
21MME019F

A thesis submitted in partial fulfillment of the requirements for the degree of
MASTER of SCIENCE in Mechanical Engineering

Department of Mechanical Engineering
CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

2024

APPROVAL

The thesis titled "**A Real- Time Monitoring System for Accurate Plant Leaves Disease Detection Using Deep Learning**" submitted by Kazi Naimur Rahman (ID:21MME019F, Session: 2021-2022) has been accepted as satisfactory by the Dissertation Committee for the partial fulfillment of the requirement for the degree of Master of Science (M.Sc.) in Mechanical Engineering, dated on 5 December, 2024.

Dissertation Committee

Dr. Sajal Chandra Banik Professor, Department of Mechanical Engineering, CUET.	Chairman (Supervisor)
Dr. Mohammad Mizanur Rahman Head, Department of Mechanical Engineering, CUET.	Member (Ex Officio)
Dr. Jamal Uddin Ahamed, Professor, Department of Mechanical Engineering, CUET.	Member
Dr. Kazi Afzalur Rahman Professor, Department of Mechanical Engineering, CUET.	Member
Dr. Nayeem Md. Lutful Huq Professor, Department of Mechanical Engineering, DUET.	Member External

Declaration

I hereby declare that the work in this thesis has not been previously submitted to meet the requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is cited. Furthermore, the Thesis compiles with PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Kazi Naimur Rahman
21MME019F
Department of Mechanical Engineering
Chittagong University of Engineering & Technology (CUET)

List of Publications

Kazi Naimur Rahman, Sajal Chandra Banik and Raihan Islam, "DeepLeafGuard: A CNN-Enabled Monitoring System for Accurate Plant Leaves Disease Detection" 6th International Conference on Mechanical, Industrial and Materials Engineering (ICMIME – 2024), Paper ID: 487, 2024, accepted in 12 November, 2024.

Kazi Naimur Rahman, Sajal Chandra Banik, Raihan Islam, and Arafath Al Fahim, "A Real-Time Monitoring System for Accurate Plant Leaves Disease Detection Using Deep Learning", Elsevier Published Scopus-Indexed Crop Design Journal (ISSN: 2772-8994), Article Reference: CROPD_100092, 2024, accepted in 18 October, 2024.

Approval by the Supervisor

This is to certify that Kazi Naimur Rahman (ID:21MME019F) has carried out this work under my supervision and that he has fulfilled the relevant Academic Ordinance of the Chittagong University of Engineering and Technology so that he is qualified to submit the following thesis in application for the degree of MASTER of SCIENCE in Mechanical Engineering. Furthermore, the Thesis compiles with PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Dr. Sajal Chandra Banik
Professor
Department of Mechanical Engineering
Chittagong University of Engineering & Technology

Abstract

Accurate and timely detection of plant diseases is crucial for sustainable agriculture and food security. This research presents a real-time monitoring system utilizing deep learning techniques to detect diseases in plant leaves with high accuracy. We combined several plant datasets, including the PlantVillage Dataset, resulting in a comprehensive dataset of 30,945 images across eight plant types (potato, tomato, pepper bell, apple, corn, grape, peach, and rice) and 35 disease classes. Initially, a custom Convolutional Neural Network (CNN) model was developed, achieving a leaf classification accuracy of 95.62%. Subsequently, the dataset was partitioned for individual plant disease detection, applying nine different CNN models (custom CNN, VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, Xception, and two hybrid models) to each plant type. The highest accuracy rates for disease detection were: 100% for potato (custom CNN), 98% for tomato (InceptionV3, custom CNN, VGG16), 100% for pepper bell (MobileNet, custom CNN), 100% for apple (MobileNet, Xception), 98% for corn (custom CNN), 99% for grape (custom CNN, VGG19, DenseNet121), 100% for peach (VGG16, custom CNN), and 98% for rice (DenseNet121). A web and mobile application were developed based on the best-performing models, allowing users to insert or capture images of plant leaves, detect diseases, and receive treatment suggestions with high confidence levels. The results demonstrate the effectiveness of deep learning models in accurately identifying plant diseases, offering a valuable tool for enhancing disease management and crop yields.

বিমূর্ত

গুণগত ও মানসম্মত কৃষি ও খাদ্য নিরাপত্তার জন্য উদ্ভিদের রোগ সঠিকভাবে ও সময়মত শনাক্তকরণ করা অত্যন্ত গুরুত্বপূর্ণ। এই গবেষণাটি একটি রিয়েল-টাইম মনিটরিং সিস্টেম উপস্থাপন করে যা ডিপ লার্নিং প্রক্রিয়ার মাধ্যমে তৈরি করা হয়েছে এবং এটি অত্যন্ত নির্ভুলতার সাথে গাছের পাতায় রোগ শনাক্ত করার জন্য উপযুক্ত। এই কাজটির জন্য প্ল্যান্টভিলেজ ডেটাসেট সহ বেশ কয়েকটি উদ্ভিদ ডেটাসেট একত্রিত করা হয়েছে। এই ডেটাসেটটিতে আট ধরনের উদ্ভিদের সর্বমোট ৩০,৯৪৫ টি পাতার ছবি রয়েছে এবং ডেটাসেটটি ৩৫টি রোগের শ্রেণীতে বিভক্ত। আট ধরণের উদ্ভিদগুলো হলো আলু, টমেটো, পেপারবেল, আপেল, ভুট্টা, আঙ্গুর, পীচ এবং ধান। প্রাথমিকভাবে, একটি কাস্টম কনভোলিউশনাল নিউরাল নেটওয়ার্ক (সিএনএন) মডেল তৈরি করা হয়েছে যেটি ৯৫.৬২% নির্ভুলতার সাথে পাতার শ্রেণিবিন্যাস করতে সক্ষম। পরবর্তীকালে, পৃথক পৃথক প্রতিটি উদ্ভিদের রোগ শনাক্তকরণের জন্য ডেটাসেটটি বিভক্ত করা হয়েছে এবং উক্ত ডেটাসেটে নয়টি ভিন্ন সিএনএন মডেল (কাস্টম সিএনএন, ভিজিজি ১৬, ভিজিজি ১৯, ইন্সেপশন ভি৩, মোবাইল নেট, ডেনসনেট ১২১, এক্সেপশন, এবং দুটি হাইব্রিড মডেল) প্রয়োগ করা হয়েছে । রোগ শনাক্তকরণের সর্বোচ্চ নির্ভুলতার হার ছিল: আলুর জন্য ১০০% (কাস্টম সিএনএন), টমেটোর জন্য ৯৮% (ইন্সেপশন ভি৩, কাস্টম সিএনএন, ভিজিজি ১৬), পেপার বেলের জন্য ১০০% (মোবাইল নেট, কাস্টম সিএনএন), আপেলের জন্য ১০০% (মোবাইল নেট, এক্সেপশন), ভুট্টার জন্য ৯৮% (কাস্টম সিএনএন), আঙ্গুরের জন্য ৯৯% (কাস্টম সিএনএন, ভিজিজি ১৯, ডেনসনেট ১২১), পীচের জন্য ১০০% (কাস্টম সিএনএন, ভিজিজি ১৬), এবং চালের জন্য ৯৮% (ডেনসনেট ১২১)। একটি ওয়েব এবং মোবাইল অ্যাপ্লিকেশন তৈরি করা হয়েছে সর্বোত্তম-পারফর্মিং মডেলের উপর ভিত্তি করে, যা ব্যবহারকারীদের উদ্ভিদের পাতার ছবি সন্নিবেশ বা ক্যাপচার করতে, রোগ সনাক্ত করতে এবং আত্মবিশ্বাসের সাথে চিকিৎসার পরামর্শ গ্রহণ করতে সাহায্য করে। ফলাফলগুলি উদ্ভিদের রোগ নির্ভুলভাবে শনাক্তকরণে ডিপ লার্নিং মডেলের কার্যকারিতা প্রদর্শন করে, রোগ ব্যবস্থাপনা উন্নত এবং ফসলের ফলন বৃদ্ধির জন্য একটি মূল্যবান হাতিয়ার হিসেবে কাজ করে।

Acknowledgment

Above all, I would like to express my gratitude to my supervisor, **Prof. Dr. Sajal Chandra Banik,** Professor, Dept. of ME, CUET for accepting me into his research group and also express my heartfelt thanks to him for his guidance, encouragement, and continuous support during my graduate studies. His enthusiasm for teaching and research offered me challenging opportunities to expand my scientific knowledge and my growing interest in the world of Industrial Engineering specifically in quality control and management.

TABLE OF CONTENTS

APPROVAL ii

Declaration iii

List of Publications iv

Approval by the Supervisor v

Abstract vi

বিমূর্ত vii

Acknowledgement viii

LIST OF FIGURES xii

LIST OF TABLES xiv

LIST OF SYMBOLS AND ABBREVIATIONS xv

Chapter 01 1

Introduction 1

1.1 Background and motivation 1

1.2 Thesis objectives 2

1.3 Context of diseases on various plant leaves 3

1.4 Context of detecting diseases on plant leaves using deep learning technique 5

1.5 Conclusion 6

Chapter 02 7

Literature Review 7

2.1 Introduction 7

2.2 Previous study about detection of diseases on various plant leaves using various DL algorithms 7

2.3 Conclusion 18

Chapter 03 19

Methodology 19

3.1 Introduction 19

3.2 Workflow of plant leaves disease detection process 19

3.2.1 Data collection 21

3.2.2 Data annotation 22
3.2.3 Data cleaning 24
3.2.4 Data pre-processing 25
3.2.5 Data augmentation 25
3.2.6 Dataset splitting and data visualization 26
3.2.7 Model building 29
3.2.7.1 Custom CNN 30
3.2.7.2 VGG16 30
3.2.7.3 VGG19 30
3.2.7.4 InceptionV3 31
3.2.7.5 MobileNet 31
3.2.7.6 DenseNet121 32
3.2.7.7 Xception 32
3.2.7.8 Hybrid models 32
3.2.8 Architectural comparison of various CNN models 34
3.2.9 Tools and system configuration 35
3.2.10 Converting model into TensorFlow serving 36
3.2.11 TF-Lite Model Conversion Process 37
3.2.12 Development Phase 37
3.3 Conclusion 39

Chapter 04 40

Results and Discussion 40
4.1 Introduction 40
4.2 Results of leaf classification process based on epochs 40
4.3 Performance Evaluation on Individual Plant's Disease Detection 41
4.3.1 Potato Plant's Performance Evaluation 41
4.3.2 Tomato Plant's Performance Evaluation 43
4.3.3 Pepper-bell Plant's Performance Evaluation 45

4.3.4 Apple Plant’s Performance Evaluation 47
4.3.5 Corn Plant’s Performance Evaluation 49
4.3.6 Grape Plant’s Performance Evaluation 51
4.3.7 Peach Plant’s Performance Evaluation 56
4.3.8 Rice Plant’s Performance Evaluation 55
4.4 Solution of overfitting issue 57
4.5 Backend API and database setup for user records 58
4.6 Web application development 59
4.7 Mobile application development 60
4.8 Optimal Hyperparameters Selection for CNN architecture 63
4.9 Conclusion 64

Chapter 05 65
Conclusion 65
5.1 Introduction 65
5.2 Summary of research contribution 65
5.3 Reason for Proposing Multiple Models in Our Research 66
5.4 Future research directions 67
5.5 Concluding remarks 68
Reference 69

LIST OF FIGURES

Figure 3.1	Workflow of plant leaves disease Detection using Deep Learning	20
Figure 3.2	Annotated dataset of various plant leaves	23
Figure 3.3	Data Visualization of Leaf Classification (Code Generated)	27
Figure 3.4	Data Visualization of Individual Plant's Leaf Disease (Code Generated)	28
Figure 3.5	Schematic Diagram of basic Convolutional Neural Network (CNN)	29
Figure 3.6	Schematic Block Diagram of Hybrid Model 1	33
Figure 3.7	Schematic Block Diagram of Hybrid Model 2	33
Figure 3.8	Flow Diagram of Mobile and Web Application Development	38
Figure 4.1	Best model's confusion matrix of potato leaves disease (Custom CNN)	42
Figure 4.2	Best model's training & validation accuracy vs epoch, loss vs epochs of potato plant's disease (Custom CNN)	43
Figure 4.3	Best model's confusion matrix of tomato leaves disease (Xception)	44
Figure 4.4	Best model's training & validation accuracy vs epoch, loss vs epochs of tomato plant's disease (Xception)	45
Figure 4.5	Best model's confusion matrix of pepper-bell leaves (Custom CNN)	46
Figure 4.6	Best model's training & validation accuracy vs epoch, loss vs epochs of pepper-bell plant's disease (Custom CNN)	47
Figure 4.7	Best model's confusion matrix of apple leaves disease (Xception)	48
Figure 4.8	Best model's training & validation accuracy vs epoch, loss vs epochs of apple plant's disease (Xception)	49

Figure 4.9	Best model's confusion matrix of corn leaves disease (Custom CNN)	50
Figure 4.10	Best model's training & validation accuracy vs epoch, loss vs epochs of corn plant's disease (Custom CNN)	51
Figure 4.11	Best model's confusion matrix of grape leaves disease (Custom CNN)	52
Figure 4.12	Best model's training & validation accuracy vs epoch, loss vs epochs of grape plant's disease (Custom CNN)	53
Figure 4.13	Best model's confusion matrix of peach leaves disease (VGG16)	54
Figure 4.14	Best model's training & validation accuracy vs epoch, loss vs epochs of peach plant's disease (VGG16)	55
Figure 4.15	Best model's confusion matrix of rice leaves disease (DenseNet121)	56
Figure 4.16	Best model's training & validation accuracy vs epoch, loss vs epochs of peach plant's disease (DenseNet121)	57
Figure 4.17	MongoDB database records of user registration	58
Figure 4.18	Homepage of web application using React framework	59
Figure 4.19	GUI of individual plant's image upload page in web application	60
Figure 4.20	Registration and login page of mobile application	61
Figure 4.21	GUI of homepage in mobile application	62
Figure 4.22	GUI of classification of diseases in mobile application	63

LIST OF TABLES

Table 3.1	Dataset details with class names	21
Table 3.2	Architectural Comparison of several CNN models	34
Table 3.3	System Configuration and Necessary Software	35
Table 4.1	Comparison of leaf classification model based on number of epochs	40
Table 4.2	Performance Evaluation for Potato Plants on Various CNN Models	42
Table 4.3	Performance Evaluation for Tomato Plants on Various CNN Models	44
Table 4.4	Performance Evaluation for Pepper-bell Plants on Various CNN Models	46
Table 4.5	Performance Evaluation for Apple Plants on Various CNN Models	48
Table 4.6	Performance Evaluation for Corn Plants on Various CNN Models	50
Table 4.7	Performance Evaluation for Grape Plants on Various CNN Models	52
Table 4.8	Performance Evaluation for Peach Plants on Various CNN Models	54
Table 4.9	Performance Evaluation for Rice Plants on Various CNN Models	56
Table 4.10	Hyperparameters selected for the proposed CNN architecture	64

NOMENCLATURE

DL	Deep Learning
CNN	Convolutional Neural Network
FCM	Fuzzy C-Means
KM	K-Means
Faster R-CNN	Faster Region Convolutional Neural Network
GLCM	Grey Level Co-occurrence Matrix
SVM	Support Vector Machine
LVQ	Learning Vector Quantization
RGB	Red, Green, Blue (color channels for images)
ReLU	Rectified Linear Unit (activation function)
SoftMax	SoftMax function (used in classification tasks)
MAP	Mean Average Precision
HOG	Histogram of Oriented Gradients
ANN	Artificial Neural Network
KNN	K-Nearest Neighbor
NB	Naive Bayes
DT	Decision Tree
VIA	VGG Image Annotator
PCA	Principle Component Analysis
EDA	Exploratory Data Analysis
FC	Fully Connected

Chapter 01

Introduction

1.1 Background and motivation

Plant disease affecting leaves, stems, roots, and fruits result in diminished crop quality and quantity, leading to global food deprivation and insecurity [1]. Globally, crop diseases contribute to an estimated annual yield loss of approximately 16%, serving as a primary factor behind both food shortages and escalated production costs [2]. According to the Food and Agriculture Organization Report (FAO), the world's population will reach approximately 9.1 billion by 2050. For a steady food supply, about 70% of food production growth is required [3]. Diseases and disorders occur in plants and their products. Biotic factors include diseases caused by algae, fungi or bacteria and abiotic factors inducing disorders comprise rainfall moisture, temperature and nutrient lack [4]. Crop disease is actually a major crisis in day-to-day life.

Agriculture is under tremendous pressure to supply the growing demand for food while maintaining sustainability due to the world's population expansion. However, plant diseases pose a serious threat to crop output since they can destroy harvests if they are not detected and treated in a timely manner. In addition to being expensive and time-consuming, the traditional method of identifying plant diseases—visual inspection by specialists—is also problematic for large-scale farming operations. These techniques frequently rely on human knowledge, which might not always be accessible in remote or resource-constrained locations. As a result, farmers suffer financial losses, and the global food supply chains are also impacted, as this leads to a major gap in disease monitoring and prompt response.

The promise of developing technology to overcome these gaps is what inspired this study. Deep learning developments have revolutionized a variety of industries by providing great accuracy and efficiency for jobs involving pattern recognition. Applying deep learning models to agriculture allows for the accurate identification of diseases that are frequently indistinguishable to the human eye by analyzing complicated visual data from plant leaves. We envisioned a real-time monitoring system that could give farmers immediate, trustworthy, and useful insights by utilizing these capabilities.

Farmers and other agricultural stakeholders are enabled to identify diseases in their early stages with the help from of real-time monitoring system. By enabling tailored treatments and drastically reducing the spread of illnesses, early diagnosis minimizes the need for pesticides and other dangerous chemicals. This encourages ecologically friendly farming methods in addition to improving crop health and output. Furthermore, even small-scale farmers, who frequently lack the funds to carry out extensive disease control procedures, can now access sophisticated agricultural solutions.

By pursuing this line of inquiry, significant difficulties are addressed which that are now being faced in contemporary agriculture through the application of innovation. The management of plant diseases is revolutionized by the incorporation of deep learning into real-time systems, which paves the way for precision agriculture. In the end, this research makes a contribution to worldwide efforts to achieve food security, improve the livelihoods of farmers, and reduce the environmental impact that agricultural practices leave behind.

1.2 Thesis objectives

Plant diseases significantly impact crop yield and quality, leading to economic losses and food security issues. Early and accurate detection of plant diseases is critical for timely intervention and reducing the spread of damage. Traditional methods rely heavily on manual inspection, which can be time-consuming and prone to error. Deep Learning (DL) techniques have emerged as a promising solution, offering efficient and automated disease detection. This study proposes an integrated approach for detecting plant leaves damage using Deep learning techniques. DL algorithms are effective for detecting damage on plant leaves rather than other techniques. The main objectives of this research work are:

1. To build CNN model for detecting diseases on various plant leaves.
2. To develop a real-time mobile application and web-based application for monitoring the plant diseases.
3. To evaluate the deployed CNN model's performance.

1.3 Context of diseases on various plant leaves

The reduction in crop productivity and quality that is caused by plant leaf diseases presents substantial problems to the agricultural industry. The following is a detailed description of illnesses that affect a wide range of crops, supported by research that were conducted recently.

- **Potato Leaves Diseases**: Early blight, caused by Alternaria solani, is characterized by concentric rings on older leaves, causing to defoliation and yield loss. It thrives in warm and humid settings [5]. When circumstances are chilly and moist, late blight, which is caused by Phytophthora infestans, emerges as lesions that are saturated with water and eventually become necrotic. This illness triggered historical events like the Irish Potato Famine [6].

- **Tomato Leaves Diseases**: Tomato early blight, similarly caused by Alternaria solani, generates dark, concentric circles on leaves, eventually causing defoliation [7]. Tomato late blight, caused to Phytophthora infestans, creates water-soaked lesions on leaves and fruits, which decay under humid circumstances [8]. Another important disease, Tomato Yellow Leaf Curl Virus (TYLCV), carried by whiteflies (Bemisia tabaci), leads to leaf curving, reduced development, and yellowing of leaf margins, significantly limiting fruit output [9].

- **Pepper-bell Leaves Disease**: Bacterial Leaf Spot is a frequent disease in pepper and bell pepper plants, caused by the bacterium Xanthomonas campestris pv. vesicatoria. It typically affects the leaves, but can also influence the stems and fruit [13].

- **Apple Leaves Disease:** Apple scab, caused by Venturia inaequalis, is identifiable by olive-green to black lesions on foliage and fruit, producing defoliation and malformations in apples [10]. Apple cedar rust, caused by Gymnosporangium juniperi-virginianae, creates yellow-orange blotches on leaves, particularly near cedar trees, which function as alternative hosts [9].

- **Corn Leaves Disease:** Northern leaf blight, caused by Setosphaeria turcica, generates cigar-shaped lesions on leaves, lowering photosynthesis and yield [11]. Common rust, owing to Puccinia sorghi, creates reddish-brown pustules upon leaflets and is more abundant in damp, temperate areas [12].

- **Grape Leaves Disease**: Grape black rot, caused by Guignardia bidwellii, creates dark, sunken lesions on leaves and fruit, leading to shriveled, inedible grapes [10]. Grape Esca (Black Measles), caused by a complex of fungus, causes to leaf scorch, black streaks on shoots, and microscopic black spots on fruit, which increase with time [9].

- **Peach Leaves Disease:** Bacterial spot, caused by *Xanthomonas arboricola pv. pruni*, produces water-soaked lesions that turn necrotic on leaves and fruits, reducing fruit marketability. Premature defoliation is common in severe infections [13].

- **Rice Leaves Disease:** Brown spot, caused by Bipolaris oryzae, appears as tiny, circular lesions on leaves, flourishing in dry, nutrient-deficient soils [14]. Bacterial leaf blight, caused by Xanthomonas oryzae, begins as water-soaked streaks on leaves and develops to necrosis, severely reducing plant development and yield [5].

In conclusion, plant leaf diseases constitute a substantial hazard to world agriculture, impacting a wide range of crops and resulting to large output losses. Early identification and appropriate therapy of these disorders are critical to reduce their effect. The continued research of disease pathogens, environmental factors, and effective management measures continues to be crucial in enhancing crop resilience and guaranteeing food security.

1.4 Context of detecting diseases on plant leaves using deep learning technique

Deep learning approaches, notably Convolutional Neural Networks (CNNs), have proved to be exceptionally successful in the identification and categorization of plant leaf diseases. These methods harness vast databases of plant photos to automate the process of recognizing signs of different illnesses, considerably boosting accuracy and efficiency compared to previous manual approaches. A recent analysis of these improvements demonstrates that CNN-based models, trained on substantial datasets, may achieve classification accuracies ranging from 85% to 98% across diverse plant species and illnesses [15].

CNN models such as VGG16, VGG19, MobileNet, InceptionV3, Xception, and DenseNet121 are commonly used for diagnosing plant leaf diseases owing to their capacity to extract complicated characteristics from pictures. VGG models, however deep and computationally costly, give excellent accuracy for illness classification, whereas MobileNet provides a more efficient option ideal for real-time applications. InceptionV3 and Xception excel in multi-scale feature extraction, making them efficient for detecting a wide range of plant diseases with diverse patterns. DenseNet121, distinguished for its dense connection, efficiently collects fine-grained data, boosting illness detection performance. These models have demonstrated encouraging outcomes in both research fields and practical applications in agriculture fields [16-17]. Several studies have proved the effectiveness of these strategies in both controlled and field settings. For example, one research found that when deep learning models were applied to photos of plant leaves, they could detect illnesses like leaf blight and powdery mildew with high precision, even in fluctuating environmental circumstances. Another study publication examined the issues encountered when deploying these models in real-world agricultural settings, including the necessity for model stability across diverse geographical areas and varied illumination conditions.

Thus, deep learning offers a viable approach for increasing the speed, accuracy, and scalability of plant disease detection, representing a huge step forward in precision agriculture. These discoveries have the potential to alter agricultural operations by allowing for speedier responses to plant health concerns, thereby enhancing crop productivity and sustainability.

1.5 Conclusion

This thesis underscores the importance of plant disease detection in agriculture for enhancing food security and optimizing crop yields. The introduction highlighted the need for early diagnosis to prevent crop losses and provided the rationale for adopting advanced techniques. Deep learning methods, particularly CNNs, were emphasized as transformative tools for achieving faster and more accurate detection of plant leaf diseases. The project aimed to create a real-time monitoring system for disease identification, addressing limitations in traditional methods and contributing to sustainable farming practices.

The thesis objectives were clearly outlined, covering the development of a CNN model, its deployment via TensorFlow and FastAPI, and its integration into web and mobile applications. The subsequent chapters provide an in-depth exploration of previous studies, the methodology for implementing the proposed system, and a detailed analysis of results, including real-time applications. The final chapter reflects on the outcomes and suggests potential future directions for improving automated plant disease detection.

Chapter 02

Literature Review

2.1 Introduction

The initial phase of this research endeavor involves a comprehensive analysis of studies about the disease detection on various plant leaves using different types of deep learning models, such as Custom CNN model, VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, EfficientNet and lots of hybrid models. This preliminary investigation aims to offer a thorough understanding of the methodologies employed by previous researchers. Thus, the identified limitations and gaps in previous research will play a vital role in shaping the objectives fulfillment and methodologies of this thesis.

2.2 Previous study about detection of diseases on various plant leaves using various DL algorithms

This section presents all the previous literature regarding deep learning algorithms.
Zhou et al. suggested a hybrid technique for speedy and reliable rice disease diagnosis that combines Fuzzy C-Means (FCM), K-Means (KM), and the Faster Region-Based Convolutional Neural Network (Faster R-CNN). The FCM-KM method was employed for image preprocessing, which efficiently segmented sick regions to improve feature extraction. Faster R-CNN was then used to identify and detect particular rice illnesses, producing faster and more accurate results than older approaches. The fusion model outperformed in accuracy, recall, and F1-score, indicating its promise for real-time agricultural applications [18].

Sharma et al. proposed a machine learning-based strategy for diagnosing plant leaf diseases, with an emphasis on the use of image preprocessing techniques to improve accuracy. The study used histogram equalization and Gaussian blur to preprocess pictures before extracting features using approaches such as Grey Level Co-occurrence Matrix (GLCM). Several machine learning techniques, including Support Vector

Machine (SVM) and Random Forest, were tested for classification, with SVM attaining the greatest accuracy rate. The suggested approach has shown substantial promise for early disease identification, which contributes to better plant health monitoring [19].

By combining Convolutional Neural Networks (CNNs) with the Learning Vector Quantization (LVQ) algorithm, Sardogan et al. were able to build a hybrid method for identifying and categorizing illnesses that affect plant leaves. In order to extract detailed characteristics from leaf pictures, the CNN was utilized, and the LVQ algorithm was utilized to efficiently classify the illnesses based on these features. In the process of illness classification, the system was evaluated using a dataset consisting of photographs of plant leaves, and it achieved an accuracy of 96.5%. The technique was acceptable for real-time agricultural disease detection as a result of this high performance, which highlighted the effectiveness of merging deep learning for feature extraction with traditional algorithms for classification [20].

Ozguven et al. published a study on the considerable impact of sugar beet leaf spot disease (Cercospora beticola Sacc.), which can lower sugar output by 10-50%, underlining the need of early identification and treatment. A modified Faster R-CNN architecture, improved by altering CNN parameters, was presented for automated illness severity identification with imaging-based expert systems. After training and testing on 155 images, the model achieved a classification accuracy of 95.48%, outperforming traditional methods. The results show that tailoring CNN settings to specific visual attributes improves detection performance. The technique is likely to speed up disease detection across wide regions, reduce human error, and increase management efficiency in sugar beet cultivation [21].

Lu et al. emphasized the vital necessity for automated systems in the identification and diagnosis of rice illnesses within the context of agricultural information systems. To address issues in vegetable pathology, the study offered a unique strategy that makes use of deep convolutional neural networks (CNNs), a strong tool in pattern recognition and machine learning. The CNN model was trained using a dataset of 500 real photos of damaged and healthy rice leaves and stems recorded in experimental rice fields. The system was built to classify ten common rice illnesses, and it achieved an accuracy of 95.48% using a 10-fold cross-validation technique. This performance greatly surpassed traditional machine learning approaches, indicating the viability and resilience of CNN-based systems for detecting rice diseases. The findings highlight the ability of deep learning models to improve disease control techniques in agriculture by delivering precise and scalable solutions [22].

Kawasaki et al. conducted a fundamental study on automating the detection of viral plant diseases using Convolutional Neural Networks (CNNs). An early use of deep learning in plant pathology was the subject of this research, which aimed to construct a CNN-based model for the purpose of identifying viral infections in plants based on picture datasets. The model attained a classification accuracy of 94.9%, highlighting the promise of CNNs for accurate and automated plant disease diagnosis. Within the context of agricultural disease control, the study lay the framework for exploiting deep learning techniques, demonstrating their capacity to minimize reliance on manual procedures and enhance diagnostic efficiency [23].

Jiang et al. present a novel deep learning-based technique, INAR-SSD, for real-time detection of five common apple leaf diseases: Alternaria leaf spot, Brown spot, Mosaic, Grey spot, and Rust. Utilizing the Apple Leaf Disease Dataset with 26,377 annotated photos, the model includes the GoogLeNet Inception structure with Rainbow concatenation to boost accuracy and speed. Experimental findings indicate a detection accuracy of 78.80% mean average precision (MAP) and a processing speed of 23.13 FPS, exceeding previous approaches. This research presents a substantial development for early detection and real-time monitoring, contributing to increased apple output and disease control [24].

Islam et al. developed a technique for detecting potato illnesses using picture segmentation and a multiclass Support Vector Machine (SVM) classifier. The study segmented potato leaf pictures to identify regions of interest, applying image processing techniques to separate sick spots successfully. Features including as shade and texture were collected from the segmented pictures and utilized to train a multiple-class SVM for diagnosing illnesses, including early blight, late blight, and healthy leaves. The suggested technique attained an overall classification accuracy of 95%, proving its usefulness in illness detection. This technique provides a reliable tool for diagnosing potato illnesses, which is critical for prompt intervention and yield enhancement [25].

Iqbal et. al. created an image processing and machine learning-based method for detecting and classifying two primary potato leaf diseases, Early Blight and Late Blight. Using 450 photos from the Plant Village database, the study used image segmentation to separate sick regions and assessed seven classifier systems for disease detection. The Random Forest classifier had the best accuracy of 97%, indicating the system's ability to discern healthy from damaged leaves. This method provides a dependable and automated option for early disease diagnosis, potentially increasing potato crop yields in places like Bangladesh [26].

Mangla et al. suggested a machine learning-based system for identifying rice leaf diseases using image processing methods. The researchers used image segmentation to preprocess photos of paddy leaves and extract attributes including texture, color, and form, which were then identified using a Support Vector Machine (SVM) technique. The system correctly identified common diseases such as bacterial blight and leaf spot. This method provides an effective and automated alternative for the early detection of rice leaf diseases, assisting farmers in disease control and crop yield [27].

Rizqi et al. studied the ability of the brown-rot fungus Daedalea dickinsii to decolorize and convert methylene blue dye, which is a prevalent environmental contaminant. The study found that the fungus could successfully destroy the dye, with decolorization rates of up to 92% under optimal circumstances. This transition was ascribed to fungal enzymatic activity, which converts the color into less toxic metabolites. The findings emphasize the potential use of D. dickinsii in bioremediation methods to clean dye-contaminated wastewater, providing an environmentally acceptable and effective solution to industrial pollution [28].

Azath et al. created a deep learning-based model utilizing CNN to detect common cotton leaf diseases and pests such as bacterial blight, spider mites, and leaf miners, which are difficult to diagnose manually. To improve generality, the model was trained and validated on a dataset of 2400 photos using K-fold cross-validation. The system, which was implemented using Python and Keras with a TensorFlow backend, achieved 96.4% accuracy in illness and pest classification. The paper emphasizes the model's suitability for real-time applications, proposing an IT-based alternative to supplement existing detection methods and enhance cotton crop management [29].

Andrew et al. offered a research on the application of deep learning, namely convolutional neural networks (CNNs), to identify plant illnesses early and increase food production quality while reducing economic losses. It focuses on fine-tuning the hyperparameters of pre-trained models—DenseNet-121, ResNet-50, VGG-16, and InceptionV4 with the PlantVillage dataset, which contains 24,305 photos from 18 plant disease categories. The performance of these models was assessed using classification accuracy, sensitivity, specificity, and F1 score. The findings indicated that DenseNet-121 had the greatest classification accuracy of 99.81%, exceeding other models and cutting-edge approaches for plant disease identification [30].

Manowarul et. al demonstrates a report that emphasizes the significance of early plant disease diagnosis in ensuring a healthy agricultural industry. To detect crop illnesses, the study used deep learning models such as CNN, VGG-16, VGG-19, and ResNet-50 on the PlantVillage dataset (10,000 pictures). The findings indicated accuracy rates of 98.60% for CNN, 92.39% for VGG-16, 96.15% for VGG-19. CNN beat the other models, making it the ideal candidate for creating a smart online application for real-time crop disease prediction. The software analyzes leaf photos to assist farmers diagnose plant illnesses, with the goal of preventing economic losses through early identification and focused treatment [31].

Omkar Kulkarni investigates the use of deep learning techniques to identify crop diseases. The study focuses on the use of neural networks, specifically convolutional neural networks (CNNs), to categorize and diagnose illnesses in plants using photographs of leaves. The research emphasizes the importance of early disease identification in crops in preventing large-scale economic losses for farmers. It addresses the creation and use of a deep learning model capable of identifying healthy from unhealthy crops. The system's disease detection accuracy is assessed, indicating the potential of deep learning models to aid in successful agricultural disease control. The findings add to the expanding corpus of research on applying AI and deep learning to better crop monitoring and disease control in agricultural operations [32].

Chohan et. al. proposes a deep learning-based model, Plant Disease Detector, aimed at detecting plant diseases from leaf images. The model utilizes a Convolutional Neural Network (CNN) with multiple convolution and pooling layers, trained on the PlantVillage dataset. To enhance model performance, data augmentation is applied to increase the sample size. The model achieved a testing accuracy of 98.3% using 15% of the dataset for validation, which includes both healthy and diseased plant images. The study demonstrates the effectiveness of the model in identifying plant diseases. Future work could involve integrating the model with drones or other systems for real-time disease detection and location reporting, aiding in targeted plant treatment [33].

Crop diseases are a serious danger to food security, yet early detection is difficult owing to poor infrastructure in many places. Deep learning advances and broad smartphone availability have paved the way for smartphone-assisted plant disease detection. Mohanty et al. present work that uses a dataset of 24,306 photos of healthy and sick plant leaves to train a deep convolutional neural network. The model obtained a stunning 97.35% accuracy on a test set, proving the efficacy of deep learning in illness diagnosis. This technique provides a scalable solution for worldwide crop disease diagnostics by utilizing huge, publicly available databases [34].

Maniyath et al.'s work "Plant Disease Detection Using Machine Learning" looks into how machine learning, especially Random Forest, may be used to diagnose plant illnesses using leaf image categorization. It describes a multi-phase technique that includes creating a dataset, extracting features with the Histogram of Oriented Gradients (HOG), training the Random Forest classifier, and categorizing photos as healthy or ill. The method illustrates that training machine learning models with huge, publicly available datasets enables scalable and effective plant disease diagnosis, making it a valuable tool for agricultural diagnostics [35].

Jung et al.'s study “Construction of Deep Learning-Based Disease Detection Model in Plants” describes a method for early and accurate agricultural disease diagnosis that can increase crop quality and productivity. The scientists created an automated system employing convolutional neural networks (CNNs), five pre-trained models, and a stepwise classification method. The method has three stages: crop categorization, illness detection, and disease classification, with an additional “unknown” category to improve generality. The model achieved excellent accuracy (97.09%) in detecting and categorizing illnesses and crops, with future improvements possible by increasing the training dataset to include more crops. This adaptive approach holds promise for smart farming applications, particularly in Solanaceae crops, and may be expanded to other crops with more diversified datasets [36].

The paper "Diagnosis and Classification of Grape Leaf Diseases Using Neural Networks" by Sannakki et al. focuses on early disease detection to reduce economic losses and increase crop quality. The researchers use image processing and artificial intelligence approaches to assess grape leaf photos with complicated backgrounds. It isolates green pixels using thresholding, then reduces noise with anisotropic diffusion. Disease segmentation is done using K-means clustering, and the sick area is recognized. A Feed Forward Back Propagation Neural Network is trained for classification, resulting in accurate diagnosis and categorization of grape leaf disorders [37].

The research "Image Processing for Smart Farming: Detection of disease and Fruit Grading" by Jhuria et al. investigates the use of image processing and artificial neural networks (ANN) to monitor crop diseases and fruit grading from plantation to harvest. The work focuses on three grape illnesses and two apple diseases, employing two picture databases: one for training pre-labeled disease photos and the other for testing with query images. Disease categorization is based on three feature vectors: color, texture, and morphology, with morphology providing the best accuracy at 90%. Backpropagation is used to update weights in the ANN model, which is implemented in MATLAB. The paper also shows algorithms for mango counting and disease spread analysis, highlighting useful tools for smart farming [38].

Wang et al. developed a unique approach for identifying vegetable illnesses and insect pests utilizing computer vision and image processing technologies that are tailored for mobile smart devices. The technique used smartphone photos to locate disease-affected regions and count insects on vegetable leaves. A novel extraction and classification technique are created to segment leaves from photos, followed by a region-labeling approach to compute disease regions and insect counts. Mathematical morphology was used to address overlapping regions in photographs. The approach was tested on mobile devices and confirmed in field testing, displaying great efficiency and excellent recognition performance. This study emphasizes the possibilities of merging smart devices with image processing for current agricultural applications [39].

Ahmed et al. created a machine learning-based method for identifying rice leaf diseases, recognizing the vital requirement for early detection to sustain healthy rice production in Bangladesh. The study concentrated on three prevalent diseases: leaf smut, bacterial leaf blight, and brown spots. Using clear photos of injured rice leaves on a white backdrop, the system used preprocessing approaches before training the dataset using several machine learning algorithms such as K-Nearest Neighbor (KNN), J48 (Decision Tree), Naïve Bayes, and Logistic Regression. Among them, the Decision Tree algorithm performed best, with over 97% accuracy after 10-fold cross-validation. The suggested method provides a time-effective and labor-saving alternative to manual illness detection [40].

Dixit et. al. discussed the challenges and developments in wheat leaf disease detection, highlighting the need of precise and early diagnosis to reduce production loss caused by diseases such as brown and yellow rust. They suggested an ML-based system for automating illness detection and categorization. The approach used a dataset obtained from several areas in Pakistan, taking into account aspects such as lighting and tilt during picture acquisition. Preprocessing included segmentation and scaling in order to properly discriminate between healthy and sick regions. ML models were then trained on the processed data, and their performance was measured using measures including accuracy, precision, recall, and F1-score. The suggested approach attained a remarkable accuracy of 99.8%, exceeding existing machine learning techniques and indicating its potential for improving wheat disease management [41].

Panigrahi et al. investigated the use of supervised machine learning algorithms to identify and classify maize plant diseases, recognizing the crucial need for early and effective disease control to avoid agricultural losses. The study used plant image data to evaluate the performance of Naive Bayes (NB), Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest (RF) classifiers. The Random Forest algorithm outperformed the other strategies, with an accuracy of 79.23%. The results show that these trained models can assist farmers in early disease identification and categorization, providing a realistic preventive tool for managing maize illnesses [42].

Phung et al. presented a high-accuracy method for cloud picture patch categorization on short datasets, solving the difficulty of training deep learning models with minimal data. They created a customized convolutional neural network (CNN) that was tuned for tiny datasets, including regularization techniques to improve generalization and reduce overfitting. In addition, a model average ensemble approach was developed to minimize prediction variance and enhance classification performance. Experiments using the Singapore Whole-Sky Imaging Categories (SWIMCAT) dataset yielded near-perfect classification accuracy for the majority of classes, demonstrating the resilience and usefulness of the suggested technique. This study highlights the possibility of integrating specialized CNN design with ensemble approaches for high-accuracy classification in data-constrained circumstances [43].

With the introduction of deep learning, plant leaf disease detection has received a lot of interest, as it allows for automation in disease identification and classification. Shivaprasad et. al. conducted their research using three models—CNN, VGG16, and VGG19, trained on a dataset of 9,127 annotated photos. The CNN beat the other models, with an accuracy of 97% and an F1 score of 95%, while VGG16 and VGG19 scored 96% and 95% respectively. This emphasizes CNN's scalability and feature extraction potential while also pointing to difficulties like as overfitting and computing requirements [44]. Similarly, Rana et al. established the viability of smartphone-assisted illness identification with a public dataset of 4,306 photos. Their CNN-based technique attained an accuracy of 97.35%, demonstrating its usefulness in areas with minimal infrastructure [45].

Expanding on multi-class detection, Pawar et al. suggested a 15-layer CNN trained on data from ten crop varieties. This algorithm not only diagnosed illnesses, but also offered contextual information such as pesticide recommendations and weather forecasts, effectively combining agricultural advice and disease detection. The study showed the need of robust datasets in improving model accuracy in uncontrolled contexts [46]. Similarly, Singh et. al. stressed the superiority of ResNet architectures in identifying potato leaf diseases, with an impressive 99.62% accuracy. Their methods included utilizing ResNet's skip connections for feature reuse, which increased training efficiency [47].

Pandian et al. exhibited further optimization in CNN designs by training a 14-layer deep CNN on 147,500 pictures. With data augmentation approaches such as neural style transfer, the model achieved an astonishing 99.96 percent accuracy. However, the authors stated that model resilience in real-world settings with noisy datasets is still an unresolved topic [48]. Hamed et al. investigated the importance of adaptation in various environmental circumstances by fine-tuning EfficientNetV2S for low-resolution and cluttered pictures, attaining excellent performance under noisy situations [49].

Peng et. al. devised a new strategy, combining YOLOv5 with metric learning to improve object localization and illness categorization. Their technique enabled the flexible inclusion of new disease classes without retraining, making it suitable for dynamic agricultural applications [50]. Addressing particular crops, Yadav et al. built AFD-Net for apple leaf diseases, exceeding state-of-the-art models with 98.7% accuracy on Plant Pathology 2020 and 92.6% on the 2021 dataset [51]. Furthermore, Nirmal et al. used CNN with LSTM layers to identify pomegranate illness, reaching 97.25% classification accuracy on a dataset with varying lighting and textural conditions [52].

Despite these advances, difficulties remain. Future research should focus on generalizing models across different climatic circumstances, assuring scalability to new crops, and minimizing computing cost for real-time applications. Collectively, these studies illustrate the revolutionary potential of deep learning in agricultural disease control, while also emphasizing areas that require further innovation.

Upon reviewing the existing literatures and the current study, several research gaps have been identified that warrant further investigation.

1. **Crop-specific focus, with limited multi-crop applicability**: Most research focus on particular crops such as rice, tomato, or sugar beet, which limits the applicability of the offered strategies to other crop types. This restricted emphasis impedes the development of adaptable solutions for broader agricultural uses.
2. **Lack of real-world field testing and scalability**: Many proposed approaches are highly accurate in controlled contexts but have not been tested across a wide range of real-world field circumstances. Additionally, scalability for large-scale agricultural implementation is sometimes underestimated, despite its need for widespread acceptance.
3. **Insufficient real-time implementation**: Several approaches do not integrate with real-time detection and monitoring systems. Real-time skills are critical for efficient disease control and fast decision-making in agricultural operations.
4. **Limited integration of IoT and advanced technologies:** Integration with IoT and cloud-based technologies for continuous illness surveillance and management is yet underexplored. This restricts the offered methodologies' practical use in real-world agricultural contexts.
5. **Data Quality and Augmentation Deficits:** Many researches have insufficient data augmentation procedures or limited access to varied, high-quality datasets. This has an influence on the robustness, reliability, and generalizability of machine learning illness detection models.

6. **Lack of standardization among models and datasets:** The procedures and datasets utilized in research vary significantly, and there is no consistent framework for evaluating and comparing model performance. This impedes the development of comprehensive solutions for plant disease detection.
7. **Challenges with user-friendliness and practical adoption**: Farmers frequently overlook the importance of usability and simplicity of adoption. To be useful in the agricultural industry, solutions should prioritize simplicity and practicality. Examples include disease progression tracking, which has been underexplored. Most models lack ways for tracking disease progression over time, limiting their applicability to long-term crop management and decision-making.

2.3 Conclusion

Despite breakthroughs in machine learning and deep learning for plant disease identification, obstacles remain in making these systems adaptable, practical, and scalable for real-world agricultural applications. Addressing these shortcomings can considerably boost the effectiveness and uptake these technologies in current farming methods.

Chapter 03

Methodology

3.1 Introduction

The first part of the methodology presents deep-learning approaches, mainly different CNN algorithms like VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, and various hybrid algorithms for leaf classification and leaf disease detection. These models have been chosen based on previous studies, as they often outperform others in plant leaf disease detection in terms of accuracy and efficiency. For any deep learning model to work, certain steps must be followed for successful completion, and many hidden layers need to be applied for different CNN algorithms. In the later part, web and mobile applications are built by integrating the frontend and backend using the React framework, React Native framework, MongoDB databases, and FastAPI.

3.2 Workflow of plant leaves disease detection process

Plant leaf disease detection proceeds methodically, starting with the choice of suitable datasets and ending with the performance assessment of the model as shown in Figure 3.1. TensorFlow Serving saves and uses a strong deep learning model created to precisely identify plant illnesses. Operating as a local server on devices including personal PCs, TensorFlow Serving hosts the trained model and responds to inference requests from outside applications, therefore enabling effective real-time predictions. This technique assures that the model may be exploited efficiently for practical applications.

The trained model is combined into a web application to provide accessibility and user-friendliness of the system. FastAPI, a high-performance web framework spanning TensorFlow server to user-facing web interface, helps to enable this connection. Users of FastAPI, including farmers and agricultural specialists, may upload plant leaf photos to the system and get real-time disease detecting findings. FastAPI guarantees minimal response times, therefore improving the scalability and usability of the system, even while it allows smooth communication between the application frontend and the machine learning backend.

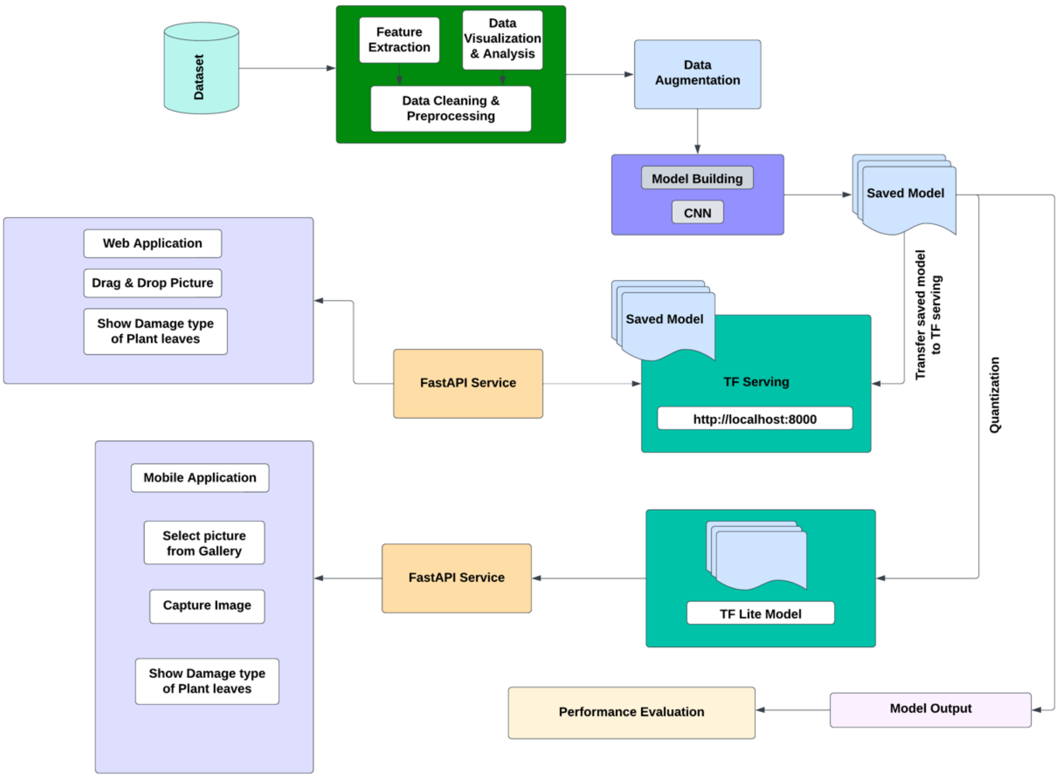


Figure 3.1: Workflow of plant leaves disease Detection using Deep Learning

Beyond its web application deployment, the technology is further enhanced for usage on mobile devices with TensorFlow Lite. Using quantization methods, the TensorFlow Lite model is customized to operate effectively on mobile devices with limited computing resources. This guarantees mobility and makes the equipment perfect for uses on-field. The mobile version employs the FastAPI service to facilitate interaction with the TensorFlow Lite model, allowing predictions to be made offline or in regions with restricted internet. This capacity is particularly important for farmers, enabling them to identify plant illnesses immediately in the field using their cellphones.

The integration of the plant disease detection model into both online and mobile platforms guarantees that the system is adaptable and accessible to a wide variety of users. Farmers may use it to identify illnesses such as potato early blight, tomato bacterial spots, and grape black rot immediately, while agricultural professionals can apply it to examine large-scale crop health data. By offering real-time diagnostic support, the system allows users to make timely decisions on crop treatments and management plans, ultimately lowering the risk of disease-related crop losses.

This dual deployment technique demonstrates the system's technological resilience and practical significance. By bridging the gap between powerful deep learning technology and real-world agricultural demands, it democratizes access to sophisticated plant health monitoring. The technology offers an effective and scalable answer to urgent agricultural difficulties, giving significant advantages for both small-scale farmers and major agricultural companies. Its design provides ease of use, scalability, and excellent performance, making it a powerful tool for increasing crop health and production in varied environments.

3.2.1 Data collection

The dataset that has been used in this study is derived from PlantVillage, as well new dataset of rice leaf diseases. This combined dataset comprised 30,945 images including eight plant species. All these photos belong to a broad range of plant diseases and healthy cases, which would make it a great hybrid dataset, having 35 distinct classes. Classes are organized by plant diseases on a particular crop or the presence of an overall healthiness level for each type of plant encountered in agricultural landscapes.

Table 3.1: Dataset details with class names

Plant Name	Number of Classes	Class names
Potato	3	Potato Early Blight, Potato Healthy, Potato Late Blight
Tomato	10	Tomato Early Blight, Tomato Healthy, Tomato Late Blight, Tomato Bacterial Spot, Tomato Leaf Mold, Tomato Mosaic Virus, Tomato Septorial

		Leaf Spot, Tomato Target Spot, Tomato Spider Mites, Tomato yellow leaf Curl Virus
Pepper-bell	2	Pepper_bell Bacterial Spot, Pepper_bell Healthy
Apple	4	Apple Black Rot, Apple Scab, Apple Healthy, Apple Cedar Rust
Corn	4	Corn Common Rust, Corn Healthy, Corn Northern Leaf Blight, Corn Cercospora Leaf Spot
Grape	4	Grape Black Rot, Grape Esca Black Measles, Grape Isariopsis Leaf Spot, Grape Healthy
Peach	2	Peach Bacterial Spot, Peach Healthy
Rice	6	Rice Brown Spot, Rice Leaf Scald, Rice Narrow Brown Spot, Rice Healthy, Rice Leaf Blast, Rice Bacterial Leaf Blight

Table 3.1 represents distribution of images and corresponding classes among various plant species. In that breakdown, the content of each plant class and disease classes can be discovered using this dataset.

3.2.2 Data annotation

Carefully annotated photos provide exact datasets needed for training and testing models of plant disease identification. This approach guarantees that machine learning systems can effectively distinguish between healthy leaves and those afflicted by different illnesses. Annotation on disease-affected areas on leaves marks them in a methodical manner, stressing minute visual characteristics as discoloration, uneven patches, or vein-specific symptoms. These exact labels help models to increase their classification accuracy and identify subtle illness signs.

For this thesis, simplicity and adaptability of the VGG Image Annotator (VIA) made it the ideal annotation tool. Detailed polygonal annotations made possible by VIA let several disease-affected areas be highlighted inside one picture. Leaves showing co-occurring illnesses or uneven patterns especially benefit from this ability. Even in cases when symptoms cross, the exact annotations enable the model to pinpoint and categorize several illness types. For instance, correct annotations let one more clearly differentiate illnesses like potato early and late blight, or bacterial spots and leaf mold in tomatoes - which may seem identical.

Figure 3.2: Annotated dataset of various plant leaves

Figure 3.2 represents the annotated dataset of different plant leaves diseases which actually helps to detect the accurate disease of leaves. Such rich datasets increase the resilience and generalizability of machine learning models, allowing them to adapt to varied crops, climatic circumstances, and disease severities. Ultimately, this meticulous annotation process assures that the plant disease detection system is both trustworthy and capable of supporting farmers and agricultural specialists in making timely and educated decisions regarding crop health and disease management.

3.2.3 Data cleaning

Feature extraction and analysis are critical processes in refining and understanding data to improve its suitability for analysis and modeling. Feature extraction selects and transforms important variables from raw data to emphasize the most relevant elements. This method includes dealing with missing values, which can affect analysis, by employing techniques such as imputation (using statistical measures such as mean or median) or eliminating incomplete information. Similarly, outlier treatment guarantees dataset consistency by recognizing and regulating extreme results using statistical approaches such as Z-scores or interquartile range criteria. Encoding categorical data is crucial since machine learning algorithms frequently demand numerical inputs. Categorical variables are converted into useable representations using techniques such as one-hot encoding and label encoding.

Scaling and normalizing also normalize numerical properties, which improves comparability and reduces noise. Dimensionality reduction techniques, such as principle component analysis (PCA), are also used to simplify datasets while retaining considerable variation, particularly in high-dimensional data. After characteristics are retrieved, the analysis focuses on identifying patterns and correlations via statistical approaches and exploratory data analysis (EDA).

To obtain insights, this technique uses visualizations, hypothesis testing, and summary statistics. Feature extraction and analysis work together to improve data quality, making it more relevant and dependable for downstream applications like predictive modeling and decision-making.

3.2.4 Data pre-processing

Pre-processing picture datasets is an important step in preparing data for machine learning tasks, especially when using image-based algorithms. Resizing photographs and normalizing pixel values are two critical image pre-processing steps. Resizing guarantees that all photos in the dataset have the same dimensions, which is critical for efficient computing and uniform input to models such as convolutional neural networks. For example, scaling all photos to a common size, such as 256x256 pixels, simplifies the architectural design and guarantees that the dataset follows a consistent input format. Uniformity in picture dimensions improves the training process and reduces mistakes caused by mismatched input sizes.

Normalizing pixel values is another critical step that includes scaling pixel intensity values (often ranging from 0 to 255) to a range of 0 to 1. This is usually accomplished by dividing all pixel values by 255. Normalization provides various advantages, including increasing the convergence rate of neural networks during training, preventing numerical instability, and ensuring that all input features (pixel values) have the same scale. Normalization stabilizes the training process while also improving the model's overall efficiency and performance.

These pre-processing stages, scaling and normalizing, work together to improve the quality and usefulness of picture data. They guarantee that the data is clean, consistent, and ready for analysis, allowing for the creation of more accurate and trustworthy machine learning models. Standardizing the inputs allows the model to focus on learning important patterns rather than being hampered by dataset irregularities.

3.2.5 Data augmentation

Data augmentation approaches are crucial for increasing datasets and boosting model resilience, notably in the context of plant leaf damage detection. These approaches artificially expand the number and variety of training datasets by performing changes to existing photos, imitating a wide range of circumstances. This not only helps minimize overfitting but also boosts the model’s capacity to generalize to unknown data. One such approach is random rotation, when photos are rotated by a random angle, for instance, between 0° and 45°, to imitate the natural variety in leaf orientations that could occur in real-world circumstances. Horizontal flipping offers another degree of diversity by producing mirror images, replicating leaves viewed from different angles. Similarly, zooming in on an image mimics differences in size, allowing the model to adjust to leaves collected at varied distances. Cropping, on the other hand, concentrates on certain features by cutting off select areas of the picture, enabling the model to concentrate on localized patterns such as spots, lesions, or discoloration.

Adjusting the brightness of photographs is another significant enhancement approach. It replicates different lighting conditions, such as strong sunshine or shadows, ensuring that the model functions effectively under diverse environmental situations. Adding noise to photos further boosts resilience by imitating distortions or defects that could be present in real-world data, such as camera artifacts or environmental influence.

These augmentation procedures collectively provide considerable ordinal variation to the dataset, increasing the model's capacity to generalize and making it more invariant to transformations, illumination changes, and noise. By integrating these methodologies, models for plant leaf damage detection become more accurate and durable, guaranteeing dependable predictions across various and demanding settings.

3.2.6 Dataset splitting and data visualization

The purpose of this research project is built around two key objectives: leaf classification and disease detection. These aims operate in combination to build a strong and accurate method for plant health evaluation. The first stage entails identifying the type of leaf to determine whether it belongs to a potato, tomato, or another plant species. This fundamental categorization ensures that future disease identification is done within the right biological context, as various plants are vulnerable to diverse sets of illnesses. To support successful model training and assessment, the dataset is carefully separated into three distinct subsets: training, testing, and validation sets. Typically, 80% of the dataset is assigned for training, while the remaining 20% is shared evenly between testing and validation. This divide guarantees that the model is exposed to a considerable percentage of the data for learning, while a separate, unseen sample is reserved to evaluate its generalizability and fine-tune its performance. The training set helps the model to learn patterns and features from the data, while the validation set is utilized during training to improve hyperparameters and avoid overfitting. Finally, the test set is applied to validate the model's accuracy and dependability on unseen data.

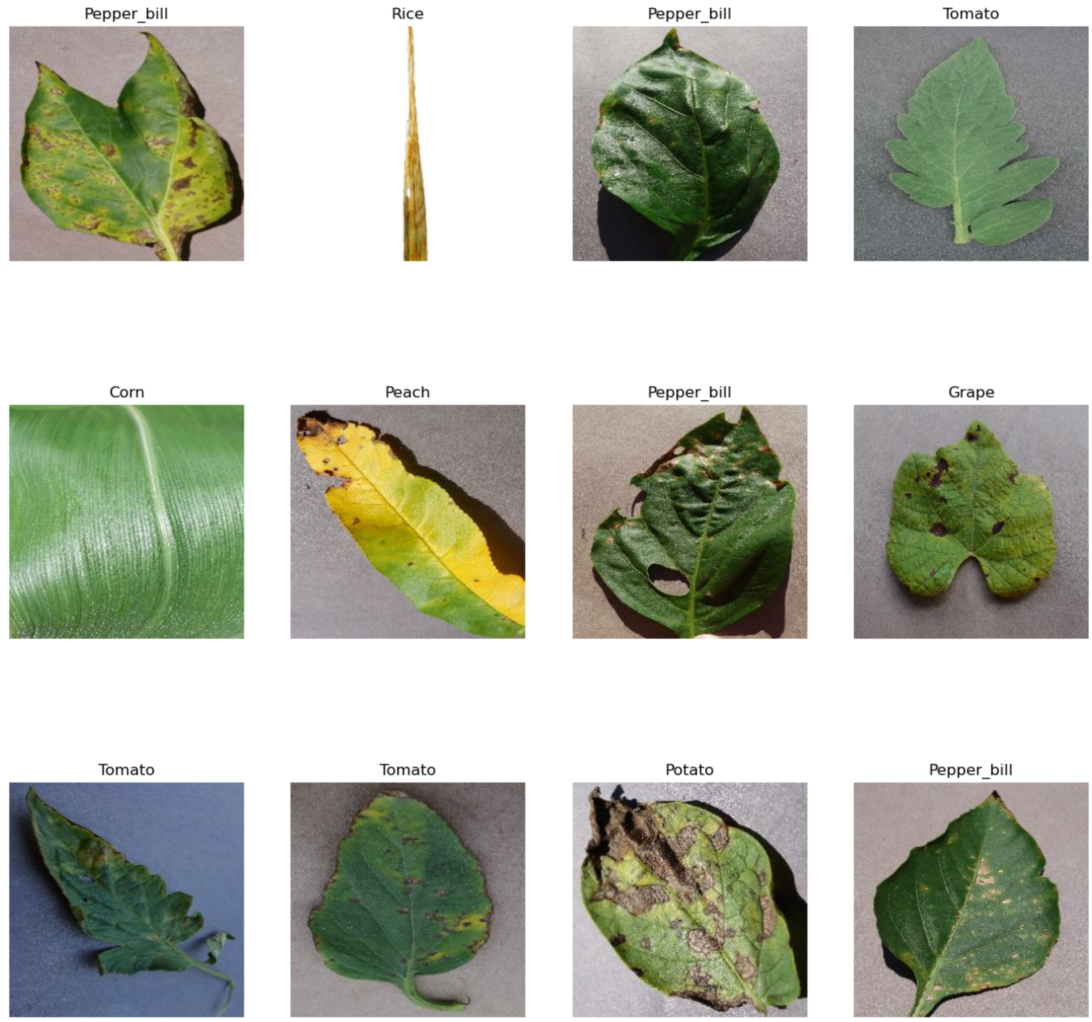


Figure 3.3: Data Visualization of Leaf Classification (Code Generated)

Figure 3.3 depicts the distribution of data across distinct leaf categories, demonstrating the visual diversity of plant leaf photos utilized for categorization. This image underlines the significance of balancing the dataset and keeping appropriate representation of each class to minimize biases during training. The diversity of plant leaf classifications means that the model is exposed to a wide range of attributes, helping it to efficiently learn and identify between leaf kinds. This extensive approach to dataset preparation, along with meticulous classification and visualization, offers the foundation for reliable illness detection. By assuring a systematic split and balanced representation of data, the study sets the framework for a viable and scalable solution for plant leaf classification and disease detection.

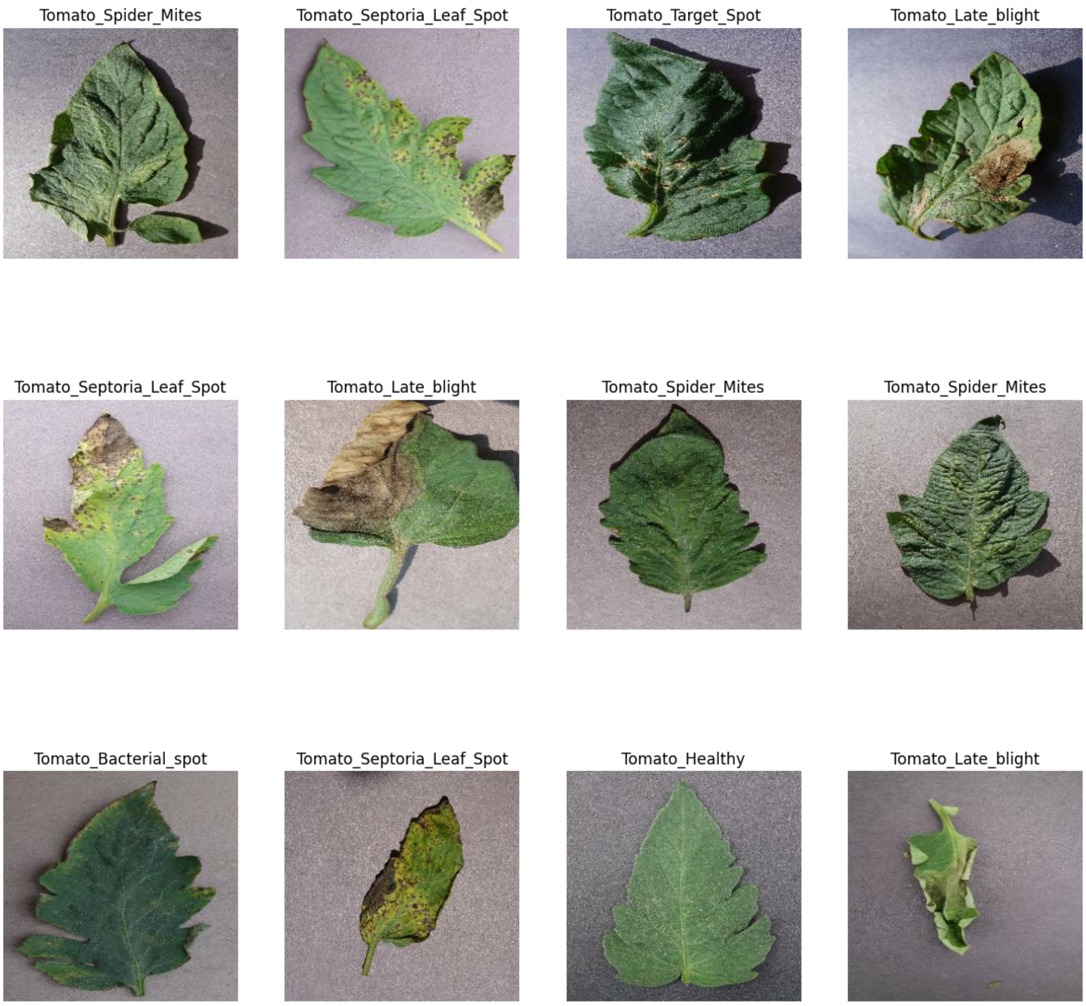


Figure 3.4: Data Visualization of Individual Plant's Leaf Disease (Code Generated)

The second part is attention to detecting diseases in leaves. It uses 80-10-10 split ratio of the same dataset. Disease Identification: subdivides the dataset into databases of individual plants. For example, if a leaf is identified as a potato leaf, the dataset specific to potato leaves, which includes three classes of diseases, is utilized. The deep learning model is then trained on this specific dataset to accurately identify which disease is present in the potato leaf. This approach ensures that the model is fine-tuned to the characteristics and disease patterns of each plant type, enhancing its precision in disease detection. Figure 3.4 represents the data visualization for Individual plant's multiple categories of leaf disease.

3.2.7 Model building

Figure 3.5 represents the schematic diagram of a basic Convolutional neural network architecture. Every convolutional neural network has several steps. Firstly, take input images from the dataset. Then some convolutional layers can be added for further steps. This convolutional layer varies because of various CNN model like VGG16, InceptionV3, EfficientNet, MobileNet etc. In every convolutional layer we can use various size matrices of filter.

After using the filter matrices of convolution layer, we can add another layer named pooling layer. Max pooling layer can also be used here for different CNN models. After that CNN model adds a Flatten layer to go further. Then dense layer can be added in CNN model building. After all of these, we can build a perfect CNN model for detecting plant leaves disease detection.

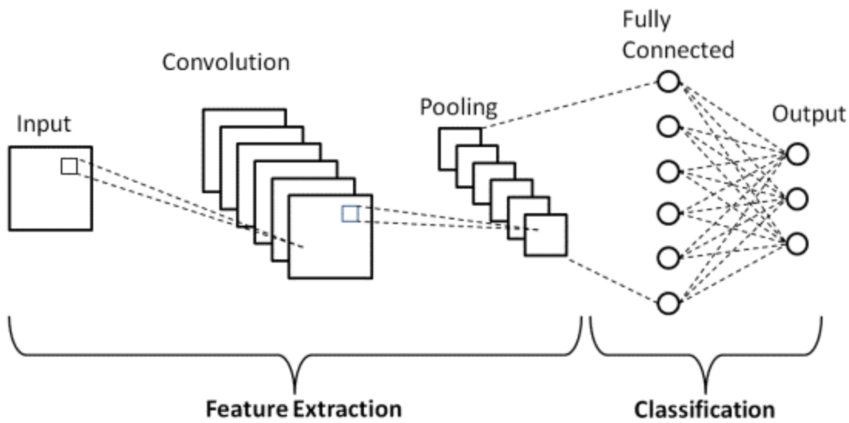


Figure 3.5: Schematic Diagram of basic Convolutional Neural Network (CNN) [30]

For this research study, we have chosen different CNN algorithms for various plant leaves disease detection like VGG16, VGG19, Custom CNN, InceptionV3, MobileNet, DenseNet121, Xception and two hybrid models. Each model's unique architecture and combination aim to enhance accuracy and performance across different types of plant leaves and diseases. In this research, the first hybrid model is concatenated with three different CNN algorithms like VGG16, DenseNet121, InceptionV3. The second hybrid model has been made by concatenating another three CNN algorithms like ResNet50, EfficientNetB0, MobileNetV2.

3.2.7.1 Custom CNN

A custom CNN typically consists of convolutional layers and pooling layers for feature extraction and spatial reduction. ReLU activation is commonly used to introduce non-linearity, combined with SoftMax for classification. In this work, the custom CNN model includes 23 convolutional layers, with 15 Conv2D and 8 MaxPooling layers optimized for extracting features and reducing dimensionality. This deeper architecture allows the model to capture more complex patterns and subtle features in plant leaves, improving the detection of diseases. The multiple Conv2D layers enable the extraction of intricate details and textures, while the 8 MaxPooling layers help in reducing the computational load and avoiding overfitting by summarizing spatial information. Various kernel sizes and filter matrices were applied to achieve accurate disease detection across diverse plant leaves [53].

3.2.7.2 VGG16

VGG16 is a deep convolutional neural network design that pays attention to the simplicity, efficiency of learning and use. The CNN has 16 layers (13 conv and 3 fc). The maxpooling layers are interleaved between the convolutional layers to reduce spatial dimensionality. All of these layers work perfectly with ReLU except for the output layer that uses SoftMax activation function when it comes to Categorical (multi-class) classification. It is well-liked by image classification people because of its highly efficient design and competitive performance on standard datasets [54].

3.2.7.3 VGG19

VGG19 is a new version of VGG which has 19 layers. It consists of 16 convolutional layers and three fully connected layers. The feature map sizes are reduced using maxpooling layers after every two or three convolutional layers. The remaining layers apply a Rectified Linear Unit (ReLU) activation function, which is key to modeling complex and nonlinear functions. Most of the time, for the case of multiple classes (like in multi-class classification tasks), SoftMax activation is used at this final layer to generate class probabilities. VGG19 has gained recognition for high accuracy as it derives from deep network layering and architecture compared to various algorithm like ResNet or Inception [55].

3.2.7.4 InceptionV3

The InceptionV3 is a convolutional neural network with 48 layers, designed with Conv layers to extract features, Pooling layers to reduce spatial dimensions between convolutional blocks, and a Fully Connected layer for output predictions [56]. Each layer uses the ReLU activation function to introduce nonlinearity and capture complex patterns in image data. This architecture includes a crucial Inception layer, which efficiently combines multiple convolutional operations to capture diverse features while maintaining computational efficiency. It is well-suited for tasks like image classification and object recognition, balancing depth and performance effectively.

3.2.7.5 MobileNet

MobileNet is a convolutional neural network architecture optimized for mobile and embedded vision applications. With 28 layers, it achieves outstanding computing efficiency by employing depth-wise separable convolutions, which divide ordinary convolutions into depth-wise and pointwise processes. This drastically decreases both the number of parameters and the computing complexity, making it appropriate for devices with limited resources. Instead of standard max pooling layers, MobileNet down samples spatial dimensions using depth-wise convolutions with strides, which preserves crucial spatial information while remaining simple [57].

The network employs the ReLU (Rectified Linear Unit) activation function throughout, resulting in nonlinearity and effective feature extraction. MobileNet's architecture stresses flexibility, with hyperparameters such as the width multiplier and resolution multiplier allowing users to trade between accuracy and computational cost, making it applicable to a wide range of applications. MobileNet's lightweight design, paired with competitive accuracy, makes it an excellent choice for real-time applications like as picture categorization and object identification on smartphones, IoT devices, and embedded systems.

3.2.7.6 DenseNet121

DenseNet121 is a kind of convolutional neural network in which each layer directly accesses the information from all the different layers that are present before it. Totally, it has 121 layers (including convolutional, pooling and fully linked layers). In a usual architecture, the use of maxpooling layers is replaced here by densely connected blocks where each layer takes as input all preceding feature maps and gives its output to subsequent convolutional blocks with concatenation. This method strengthens the feature propagation and enhances the feature learning, which in turn improves model performance. However, in this implementation ReLU has been used which works best for the activation function and allows us to learn 2D patterns with dense inputs [58].

3.2.7.7 Xception

Xception is a deep and complex convolutional neural network architecture. It consists of 71 layers, in which there are both convolution layers and depth wise separable convolution layers as well. Regarding the fact that Xception has depth wise separable convolutions, as opposed to standard CNNs, which learn spatial and channel-wise correlations. This architecture aims to capture the fine-grained detail in data while still being computationally efficient. Xception utilizes linear activation functions for its convolutional layers and uses ReLU activation in other parts of the network to bring nonlinearity necessary for learning features, classifying objects.

3.2.7.8 Hybrid models

The first hybrid model combines VGG16, InceptionV3, and DenseNet121 architectures to leverage their individual strengths for improved performance as shown in Figure 3.6. VGG16 provides deep feature extraction, InceptionV3 efficiently captures multi-scale patterns, and DenseNet121 ensures better gradient flow and feature reuse across layers. Merging these three models allows to benefit from their collective capabilities, resulting in higher accuracy, better generalization, and more robust detection of plant leaf diseases, outperforming each model when used separately.

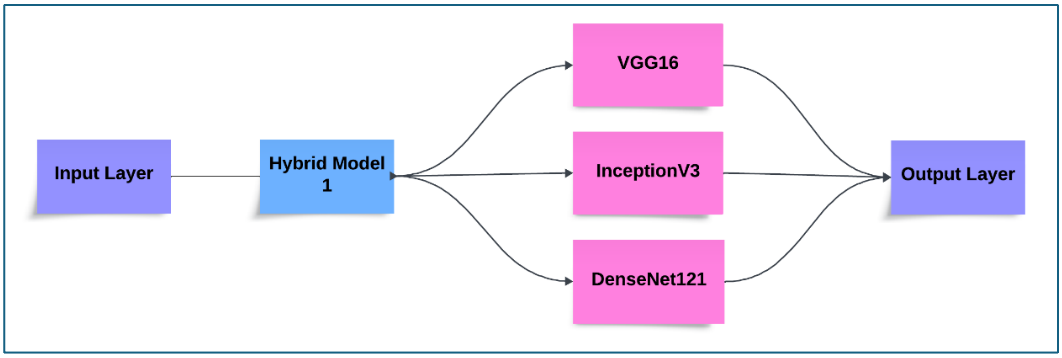


Figure 3.6: Schematic Block Diagram of Hybrid Model 1

Based on what architecture is used, the total number of layers in that hybrid model varies. Maxpooling layers are added to each component architecture to decrease the spatial dimensions of feature maps increase model efficiency and performance.

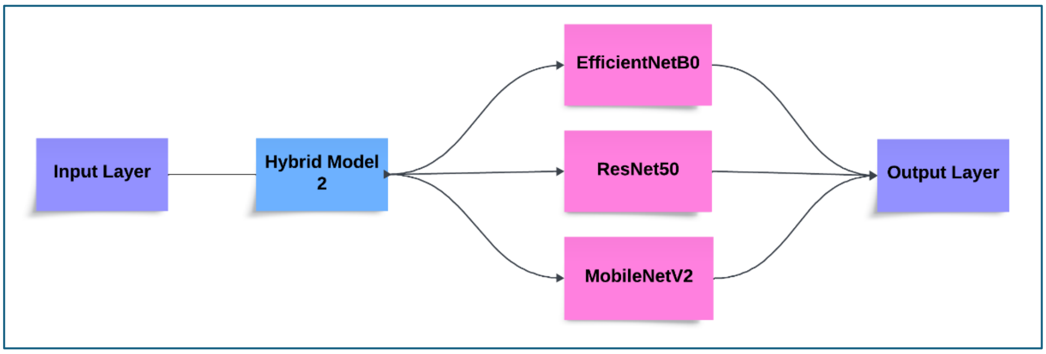


Figure 3.7: Schematic Block Diagram of Hybrid Model 2

Figure 3.7 shows the second hybrid model, which combines EfficientNetB0, ResNet50, and MobileNetV2, and incorporates the ReLU activation function across the network. The ReLU function introduces nonlinearity, essential for efficient deep neural network training and learning complex input patterns. EfficientNetB0 was selected for its scalability and efficient resource utilization, balancing accuracy with computational cost. ResNet50's residual connections effectively address the vanishing gradient problem, ensuring the network maintains performance as depth increases. MobileNetV2 adds computational efficiency with depthwise separable convolutions, making it ideal for resource-constrained environments. This hybrid design capitalizes on the complementary strengths of these models, enabling robust feature extraction, enhanced generalization, and improved accuracy for plant disease detection, advancing precision agriculture using cutting-edge machine learning techniques.

3.2.8 Architectural comparison of various CNN models

Today Convolutional Neural Networks (CNN) has become a basic building block in computer vision, and CNNs have provided the solution for many tasks like Image Classification, Object Detection or Semantic Segmentation. CNN is an evolving project, and several different architectures have been developed, each with their own design philosophy.

Table 3.2 demonstrates an in-depth architectural comparison between various CNN models, illustrating key attributes such as image size, model size, layers count depth-wise and involutional, parameter total. This comparison provides a deep dive into the pros and cons of each architecture, which may be helpful in selecting one for a specific applications.

Table 3.2: Architectural Comparison of several CNN models

	VGG16	**VGG19**	**Inception V3**	**Mobile Net**	**DenseNet 121**	**Xception**	**Custom CNN**
Model Size	528MB	549MB	92MB	16MB	30.5MB	88MB	58MB
Total Layer	16	19	48	28	121	71	23
Conv. Layer	13	16	23	27	117	36	15
Parameter	138.4 million	143.7 million	23.9 million	13 million	7.2 million	22.8 million	32.3 million

3.2.9 Tools and system configuration

This section gives a brief recap of the high-end hardware and software components selected in this research work, highlighting each element role and contribution to the project. The selective nature of this paradigm permits handling massive data, efficient model training, and rapid build-up of rich web + mobile applications. Therefore, the powerful architecture shows in Table 3.3 is very important to plant leaf damage detection system because it needs a high quality of performance and reliability. Several kernel crashes are observed when any CNN model is trained on a low-end machine.

Table 3.3: System Configuration and Necessary Software

Component	Specification	Role/Description
Graphics Card	Asus ROG Strix GeForce RTX 3090 OC Edition 24GB	Handles computationally intensive tasks of deep learning models without kernel crash issues
Processor	Intel 11th Gen Core i9-11900K Rocket Lake	Provides high-speed processing power for data handling and multitasking
RAM	Corsair VENGEANCE 16GB DDR5 6000MHz	Supports high-speed data processing and multitasking
Motherboard	Asus ROG CROSSHAIR X670E HERO DDR5 AMD AM5 ATX	Ensures optimal compatibility and performance of all components
Storage	Seagate FireCuda 530 2TB Gen4 M.2 2280 PCIe NVMe Gaming SSD	Offers fast read/write speeds for efficient handling of large datasets
Development Environment	Visual Studio Code	Provides an integrated environment for web and mobile application development
Frontend (Web)	React	Utilized for its component-based architecture and efficient rendering

Frontend Framework (Mobile)	React Native	Enables cross-platform development, ensuring compatibility with both Android and iOS
Backend Framework	FastAPI	Modern web framework used for building high-performance backend services
Database	MongoDB	Scalable and flexible NoSQL database used for storing user records
Interactive Environment	Jupyter Notebook	Used for developing and executing deep learning algorithms
GPU Toolkit	CUDA Toolkit and cuDNN	Facilitates efficient computation by leveraging GPU capabilities

The architecture was selected to meet the resource demands of deep learning model training, ensuring hardware that wouldn't hold up research progress while making resulting applications more reliable and scalable for further developments.

3.2.10 Converting model into TensorFlow serving

There are several key steps required to convert a model into a state that can be served with TensorFlow Serving such as fast deployment and real-time inference. Once training of the model is done, the first thing to do would be saving the created model using keras i.e. Keras format. The next step is to perform the implementation of TensorFlow Serving, which provides comprehensive and high-performance serving for machine learning models. Especially the operational part, this is recommended to use Docker since its dynamic and very elastic deployment helps at using it. The saved model is in a directory structure organized by version, with each version at the subdirectory of its base model path. This versioning gives the ability to perform simple updates and rollbacks without downtime. To start the TensorFlow Serving container, we map the base model directory to it and then provide the name of our models. This organization also enables us to provision many versions of the model, facilitating continual innovation by delivery and testing new models.

3.2.11 TF-Lite Model Conversion Process

TFlite model conversion is an essential component to implement machine learning models within devices at the edge, such as mobile phones with limited processing power. TensorFlow model needs to be trained, which is usually saved in the Saved Model format. Then the TensorFlow Lite model is added to TensorFLow Lite Converter, which is a special tool inside the library itself. There are many optimizations with conversion process, major one is quantization, which reduces the model size and improves inference time as it converts floating-point data to integer representations. Quantization can be done in different ways depending on the desired performance/accuracy trade-off: dynamic range, full integer or float16. After converting the model to a tflite file, the model is tested so that it functions correctly and gives consistent results as of original TensorFlow Model. This involves executing the TFLite model on test data and verifying it against that of original model. After validation, the TFLite model is used with TensorFlow Lite Interpreters and APIs for Android interfaces or iOS interfaced as well as embedded Linux through which it can be integrated into a mobile application in the end, testing it on target device is achieved of all those performance and accuracy criteria. Further optimizations like model pruning and hardware acceleration could be employed to increase performance. This complete conversion process ensures that the model becomes efficient, lightweight and works well in mobile devices.

3.2.12 Development Phase

After converting the TensorFlow model to TensorFlow Serving and as well a lightweight format version for mobile called ‘TensorFlow Lite’, these models were utilized by us to create robust backend API service that allowed users with confidence score. The backend was tested extensively using Postman for accuracy and stability before being weaved together with the frontend components. The backend API was carefully crafted with FastAPI, a modern and fast web framework for building APIs with Python. The front-end of the web application was created with React, a widely employed JavaScript framework lauded for its productivity and expandability.

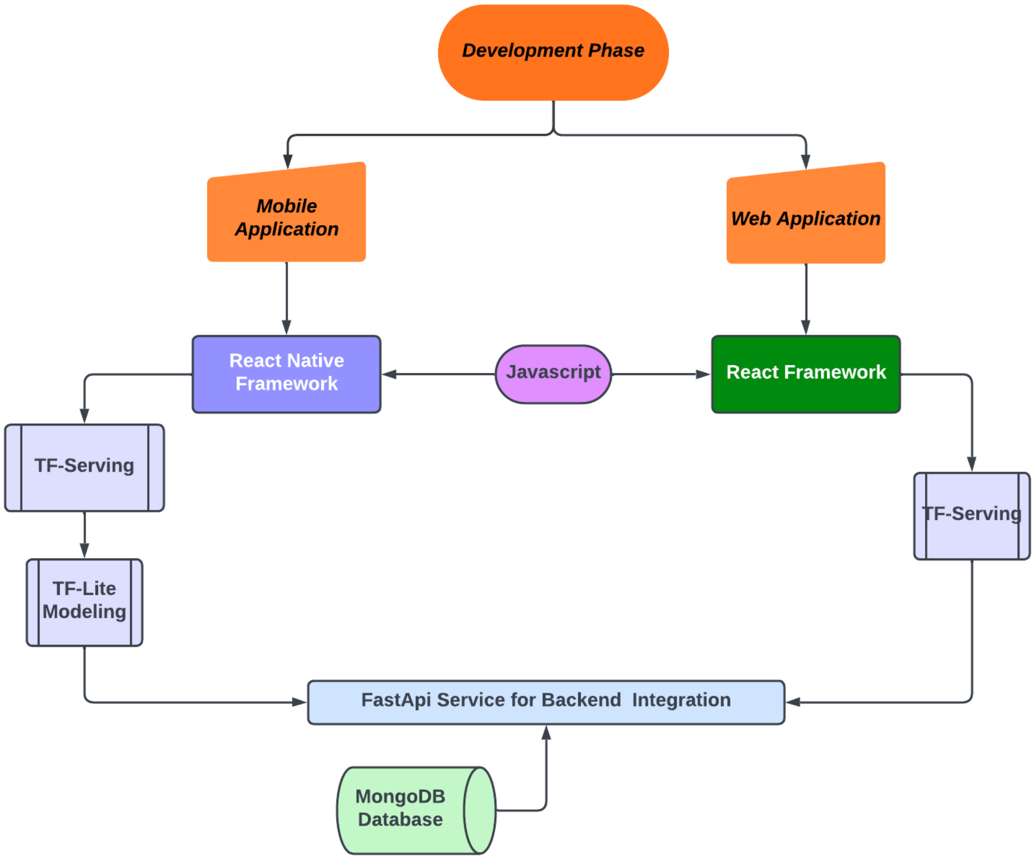


Figure 3.8: Flow Diagram of Mobile and Web Application Development

The backend api is successfully connected with react frontend, so users could interact using drag and drop-leaf images. In addition, a mobile application is developed using the React Native Expo framework that supports Android and iOS. This mobile application directly connects to the RESTful backend API also made more effective through ngrok for allowing real-time interaction. A secure database connection with MongoDB was established to maintain user data records and for reliable data storage & retrieval. This comprehensive approach shown in Figure 3.8 with extensive backend testing, frontend integration and cross-platform mobile development proves power of the system.

3.3 Conclusion

This chapter demonstrates all the workflow of research methodology. It starts from dataset collection, data annotation, data cleaning, data pre-processing, data augmentation technique. Then it goes to model building section for leaf classification and disease detection using various DL models. Development phase is the last stage.

Chapter 04

Results and Discussion

4.1 Introduction

This chapter focuses on the results obtained from applying deep learning techniques for detection of leaves disease on various plants. It showcases the performance of nine models: Custom CNN, VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, Xception and two hybrid models. Different types of CNN models demonstrate superior performance for specific plant leaf disease detection. It is because of dataset diversity, dataset's feature extraction process, data augmenting criteria, internal architecture of different CNN models and so on.

4.2 Results of leaf classification process based on epochs

Numerous CNN models have been trained on the dataset before evaluating it by technical aspects such as accuracy, precision recall and F1 score. The training process are divided into 2 phases: leaf classification and plant level disease detection. Leaf classification was done using custom CNN model that has been trained on a dataset containing 30,945 images of different plants.

Table 4.1: Comparison of leaf classification model based on number of epochs

Comparative Parameter	Number of epochs (20)	Number of epochs (40)
Accuracy	95.62%	97.28%
Loss	0.1343	0.0982
Batch Size	32	32
Time/steps per epoch	753s	738s
Precision	0.9578	0.9739
Recall	0.9522	0.9678
F1-score	0.9643	0.9837

Table 4.1 represents the comparison of custom CNN model for leaf classification on large dataset based on various number of epochs. From that table it can be concluded that a greater number of epochs causes more accuracy in this case. Due to the dataset's large size, initially 20 epochs have been set for training the dataset, achieving a time/step of 753 seconds per epoch. This training phase has yielded an accuracy of 95.62% and a loss of 0.1343. To enhance the model's performance, training epochs have been extended to 40, which results in an improved accuracy of 97.28% and a reduced loss of 0.0982. The batch size for this training is set to 32.

4.3 Performance Evaluation on Individual Plant's Disease Detection

After the successful leaf classification, the next step involved proceeding to individual plant disease detection. This phase required partitioning the dataset into specific plant disease subsets, which subsequently reduced the time/steps per epoch, thereby shortening the overall model training time. There is eight individual plant's leaves disease dataset which is to be evaluated on various CNN model like VGG16, VGG19, Custom CNN etc. For this research, seven CNN model algorithms have been implemented on each plant's leaves and also applied two hybrid models. From all that model's performance evaluation, the best model among different CNN algorithms has been selected for the development purpose on various plant.

4.3.1 Potato Plant's Performance Evaluation

For the classification of potato plant diseases, a curated dataset of 2,152 potato leaf images was utilized, divided into three classes: early blight, healthy, and late blight, as shown in Table 3.1. The dataset includes 1,000 images each for early and late blight and 152 healthy leaf images. Table 4.2 summarizes the performance of various CNN models using metrics like accuracy, precision, recall, and F1-score. The custom CNN model outperformed others due to its tailored architecture, fine-tuned to capture the unique patterns in the dataset, its ability to mitigate overfitting by optimizing hyperparameters, and its efficient use of feature extraction layers, enabling it to better distinguish subtle differences between disease symptoms.

Table 4.2: Performance Evaluation for Potato Plants on Various CNN Models

CNN Models	Accuracy	Precision	Recall	F1-score
Custom CNN	100%	0.9967	0.9967	1.0000
Hybrid Model 1	98.84%	0.9884	0.9884	0.9880
Hybrid Model 2	99.53%	0.9953	0.9953	0.9951
VGG16	97%	0.95	0.9633	0.9533
VGG19	97%	0.9467	0.9467	0.9467
InceprionV3	97%	0.98	0.93	0.9533
MobileNet	98%	0.9733	0.9633	0.97
DenseNet121	98%	0.99	0.96	0.9733
Xception	96%	0.9633	0.9233	0.9433

The confusion matrix shown in the Figure 4.1 depicts the performance of a classification model in diagnosing potato leaf conditions. The matrix provides reliable predictions for three classes: potato early blight, potato healthy, and potato late blight. From the confusion matrix, actual class and predicted class have been found for detecting diseases on potato plant leaves. By using this classes, accuracy, precision, f1-score and recall can be measured as shown in Table 4.2.

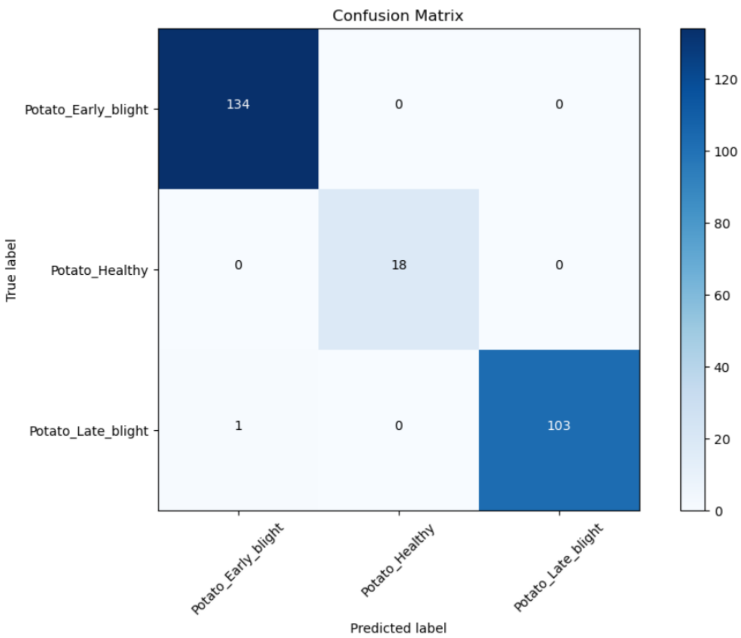


Figure 4.1: Best model's confusion matrix of potato leaves disease (Custom CNN)

Figure 4.2 represents the best model's accuracy vs epochs curve and loss vs epochs curve of potato plant leaves diseases. The best model for potato plant leaves disease is custom CNN which is obtained after implementing nine CNN models on it.

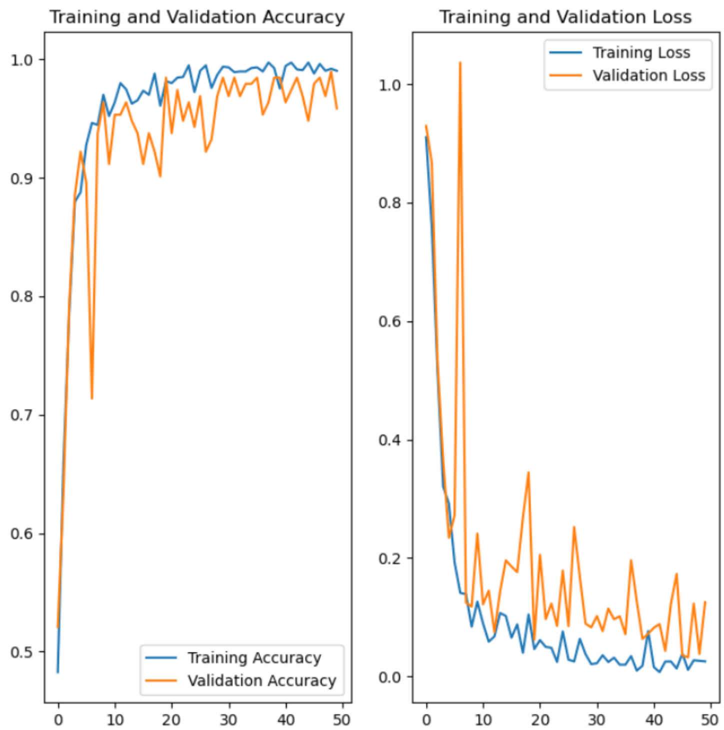


Figure 4.2: Best model's training & validation accuracy vs epoch, loss vs epochs of potato plant's disease (Custom CNN)

4.3.2 Tomato Plant's Performance Evaluation

For the classification of tomato plant diseases, a meticulously curated dataset comprising 13,226 images of tomato plant leaves was utilized. The dataset includes 2,127 images of bacterial spot, 1,000 of early blight, 1,592 healthy leaves, 1,909 of late blight, 952 of leaf mold, 373 of mosaic virus, 1,771 of Septoria leaf spot, 1,676 of spider mites, 1,404 of target spot, and 422 of yellow leaf curl virus. From Table 4.3, the Xception model was selected as the best performer for tomato disease detection due to its depthwise separable convolutions, which enhance efficiency and maintain high accuracy. The custom CNN model, almost parallel to Xception in performance, excelled with its tailored architecture, offering flexibility for capturing subtle patterns specific to multi-class classification problems.

Table 4.3: Performance Evaluation for Tomato Plants on Various CNN Models

CNN Models	Accuracy	Precision	Recall	F1-score
Custom CNN	98%	0.9820	0.9750	0.9790
Hybrid Model 1	96.02%	0.9602	0.9600	0.9598
Hybrid Model 2	97.03%	0.9699	0.9710	0.9657
VGG16	98%	0.9800	0.9810	0.9800
VGG19	97%	0.9750	0.9710	0.9720
InceprionV3	98%	0.9780	0.9770	0.9800
MobileNet	96%	0.9620	0.9510	0.9560
DenseNet121	97%	0.9710	0.9670	0.9670
Xception	98%	0.9850	0.9800	0.9820

The confusion matrix shown in the Figure 4.3 depicts the performance of a classification model in diagnosing tomato leaf conditions. The matrix provides reliable predictions for ten classes. From the confusion matrix, actual class and predicted class have been found for detecting diseases on tomato plant leaves. By using this classes, accuracy, precision, f1-score and recall can be measured as shown in Table 4.3.

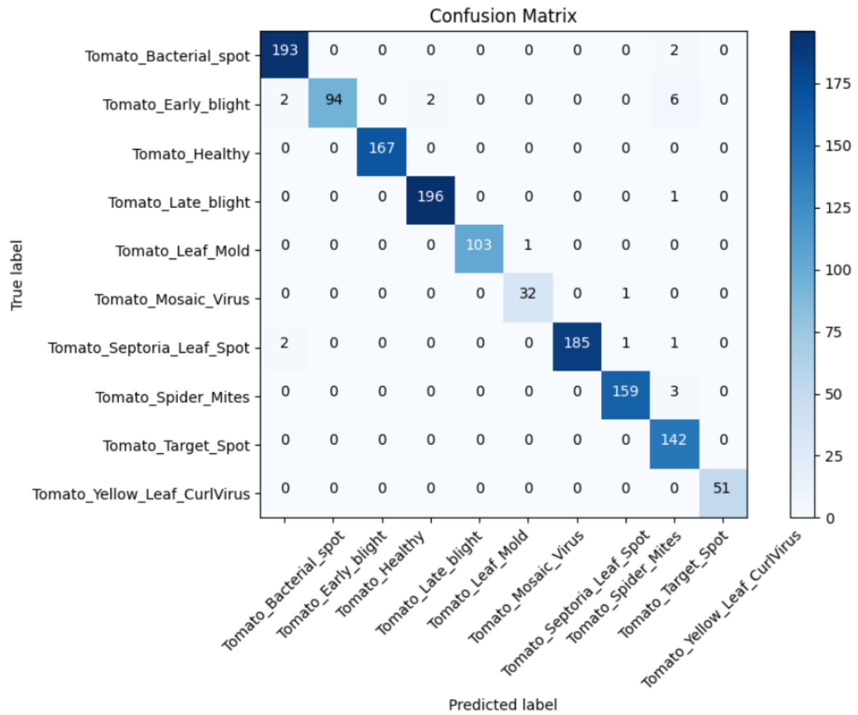


Figure 4.3: Best model’s confusion matrix of tomato leaves disease (Xception)

Figure 4.4 represents the best model's accuracy vs epochs curve and loss vs epochs curve of tomato plant leaves diseases. The best model for tomato plant leaves disease is Xception which is obtained after implementing nine CNN models on it.

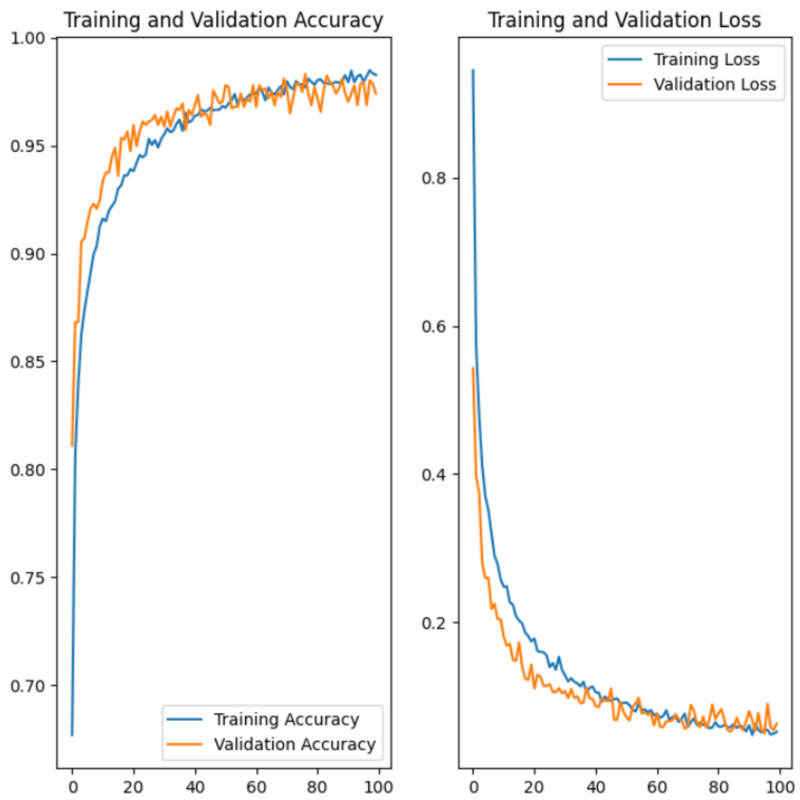


Figure 4.4: Best model's training & validation accuracy vs epoch, loss vs epochs of tomato plant's disease (Xception)

4.3.3 Pepper-bell Plant's Performance Evaluation

For pepper bell plant disease classification, a dataset of 2,475 images across 2 classes was used, as shown in Table 3.1. This includes 997 images of bacterial spot and 1,478 healthy leaves. From Table 4.4, the custom CNN model was selected as the best performer. The choice of custom CNN was driven by its ability to adapt specifically to the dataset's unique characteristics, ensuring precise feature extraction for subtle variations in leaf patterns. Additionally, its flexibility in layer configuration allowed optimization for better accuracy, while its efficient training process minimized computational requirements, making it well-suited for detecting pepper bell plant diseases.

Table 4.4: Performance Evaluation for Pepper-bell Plants on Various CNN Models

CNN Models	Accuracy	Precision	Recall	F1-score
Custom CNN	100%	1.0000	1.0000	1.0000
Hybrid Model 1	99.60%	0.9960	0.9960	0.9956
Hybrid Model 2	97.57%	0.9757	0.9757	0.9760
VGG16	98%	0.9750	0.9700	0.9700
VGG19	95%	0.9600	0.9450	0.9500
InceprionV3	99%	0.9850	0.9900	0.9850
MobileNet	100%	1.0000	1.0000	1.0000
DenseNet121	92%	0.9200	0.9200	0.9150
Xception	93%	0.9400	0.9300	0.9300

The confusion matrix shown in the Figure 4.5 depicts the performance of a classification model in diagnosing pepper-bell leaf conditions. The matrix provides reliable predictions for 2 classes. From the confusion matrix, actual class and predicted class have been found for detecting diseases on pepper-bell plant leaves. By using this classes, accuracy, precision, f1-score and recall can be measured shown in Table 4.4.

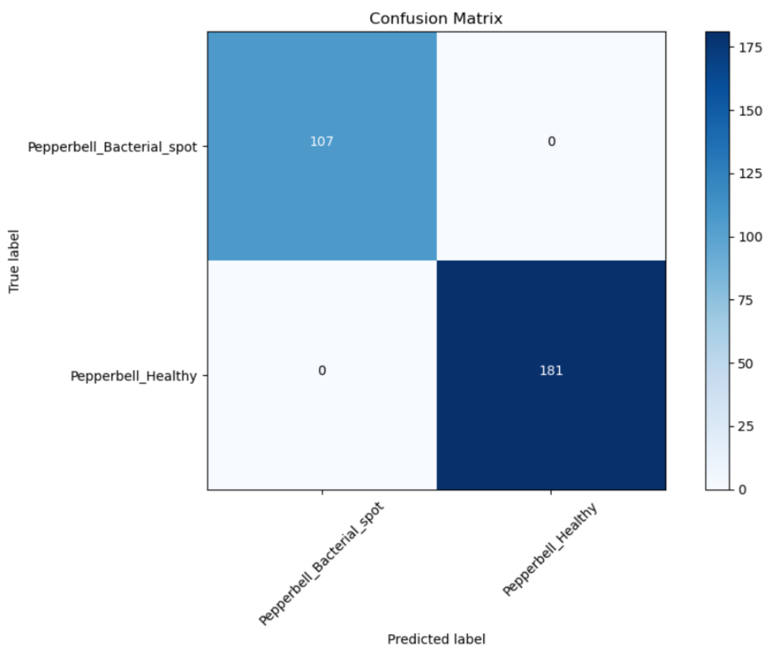


Figure 4.5: Best model’s confusion matrix of pepper-bell leaves (Custom CNN)

Figure 4.6 represents the best model's accuracy vs epochs curve and loss vs epochs curve of pepper-bell plant leaves diseases. The best model for pepper-bell plant leaves disease is custom CNN which is obtained after implementing nine CNN models on it.

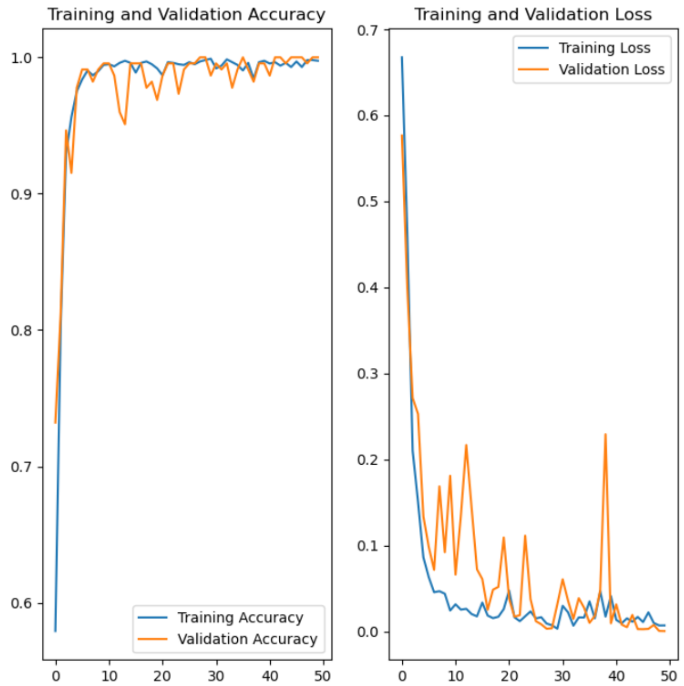


Figure 4.6: Best model's training & validation accuracy vs epoch, loss vs epochs of pepper-bell plant's disease (Custom CNN)

4.3.4 Apple Plant's Performance Evaluation

For apple leaf disease classification, a dataset of 2,536 images across 4 classes was utilized, as shown in Table 3.1. This includes 496 images of black rot, 220 of cedar rust, 1,316 healthy leaves, and 504 showing apple scab symptoms. From Table 4.5, the Xception model was selected as the best performer due to its deep architecture and efficient feature extraction capabilities. While the custom CNN model showed good performance, it struggled with blurrish dataset images, which hindered accurate pattern detection. Additionally, the custom model's limited adaptability made it less robust against variations in leaf textures and lighting conditions. Interestingly, MobileNet also delivered comparable results in some cases for apple plant classification due to its efficient architecture and lightweight design, which maintained a balance between speed and accuracy.

Table 4.5: Performance Evaluation for Apple Plants on Various CNN Models

CNN Models	Accuracy	Precision	Recall	F1-score
Custom CNN	84%	0.8400	0.8125	0.8225
Hybrid Model 1	99.80%	0.9980	0.9980	0.9976
Hybrid Model 2	99.60%	0.9960	0.9960	0.9957
VGG16	97%	0.9800	0.9500	0.9625
VGG19	98%	0.9825	0.9600	0.9725
InceprionV3	98%	0.9900	0.9700	0.9800
MobileNet	100%	0.9975	0.9925	0.9975
DenseNet121	95%	0.9625	0.9325	0.9425
Xception	100%	1.0000	1.0000	1.0000

The confusion matrix shown in the Figure 4.7 depicts the performance of a classification model in diagnosing apple leaf conditions. The matrix provides reliable predictions for 4 classes. From the confusion matrix, actual class and predicted class have been found for detecting diseases on apple plant leaves. By using this classes, accuracy, precision, f1-score and recall can be measured shown in Table 4.5.

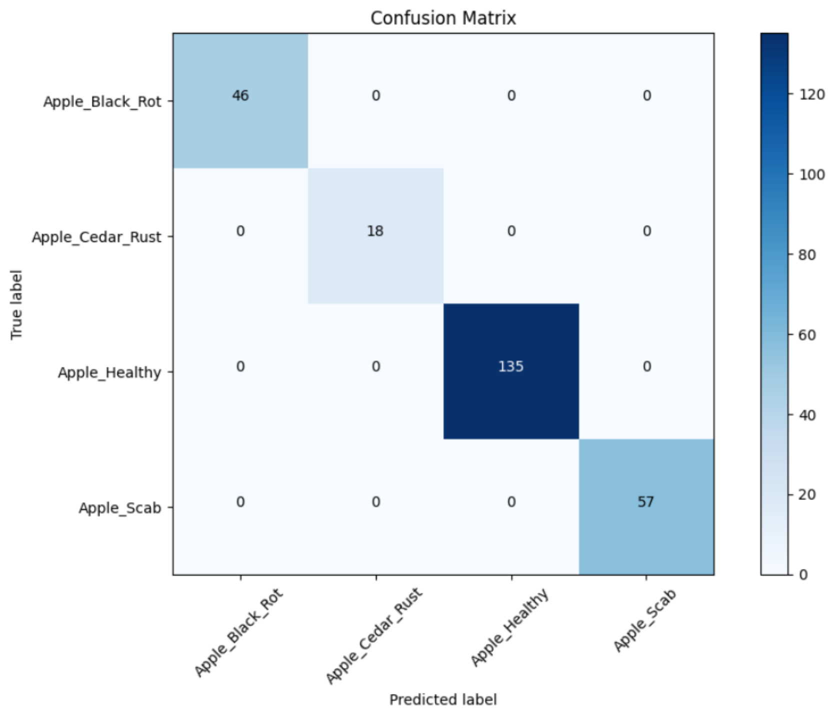


Figure 4.7: Best model's confusion matrix of apple leaves disease (Xception)

Figure 4.8 represents the best model's accuracy vs epochs curve and loss vs epochs curve of apple plant leaves diseases. The best model for apple plant leaves disease is Xception which is obtained after implementing nine CNN models on it. From this figure, it can be concluded that accuracy increases over epoch and loss simultaneously decreases over epochs.

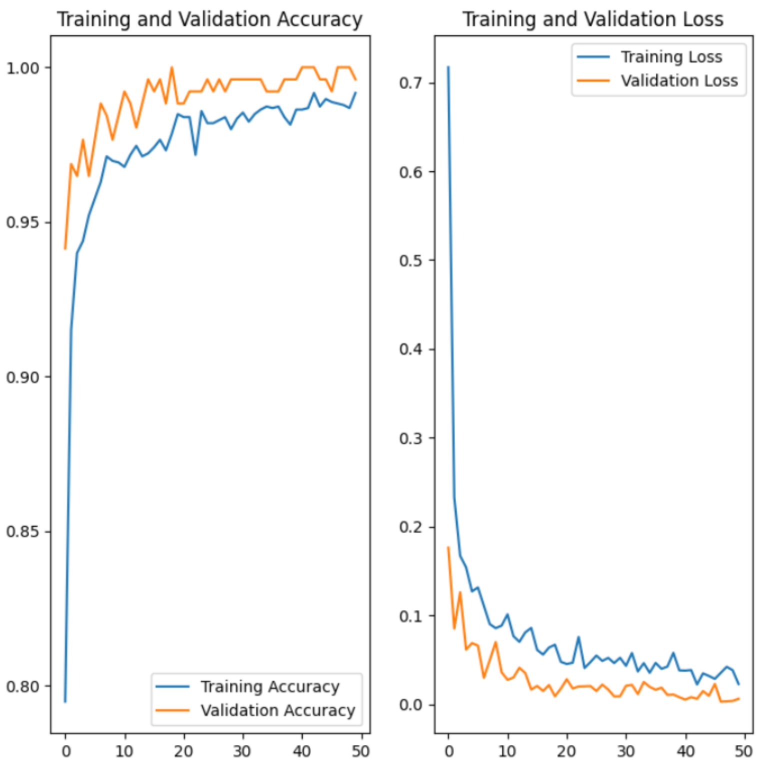


Figure 4.8: Best model's training & validation accuracy vs epoch, loss vs epochs of apple plant's disease (Xception)

4.3.5 Corn Plant's Performance Evaluation

For the classification of corn leaf diseases, a meticulously curated dataset comprising 3,080 images of corn leaves was employed, featuring 4 distinct classes as outlined in Table 3.1**.** Specifically, this dataset includes 410 images of corn leaves afflicted by cercospora leaf spot, 953 images exhibiting common rust, 929 images of healthy corn leaves, and 788 images displaying symptoms of northern leaf blight. From the analysis of Table 4.6, the custom CNN model is selected as the best model for future prediction in real time.

Table 4.6: Performance Evaluation for Corn Plants on Various CNN Models

CNN Models	Accuracy	Precision	Recall	F1-score
Custom CNN	98%	0.9675	0.9700	0.9700
Hybrid Model 1	96.74%	0.9674	0.9674	0.9670
Hybrid Model 2	95.28%	0.9528	0.9528	0.9525
VGG16	96%	0.9450	0.9450	0.9450
VGG19	95%	0.9300	0.9325	0.9325
InceprionV3	92%	0.8950	0.8900	0.8925
MobileNet	95%	0.9300	0.9325	0.9325
DenseNet121	94%	0.9225	0.9250	0.9250
Xception	97%	0.9525	0.9675	0.9600

The confusion matrix shown in the Figure 4.9 depicts the performance of a classification model in diagnosing corn leaf diseases. The matrix provides reliable predictions for 4 classes. From the confusion matrix, actual class and predicted class have been found for detecting diseases on corn plant leaves. By using this classes, accuracy, precision, f1-score and recall can be measured shown in Table 4.6.

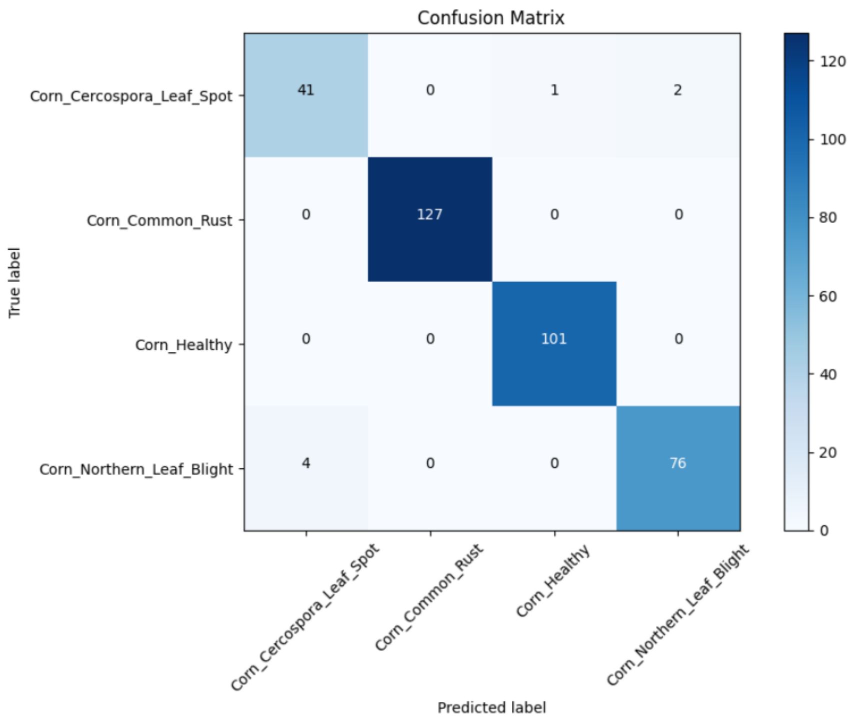


Figure 4.9: Best model’s confusion matrix of corn leaves disease (Custom CNN)

Figure 4.10 represents the best model's accuracy vs epochs curve and loss vs epochs curve of corn plant leaves diseases. The best model for corn plant leaves disease is custom CNN. This model captures fine patterns and subtle textures. It also included specialized preprocessing to handle varying leaf orientations and partial occlusions, ensuring better feature extraction and detection accuracy.

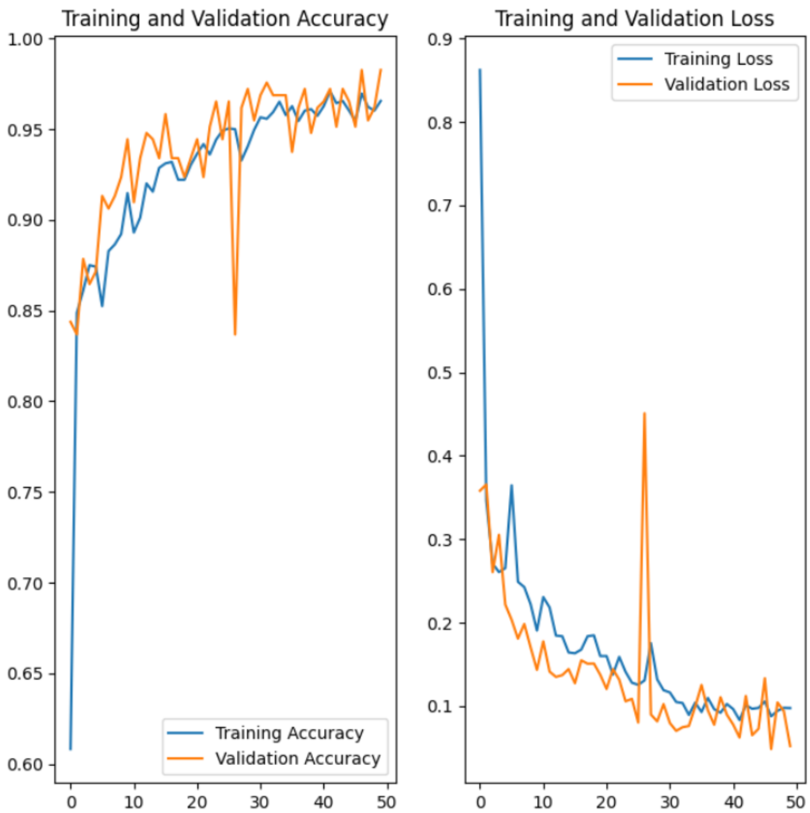


Figure 4.10: Best model's training & validation accuracy vs epoch, loss vs epochs of corn plant's disease (Custom CNN)

4.3.6 Grape Plant's Performance Evaluation

For the classification of grape leaf diseases, a meticulously curated dataset comprising 3,251 images of grape leaves was utilized, featuring 4 distinct classes as outlined in Table 3.1. Specifically, this dataset includes 944 images of grape leaves afflicted by black rot, 1,107 images exhibiting symptoms of esca black measles, 339 images of healthy grape leaves, and 861 images showing signs of isariopsis leaf spot. From Table 4.7, best model custom CNN was selected for grape plant leaves disease detection.

Table 4.7: Performance Evaluation for Grape Plants on Various CNN Models

CNN Models	Accuracy	Precision	Recall	F1-score
Custom CNN	99%	0.9950	0.9925	0.9925
Hybrid Model 1	98.46%	0.9846	0.9846	0.9845
Hybrid Model 2	98.30%	0.9830	0.9830	0.9832
VGG16	93%	0.9225	0.9300	0.9200
VGG19	99%	0.9925	0.9900	0.9900
InceprionV3	93%	0.9150	0.9425	0.9250
MobileNet	96%	0.9650	0.9650	0.9650
DenseNet121	99%	0.9900	0.9900	0.9900
Xception	98%	0.9875	0.9850	0.9850

The confusion matrix shown in the Figure 4.11 depicts the performance of a classification model in diagnosing grape leaf diseases. The matrix provides reliable predictions for 4 classes. From the confusion matrix, actual class and predicted class have been found for detecting diseases on grape plant leaves. By using this classes, accuracy, precision, f1-score and recall can be measured shown in Table 4.7.

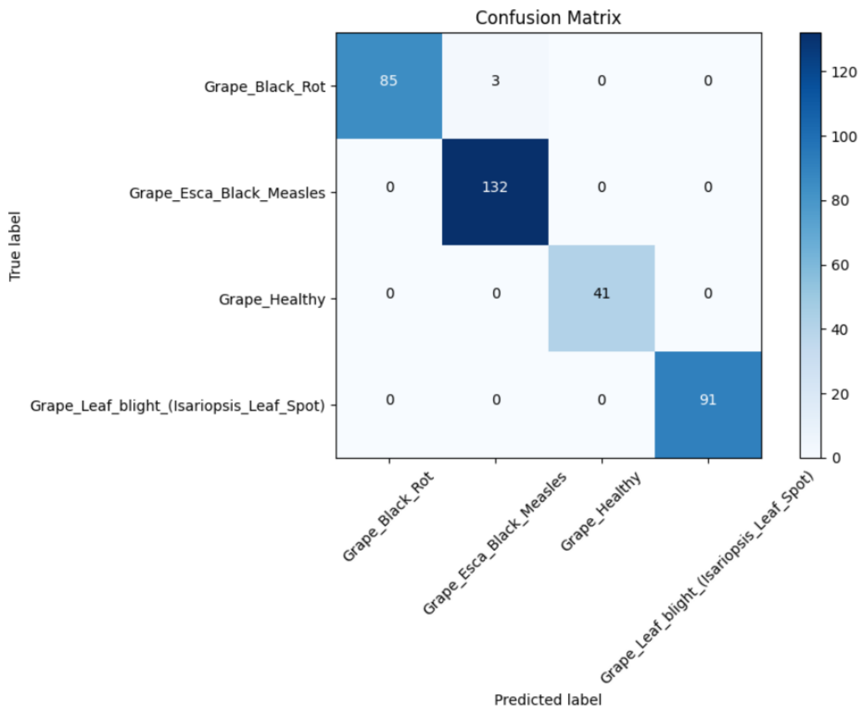


Figure 4.11: Best model's confusion matrix of grape leaves disease (Custom CNN)

Figure 4.12 represents the best model's accuracy vs epochs curve and loss vs epochs curve of grape plant leaves diseases. The best model for grape plant leaves disease is custom CNN which is obtained after implementing nine CNN models on it. From this figure, it can be concluded that accuracy increases over epoch and loss simultaneously decreases over epochs.

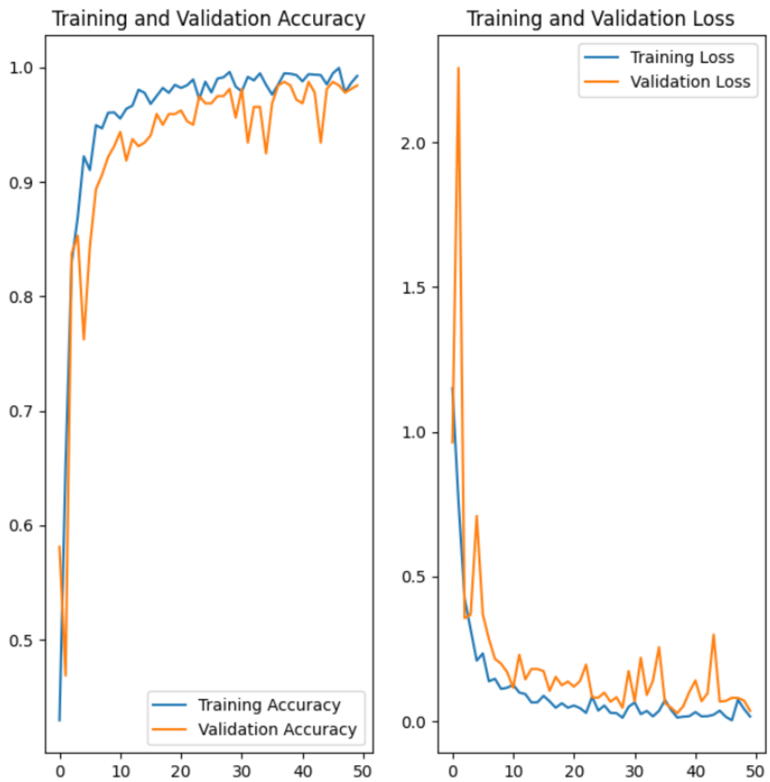


Figure 4.12: Best model's training & validation accuracy vs epoch, loss vs epochs of grape plant's disease (Custom CNN)

4.3.7 Peach Plant's Performance Evaluation

To classify peach leaf illnesses, a rigorously curated dataset of 2,126 photos of peach leaves was used, with two unique classifications as shown in Table 3.1. This dataset contains 1,838 photos of peach leaves with bacterial spots and 288 images of healthy peach leaves. This large dataset was critical in training the algorithm to effectively distinguish between healthy leaves and those with bacterial spot, ensuring precise and consistent disease diagnosis. Table 4.8 shows that the best model for detecting illness in peach plant leaves is VGG16 because of its small categorized dataset.

Table 4.8: Performance Evaluation for Peach Plants on Various CNN Models

CNN Models	Accuracy	Precision	Recall	F1-score
Custom CNN	100%	0.9728	0.9850	0.9900
Hybrid Model 1	99.76%	0.9976	0.9976	0.9972
Hybrid Model 2	99.76%	0.9976	0.9976	0.9969
VGG16	100%	0.9850	1.0000	0.9900
VGG19	99%	0.9800	0.9750	0.9750
InceprionV3	97%	0.9400	0.9450	0.9400
MobileNet	99%	0.9750	0.9950	0.9850
DenseNet121	99%	0.9750	0.9950	0.9850
Xception	99%	0.9450	0.9950	0.9650

The confusion matrix shown in the Figure 4.13 depicts the performance of a classification model in diagnosing peach leaf diseases. The matrix provides reliable predictions for 2 classes. Custom CNN performs almost similar. By using this classes, accuracy, precision, f1-score and recall can be measured shown in Table 4.8.

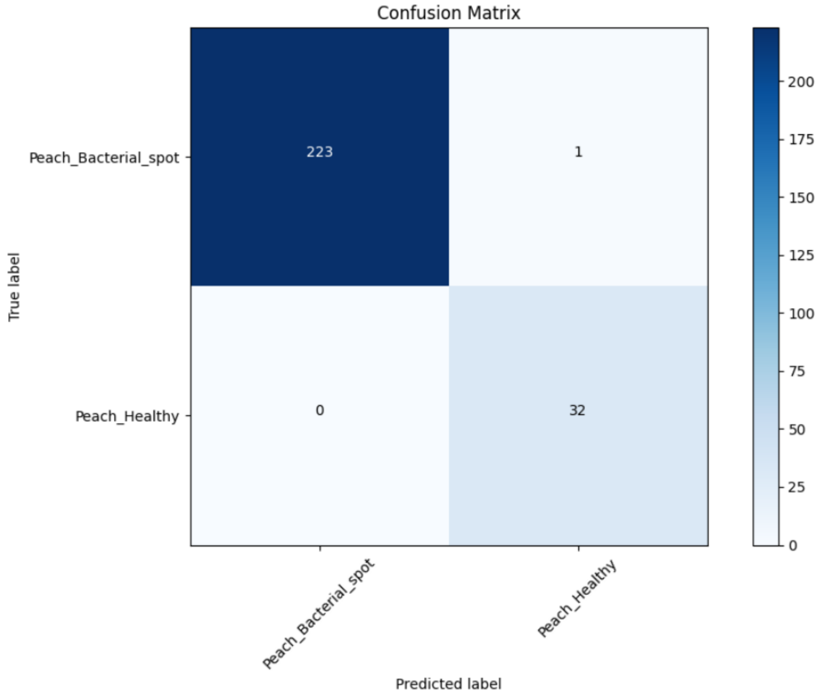


Figure 4.13: Best model’s confusion matrix of peach leaves disease (VGG16)

Figure 4.14 represents the best model's accuracy vs epochs curve and loss vs epochs curve of peach plant leaves diseases. The best model for peach plant leaves disease is VGG16.

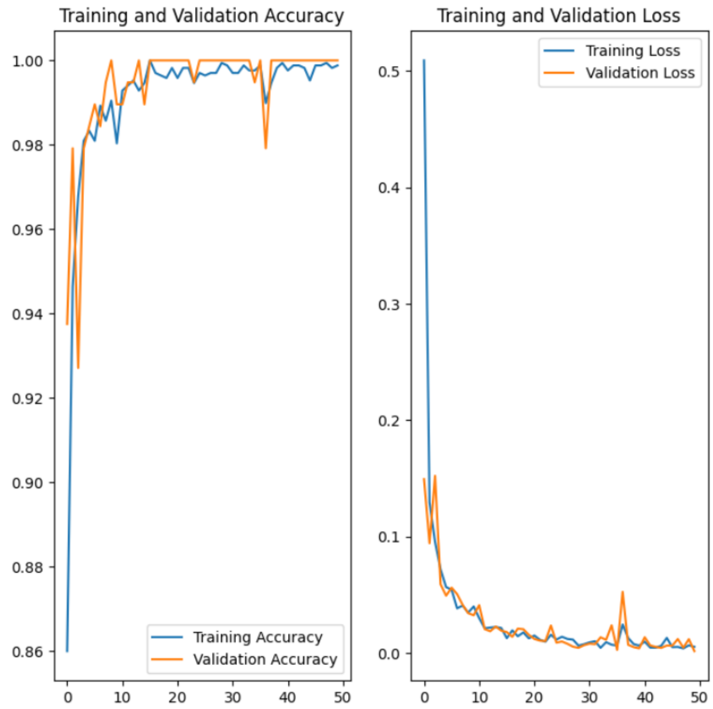


Figure 4.14: Best model's training & validation accuracy vs epoch, loss vs epochs of peach plant's disease (VGG16)

4.3.8 Rice Plant's Performance Evaluation

A rigorously curated collection of 2,100 rice leaf pictures was used for disease classification, with six unique classifications as shown in Table 3.1. This dataset includes 350 photos of rice leaves infected with bacterial leaf blight, brown spot, healthy leaves, leaf blast, leaf scald, and narrow brown spot. This comprehensive dataset played a crucial role in training the model to accurately differentiate between different diseases affecting rice leaves, ensuring robust and accurate disease detection capabilities. The DenseNet121 model was selected as the best performer from the analysis of Table 4.9 which enhances feature reuse, improves gradient flow, and reduces the vanishing gradient problem, enabling more efficient and robust learning for capturing subtle patterns in rice leaf textures.

Table 4.9: Performance Evaluation for Rice Plants on Various CNN Models

CNN Models	Accuracy	Precision	Recall	F1-score
Custom CNN	90%	0.9117	0.9033	0.9050
Hybrid Model 1	92.86%	0.9286	0.9286	0.9279
Hybrid Model 2	88.57%	0.8857	0.8857	0.8859
VGG16	91%	0.9200	0.9100	0.9083
VGG19	95%	0.9533	0.9433	0.9433
InceprionV3	91%	0.9150	0.9067	0.9100
MobileNet	93%	0.9200	0.9300	0.9217
DenseNet121	98%	0.9833	0.9850	0.9850
Xception	94%	0.9283	0.9267	0.9283

The confusion matrix shown in the Figure 4.15 depicts the performance of a classification model in diagnosing rice leaf diseases. The matrix provides reliable predictions for 6 classes. From the confusion matrix, actual class and predicted class have been found for detecting diseases on grape plant leaves. By using this classes, accuracy, precision, f1-score and recall can be measured shown in Table 4.9.

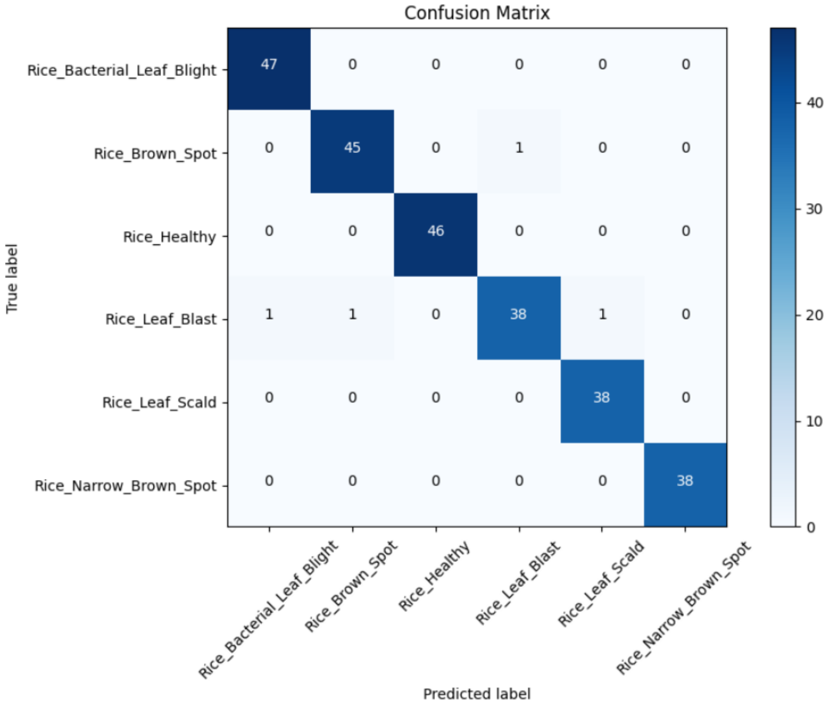


Figure 4.15: Best model's confusion matrix of rice leaves disease (DenseNet121)

Figure 4.16 represents the best model's accuracy vs epochs curve and loss vs epochs curve of rice plant leaves diseases. The best model for rice plant leaves disease is DenseNet121 which is obtained after implementing nine CNN models on it. From this figure, it can be concluded that accuracy increases over epoch and loss simultaneously decreases over epochs.

Figure 4.16: Best model's training & validation accuracy vs epoch, loss vs epochs of peach plant's disease (DenseNet121)

4.4 Solution of overfitting issue

The concern of overfitting with large models like VGG16, DenseNet121, and hybrid models was addressed in my research which was mitigated through several strategies to ensure robust performance across plant species and diseases. Key strategies included extensive data augmentation (e.g., random rotations, flipping, and scaling) to enhance dataset diversity, regularization techniques like L2 regularization and Dropout to reduce model complexity, and k-fold cross-validation to assess model performance across different data subsets. Additionally, batch normalization helped stabilize training, while transfer learning from pre-trained models like ImageNet improved generalization. Early stopping was also implemented to halt training when validation performance plateaued, preventing overfitting. Together, these techniques effectively controlled overfitting, enabling large models to generalize well across different plant disease categories, as demonstrated by the high accuracy achieved in both training and validation phases.

4.5 Backend API and database setup for user records

In this thesis, the backend service for the plant disease detection system has been developed utilizing FastAPI and MongoDB. The project architecture is structured to separate models, schemas, database connections, and CRUD operations, ensuring a clean and maintainable codebase. FastAPI is chosen for its performance and ease of use, while MongoDB provided a scalable NoSQL database solution.

To manage user authentication, a registration system is implemented requiring users to provide a username, email, and password during registration. Using custom utility functions imported from 'app.utils', passwords are hashed before storage to ensure security. The MongoDB driver has facilitated efficient database interactions. Pydantic models have been used to validate and serialize data, maintaining data integrity. This robust backend setup enabled secure storage of user credentials with hashed passwords, providing a reliable foundation for integrating the plant disease detection models with web and mobile applications. The database name has been chosen as Plant_manage. Inside it, a directory has been created whose name is '*USERS*' for keeping user records. Figure 4.17 represent the database of users.

```
_id: ObjectId('6689abc59c66d99c79405e33')
username: "naimur"
email: "naimur@gmail.com"
hashed_password: "$argon2id$v=19$m=65536,t=3,p=4$LiXEWMv5n1Pq/f8fw9h7zw$rdgMRwCNtDNBajyp…"

_id: ObjectId('66935fcf1bc07b61767a5fd6')
username: "raihan islam"
email: "raihanislam@gmail.com"
hashed_password: "$argon2id$v=19$m=65536,t=3,p=4$XetdqxUCYGwN4Vzr3TtnLA$8bJ0UecyyaG0Ognn…"
```

Figure 4.17: MongoDB database records of user registration

4.6 Web application development

For the development of the web application, the React framework has been utilized for its robust frontend capabilities. The web application was designed with a user-friendly interface featuring a drag-and-drop functionality, allowing users to easily upload images of plant leaves. Upon image upload, the application interfaces with the backend service to detect and identify the disease present in the leaf. Figure 4.18 represents the homepage graphical user interface of web application where eight several plants portion exists. If upload image button is clicked, then user can go to the individual plant's portion to upload their plant image.

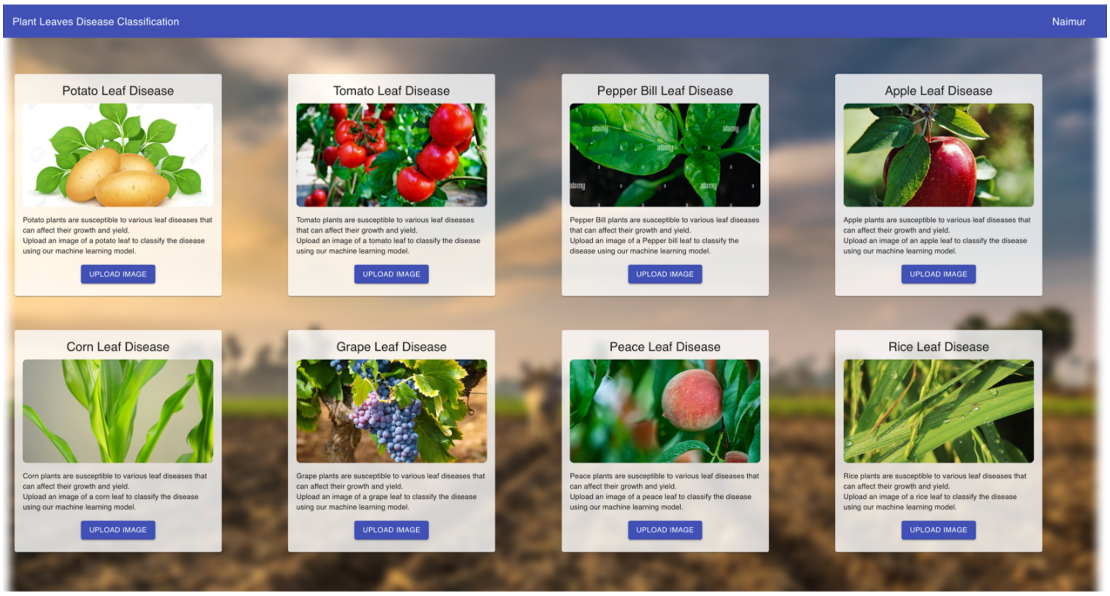


Figure 4.18: Homepage of web application using React framework

Inside every plant leaves portion, user can drag image or upload image of plant leaves. After uploading plant image, the application displays the disease name along with the confidence level of the detection, providing users with an accurate assessment. Additionally, it offers detailed instructions for disease management and treatment. This comprehensive approach not only aids in the rapid identification of plant diseases but also empowers users with actionable information to address the detected issues, thereby enhancing the overall utility of the web application in agricultural disease management. By early detection of plant leaves disease, farmers can easily take necessary steps to mitigate the problem which enhances the food security. Figure 4.19

represents the individual plant leaves portion to upload image, detect disease and also cure instruction.

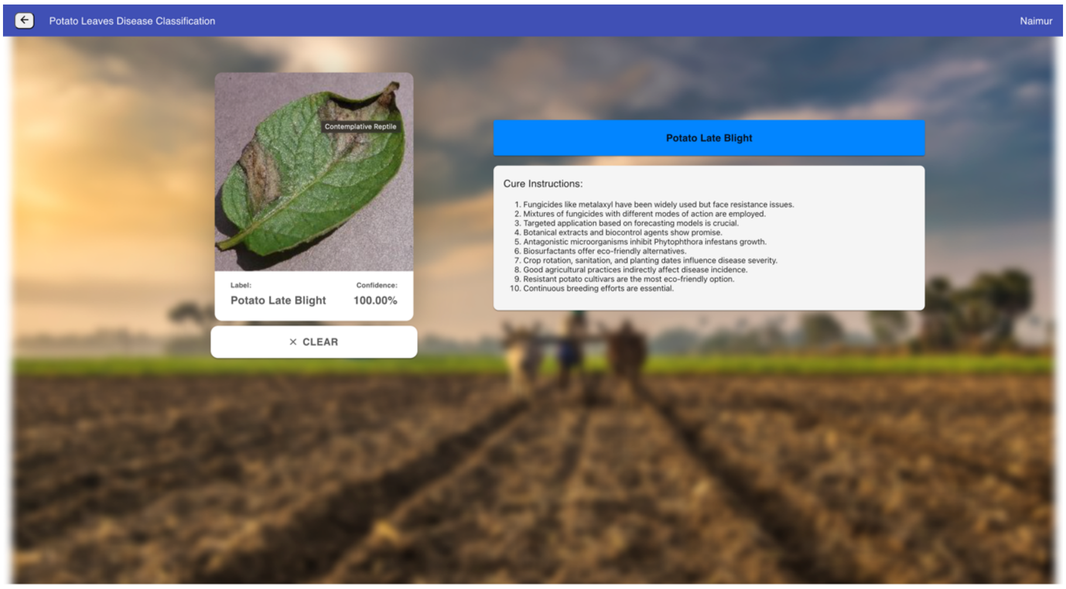


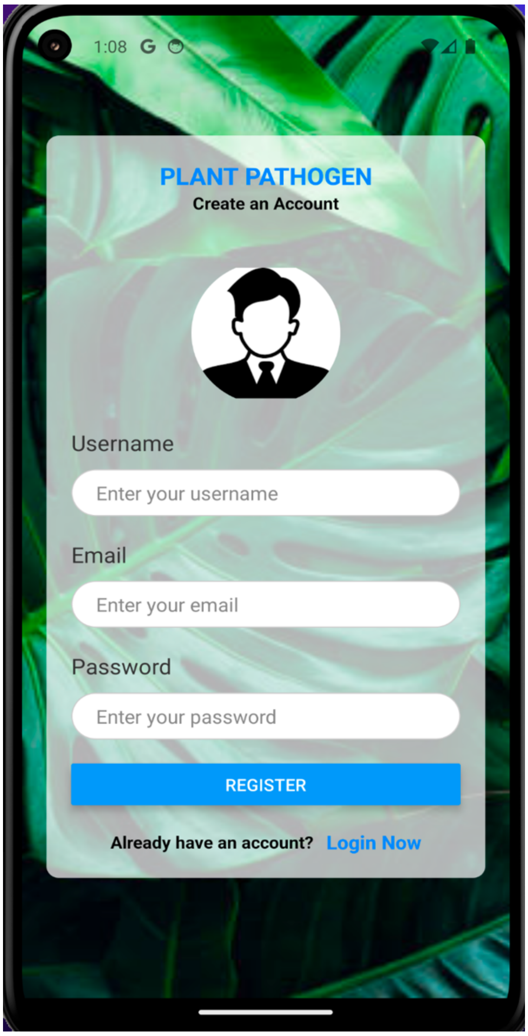
Figure 4.19: GUI of individual plant's image upload page in web application

4.7 Mobile application development

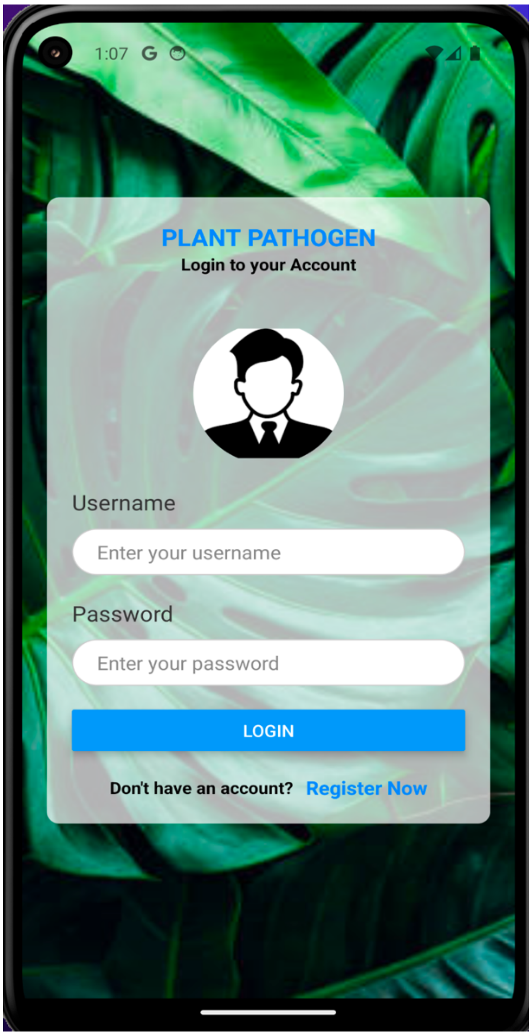
For the development of the mobile application, the React Native framework has been utilized which is a cross-platform JavaScript framework designed to enable seamless operation on both Android and iOS devices. This method offers broad accessibility and a uniform user experience across multiple platforms. User Registration and Login, which is essential to have secure authentication took place in our database in back-end side inside the mobile application. Users need to created account and have successfully logged in using application functionalities as shown in Figure 4.20.

Once logged in, users can upload images of plant leaves to identify the invading diseases. Leaf classification is performed when a user submits the image via backend API service. Following that, an additional API request is immediately launched to detect specific diseases affecting plant leaves. This two-step process ensures accurate and detailed disease detection.

The mobile application's graphical user interface (GUI) is designed for ease of use, providing an intuitive and streamlined user experience. The practical implementation of the mobile application, alongside the web application, is documented in this section. The GUI of mobile application contains registration page, login page, homepage, individual plant leaves disease details with video and written description, upload plant image page with disease detection and cure instruction also. When user uploads an image in mobile application, at first it detects actually which leaf it is and then it goes for that plant's disease detection process with confidence level.



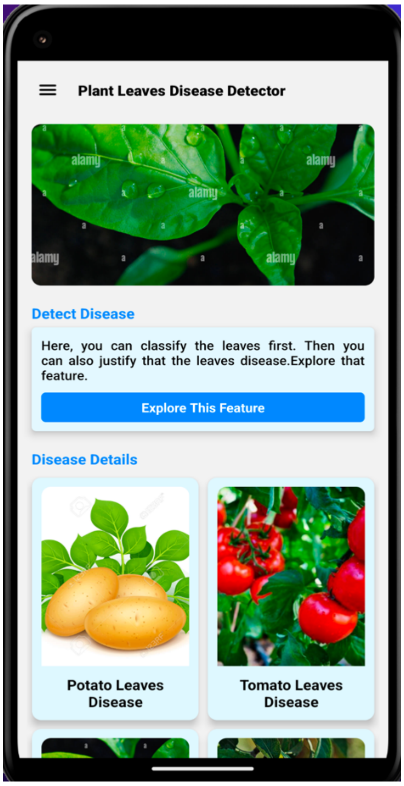
(a)Registration page



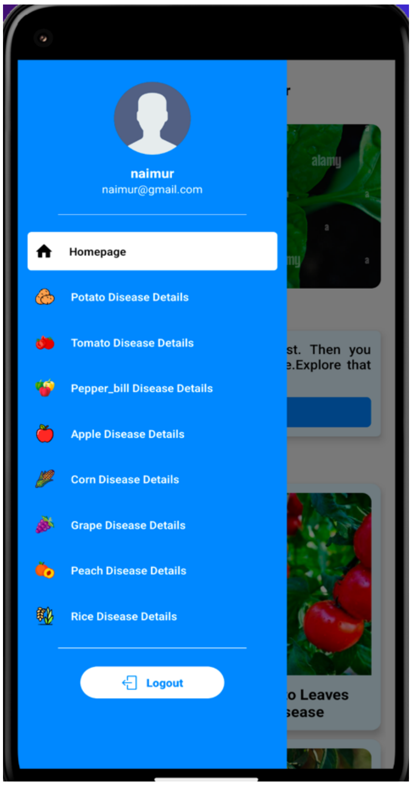
(b)Login page

Figure 4.20: Registration and login page of mobile application

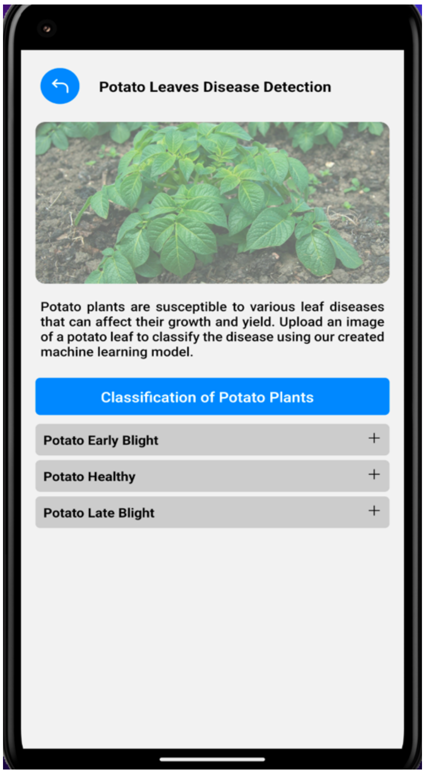
After the registration and login process in mobile application, it goes to home page which consists of a slider, individual plant leaves portion and also a drawer. Inside the individual plant leaves portion, there are several diseases. Every disease consists of symptoms, causes and cure instruction. Figure 4.21 represents the GUI of homepage.



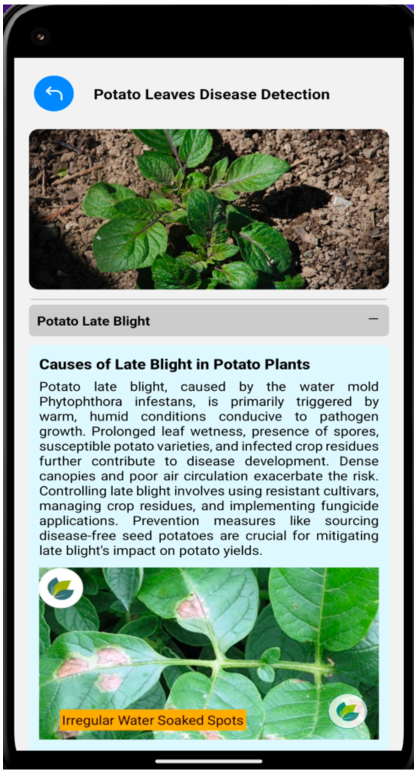
(a)Homepage viewer



(b)Homepage drawer



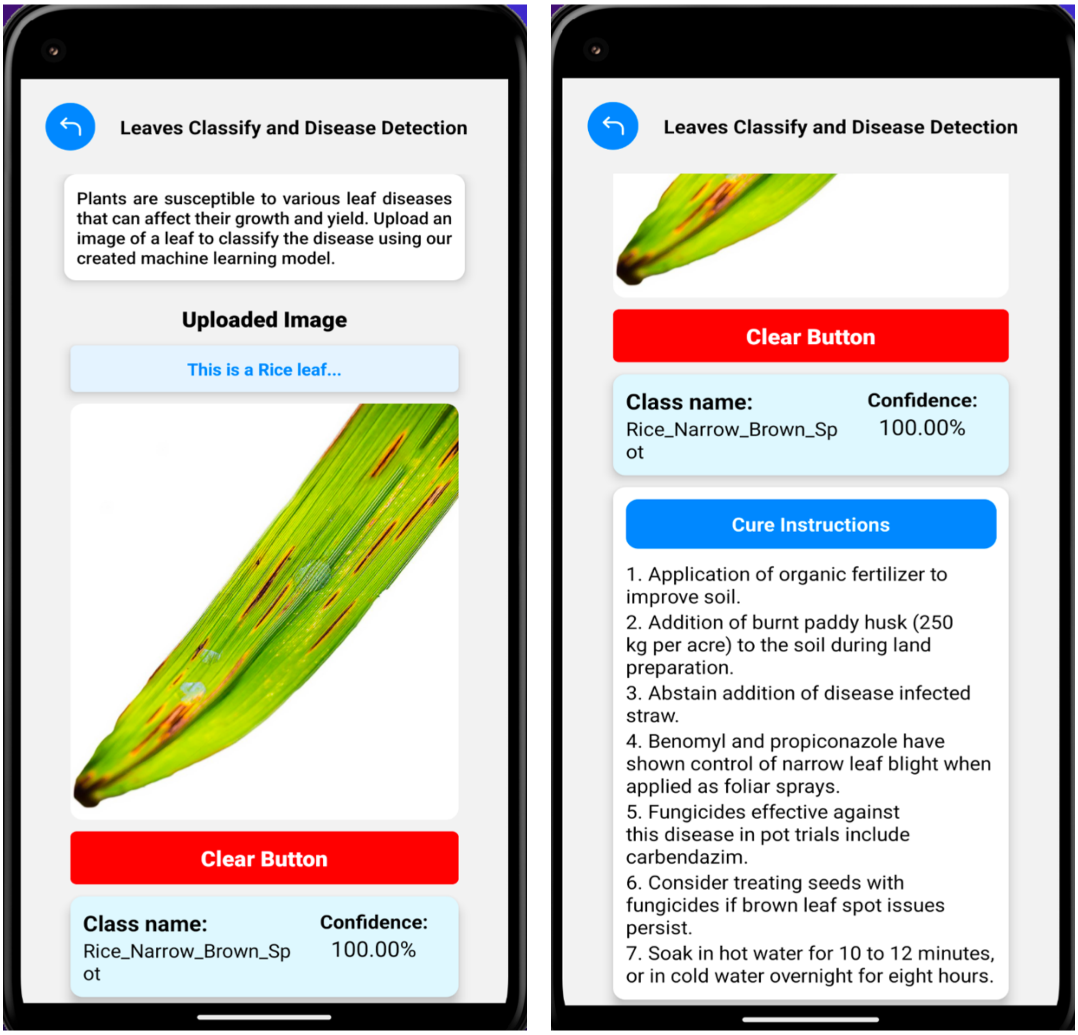
(c)Individual plant's portion



(d)Individual disease details

Figure 4.21: GUI of homepage in mobile application

In homepage, there is a section where user can upload image of plant's leaves or can take photo by mobile phone's camera. When the image is uploaded, then leaf classification model is triggered. Then FastAPI service is activated to get the actual classification of leaf and displayed it in mobile application. After the leaf classification, it triggers the second api which is used for disease detection of that particular leaf as shown in Figure 4.22.



(a)Upload image segment (b)Cure instruction showcasing

Figure 4.22: GUI of classification of diseases in mobile application

4.8 Optimal Hyperparameters Selection for CNN architecture

In deep learning, hyperparameters are crucial settings that influence how a model is trained and its ultimate performance. These include the learning rate, batch size, number of epochs, and network architecture details like the number of layers and nodes per layer. Unlike model parameters, hyperparameters are not learned during training; they must be set beforehand. Finding the optimal hyperparameters often involves a trial-and-error process, as their values significantly affect the model's accuracy, convergence speed, and ability to generalize. Table 4.10 represents the optimal hyperparameters for our proposed CNN architectures of various plant leaves according to our research.

Table 4.10: Hyperparameters selected for the proposed CNN architecture

Hyperparameters	**Optimal Hyperparameter**
Learning Rate	0.0001
Batch Size	32 (potato, pepper bell, apple, corn, grape, peach, rice) 16 (tomato)
Epochs	50 (potato, pepper bell, apple, corn, grape, peach) 100 (tomato, rice)
Number of Dense Layers	2
Number of Nodes in Dense Layers	64 (potato, pepper bell, corn, grape) 512 (tomato, apple, peach, rice)
Activation Function	SoftMax

4.9 Conclusion

This results and discussion section hold the full experimental results of various CNN models which are implemented on various individual plant leaves. This section consists of best model’s accuracy curve, loss curve, confusion matrix of individual plant leaves. Development phase like mobile application development and web application development are also a part of this section. Database maintaining for user records are also included in this section.

Chapter 05

Conclusion

5.1 Introduction

The objective of this thesis was to explore the effectiveness of deep learning techniques in detecting disease in various plant leaves. Given the critical role that plant leaves quality plays in agricultural sector, automating the detection of diseases is an essential step toward improving operational efficiency and reducing costs. This chapter aims to summarize the key findings of the research, discuss their significance, and propose potential avenues for future research. The conclusions presented here are based on a comprehensive analysis of experimental results using various deep learning models, including custom CNN, VGG16, VGG19, InceptionV3, MobileNet, DenseNet121, Xception and hybrid models.

5.2 Summary of research contribution

This study addresses the objectives outlined in the thesis by employing a comprehensive approach to plant disease detection using deep learning. Each objective is effectively met through the contributions presented below:

- **To build a CNN model for detecting diseases on various plant leaves:**
    - A robust dual-stage classification framework was developed, ensuring accurate detection of plant diseases across multiple species (potato, tomato, corn, etc.). Various deep learning models, such as custom CNN, VGG16, Xception, DenseNet121 etc. were employed to capture fine-grained patterns and effectively classify diseases on plant leaves. This ensures the objective of creating a CNN model for plant disease detection is successfully achieved.
- **To develop a real-time mobile application and web-based application for monitoring plant diseases:**

  - The study deployed the top-performing CNN models using FastAPI, a lightweight and efficient framework. This enabled the development of web and mobile applications that allow farmers and agricultural specialists to monitor plant health in real-time. The integration of frontend and backend technologies (React, React Native, MongoDB) ensures a user-friendly and scalable solution accessible across different devices and environments.

- **To evaluate the deployed CNN model's Performance:**

  - A comprehensive comparative analysis was conducted across various plant species and disease categories, evaluating multiple deep learning models (VGG16, InceptionV3, custom CNN, hybrid models, etc.). Performance metrics such as accuracy, precision, recall, and F1-score were analyzed to ensure optimal model selection. This rigorous evaluation confirmed the models' robustness and reliability in real-world scenarios, thus meeting the objective of assessing the deployed CNN model's performance comprehensively.

Through the development of a robust dual-stage classification framework, deployment on scalable applications with FastAPI, and a thorough evaluation of models across plant species, this study fully addresses all the objectives of building a CNN model, creating real-time applications, and rigorously evaluating the performance of deployed models.

5.3 Reason for Proposing Multiple Models in Our Research

In this research, multiple models were proposed for the following reasons:

- **Comprehensive Performance Evaluation**: Different models have unique strengths in feature extraction, computational efficiency, and generalizability. Evaluating several models helped to identify the most robust architecture for diverse plant species and diseases.

- **Optimization for Deployment**: Models like MobileNet are lightweight and ideal for mobile devices, while more complex models like DenseNet121 and InceptionV3 provide enhanced feature extraction. This allows adaptability for various platforms (web, mobile).
- **Generalizability and Robustness**: Using multiple models ensures the system performs well under different real-world conditions, such as lighting and image quality variations, improving overall robustness.
- **Facilitating Future Research**: The comparative analysis provides insights for future studies, offering a foundation for further refinement of model architectures or development of hybrid approaches.
- **Addressing Plant and Disease Variations**: Certain models excel for specific plants or diseases. By testing multiple models, we ensured higher accuracy and reliability across a range of agricultural applications.

This strategic approach ensures flexibility, scalability, and the highest possible accuracy in plant disease detection.

5.4 Future research directions

The findings of this study lay the groundwork for future improvements and research in plant leaves disease detection using deep learning. Future research directions are:

- **Dataset Expansion**: Future work will aim on expanding the dataset to cover a larger range of plant species and disease classes, hence increasing the system's adaptability and capacity to generalize across varied agricultural contexts.
- **Advanced Image Augmentation**: Additional image augmentation techniques will be investigated to increase the resilience and accuracy of deep learning models.
- **Synthetic Data Generation**: The use of generative adversarial networks (GANs) will be researched to produce synthetic data, overcoming the issues provided by the scarcity of real-world data and resulting in a bigger, more diversified training dataset.

- **Real-Time Video Analysis**: The system's capabilities will be expanded to include real-time video analysis for continuous crop monitoring, allowing for dynamic disease diagnosis and full management solutions.

5.5 Concluding remarks

The integration of deep learning into detection of diseases in various plant leaves represents a promising solution for agriculture looking to optimize the loss of growing diseased food. By automating the disease detection process, deep learning models reduce the reliance on human inspections, which can be error-prone and inefficient. This research demonstrates that leveraging models such as deep learning-based several architectures which can significantly improve the accuracy and efficiency of disease detection, leading to better quality control, reduced costs, and enhanced operational efficiency in agricultural sector. As technology continues to evolve, the integration of AI-driven solutions will undoubtedly play a critical role in the future of agricultural sector.

References

[1] Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shah, M. H. ur Rehman, and K. Javed, “An automated detection and classification of citrus plant diseases using image processing techniques: A review,” *Computers and Electronics in Agriculture*, vol. 153, pp. 12–32, Oct. 2018, doi: 10.1016/j.compag.2018.07.032.

[2] E.-C. Oerke, “Crop losses to pests,” *J. Agric. Sci.*, vol. 144, no. 1, pp. 31–43, Feb. 2006, doi:10.1017/S0021859605005708.

[3] K. Golhani, S. K. Balasundram, G. Vadamalai, and B. Pradhan, “A review of neural networks in plant disease detection using hyperspectral data,” *Information Processing in Agriculture*, vol. 5, no. 3, pp. 354–371, Sep. 2018, doi: 10.1016/j.inpa.2018.05.002.

[4] S. S. Chouhan, A. Kaul, U. P. Singh, and S. Jain, “Bacterial Foraging Optimization Based Radial Basis Function Neural Network (BRBFNN) for Identification and Classification of Plant Leaf Diseases: An Automatic Approach Towards Plant Pathology,” *IEEE Access*, vol. 6, pp. 8852–8863, 2018, doi: 10.1109/ACCESS.2018.2800685.

[5] T. Nazir, M. Iqbal, S. Jabbar, A. Hussain, and M. Albathan, "EfficientPNet—An Optimized and Efficient Deep Learning Approach for Classifying Disease of Potato Plant Leaves," *Agriculture*, vol. 13, 2023.

[6] R. Kumar, R. Tiwari, A. Jeevalatha, S. Siddappa, M. A. Shah, S. Sharma, V. Sagar, M. Kumar, and S. Chakrabarti, "Potato Apical Leaf Curl Disease: Current Status and Perspectives," *Journal of Plant Diseases and Protection*, vol. 128, pp. 897–911, 2021.

[7] M. Agarwal, S. Gupta, and K. K. Biswas, "Development of Efficient CNN Model for Tomato Crop Disease Identification," *Sustainable Computing: Informatics and Systems*, vol. 28, p. 100407, 2020.

[8] Y. Huang, C. Jones, H. Urbina, and S. Zhang, "First Report of Leaf Blight Caused by Curvularia aeria and C. senegalensis on Tomato," *Plant Disease*, vol. 107, 2023.

[9] E. Bogodukhova, V. V. Britvina, A. A. Mukhanova, V. N. Gelmiyarova, and A. C. Agostinho, "Creating a Digital Platform with Deep Neural Network for Detecting Plant Diseases," *BIO Web of Conferences*, vol. 71, 2023.

[10] M. N. Ahil, V. Vanitha, and N. Rajathi, "Apple and Grape Leaf Disease Classification Using MLP and CNN," in *Proc. 2021 Int. Conf. Advancements in Electrical, Electronics, Communication, Computing, and Automation (ICAECA)*, pp. 1–4, 2021.

[11] X. I. Xu, X. Yang, L. Zhang, H. Cao, P. Cao, X. Wang, W. Xiang, and J. Zhao, "First Report of Limonomyces roseipellis Causing Leaf Blight on Maize," *Plant Disease*, vol. 105, 2021.

[12] J. Lai, T. Liu, B. Liu, W. Kuang, and S. Song, "First Report of Curvularia Plantarum Causing Leaf Spot on Sweet Potato," *Plant Disease*, vol. 105, 2021.

[13] N. Ullah, J. A. Khan, S. Almakdi, M. S. Alshehri, M. Al Qathrady, N. El-Rashidy, S. El-Sappagh, and F. Ali, "An Effective Approach for Plant Leaf Disease Classification Based on DeepPlantNet," *Frontiers in Plant Science*, vol. 14, 2023.

[14] S. V. Shetty, N. A. Hemantha, A. Devaganamath, R. Ananth, and A. M. Alur, "Detection and Classification of Multiple Plant Diseases Using CNN," in *Proc. 2023 Int. Conf. Applied Intelligence and Sustainable Computing (ICAISC)*, pp. 1–6, 2023.

[15] A. S. Doutoum and B. Tugrul, "A Review of Leaf Diseases Detection and Classification by Deep Learning," in *IEEE Access*, vol. 11, pp. 119219-119230, 2023, doi: 10.1109/ACCESS.2023.3326721.

[16] N. Revathy and A. Meenakshi, "Classification of Plant Leaf Diseases Identification Using Deep Learning," *2nd International Conference on Networking and Communications (ICNWC)*, Chennai, India, 2024, pp. 1-9, doi: 10.1109/ICNWC60771.2024.10537326.

[17] H. B. Mureşan, A. M. Coroiu and A. D. Călin, "Detecting Leaf Plant Diseases Using Deep Learning: A Review," *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 2020, pp. 1-6, doi: 10.23919/SoftCOM50211.2020.9238318.

[18]] G. Zhou, W. Zhang, A. Chen, M. He, and X. Ma, “Rapid Detection of Rice Disease Based on FCM- KM and Faster R-CNN Fusion,” *IEEE Access*, vol. 7, pp. 143190–143206, 2019, doi: 10.1109/ACCESS.2019.2943454.

[19] P. Sharma, P. Hans, and S. C. Gupta, “Classification of Plant Leaf Diseases Using Machine Learning and Image Preprocessing Techniques,” in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India: IEEE, Jan. 2020, pp. 480–484. doi: 10.1109/Confluence47617.2020.9057889.

[20] M. Sardogan, A. Tuncer, and Y. Ozen, “Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm,” in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo: IEEE, Sep. 2018, pp. 382–385. doi: 10.1109/UBMK.2018.8566635.

[21] M. M. Ozguven and K. Adem, “Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms,” *Physica A: Statistical Mechanics and its Applications*, vol. 535, p. 122537, Dec. 2019, doi: 10.1016/j.physa.2019.122537.

[22] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, “Identification of rice diseases using deep convolutional neural networks,” *Neurocomputing*, vol. 267, pp. 378–384, Dec. 2017, doi: 10.1016/j.neucom.2017.06.023.

[23] Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, “Basic Study of Automated Diagnosis of Viral Plant Diseases Using Convolutional Neural Networks,” in *Advances in Visual Computing*, vol. 9475, Springer International Publishing, 2015, pp. 638–645. doi: 10.1007/978-3-319-27863-6_59.

[24] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, “Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks,” *IEEE Access*, vol. 7, pp. 59069–59080, 2019, doi: 10.1109/ACCESS.2019.2914929.

[25] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, “Detection of potato diseases using image segmentation and multiclass support vector machine,” in *IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, Apr. 2017, pp. 1–4. doi: 10.1109/CCECE.2017.7946594.

[26] M. A. Iqbal and K. H. Talukder, “Detection of Potato Disease Using Image Segmentation and Machine Learning,” in *IEEE International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, Chennai, India, Aug. 2020, pp. 43–47. doi: 10.1109/WiSPNET48689.2020.9198563.

[27] N. Mangla, P. B. Raj, Soumya G. Hedge, and P. R, “Paddy Leaf Disease Detection Using Image Processing and Machine Learning,” *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, vol. 7, no. 2, pp. 97–99, Feb. 2019, doi: 10.17148/IJIREEICE.2019.7220.

[28] H. D. Rizqi and A. S. Purnomo, “The ability of brown-rot fungus Daedalea dickinsii to decolorize and transform methylene blue dye,” *World J Microbiol Biotechnol*, vol. 33, no. 5, p. 92, May 2017, doi: 10.1007/s11274-017-2256-z.

[29] A. M., M. Zekiwos, and A. Bruck, “Deep Learning-Based Image Processing for Cotton Leaf Disease and Pest Diagnosis,” *Journal of Electrical and Computer Engineering*, vol. 2021, pp. 1–10, Jun. 2021, doi: 10.1155/2021/9981437.

[30] J. Andrew, D. E. Popescu, M. K. Chowdary, and J. Hemanth, “Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications,” *Agronomy*, vol. 12, no. 10, p. 2395, Oct. 2022, doi: 10.3390/agronomy12102395.

[31] M. Islam, M. Adil, M. A. Talukder, M. K. Ahamed, M. A. Uddin, M. Hasan, S. Sharmin, M. Rahman, and S. Debnath, "DeepCrop: Deep learning-based crop disease

prediction with web application," *Journal of Agriculture and Food Research*, vol. 14, p. 100764, 2023, doi: 10.1016/j.jafr.2023.100764.

[32] O. Kulkarni, "Crop Disease Detection Using Deep Learning," in *Proceedings of the 2018 International Conference on Current Trends toward Converging Technologies (ICCTCT)*, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697390.

[33] M. Chohan, A. Khan, R. Chohan, S. Katper, and M. Mahar, "Plant Disease Detection using Deep Learning," *International Journal of Recent Technology and Engineering*, vol. 9, pp. 909-914, 2020, doi: 10.35940/ijrte.A2139.059120.

[34] S. P. Mohanty, D. P. Hughes, and M. Salathe, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, 2016, doi: 10.3389/fpls.2016.01419.

[35] S. Ramesh, R. Hebbar, M. Niveditha, R. Pooja, N. Shashank, and P. V. Vinod, "Plant disease detection using machine learning," in *2018 International Conference on Design Innovations for 3Cs: Compute Communicate Control (ICDI3C)*, pp. 41–45, 2018, doi: 10.1109/ICDI3C.2018.00017.

[36] M. Jung, J. Song, A.-Y. Shin, B. Choi, S. Go, S.-Y. Kwon, J. Park, S. Park, and Y.-M. Kim, "Construction of deep learning-based disease detection model in plants," *Scientific Reports*, vol. 13, 2023, doi: 10.1038/s41598-023-34549-2

[37] S. Sannakki, V. Rajpurohit, V. Nargund, and P. Kulkarni, "Diagnosis and classification of grape leaf diseases using neural networks," in *4th International Conference on Computing, Communications and Networking Technologies (ICCCNT 2013)*, 2013, pp. 1-5, doi: 10.1109/ICCCNT.2013.6726616.

[38] M. Jhuria, A. Kumar and R. Borse, "Image processing for smart farming: Detection of disease and fruit grading," *IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Shimla, India, 2013, pp. 521-526, doi: 10.1109/ICIIP.2013.6707647.

[39] K. Wang, S. Zhang, Z. Wang, Z. Liu, and F. Yang, "Mobile smart device-based vegetable disease and insect pest recognition method," *Intelligent Automation & Soft Computing*, vol. 19, 2013, doi: 10.1080/10798587.2013.823783.

[40] K. Ahmed, T. Shahidi, S. I. Alam, and S. Momen, "Rice Leaf Disease Detection Using Machine Learning Techniques," *International Conference on Sustainable Technologies for Industry 4.0 (STI),* Dhaka, Bangladesh, 2019, pp. 1-5, doi: 10.1109/STI47673.2019.9068096.

[41] A. Dixit and S. Nema, "Wheat leaf disease detection using machine learning method-a review," *International Journal of Computer Science and Mobile Computing*, vol. 7, no. 5, pp. 124-129, 2018.

[42] K. Panigrahi, H. Das, A. Sahoo, and S. Moharana, "Maize Leaf Disease Detection and Classification Using Machine Learning Algorithms," in *Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications*, 2020, pp. 66. doi: 10.1007/978-981-15-2414-1_66.

[43] D. M. Phung and J. T. Rhee, "A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets," *Applied Sciences*, vol. 9, no. 21, p. 4500, 2019, doi: 10.3390/app9214500.

[44] K. Shivaprasad and A. Wadhawan, "Deep Learning-based Plant Leaf Disease Detection," in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2023, pp. 360-365. doi: 10.1109/ICICCS56967.2023.10142857.

[45] M. Rana, T. A. Tithy, N. R. Mamun, and H. K. Sharker, "Plant Leaf Diseases Identification in Deep Learning," *Computer Science & Engineering: An International Journal*, vol. 12, no. 5, pp. 1-10, 2022. doi: 10.5121/cseij.2022.12501.

[46] S. Pawar, S. Shedge, N. Panigrahi, A. P. Jyoti, P. Thorave, and S. Sayyad, "Leaf disease detection of multiple plants using deep learning," in *2022 International*

Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), vol. 1, pp. 241–245, 2022.

[47] G. Singh and K. K. Yogi, "Performance evaluation of plant leaf disease detection using deep learning models," *Archives of Phytopathology and Plant Protection*, vol. 56, no. 3, pp. 209-233, 2023. doi: 10.1080/03235408.2023.2183792.

[48] J. A. Pandian, V. D. Kumar, O. Geman, M. Hnatiuc, M. Arif, and K. Kanchanadevi, "Plant disease detection using deep convolutional neural network," *Applied Sciences*, vol. 12, no. 14, p. 6982, 2022.

[49] B. S. Hamed, M. M. Hussein, and A. M. S. Mousa, "Plant Disease Detection Using Deep Learning," *International Journal of Intelligent Systems and Applications*, vol. 15, no. 6, pp. 32-42, 2023. doi: 10.5815/ijisa.2023.06.04.

[50] Y. Peng and Y. Wang, "Leaf disease image retrieval with object detection and deep metric learning," *Frontiers in Plant Science*, vol. 13, 2022. doi: 10.3389/fpls.2022.963302.

[51] A. Yadav, U. Thakur, R. Saxena, V. Pal, V. Bhateja, and J. C.-W. Lin, "AFD-Net: Apple Foliar Disease multi-classification using deep learning on plant pathology dataset," *Plant and Soil*, vol. 477, no. 1, pp. 595–611, 2022.

[52] M. D. Nirmal, P. Jadhav, S. Pawar, and M. Kharde, "Deep Learning-based Disease Detection using Pomegranate Leaf Image," in *2022 Smart Technologies, Communication and Robotics (STCR)*, pp. 1–5, 2022.

[53] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, 2021.

[54] G. Kaur, N. Sharma, S. Malhotra, S. Devliyal, and R. Gupta, "Leveraging VGG16 Deep Learning for Photovoltaic Panel Fault Classification," in *2024 2nd World Conference on Communication & Computing (WCONF)*, 2024, pp. 1–5.

[55] P. Manjula and S. B. Priya, "An effective network intrusion detection and classification system for securing WSN using VGG-19 and hybrid deep neural network techniques," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 5, pp. 6419–6432, 2022.

[56] M. Ahmed, N. Afreen, M. Ahmed, M. Sameer, and J. Ahamed, "An inception V3 approach for malware classification using machine learning and transfer learning," *International Journal of Intelligent Networks*, vol. 4, pp. 11–18, 2023.

[57] S. M. M. Hossain, K. Deb, P. K. Dhar, and T. Koshiba, "Plant leaf disease recognition using depth-wise separable convolution-based models," *Symmetry*, vol. 13, no. 3, p. 511, 2021.

[58] T. Zhou, X. Ye, H. Lu, X. Zheng, S. Qiu, and Y. Liu, "Dense convolutional network and its application in medical image analysis," *BioMed Research International*, vol. 2022, no. 1, p. 2384830, 2022.

[59] H. Van Thanh, N. D. Quang, and K.-H. Jo, "Modern Deep Learning Models for Skin Disease Diagnosis," in *2023 International Workshop on Intelligent Systems (IWIS)*, pp. 1–6, 2023.

plant leaves disease detection using deep learning

ORIGINALITY REPORT

18%	12%	15%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	1%
2	de.overleaf.com Internet Source	1%
3	ebin.pub Internet Source	<1%
4	Hasan Abed Hasan, Abdullahi Abdu Ibrahim. "Hybrid Detection Techniques for Skin Cancer Images", 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2020 Publication	<1%
5	Sartini, Subiyanto, M M F Alim. "A sentiment analysis study for twitter using the various model of convolutional neural network", Journal of Physics: Conference Series, 2021 Publication	<1%
6	www.researchgate.net Internet Source	<1%
7	www.techscience.com	