Real-Time Detection & Classification of Potato and Guava Leaf Species & Diseases using Deep Learning Techniques

Ph.D Thesis

By

Javed Rashid

155-FBAS/PhDCS/F16

Supervisor

Dr. Imran Khan

Assistant Professor, CS&SE, IIU

Co-Supervisor

Dr. Ghulam Ali

Assistant Professor, DCS, University of Okara, Pakistan

Department of Computer Science & Software Engineering

International Islamic University, Islamabad

December 2022

Computer vision

Deep learning (Machine learning)

Pattern recognition system

Image processing Digital techniques

Leaves - Indian [illegible] - Data processing

Potatoes - Diseases [illegible] India food [illegible]

Guava -

A dissertation submitted to the
Department of Computer Science & Software Engineering,
International Islamic University, Islamabad
as a partial fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy in Computer Science.

INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD
FACULTY OF BASIC & APPLIED SCIENCES
DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING

Date: 28-08-2023

Final Approval

It is certified that we have read this thesis, entitled **"Real-Time Detection & Classification of Potato and Guava Leaf Species and Diseases using Deep Learning Techniques"** submitted by **Mr. Javed Rashid Registration No. 155-FBAS/PHDCS/F16**. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for the award of the degree of PhD in Computer Science.

Committee

External Examiners:
Dr. Basit Raza,
Associate Professor
COMSATS University, Islamabad

Dr. Asif Nawaz,
Assistant Professor
Barani Institute of Information Technology

Internal Examiner:
Dr. Syed Musharaf Ali,
Assistant Professor
Department of Computer Science & IT
FCIT, IIUI

Supervisor:
Dr. Imran Khan,
Assistant Professor
Department of Computer Science & IT
FCIT, IIUI

Co-Supervisor:
Dr. Ghulam Ali,
Assistant Professor
Department of Computer Science
University of Okara

Declaration

I hereby declare that this thesis, neither as a whole nor any part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Javed Rashid

Dedication

This thesis is dedicated to my beloved Prophet *Hazrat Muhammad (Peace be upon Him).*

Javed Rashid

Acknowledgments

This thesis and all my efforts are fruitful only due to ALLAH Almighty, the Most Merciful and Beneficent, Who gave me strength to complete this task to the best of my abilities and knowledge. I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of this thesis. This journey has been a challenging and rewarding one, and I could not have reached this milestone without the support, encouragement, and guidance of many individuals.

First and foremost, I would like to thank my thesis supervisor, Dr Imran Khan, and my co-supervisor, Dr Ghulam Ali, for their unwavering support, expertise, and mentorship throughout this research. Their valuable insights, patience, and dedication have been instrumental in shaping this work. I want to thank my mother and all my beloved family members, who constantly intercede *Dr. Imran Khan* and Co-Supervisor *Dr. Ghulam Ali.* I would also like to thank my wife, who has supported me patiently and firmly during the completion of my task. I am also sorry to my son *Muhammad Umar* because I did not properly time him in his precious years.

I would also like to acknowledge my brothers, friends, students and colleagues especially *Dr. Muhammad Shoaib Saleem, Dr. Riaz ul Amin, Dr. Hamood ur Rehman, Dr. Faheem Arshad, Dr. Rai Imtiaz Hussain, Dr. Rai Shahbaz Hussain, Mr. Sabookh Usman, Mr. Rizwan Saeed, Dr. Rizwan Ali, Mr. Nafees Ahmad Khan, Mr. Israr Ahmad, Mr. Noor Samand Nahra, Mr. Zahid Ali, Mr. Shahyar Shahid* and *all others.* All of them encouraged and provided logistic and technical help during this research.

Last but not the least, I would like to admit that I owe all my achievements to my truly, sincere and most loving parents and friends who mean the most to me, and whose prayers have always been a source of determination for me. I especially, thanks to my father Rana Muhammad Asad Khan Advocate and my elder brother Rana Perveze Akhtar for supporting me in entire life. I must admit without their sincere support, I was not able to complete this task.

Abstract

Plant disease detection is a complex area of study in agriculture since there are so many different species grown in so many different environments across the world. Plant disease detection involves various processes including classification and segmentation. Due to the striking resemblance in morphology among many plant species, it is challenging to accurately segment and classify not only the various species of plants but also the diseases plant. The accurate classification of plant species, make it possible to develop tools and techniques for early disease detection. There are various efforts made in order to achieve satisfactory performance for segmentation and classification processes, however, using machine learning in general and deep learning in specific faces several challenges. Considering the two types of the plants one growing with fruiting in the soil as plant and the other one fruiting above the soil as tree, this research defines its scope to consider a case from each of the mentioned classes. One such case is segmentation and classification of guava leaves. The guava plant has achieved viable significance in subtropics and tropics owing to its flexibility to climatic environments, soil conditions, and higher human consumption. It is cultivated in vast areas of Asian and Non-Asian countries, including Pakistan. The guava plant is vulnerable to diseases, specifically the leaves and fruit, which result in massive crop and profitability losses. The existing plant leaf disease detection techniques can detect only one disease from a leaf. However, a single leaf may contain symptoms of multiple diseases. The other case considered in this research is of the potato leaves. Variations in crop species, crop disease signs, and environmental conditions may contribute to the difficulty of early disease detection in potato leaves. Several machine learning methods have been employed for disease detection in potato leaves. There exist significant variance in geometry, color shape and other attributes of guava and potato leaves in Pakistan region with that of rest of the world whose dataset for the same is available. However, the current techniques are limited in identifying crop species and its illnesses in Pakistan. The limitations include size and scale of the dataset and the application of the models tested on images of a small subset of plant leaf regions. Robust methods are urgently needed for identifying plant leaf species, segmentation, and diseases that may be applied in practical settings. In this thesis, considering identification of the guava and potato species and diseases native to the Pakistani region as primary problem, We overcome the issue of limited size and scale of the region specific dataset and use deep learning methods to develop a system for real-time detection and classification of potato and guava leaf species and illnesses.

The state-of-the-art plant leaf species, disease identification, and plant segmentation techniques have been reviewed to identify the research gaps for contribution. Because all existing algorithms were trained on only the specific region dataset (PlantVillage dataset), they may be more accurate at recognizing guava and potato plant leaf species and illnesses in Pakistan. Variations in shape, variety, and environmental conditions contribute to regional differences in the prevalence of illnesses that affect potatoes and guavas. Because of this, there are many false positives when using these techniques to diagnose diseases in potatoes and guavas in the Pakistani region. To overcome this issue, three techniques have been proposed.

The first hybrid deep learning method classifies and segments guava, potato, and java plum leaf species. The proposed hybrid model consists of two novel methods. The first model using MobileNetV2-UNet architecture has been developed to segment the guava, potato, and java plum plant species. Plant species detection stacking ensemble deep learning model (PSD-SE-DLM) to detect the potato, java plum, and guava species as a second model. The proposed PSD-SE-DLM and MobileNetV2-UNet models were obtained with 99.84% and 96.74% accuracies, respectively.

The second method detects multiple diseases in guava leaves in real-time. A hybrid deep learning-based framework for real-time various disease identification from a single guava leaf in numerous phases is used for this. Guava Infected Patches Modified MobileNetV2 and U-Net (GIP-MU-NET) segment infected guava patches. The encoder is modified MobileNetv2, and the decoder is the U-Net model's up-sampling layers. Second, the Guava Leaf Segmentation Model (GLSM) divides healthy and diseased leaves. In the last step, the YOLOv5-based Guava Multiple Leaf illnesses Detection (GMLDD) model detects multiple illnesses from guava leaves. The proposed approach detected five fault classes: anthracnose, insect assault, nutrition shortage, wilt, and healthy. The GIP-MU-Net model has 92.41% accuracy, the GLSM 83.40%, and the proposed GMLDD approach 73.3% precision, 73.1% recall, 71.0% mAP@0.5, and 50.3 mAP@0.5:0.95 scores for all classes.

The third method detects potato leaf disease in Central Punjab, Pakistan. A multi-level deep learning model for potato leaf disease recognition is created. YOLOv5 image segmentation first removes potato leaves from the potato plant image. A convolutional neural network can detect early and late blight potato diseases from potato leaf images at the second level. The potato leaf disease dataset was 99.75% accurate using deep learning. The performance of the proposed methods was also evaluated on the PlantVillage dataset. Concerning accuracy and computational cost, the suggested method greatly outperforms state-of-the-art models.

In addition, to validate the robustness of the proposed methods in real scenarios, new datasets are developed. For plant leaf species identification and segmentation, two datasets (Plant Leaf Species

Dataset (PLSD) and Plant Leaf Species Segmentation Dataset (PLSSD)) for guava, java plum, and potato healthy and diseased leaves are created in Central Punjab, Pakistan. To validate and train the hybrid deep learning-based framework for the real-time detection of multiple diseases from a single guava leaf, two self-collected datasets (the Guava Patches Dataset and the Guava Leaf Diseases Dataset) are used. The proposed potato leaf disease detection model is trained and tested on a self-created potato leaf disease dataset. The potato leaf disease dataset contains 4062 images collected from the Central Punjab region of Pakistan. To further validate the robustness of the proposed methods, cross-dataset experiments are performed to analyze the applicability of unseen images.

Contents

List of Figures xiii

List of Tables xvii

1 Introduction 1

1.1 Motivation 6

1.2 Objectives 9

1.3 Relevance and Effectiveness 9

1.4 Contributions 10

1.5 Problem Statement 11

1.6 Proposed Solution 12

1.7 Thesis Organization 13

2 Literature Review 15

2.1 Plant Species Detection 15

2.1.1 Flower Species Recognition 15

2.1.2 Grapes Species Recognition 16

2.1.3 Multiple Species Recognition 16

2.2 Plant Leaf Diseases 19

2.2.1 Soybean Leaf Diseases 20

2.2.2 Cucumber Leaf Diseases 20

2.2.3 Peach Leaf Diseases 20

2.2.4 Rice Leaf Diseases 21

2.2.5 Wheat Leaf Diseases 22

2.2.6 Rice and Wheat Leaf Disorders 22

2.2.7 Corn/Maize Leaf Diseases 22
2.2.8 Apple Leaf Diseases 23
2.2.9 Banana Leaf Diseases 23
2.2.10 Apple and Banana Leaf Diseases 24
2.2.11 Tomato Leaf Diseases 24
2.2.12 Tomato and Potato Leaf Disorders 26
2.2.13 Tomato and Grape Leaf Diseases 26
2.2.14 Tea Leaf Diseases 26
2.2.15 Coffee Leaf Diseases 27
2.2.16 Coffee and Apple Leaf Diseases 27
2.2.17 Cotton Leaf Diseases 27
2.2.18 Citrus Leaf Diseases 28
2.2.19 Grapes Leaf Diseases 28
2.2.20 Grapes and Tomato Leaf Diseases 29
2.2.21 Grapes and Cucumber Leaf Diseases 29
2.2.22 Peanut Leaf Diseases 29
2.2.23 Olive Leaf Diseases 30
2.2.24 Mango Leaf Diseases 30
2.2.25 Kiwifruit Leaf Diseases 30
2.2.26 Cassava Leaf Diseases 31
2.2.27 Beans Leaf Diseases 32
2.2.28 Squash Leaf Diseases 32
2.2.29 Palm Leaf Diseases 33
2.2.30 Leaf Diseases of Multiple Crops 33
2.2.31 Potato Leaf Diseases 35
2.2.32 Guava Leaf Diseases 36
2.3 Plant Leaf Segmentation Techniques 39
2.4 Publicly Available Datasets 41
2.4.1 Potato Leaf Disease Datasets 41
2.4.2 Guava Leaf Disease Datasets 42
2.4.3 Plant Leaf Species Datasets 43
2.5 Convolutional Neural Network (CNN) Architecture 43
2.5.1 Feature Learning 44
2.5.2 Classification 50

2.5.3 Training and Optimization . . . 52

2.6 YOLOv5 Architecture . . . 53

2.7 Open Issues . . . 56

3 A Hybrid Deep Learning Approach to Classify the Plant Species 58

3.1 Overview . . . 58

3.2 The Proposed Methodology . . . 60

3.2.1 Dataset Preperation . . . 60

3.2.2 Image Pre-Processing . . . 63

3.2.3 Dataset Splitting . . . 63

3.2.4 The MobileNetV2-UNet Segmentation Model Architecture For Plant Leaf Species . . . 64

3.2.5 The Proposed Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*) Methodology . . . 65

3.2.6 Experimental Setup . . . 70

3.2.7 Evaluation Measures . . . 70

3.3 Results and Discussion . . . 72

3.3.1 The Performance Analysis of the MobileNetV2-UNet Plant Species Segmentation . . . 72

3.3.2 The Performance Analysis of the Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*) . . . 74

3.3.3 Comparison Analysis of the Proposed *PSD-SE-DLM* Model with Existing Techniques . . . 78

3.3.4 Comparison Analysis of the Proposed *PSD-SE-DLM* Model with State-of-the-Art Techniques . . . 78

3.4 Chapter Summary . . . 79

4 Real–Time Multiple Guava Leaf Disease Detection from a Single Leaf Using Hybrid Deep Learning Technique 81

4.1 Overview . . . 81

4.2 Proposed Work . . . 83

4.2.1 Data Preparation . . . 84

4.2.2 Image Preprocessing (Data Division) . . . 87

4.2.3 The Proposed Network . . . 87

4.2.4 Experimental Setup . . . 90

4.2.5 Evaluation Measures 90
4.3 Results and Discussion 91
4.3.1 Proposed Guava Infected Patches Modified MobileNetV2 and U-Net (*GIP-MU-NET*) Performance 91
4.3.2 Proposed Guava Leaf Segmentation Model (*GLSM*) Performance 96
4.3.3 Proposed Guava Multiple Leaf Diseases Detection (*GMLDD*) Model Performance 101
4.3.4 Comparison of Proposed *GMLDD* Model with Other *YOLO* Variants 109
4.4 Chapter Summary 109

5 Multi-Level Deep Learning Model for Potato Leaf Disease Recognition 112
5.1 Overview 112
5.2 Proposed Methodology 113
5.2.1 Dataset 115
5.2.2 Image Pre-Processing 117
5.2.3 Dataset Split 118
5.2.4 Potato Leaf Segmentation and Extraction Technique Using YOLOv5 . . . 119
5.2.5 Potato Leaf Disease Detection Convolutional Neural Network (*PDDCNN*) 120
5.2.6 Experimental Setup 122
5.3 Results and Discussion 122
5.3.1 Proposed *PDDCNN* Model Performance on *PLD* Dataset 123
5.3.2 The Proposed *PDDCNN* Model Performance on the PlantVillage Dataset . 133
5.3.3 Cross Dataset Performance 142
5.3.4 Accuracies Comparison of Proposed Method with State-of-the-Art Methods 143
5.3.5 Accuracies Comparison of Proposed Method with Existing Studies 143
5.4 Chapter Summary 144

6 Conclusions and Future Directions 146
6.1 Suggestions for Future Work 148

Appendix A Dataset Authentication, Acceptance Letter, Proof of Submission, & Research Statistics 150
A.1 Dataset Authentication Letters 150

Bibliography 155

List of Figures

1.1 (a) Potato Species of PlantVillage Dataset (b) Potato Leaf Species of Pakistani Environment (c) Guava Leaf Species of Pakistani Region (d) Plant Leaf Species of Bangladesh. 8
1.2 (a) Potato Leaf Diseases of PlantVillage Dataset (b) Potato Leaf Diseases of Pakistani Environment (c) Guava Leaf Diseases of Bangladesh (d) Plant Leaf Diseases of Pakistan Region. 8
1.3 The Proposed Methodology. 13

2.1 Convolutional Neural Network (CNN) Architecture. 44
2.2 Feature Hierarchy. 44
2.3 Feature Hierarchy. 46
2.4 Same Padding Process. 47
2.5 (a) ReLU, (b) Sigmoid, and (c) Hyperbolic Tangent. 48
2.6 Examples of Pooling Operations. 49
2.7 Fully Connected Layer. 51
2.8 Fully Connected Layer Flatten. 51
2.9 YOLOv5s Model Architecture. 54

3.1 Flowchart of the Proposed Hybrid Deep Learning Model. 60
3.2 (a) Guava Leaf (b) Potato Leaf (c) Java Plum Leaf Species. 61
3.3 (a) Guava Leaf (b) Guava Leaf Mask (c) Java Plum Leaf (d) Java Plum Mask (e) Potato Leaf (f) Potato Mask. 62
3.4 MobileNetV2-UNet Architecture. 65
3.5 Ensemble learning techniques 65
3.6 The Proposed Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*) 66

3.7 MobileNetV2 Architecture 67

3.8 InceptionV3 Architecture 68

3.9 The ResNet50 Architecture 69

3.10 The MobileNetV2-UNet Plant Leaf Segmentation Model Dice Score and Loss Graph. 73

3.11 The MobileNetV2-UNet Plant Leaf Segmentation Model Prediction on Test Set. . . 74

3.12 The Proposed *PSD-SE-DLM* Model Accuracy and Loss Graph 75

3.13 The Proposed *PSD-SE-DLM* Model Confusion Matrix on Test Set 76

3.14 The Proposed *PSD-SE-DLM* Model *ROC* Graph on Test Set 77

3.15 The Proposed *PSD-SE-DLM* Model's Comparison with State-of-the-Art Models. . 79

4.1 Diagram Illustrating the New Hybrid *DL* Approach. 84

4.2 (a) Infected Patches Images (b) Infected Patches Masks. 84

4.3 (a) Examples of Guava Leaf Images and (b) Examples of Guava Images Leaves' Masks. 85

4.4 Examples of Guava Leaf Multiple Diseases Images. 86

4.5 Architecture of the Proposed GIP-MU-Net Model. 88

4.6 Architecture of Proposed GLSM Method. 89

4.7 GIP-MU-NET Model Dice Score Graph using Data Augmentation. 92

4.8 The Proposed *GIP-MU-NET* Dice Score Comparison Graph on Test Set. 93

4.9 (a) Original Images, (b) Ground Truth, and (c) Infected Patches Predictions Applying Data Augmentation Techniques. 93

4.10 The Proposed *GIP-MU-NET* Dice Score Graph Without Data Augmentation. . . . 94

4.11 (a) Original Images, (b) Ground Truth, and (c) Infected Patches Predictions Without Data Augmentation Techniques. 95

4.12 GLSM Model Dice Score Graph Using Adam Optimizer. 96

4.13 (a) Original Image (b) Generated Mask (c) Predicted Mask of the *GLSM* Using *Adam* Optimizer. 97

4.14 The Proposed *GLSM* Dice Score Comparison Graph on Test Set. 98

4.15 *GLSM* Model Dice Accuracy Graph Using *SGD*. 99

4.16 (a) Original Image (b) Generated Mask (c) Predicted Mask of the *GLSM* Using *SGD*.100

4.17 Training and Validation Graph of the GMLDD Technique. 101

4.18 The Proposed GLMDD Model Predictions. 102

4.19 The Proposed GLMDD Model Field Level Predictions Under Different Illumination and Complex Backgrounds. (a) Original Images (b) Predictions. 103

4.20 The Proposed GMLDD Model Confusion Matrix. 105

4.21 (a) Precision Curve (P Curve) (b) Recall Curve (R Curve) (c) Precision Recall Curve (PR Curve) (d) F1 Curve. 106

4.22 The Proposed *GMLDD* Model Misclassification Examples. 108

5.1 The System Flow Diagram of the Proposed *PDDCNN* Technique. 114

5.2 The Block Diagram of the Proposed *PDDCNN* Technique. 115

5.3 Examples of Potato Leaf Images: (**a**) Early Blight, (**b**) Healthy and (**c**) Late Blight. 117

5.4 The Proposed Potato Leaf Segmentation and Extraction Technique Using YOLOv5 Model Architecture. 119

5.5 The Architecture of the Proposed PDDCNN Model. 120

5.6 Accuracies Graph of *PDDCNN* on *PLD* with Data Augmentation. 124

5.7 **a**) Accuracy Graph of *PDDCNN* on *PLD* using Data Augmentation. (**b**) Loss Graph of *PDDCNN* on *PLD* using Data Augmentation. 125

5.8 *PDDCNN* Precision, Recall and F1-Score on the *PLD* Dataset with Data Augmentation. 126

5.9 The Confusion Matrix of *PDDCNN* on *PLD* with Data Augmentation. 127

5.10 *ROC* Curve on the *PLD* Dataset with Augmentation Techniques. 128

5.11 Accuracies Graph of *PDDCNN* on *PLD* without Data Augmentation. 129

5.12 (**a**) Accuracy Graph of *PDDCNN* on *PLD* without Data Augmentation. (**b**) Loss Graph of *PDDCNN* on *PLD* without Data Augmentation. 130

5.13 *PDDCNN* Precision, Recall and F1-Score on the *PLD* Dataset without Data Augmentation. 131

5.14 The Confusion Matrix of *PDDCNN* on *PLD* without Using Data Augmentation. . . 132

5.15 *ROC* Curve on the *PLD* Dataset Without Applying Augmentation Techniques. . . . 132

5.16 Accuracies Graph of PDDCNN on PlantVillage with Data Augmentation. 134

5.17 (**a**) Accuracy Graph of PDDCNN on PlantVillage using Data Augmentation. (**b**) Loss Graph of PDDCNN on PlantVillage using Data Augmentation. 135

5.18 PDDCNN Precision, Recall and F1 Score on PlantVillage with Data Augmentation Techniques. 136

5.19 PDDCNN Confusion Matrix on PlantVillage with Data Augmentation Techniques. 137

5.20 PDDCNN ROC Curve on PlantVillage with Applying Data Augmentation. 138

5.21 Accuracies Graph of PDDCNN on PlantVillage without Applying Data Augmentation. 138

5.22 (**a**) Accuracy Graph of PDDCNN on PlantVillage without Data Augmentation. (**b**) Loss Graph of PDDCNN on PlantVillage without Data Augmentation. 139
5.23 PDDCNN Precision, Recall and F1 Score on PlantVillage without Data Augmentation Techniques. 140
5.24 PDDCNN Confusion Matrix on PlantVillage without Data Augmentation Techniques. 141
5.25 PDDCNN ROC Curve on PlantVillage without Applying Data Augmentation. . . . 141
5.26 PDDCNN Cross Dataset Classification Accuracies. 142

List of Tables

2.1 Summary of Plant Leaf Species Related Work. 18
2.2 Summary of Potato Leaf Diseases Related Work. 37
2.3 Summary of Guava Leaf Diseases Related Work. 38
2.4 Summary of Plant Leaf Segmentation Related Work. 41

3.1 Exposition of the Plant Leaf Species Dataset (*PLSD*). 62
3.2 Summary of the Plant Species Segmentation Dataset (PLSSD) Splitting. 63
3.3 Summary of the Plant Leaf Species Dataset (*PLSD*) Splitting. 64
3.4 *PSD-SE-DLM* Experimental Setup Configurations. 70
3.5 Classification Accuracies, Precision, Recall & F1 Score of the Suggested *PSD-SE-DLM* Technique. 76
3.6 Comparison of the Proposed *PSD-SE-DLM* Model with other Existing Models. . . 78
3.7 Comparison of the Proposed *PSD-SE-DLM* Model with other State-of-the-Art Models. 79

4.1 Guava Patches Dataset (GPD) Summary 85
4.2 Guava Leaf Diseases Dataset (GLDD) Summary 86
4.3 Configuration Parameters of the Proposed GMLDD Model 90
4.4 The Proposed *GIP-MU-NET* Dice Score Comparison on Test Set. 93
4.5 The Proposed *GLSM* Dice Score Comparison on Test Set. 97
4.6 The proposed GMLDD model performance 104
4.7 The Proposed GMLDD comparison with YOLOv3, YOLOv4, and YOLOv4-Tiny . 109

5.1 Summary of the PlantVillage Dataset. 115
5.2 Classification Accuracies of the Proposed PDDCNN Model Training on PlantVillage and Testing on the PLD Dataset. 116

5.3 Summary of PLD Dataset. 117
5.4 The Proposed *PDDCNN* Model Configuration Details of Various Parameters. . . . 122
5.5 Classification Accuracies on Different Sets of Data Augmentations of the Proposed *PDDCNN* Model on the *PLD* Dataset. 124
5.6 Classification Accuracies, Precision, Recall and F1-Score of the Proposed *PDDCNN* Model on the *PLD* Dataset Using Data Augmentation Techniques. 126
5.7 Classification Accuracies, Precision, Recall and F1-Score of the Proposed *PDDCNN* Model on the *PLD* Dataset without Applying Data Augmentation Techniques. 129
5.8 Classification Accuracies, Precision, Recall & F1-Score of the Proposed PDDCNN Model on the PlantVillage Dataset Using Data Augmentation Techniques. 136
5.9 Classification Accuracies, Precision, Recall & F1-Score of the Proposed PDDCNN Model on the PlantVillage Dataset. 140
5.10 Classification Accuracies of the Proposed PDDCNN Model Training on PlantVillage and Testing on PLD and Training on PLD and Testing on the PlantVillage Dataset. 142
5.11 Comparison with State-of-the-Art Techniques. 143
5.12 Comparison with Existing Studies. 144

Acronyms

AI Artificial Intelligence.

Adam Adaptive Moment Estimation.

BN Batch Normalization.

BRISK BRISK Binary Robust Invariant Scalable Keypoints.

BT Bagged Tree.

BUGL2018 BU Guava Leaf 2018.

C-GAN Conditional Generative Adversarial Network.

CAD Computer-Aided Diagnosis.

CAE Convolutional Auto Encoder.

CASC-IFW Comprehensive Automation for Specialty Crops-Internal Feeding Worm.

CBB Cassava Bacterial Blight.

CBSD Cassava Brown Streak Disease.

CCDF Correlation Coefficient & Deep Features.

CGM Cassava Green Mite.

CHT Circular Hough Transform.

CIAT International Center for Tropical Agriculture Dataverse.

CIFAR Canadian Institute For Advanced Research.

CMD Cassava Mosaic Disease.

CMYK Cyan, Magenta, Yellow, Key.

CM Confusion Martrix.

CNN Convolutional Neural Network.

CVPPP Computer Vision Problems in Plant Phenotyping.

CV Computer Vision.

DCNN Deep Convolutional Neural Network.

DIR-BiRN Disease Image Recognition Method based on Bilinear Residual.

DL Deep Learning.

DNA Deoxyribonucleic Acid.

DNN Deep Neural Network.

DSCPLD Depth-Wise Separable Convolutional PLD.

DSC Depthwise Separable Convolution.

DSLR Digital Single-Lens Reflex.

DT Decision Tree.

E-MMC Elliptical-Maximum Margin Criterion.

ECNN Enhanced Convolutional Neural Network.

ELM Extreme Learning Machine.

EXIF EXchangeable Image File Format.

FAO Food and Agriculture Organization.

FAST Features from Accelerated Segmented Test.

FBD Foreground-Background Dice.

FCM-KM Fuzzy C-Mean-K-Mean.

FFNN Feed-Forward Neural Network.

FPN Feature Pyramid Networks.

FPR False Positive Rate.

GDP Gross Domestic Product.

GIP-MU-NET Guava Infected Patches Modified MobileNetV2 and U-Net.

GLCM Grey Level Co-Occurrence Matrix.

GLDD Guava Leaf Disease Dataset.

GLSM Guava Leaf Segmentation Model.

GMLDD Guava Multiple Leaf Diseases Detection.

GPDCNN Global Pooling Dilated Convolutional Neural Network.

GPD Guava Patches Dataset.

GPUs Graphical Processing Units.

GPU Graphical Processing Unit.

HOG Histogram of Gradient.

HSV Herpes Simplex Virus.

ICRMBO Improved Crossover Based Monarch Butterfly Optimization.

ILSVRC ImageNet Large Scale Visual Recognition Challenge.

INAR-SSD Inception and Rainbow-Single Shot Multibox Detector.

IoT Internet of Things.

KNN K-Nearest Neighbour.

LBP Local Binary Pattern.

LR Logistic Regression.

LSC Leaf Segmentation Challenge.

MCNN Multilayer Convolutional Neural Network.

MLR Multinomial Logistic Regression.

ML Machine Learning.

ODC Optimal Deep CNN.

OMNCNN Optimal Mobile Network-based CNN.

PCA-LDA Principal Component Analysis-Linear Discriminant Analysis.

PDDB Plant Disease Database.

PDDCNN Potato Leaf Disease Detection Convolutional Neural Network.

PLD Potato Leaf Dataset.

PLSD Plant Leaf Species Dataset.

PLSSD Plant Leaf Species Segmentation Dataset.

PR-Curve Precision-Recall Curve.

PSD-SE-DLM Plant Species Detection using Stacking Ensemble Deep Learning Model.

R-Curve Recall Curve.

RCNN Region based Convolution Neural Network.

RF Random Forest.

RGB Red, Green, Blue.

ROC Receiver Operating Characteristic.

ROI Region of Interest.

ReLu Rectified Linear Unit.

ResNet Residual Neural Network.

RoCoLe Robusta Coffee Leaf.

S-CNN Siamese Convolutional Neural Network.

SBCNN Self-Build CNN.

SGD Stochastic Gradient Descent.

SIFT Scale-Invariant Feature Transform.

SPP Spatial Pyramid Pooling.

SSD Single Shot MultiBox Detector.

SVM Support Vector Machine.

SegCNN Segmentation Convolutional Neural Network.

TL Transfer Learning.

TMV Tomato Mosaic Virus.

TPR True Positive Rate.

TP True Positive.

VGG-FCN-S Visual Geometry Group-Fully Convolutional Network-Shallow.

VGG Visual Geometry Group.

WDD2017 Wheat Disease Dataset 2017.

XGBoost Extreme Gradient Boosting.

YOLO You Only Look Once.

mAP mean Average Precision.

List of Publications

1. **Published Article(s)**

   (a) Rashid, J., Khan, I., Ali, G., Almotiri, S.H., AlGhamdi, M.A., Masood, K., "Multi-Level Deep Learning Model for Potato Leaf Disease Recognition," *Electronics* **2021**, *10(17)*, 2064.

   (b) Rashid, J., Khan, I., Ali, G., Rehman, S.U., Alkhalifah, T., Alturise, F., "Real-Time Multiple Guava Leaf Disease Detection from a Single Leaf Using Hybrid Deep Learning Technique," *Computers, Materials & Continua* **2023**, *74(1)*, 1235-1257.

   (c) Rashid, J., Khan, I., Abbasi I. A., Saeed M.R., Saddique M., Abbas, M., "A Hybrid Deep Learning Approach to Classify the Plant Species." *Computers, Materials & Continua* **2023**. [Accepted on 30-05-2023]

Chapter 1

Introduction

Agriculture is the backbone of human survival on this planet as it provides life necessities, including food, fire, fibre, fodder, fruits and fertilizers, etc. Some of the world's most important crop plants and animal species were domesticated thousands of years ago when agriculture began to flourish. Crop protection has been a challenging task in crop husbandry as it also carries a long history of plant destruction and famines, leading to millions of casualties globally [1]. Plant diseases are a vital cause of food insecurity, a severe global problem today [2]. According to one estimate, 16% of worldwide food yields are lost due to plant diseases. Pests are predicted to have a potential global loss of roughly 50% for wheat and 26-29% for soybeans [3]. A country's economy relies heavily on its agricultural sector, of which small farmers make up the vast majority (around 80%). Crop pests and illnesses account for at least half of this production's losses [4]. According to the UN's *FAO* report, there will be an estimated 9.1 billion people on the planet by the year 2050. Approximately 70% of the increase in agricultural output is required for a safe and sufficient food supply. [5].

Identifying plant disease is the first step toward a successful plant protection program, which follows the employment of appropriate control measures to achieve agricultural production and food security goals. Providing such service at the farmer level is a significant hindrance for which various service structures have been designed in the agriculture departments in Pakistan. Mostly small scale farmers do not know much about the nature and severity of diseases. Application timing of control measures also matters a lot, especially in controlling the disease, as it drastically and robustly spoils the whole crop yield within a short period. Further, symptoms of many diseases and nutritional deficiencies look similar, sometimes misleading, when visually observed. In such circumstances, farmers confuse or make wrong choices about pesticides and fertilizers. According to the United Nations Environment Programme report [6], about 50% of crop yield losses are due to incorrect determination of crop diseases and wrong usage of pesticides and fertilizers.

Many countries cultivate potatoes as a significant crop across various climatic zones, including temperate regions, subtropics, and tropics, as well as lowlands and highlands with vastly varying agroecological zones. There are several climates where the potato (*Solanum tuberosum*) can grow, but it cannot flourish in extreme situations like intense heat or humidity [7]. About 130 countries now cultivate potatoes, and there are more than 5000 varieties for selection in various agroecological regions [8]. When considering global food consumption, potatoes rank as the third most significant non-grain crop. In areas where food availability is a problem, the *FAO* has designated the potato as a food security crop [9]. With their high nutritional contents (vitamins like Niacin, thiamin, riboflavin, and vitamin C, minerals and carbohydrates) and excellent yield, potatoes have overtaken wheat and rice as the fourth most significant worldwide food crop.

Almost three-quarters of Pakistan's population lives on less than $2 daily earnings for accommodating daily domestic requirements [10]. Many people cannot buy and eat healthy food because of the lack of resources. About 70% of the population lives off of agricultural income, while at least 43.7% of the workforce is involved in farming [10]. Even while agriculture plays an essential role in the economy, production is less than the threshold. Low productivity can be attributed to various factors, including small farm holdings, insufficient farmer purchasing power, little or suboptimal input utilization, and others. Farmers are compelled to develop short-lived crops to make a profit [11].

Potato cultivation is considered one of the most practical among short-duration crops. With the establishment of Pakistan as a sovereign government in 1947, the country only had enough land to grow potatoes on a few thousand acres (ha), with an annual production of fewer than 30,000 tonnes [12]. The significant increases in cultivated areas and average yield attained in the decades after independence, the potato has become the fastest-growing primary food crop in the country. Higher irrigation efficiency in Pakistan is mainly responsible for expanding cultivated land and increasing crop yield across the country, with Punjab showing solid growth. In 2020-21 agricultural statistics exhibited that the cultivation of potatoes in Pakistan was on 234491 ha of land area, yielding a total of around 5854208 tonnes, and that Punjab (Pakistan) cultivates potatoes on 220550 ha, generating a capacity of 5682000 tonnes of which the district Okara of Punjab Province contributed 159,5425 metric tonnes ($1/4th$) in the yield of potatoes [13]. In 2020-21, the Okara district of Punjab produced 159,5425 metric tonnes of potatoes. It is the most prosperous region in Punjab and Pakistan [13]. The utilization of potatoes generally in domestically usages as raw in food preparation, industries like snacks, flakes, or fries, or utilized in the sizing of garments (starch).

The guava, or *Psidium guajava L.*, is a popular fruit grown worldwide, especially in the tropics and subtropics. Its year-round availability, low price relative to other fruit crops, and high nutrient density make it desirable for forward-thinking farmers [14, 15]. The flavour of guava when it is fully ripened and newly picked is sweet and enticing. Aside from fresh consumption, it is also utilized in producing jams and jellies. It is high in nutrients containing high contents of vitamin A, B, and C, carbohydrates (11%), protein (0.7%), and water (82%). Soluble fiber, phosphorus, nicotinic acid, and calcium. Guava is anti-diabetic, aids in increasing the human immune system, reduces the risk of cardiac arrest and cancer, lowers stress, and is a perfect weight reduction tool [16].

Among Asian countries, India and Pakistan produce the guava each year, with a total of 4.10 million tonnes and 0.52 million, respectively. Guava in Pakistan is the fourth-largest fruit crop in production and land area, after only citrus, mango, and apple [17]. A large percentage of Punjab's agricultural land (about 80%) is dedicated to growing guava because of the fruit's adaptability to various climates [18, 19]. Guava production is concentrated in the districts of Lahore, Sheikhupura, Faisalabad, Nankana Sahib, Kasur, Sahiwal, and Okara in central Punjab [20].

Plants mainly various crops in agricultural lands are usually much susceptible to various environmental factor leading to the outbreak of various disorders. Such diseases and symptoms are attributed because of disruption of a plant's vital biochemical or physiological processes [21]. Since the beginning of recorded history, diseases like rust, mildews, and blights have been responsible for hunger, malnutrition as well as for starvation thus altering the economic landscape of nations e.g., Ireland's late potato blight (1845-60), France's powdery and downy grape mildews (1851 and 1878), Ceylon (now Sri Lanka's coffee) rust (the 1870s), cotton and flax wilts due to Fusarium, tobacco's southern bacterial wilt (early 1900s), banana's Sigatoka leaf spot (1960) and Panama disease (2012-present) [21].

The disease must be diagnosed appropriately to recommend particular treatments. In the research of any disease, it is the first step. The diseased plant's symptoms play a significant role in determining the disease's identity. Diagnosis cannot be accomplished without first determining the pathogen's identity. Careful observation and categorization of evidence, appraisal of facts and logical conclusion to the cause are three processes needed in diagnosing [21].]. Identifying illnesses and their causative agents is critical to develop practically controlled approaches. The ability to accurately diagnose plant diseases is thus crucial to training any aspiring plant pathologist. If the sickness and the agent causing it are not correctly diagnosed, efforts to control the condition could be wasted.

For detecting plant diseases, traditional methods rely heavily on the farmer's experience, which are highly subjective and inaccurate. Visual estimation is one of the quickest and most common approaches when identifying plant diseases. Researchers have proposed the spectrometer as an alternative to traditional methods of analyzing plant diseases for determining healthy or infected crops [22]. Deoxyribonucleic Acid (*DNA*) can also be extracted from leaves using polymerase chain reaction [23] or real-time polymerase chain reaction [24]. The employment of numerous crop protection procedures and the guidance of experienced experts make these approaches difficult, time-consuming, and expensive. As a result, an automatic plant leaf species and disease identification system capable of early detection of diseases affecting potato and guava leaf species are urgently required.

The invention of artificial intelligence (*AI*) [25] has revolutionized many fields, such as speech recognition, pattern recognition, meteorology, livestock, etc. [26–28], throughout the world, including the agriculture sector [21, 25]. Artificial intelligence applications in agriculture have given rise to the idea of smart agriculture [29, 30]. It can successfully solve the issues of farmers at their doorstep. However, its utilization in the agriculture sector has not yet been fully explored in Pakistan. Further, there is no knowledge or minimal services achieved by applying this system of AI for crop disease management in Pakistan, which necessitates research work in this regard. Recent advancements in machine learning and image processing approaches have made it possible to diagnose plant diseases in real time using film collected in the field [31, 32]. Even in the area, plant diseases may be accurately detected from photographs of plants using convolutional networks and Deep Learning (*DL*) [33]. In locations where technical expertise is few or continuous on-site monitoring is impractical, image segmentation and *DL* algorithms for autonomous plant disease diagnosis may become a vital source of decision-making on farms. Plant species recognition and disease detection have made significant strides using *DL*-based algorithms [34, 35]. The ultimate goal of this study is to design a system that employs *DL* methods to assess problems in various potato and guava varieties in real-time.

The accurate classification of plant species plays a crucial role in the effective detection and management of plant diseases. Plant diseases severely damage the crop yields, which leads to economic loss and effect food security. Therefore, early and accurate detection of plant diseases is vital to minimize the disease spread and mitigate their effects. The classification of plant species is a fundamental step in plant disease detection as it provides a foundation to identify and understand the specific diseases that affects particular plant species. However, different plant species are susceptible to different diseases, and the symptoms of plant diseases vary significantly on the basis of plant species. The accurate classification of plant species, help the plant pathologists and agricultural experts to develop disease detection methods that are tailored to specific plant species. It allows the farmers to identify disease symptoms that are unique to each plant species and vary from other non-disease-related symptoms. This specificity enhances the accuracy and reliability of plant disease detection, that leads to develop timely and appropriate management strategies. The accurate classification of plant species, make it possible to develop tools and techniques for early disease detection. Early detection allows for prompt intervention measures, such as applying fungicides or adjusting irrigation and fertilization practices, to prevent disease spread and minimize crop losses. Plant species classification also enables precision management of plant diseases. Accurate plant species classification helps the experts to determine the appropriate dosage and timing of treatments. This ensures that the right interventions are applied only to the affected plant species, reducing unnecessary use of agrochemicals and minimizing the risk of environmental contamination. Therefore, it helps to optimize disease control measures, reduce costs, and minimize negative impacts on human health and the environment. It also plays a critical role in disease forecasting and risk assessment. The accurate information about the specific plant species in a field, plant pathologists and agronomists can assess the vulnerability of a crop to specific diseases based on historical disease data and weather conditions. This allows for proactive disease management strategies, such as adjusting planting dates, crop rotations, or using resistant cultivars, to mitigate disease risks. Disease forecasting and risk assessment based on accurate plant species classification enable proactive planning and decision-making, which saves time, resources, and reduce crop losses. We believe that accurate plant species classification is a fundamental aspect of plant disease detection and management. It enables disease-specific detection, early detection, precision management, and disease forecasting and risk assessment.

In light of the recent advances in machine learning and image processing, researchers have created new methods for real-time detection and classification of potato and guava leaf species and diseases using *DL*. The proposed method seeks to develop cutting-edge machine learning and image-processing algorithms that quickly and accurately aid farmers in making decisions to limit the spread of diseases. This study is limited to potatoes and guava from central Punjab, Pakistan, for experimentation and validation. The experiments are performed using publicly available benchmarks and self-contained datasets. We have developed the aerial image datasets of potato and guava leaf images from the Central Punjab of Pakistan. We have built datasets, one for identifying a specific disease on a single leaf and the other for identifying many disorders on a single leaf. From the Central Punjab region of Pakistan comes a potato leaf dataset (PLD) for studying single-leaf diseases and a Guava leaf diseases dataset (GLDD) for studying multiple-leaf diseases.

Three models have been established for this research:

1. A Hybrid Deep Learning Approach to Classify the Plant Leaf Species has been developed for classifying guava, potato, and java plum leaf species. It consists of two novel methods. The first model using MobileNetV2-UNET architecture has been developed to segment the guava, potato, and java plum plant species. A Plant species detection stacking ensemble deep learning model (PSD-SE-DLM) to detect the potato, java plum, and guava species as a second model.

2. Identifying Multiple Diseases on a Single Guava Leaf in Real-Time It has been discovered that a single guava leaf may diagnose several diseases in real-time using a three-step process called the hybrid deep learning technique. A method for dividing up contaminated guava patches called Guava Infected Patches Modified MobileNetV2 and U-Net (GIP-MU-NET) has been presented in the first part. The encoder of the suggested model is based on a tweaked version of MobileNetv2, while the decoder uses the up-sampling layers from the U-Net architecture. For the second part, we propose the Guava Leaf Segmentation Model (GLSM) to divide the leaves into diseased and healthy parts. Finally, a Guava Multiple Leaf Diseases Detection (GMLDD) technique built on top of the YOLOv5 method is used to identify a wide range of illnesses in a guava leaf.

3. A multi-stage *DL* technique for potato disease classification was created to tell the difference between normal potato leaves and those infected with late blight or early blight. First, it takes a picture of a potato plant and utilizes the YOLOv5 picture segmentation method to separate the leaves. The second level uses a novel *DL* method established with a convolutional neural network to identify several potato disorders, such as early and late blight, in real-time from images of potato leaves.

1.1 Motivation

Despite tremendous breakthroughs in food production, plant diseases [36] continue to pose a hazard to the sustainability and availability of the food supply [37]. Disease detection from plant leaves is challenging, and developing countries that lack the essential infrastructure in early disease detection from plants lost more than 50% of crop production. Crop output is also negatively impacted by catastrophes; in 2017, 23% of food production in poor nations was lost due to medium to large-scale disasters [38]. Food and the other sectors were also dented and approximately 80% of collective damage in transit, health, and infrastructure were reported as most of the developing countries' economies depend on the agriculture sector solely.

The developing country like Pakistan has an 18.9% contribution to the Gross Domestic Product (*GDP*) and absorbs 42% of the workforce across the country [39]. These stats will be devastating if we combine the impact of both challenges but still, the diseases damage more. According to the United Nations Environment Program report [38], about 50% of crop yield losses are due to incorrectly detecting crop diseases first and using the wrong pesticides and fertilizers after. The first stage in developing an effective plant protection program is accurately diagnosing plant diseases; the second is implementing effective management measures to ensure food security and crop yields. Plant diseases cause damage and reduce crop yield by manifesting symptoms on plant buds, stems, or leaves. Furthermore, these disorders spread to other plants. Product loss can be avoided if these illnesses are caught early enough to be treated. Therefore, it is critical to identify the disease early on and take the required precautions, which may prevent the loss of productivity and amount of agricultural products.

Many methods have been developed to identify plant disease from leaf samples; however, these approaches need to be more sensitive to pick up on the plant diseases prevalent in the Pakistan region. A plant disease detection website [40] is also publicly available, but this site also overlooks the diseases of the Pakistan region. Figures 1.1 (a) and 1.1 (b) show that the potato plant has a distinct set of leaf species from its Pakistani counterpart and the PlantVillage dataset. Figures 1.1 (c) and 1.1 (d) illustrate that the guava leaf species native to Bangladesh are distinct from their Pakistani counterparts. It is because, as shown in Figure 1.1, crop species diversity is present everywhere. Figure 1.2 (a) and (b) depict the varying symptoms of early blight and late blight potato leaf infections in Pakistan and the PlantVillage dataset, respectively, due to regional climate and global warming differences. However, current methods are limited in identifying crop species and agricultural diseases since the underlying models are only tested and trained on photographs of plant leaves from a specific geographical area. Modern methods avoid generalization by relying on handcrafted and dynamic features. Thus, relying on the extreme of bias-variance does not allow them to detect crop diseases with subtle perceptual differences. Usually, botanists and agricultural engineers apply the manual process of detection of diseases in plant leaves in two stages, i.e., (i) visual inspection and (ii) laboratory environment, which is time-consuming, not robust [41], but these methods require continuous inspection of specialists, which is expensive and time-consuming. Further, symptoms of many diseases and nutritional deficiencies look similar, sometimes misleading when observed with the naked eye. In such situations, farmers get confused or make wrong choices about pesticides and fertilizers.

Given these constraints, researchers are developing novel image processing-based algorithms for precisely detecting and classifying potato and guava leaf species and disorders.

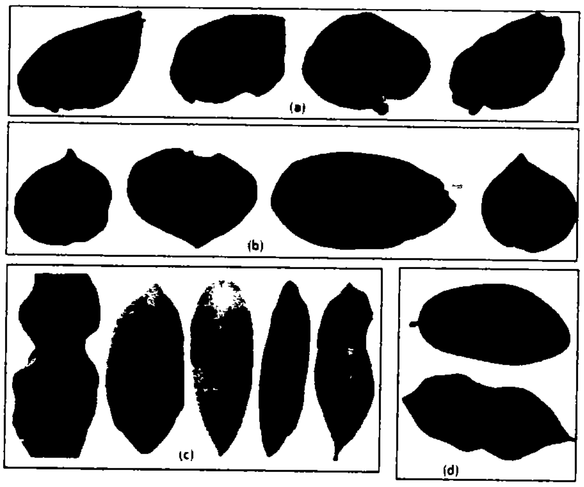


Figure 1.1: (a) Potato Species of PlantVillage Dataset (b) Potato Leaf Species of Pakistani Environment (c) Guava Leaf Species of Pakistani Region (d) Plant Leaf Species of Bangladesh.

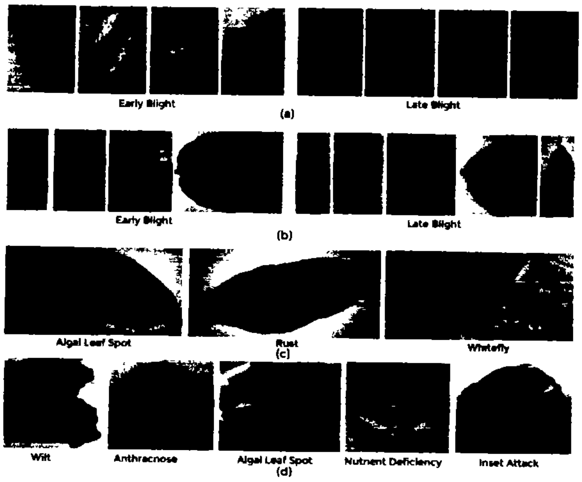


Figure 1.2: (a) Potato Leaf Diseases of PlantVillage Dataset (b) Potato Leaf Diseases of Pakistani Environment (c) Guava Leaf Diseases of Bangladesh (d) Plant Leaf Diseases of Pakistan Region.

1.2 Objectives

Some goals of this study are as follows:

1. To develop new benchmark plant disease datasets of potato and guava in the Central Punjab region of Pakistan.
2. To devise and employ an image segmentation technique in particular for guava and potato plant leaves with different illumination and complex backgrounds to detect the infected region from a crop field.
3. To craft an efficient deep-learning based classification approach to classify the potato and guava leaf species from Central Punjab, Pakistan.
4. To employ an efficient deep learning approach for potato and guava leaf disease detection for the Central Punjab region of Pakistan.
5. To enhance the efficiency of the employed methods within the context of accuracy.

1.3 Relevance and Effectiveness

To improve the growth and quality of yield in agricultural sectors, Computer-Aided Diagnosis (*CAD*) plays an important role. This research is allied with great significance and has a wide variety of scope. Few are below:

1. In the current era, indirect methods of detecting disease from plant leaves are crucial in agricultural practice. Computer-based detection has earned a stellar reputation for addressing production crises and quality issues.
2. Improved disease diagnosis will allow for more effective treatment of leaf abnormalities in potato and guava plants, increasing the likelihood that harm will be halted or reversed.
3. An additional perk of using an automated disease detection system is that it can serve as a preventative step by analyzing the source of the disease and offer advice on how to cure it.
4. Clear and high-quality imaging is critical for many decisions in plant disease detection. This method is safe, cost-effective, time-saving, and easily adaptable, becoming increasingly important with each passing day. Automated diagnosis is on the rise with improved production and quality policies and increased access to computer hardware worldwide.

5. The created databases will aid researchers in identifying single-leaf diseases on individual potato leaves and multiple-leaf diseases on single guava leaves. The different potato and guava types' plant species and diseases can be identified with its aid.

1.4 Contributions

In this thesis, we set out to construct real-time detection and categorization of species and pathogens in potato and guava leaf samples using deep learning techniques. What this thesis contributes to the discussion is:

1. Plant Species Recognition using Ensemble Technique is developed to recognize the plant species of potato, java plum, and guava crops.
2. We have developed the following Segmentation Techniques:
   (a) A Guava Infected Patches Modified MobileNetV2 and U-Net (*GIP-MU-NET*) segmentation technique is developed to segment the infected guava patches. The suggested *GIP-MU-NET* technique is made up of modified MobileNetv2 as the encoder and upsampling layers of U-Net as the decoder.
   (b) A Guava Leaf Segmentation Model (*GLSM*) is developed to segment the guava leaves into infected and healthy ones.
   (c) A *YOLO*v5-based approach is proposed for real-time efficient segmentation and extraction of potato leaf data.
   (d) A Plant Leaf Segmentation based on MobileNetV2-UNet based technique is developed to segment the guava, java plum, and potato healthy and diseased leaves. The MobileNetV2 is used as an encoder par and UNet as a decoder part.
3. Two new benchmark datasets have been developed for potato and guava plant leaf diseases from the Central Punjab of Pakistan:
   (a) We have created the first Guava Leaf Disease Dataset (*GLDD*) dataset to identify multiple diseases on a single leaf. Anthracnose, nutritional shortage, wilt illnesses, insect attack, and healthy comprise the five groups represented in the dataset.
   (b) A new Potato Leaf Dataset (*PLD*) dataset has been developed to identify the late and early blight disorders along with healthy potato leaves originating in the Central Punjab

region of Pakistan.

4. For potato and guava leaf diseases identification, the following two methods have been proposed:

   (a) A real-time Guava Multiple Leaf Diseases Detection (*GMLDD*) method is developed based on YOLOv5 to detect and localize multiple diseases from a single guava leaf. It is the only model we know that can identify many disorders from a single image of a guava leaf. The proposed method can detect and localize multiple diseases from a single image, including anthracnose, nutrient deficiency, wilt diseases, and insect attacks.

   (b) A novel deep-learning approach called *PDDCNN* has been created to assist in the identification of late and early blight on potato leaves.

5. The proposed models now have improved accuracy and performance. It is the most accurate if we compare the *PDDCNN* technique to other cutting-edge techniques. However, the Guava Multiple Leaf Diseases Detection (*GMLDD*) model has also obtained good mAP to identify multiple diseases on a single leaf.

1.5 Problem Statement

Diseased plant detection is one important task in agricultural processes. There exist various approaches to detect the diseased plants. One such approach is to employ machine learning techniques to detect the disease from leaves of the plant. This process requires classification and segmentation of plant leaves. Researchers have already explored these approaches, however, the accuracy, scale, scope and cost of these approaches remains questionable. One major problem is limited scale of the dataset used in existing approaches that lacks in region specific data. To address the problem at regional level in particular in Pakistan, considering the two types of the plants one growing with fruiting in the soil as plant and the other one fruiting above the soil as tree, this research defines its scope to consider a case from each of the mentioned classes. One such case is segmentation and classification of guava leaves. The guava plant is vulnerable to diseases, specifically the leaves and fruit, which result in massive crop and profitability losses. The existing plant leaf disease detection techniques can detect only one disease from a leaf. However, a single leaf may contain symptoms of multiple diseases. The other case considered in this research is of the potato leaves. Variations in crop species, crop disease signs, and environmental conditions may contribute to the difficulty of early disease detection in potato leaves. Several machine learning methods have been employed for disease detection in potato leaves. There exist significant variance in geometry, color shape and other attributes of guava and potato leaves in Pakistan region with that of rest of the world whose dataset for the same is available. To address the problem at regional level, the existing studies of potato leaf disease classification trained on a specific region dataset (PlantVillage) do not detect the diseases of Pakistan region with satisfactory accuracy. This research provides an efficient deep learning technique using CNN to detect the different potato leaf diseases for the Central Punjab region of Pakistan. Finally, the detection of multiple diseases from plant leaves in real-time is another challenging task that requires a comprehensive dataset and effective approaches to process the data.

In the given described context, there are several issues that needs to be addressed for effective solution of the problem. Firstly, there exists no comprehensive dataset that can support detection of the infected patch of guava tree from its leaves. Secondly, the segmentation of guava and potato leaves from the plant image having different illumination and complex background is required. Thirdly, as there is currently no literature on the detection of the potato and guava leaf species, the classification of guava and potato species through application of deep learning is required. Fourthly, an effective deep learning approach for detection of single or multiple diseases from potato and guava leaves is required. Finally the real-time multiple leaf diseases on a single leaf is also a challenging and complex however a required task. The real challenge is to address these issues in cost effective and efficient manner.

1.6 Proposed Solution

In agriculture studies, plant disease detection is a complex area due to diverse species cultivation in different environments around the globe. It is even more challenging to classify and segment plant species and their diseases using deep learning as their resemblance changes across various countries in different environmental conditions. The potato and guava plant crops, which are farmed at a vast scale in Pakistan, may vary from the plants of the same genera in other countries, and so does the shape symptoms and occurrence of their diseases. Existing studies were conducted on potatoes and guava but for different regions. This study presents three sub-model-based solutions for plant species identification and disease segmentation and classification for guava and potato. A hybrid deep learning approach with MobileV2 and UNET was proposed to classify the plant species, which captures video, converts it into frames and segments and classifies them according to their corresponding classes. The second sub-model was also a hybrid deep learning technique for the real-time multiple guava leaf disease detection from a single image by capturing video of a guava tree and then converting its frames and performing detection and localization of diseases employing a modified MobileNetV2 along with UNET. The third model used deep multilevel learning for potato leaf disease recognition; a CNN classifier classifies different potato leaf diseases at a higher accuracy. The block diagram for the presented methodology is presented in Figure 1.3.

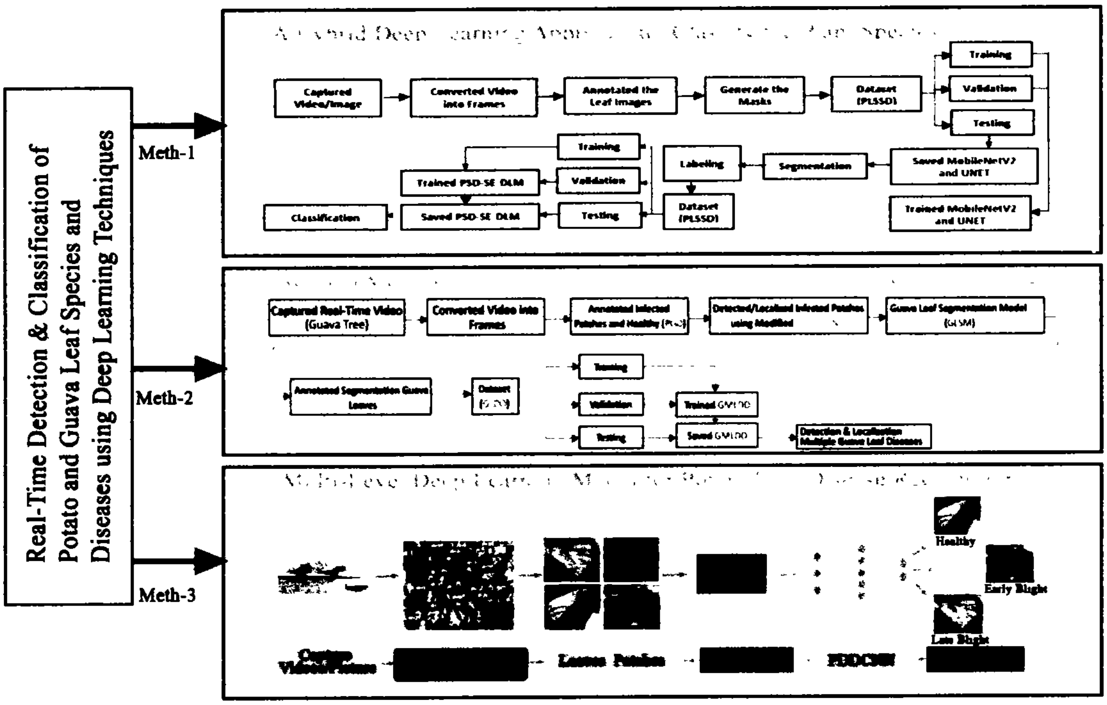

Real-Time Detection & Classification of Potato and Guava Leaf Species and Diseases using Deep Learning Techniques
Meth-1
Meth-2
Meth-3
Captured Video/Image
Converted Video into Frames
Annotated the Leaf Images
Generate the Masks
Dataset (PLSSD)
Training
Validation
Testing
Saved MobileNetV2 and UNET
Trained MobileNetV2 and UNET
Segmentation
Labeling
Dataset (PLSSD)
Training
Validation
Testing
Trained PSD-SE DLM
Saved PSD-SE DLM
Classification
Captured Real-Time Video (Guava Tree)
Converted Video into Frames
Annotated Infected Patches and Healthy
Detected/Localized Infected Patches using Modified
Guava Leaf Segmentation Model (GLSM)
Annotated Segmentation Guava Leaves
Dataset
Training
Validation
Testing
Trained GMLDD
Saved GMLDD
Detection & Localization Multiple Guava Leaf Diseases
Healthy
Early Blight
Late Blight
Leaves Patches

Figure 1.3: The Proposed Methodology.

1.7 Thesis Organization

The following are the next six chapters of this thesis:

- Chapter 2 covers the most up-to-date research on classifying plant leaves by species, analyzing leaf structure, and identifying leaf diseases. It includes the problem statement for this thesis and a discussion of the unanswered questions raised by the existing literature.
- In Chapter 3, we dive deep into the hybrid deep learning method, the process of building a dataset of plant leaf species, and a comparison to other cutting-edge approaches. The proposed hybrid model combines two cutting-edge methods. Model 1 uses the MobileNetV2-

UNet architecture for segmentation, and Model 2 identifies the different plant species using a Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*).

- The focus of Chapter 4 is on creating the first-ever Guava Leaf Disease Dataset (*GLDD*) as well as the Guava Infected Patches Modified MobileNetV2 and U-Net (*GIP-MU-NET*) segmentation technique, Guava Leaf Segmentation Model (*GLSM*), and real-time Guava Multiple Leaf Diseases Detection (*GMLDD*) method. Also, the suggested Guava Multiple Leaf Diseases Detection (*GMLDD*) model is as opposed to other members of the *YOLO* family.
- In Chapter 5, we discuss the Potato Leaf Dataset (*PLD*), the Potato Leaf Disease Detection Convolutional Neural Network (*PDDCNN*), and the efficacy of the suggested *PDDCNN* technique, in addition a revolutionary deep learning technique termed real-time potato leaf segmentation and extraction using *YOLO*v5. The suggested technique is rigorously evaluated against existing techniques and benchmarked against numerous datasets to guarantee its precision.
- The final verdict is presented in Chapter 6.

Chapter 2

Literature Review

The chapter below provides a synopsis of the relevant literature. This section examines the extensive and existing image-based approaches developed so far to identify patterns and issues in recognizing plant species, plant leaf disorders, and plant leaf segmentation.

2.1 Plant Species Detection

Today's AI systems, created with multiple processing layers to facilitate representation learning at different levels of data abstraction, have earned the catch-all term "deep learning". Deep learning allows for the automatic generation and extrapolation of features from unprocessed representations of input data, eliminating the need for human intervention typically required by traditional algorithmic methods. Several papers employ deep learning methods to determine what kind of plant is depicted in a given leaf image.

2.1.1 Flower Species Recognition

Nguyen et al. [42] presented flower species identification using GoogLeNet, CaffeNet, and AlexNet deep learning models. Firstly, this implementation achieved the maximum accuracy (66.6%) by GoogLeNet using the PlantCLEF2015 dataset. However, the accuracy achieved in this research could be better. Secondly, the methodology used for identification can be improved using other methods, such as in our scenario ensemble technique. Finally, the scope of this work was focused on flowers, whereas potato and guava leaf species were not explored. Gogul and Kumar [43] investigated multiple flower species identification using InceptionV3, OverFeat, and Xception deep learning methods. Employing InceptionV3 to identify flower species could obtain the maximum accuracy of 92.41% with the Flowers28 dataset. It can be seen that the size of the dataset in each category needs to be bigger. Additionally, the dataset belongs to different domains and objectives from our focus. Xiao et al. [44] looked at using deep learning models to identify the flowers, specifically InceptionV3 and Residual Neural Network50 (*ResNet50*). The datasets PlantCLEF and Oxford Flower were used in this study. Compared to *ResNet50*, the InceptionV3 model achieved maximum accuracy of 69.5% On PlantCLEF, whereas, on the Oxford Flower dataset, it reached a maximum accuracy of 92.8%. This study only looked at flowers; therefore, similar phenomena in potato and guava leaf species were ignored.

2.1.2 Grapes Species Recognition

In a study by Pereira et al. [45], the authors used the AlexNet model to determine the species of grape leaves. The datasets used by the researchers included DRVG, DRVG2018, and the Flavia leaf dataset. AlexNet attained maximum accuracies of 77.30% on combined DRVG datasets and 89.75% on the Flavia leaf dataset. Other methods, such as ensemble, can be used to boost the accuracy. This study focused on grapes; thus, it did not look at other plant species like potatoes or guava leaves.

2.1.3 Multiple Species Recognition

Various researchers [46, 47] have used the LifCLEF2017 dataset to detect the different plant leaf species. Lasseck [46] used the Ensemble model to identify the multiple plant leaf species using LifCLEF2017 dataset and obtained 92.6% accuracy. In another research [47], the researchers employed InceptionResNetV2, VGG16, and MobileNetV2 deep learning techniques to identify the various plant leaf species using the LifCLEF2017 dataset. As compared to the other two models, the MobileNetV2 achieved the highest accuracy of 95.6%. Both pieces of research focused on other plant leaf species, but the potato and guava leaf species were not explored. Although Lasseck [46] employed the ensemble model that can be characterized to address overfitting problems and enhance accuracy yet the low accuracy attained in the research is due to the choice of base models. Contrary to this research, We have used the ensemble model with three base models.

Some researchers [48, 49] used different plant leaf species using the Swedish Leaf dataset. Kaur and Kaur [48] used Support Vector Machine (*SVM*) to identify the 15 plant leaf species using Swedish Leaf dataset and obtained 93.26% accuracy. On the other hand, Bisen [49] proposed a

CNN-based model to identify 75 plant leaf species using the Swedish Leaf dataset with an accuracy of 97%. The problem with both methods was that they could only perform better on large-scale datasets. Secondly, neither of these techniques has been used to identify the guava and potato leaf species.

Many researchers [50–53] identified the multiple leaf species using a self-created dataset. Wei Tan et al. [50] employed D-Leaf, Pre-trained AlexNet, and Fine-tuned AlexNet to classify the 43 species of tropical trees using a self-created dataset. Compared to the other models AlexNet fine-tuned model obtained the highest accuracy of 95.4%. Another research [51] compared the deep-learning models, such as InceptionResNetV2, ResNetV2, VGG16, and MobileNetV2, to identify Vietnamese plant species using a self-created dataset. MobileNetV2 obtained the highest accuracy of 83.9% as compared to other models. Research conducted by Bambil et al. [52] proposed DNN and SVM-based models to identify the 30 plant leaf species using a self-created dataset. Compared to DNN, the SVM-based model achieved a maximum accuracy of 98.4%. Hati and Singh [53] developed a ResNetV2 model to classify the various plant leaf species and monitor their health using a self-collected dataset. The proposed model achieved 90.53% accuracy. These approaches have several problems and performance limitations, such as all the studies used small-scaled datasets. In the case of a small-scale dataset, the chance of overfitting remains an issue. Furthermore, alternative methodologies could be exploited to achieve effective accuracy. It can also be observed that those studies did not consider guava and potato leave species in the scope of the problem.

Various researchers [54–58] used multiple datasets to identify the different plant leaf species. In a research conducted by Barré et al. [54], the researchers used the LeafSnap, Foliage, and Flavia datasets to identify the different plant leaf species employing the LeafNet model. The proposed method obtained 97.9% top-1 and 99.9% top-5 accuracy on the Flavia dataset. Araújo et al. [55] developed the VGG16 and Siamese Convolutional Neural Network (*S-CNN*) models to identify the 424 plant leaf species using PlantCLEF2015 and LeafSnap datasets. The models trained on the LeafSnap dataset obtained the highest accuracy of 88% (VGG16) and 96% using the *S-CNN* model. Hassan and Maji [56] proposed a CNN model to classify the various plant leaf species using Flavia, MK-D1, MK-D2, and LeafSnap datasets. The proposed model obtained the highest accuracy of 99.75% on the Flavia dataset. Another research conducted by Erkan et al. [57] developed an Optimal Deep CNN (*ODC*) method, which can identify the different plant leaf species and handwritten digits using the Folio Leaf species dataset and MNIST handwritten dataset. The proposed method obtained excellent accuracy of 98.2% on Folio and 99.21% on the MNIST dataset.

Pearline and Kumar [58] employed different machine learning classifiers to classify the various plant leaf species using Flavia, Folio, Swedish, and an internal dataset. Among all the classifiers, MLR achieved the highest accuracy of 99.38% on the internal dataset. The problem with all the methods was that they could only perform better on small-scale datasets. Secondly, neither of these techniques has been used to identify the guava and potato leaf species.

Table 2.1: Summary of Plant Leaf Species Related Work.

Reference	Methodology	Plant Species	Dataset	Accuracy
[42]	GoogLeNet, CaffeNet, and AlexNet	Flowers	PlantCLEF 2015	Alexnet: 50.6%, Caffenet: 54.84%, Googlenet: 66.6%
[43]	InceptionV3, OverFeat, and Xception	Flowers	Flowers28 Oxford Flower	OverFeat: 85.71%, InceptionV3: 92.41%, Xception: 90.18%
[44]	InceptionV3 and ResNet50	Flowers	PlantCLEF, Oxford Flower	InceptionV3: 69.5% (P), 92.8% (O) ResNet50: 68% (P), 92.4% (F)
[45]	AlexNet	Grapes	DRGV2018, Flavia Leaf	DRGV2018: 77.30%, Flavia: 89.75%
[46]	Ensemble	Multiple	LifCLEF2017	92.6%
[47]	VGG16, InceptionResNetV2, and MobileNetV2	Multiple	PlantCLEF	VGG16: 75.5% InceptionResNetV2: 90.9% MobileNetV2: 95.6%
[48]	SVM	Multiple	Swedish Leaf	93.26%
[49]	CNN	Multiple	Swedish Leaf	97%
[50]	D-Leaf, Pre-tuned AlexNet, and Fine-tuned AlexNet	Multiple	Self-Created	D-Leaf: 94.88% Pre-AlexNet: 93.26% Fine-AlexNet: 95.54%
[51]	InceptionResNetV2,	Multiple	Self-Created	InceptionResNetV2: 81.2%

	ResNetV2, VGG16 and MobileNetV2			ResNetV2: 74.6%, VGG16: 70.6%, MobileNetV2: 83.9%
[52]	DNN, SVM	Multiple	Self-Created	DNN: 92.5%, SVM: 98.4%
[53]	ResNetV2	Multiple	Self-Created	90.53%
[54]	LeafNet	Multiple	LeafSnap Foliage Flavia	LeafSnap: 97.8% Foliage: 99.6% Flavia: 99.9%
[55]	S-CNN, VGG16	Multiple	PlantCLEF 2015, LeafSnap	VGG16: 78%, S-CNN: 87% VGG: 88%, S-CNN: 96%
[56]	CNN	Multiple	Flavia, MK-D1, MK-D2 LeafSnap	Flavia: 99.75% MK-D1: 99.15%, MK-D2: 99.43% LeafSnap: 89.17%
[57]	ODC	Multiple	Folio Leaf MNIST	Folio: 98.2% MNIST: 99.21%
[58]	DenseNet-121, MobileNet, Xception, SVM, MLP, BC, RF Cart, KNN, NB MLR, LDA	Multiple	Folio Leaf, Flavia, Swedish Leaf, and Self-Created	MLR achieved best on Flavia: 98.71% Folio: 96.38% Swedish: 99.14% Self: 99.38%

2.2 Plant Leaf Diseases

Over the past few decades, many scientists have employed deep learning techniques to study various agricultural illnesses. Most of the studies relied on data from PlantVillage research. Some researchers also developed their datasets to track down the numerous diseases affecting plant leaves across different crops. Here, we look at the most recent research to be conducted.

2.2.1 Soybean Leaf Diseases

Many researchers [59–62] used different deep-learning models to identify soybean leaf diseases. Nagasubramanian et al. in [59] developed a 3D-DCNN model to detect soybean leaf diseases. Wu et al. [60] conducted research in which transfer learning-based models such as GoogLeNet, AlexNet, and Residual Neural Network (ResNet) architectures were used to classify soybean leaf diseases. Jadhav et al. [61] deployed GoogLeNet and AlexNet pre-trained models to identify the soybean leaf diseases using a self-created dataset. A research conducted by Karlekar and Seal [62] in which soybean leaf diseases were classified using deep CNN using *PDDB* [63] All the studies were trained on small-scale datasets. That might make the above studies less stable when used on more extensive datasets or in more realistic settings. The models also did not find diseases of potato and guava leaves. The models also did not find diseases of potato and guava leaves. The models also trained on specific region datasets besides Pakistan; therefore, they failed to detect the diseases of Pakistan region soybean diseases.

2.2.2 Cucumber Leaf Diseases

Different studies [64–66] were proposed to distinguish cucumber leaf diseases using different deep-learning models. Specifically, Zhang et al. [64] presented a *GPDCNN* to classify the cucumber leaf diseases using a self-created dataset. Using a feature fusion selection technique, Kianat et al. [65] created a hybrid framework to diagnose cucumber leaf diseases. Features from Features from Accelerated Segmented Test (*FAST*), *BRISK*, and a Histogram of Gradient (*HOG*) were built using the preprocessed picture samples. Khan et al. [66] researched to create a *DL*-based system for categorizing cucumber disorders using Cucumber Leaf Diseases publicly available dataset. All the above studies are restricted to a single region, which could prevent the proposed model from being applied to other regions. The studies are limited to cucumber leaf disease detection and failed to detect potato and guava leaf diseases. The models are also trained on a small dataset. Therefore, there was a chance of overfitting. The suggested approaches have a high computational cost, which may reduce their use in low-resource environments.

2.2.3 Peach Leaf Diseases

There existed many studies [67–69] that utilized *CNN* deep learning models to classify peach leaf diseases using the PlantVillage dataset. In recent research, Zhang et al. [67] reported utilizing a *CNN* to identify peach leaf disease. Yadav et al. [68] developed a CNN model to classify the isolated bacteria that cause illnesses in peach leaves. Bedi and Gola [69] proposed a hybrid method combining the Convolutional Auto Encoder (*CAE*) network and *CNN* for autonomously identifying peach diseases. All the studies used a specific region dataset (PlantVillage) that failed to detect the diseases of Pakistani region peach leaf diseases. The studies are limited to peach leaf disease detection and failed to detect potato and guava leaf diseases. The dataset employed in the study is broad and diversified, improving the suggested model's stability.

2.2.4 Rice Leaf Diseases

Researchers employed different deep-learning models to distinguish rice diseases [70–79]. Lu et al. [70] introduced a *CNN* model to identify the ten rice diseases. For early diagnosis of rice diseases, Guoxiong Zhou et al. [71] suggested a system that uses Fuzzy C-Mean-K-Mean (*FCM-KM*) clustering and a Region based Convolution Neural Network (*RCNN*). In [72], Ramesh and Vydeki proposed an improved *DNN* with the Jaya technique to identify rice leaf disorders. Azim et al. [73] utilized color, shape, and texture features to classify the different rice diseases using Extreme Gradient Boosting (*XGBoost*) for classification. Improving the learning ability to recognize subtle lesion characteristics, Chen et al. [74] employed the MobileNetV2 pre-trained on ImageNet as the backbone network and incorporated the attention mechanism to distinguish the different rice diseases. InceptionResNetV2 was a kind of *CNN* model employed in a study [75] to recognize disorders in rice. Upadhyay and Kumar [76] built a *CNN* approach using deep learning to identify rice illnesses. Archana et al. [77] employed an adjusted version of the k-means segmentation technique to isolate the *ROI* from the surrounding images of rice. Sharma et al. [78] considered InceptionV3, MobileNet, ResNet50, Xception, and InceptionResNetV2 transferred learning models to detect the diseases of rice. The rice disorders were proposed by Patil and Kumar [79] using a multi-model data fusion model called Rice-Fusion. The datasets utilized in all the studies were developed in a lab and may not have captured real-world circumstances in rice fields. The studies do not consider environmental elements like weather and soil quality. Any inaccuracies or unfairness in the results could be due to biases in the dataset or the model, neither of which are addressed in the study. The study's findings may be limited in their applicability because of the tiny size of the dataset used. All the models are trained on a specific region dataset; therefore, these models fail to detect Pakistani region diseases. The studies are limited to rice leaf disease detection and failed to detect potato and guava leaf diseases.

2.2.5 Wheat Leaf Diseases

Weekly supervised deep learning frameworks, i.e., *VGG-FCN-S* and VGG-FCN-VD16, were proposed for automatic in-field wheat disease diagnosis [32] using *WDD2017*. To create a new dataset, Bao et al. [80] compiled 360 photos of wheat infected with two common Chinese diseases—powdery mildew and mild stripe rust. In this study, the authors suggested a *DL* system for diagnosing the severity of wheat leaf disorders using the Elliptical-Maximum Margin Criterion (*E-MMC*) metric. Due to the need for specialized hardware and human operators, the system can be expensive to set up and maintain. Small-scale farmers or those working in areas with few resources may need help to afford the system. The models are trained on small-scale datasets. When used in large-scale field circumstances, the system's performance may suffer from obstacles such as variable illumination and weather. Due to variations in disease frequency, crop variety, and cultural behaviors, the system's performance may change among areas or situations. All the models are trained on a specific region dataset; therefore, these models fail to detect Pakistani region diseases. The studies are limited to wheat leaf disease detection and failed to detect potato and guava leaf diseases.

2.2.6 Rice and Wheat Leaf Disorders

Based on the concept of multitask learning, Jiang et al. [81] employed the pre-training method on ImageNET for alternate learning and *TL* to enhance the VGG16 technique. A new dataset of leaf blast, brown spot, bacterial blight diseases of rice, and powdery mildew and rust diseases of wheat was developed of 40 images each for each class. Because of its high accuracy, non-destructiveness, multi-task learning, and transferability, a study is a promising method for automated crop disease identification. Limitations in its interpretability, effectiveness for rare diseases, scalability, generalizability, and accessibility are only some of its difficulties. The model is trained on a specific region dataset; therefore, it fails to detect Pakistani region diseases. The study is limited to rice leaf disease detection and fails to detect potato and guava leaf diseases.

2.2.7 Corn/Maize Leaf Diseases

Many researchers [82–87] researched corn/maize disease identification using different deep-learning models. Barbedo [82] investigated the various factors, i.e., annotation, covariate shift, image capture conditions, symptom representation, image background, variations in symptoms, and disorder variations with similar and different symptoms, that affect the performance and design of plant pathology automation, based on deep learning. For this purpose, the researcher used a benchmark dataset, i.e., an image database of *PDDB* available publicly for different crops. Zhang et al. [83] developed an automated system based on pre-trained *CIFAR*10 and GoogLeNet to detect maize leaf diseases. In [84], Priyadharshini et al. introduced an algorithm dependent upon modified LeNet for analyzing leaf pictures for signs of maize illness. Regarding disease classification in maize, Setiawan et al. [85] compared two *CNN*-based networks, AlexNet and Squeezenet. Multiple *DL* methods, including VGG19, VGG16, GoogLeNet, and AlexNet, were utilized in research by Pan et al. [86] to detect illnesses in Chinese maize leaves. Lu et al. [87] created a technique for diagnosing maize leaf diseases using multi-channel ResNet, Wavelet threshold-guided bilateral filtering and attenuation factor. Low detection, interference, background, and noise are all issues that the proposed technique can address. All the models used small-scale datasets. The proposed methodologies did not perform well on Pakistani region corn/maize plant leaf diseases.

2.2.8 Apple Leaf Diseases

Various studies [88–91] worked on apple leaf disease detection using different deep-learning models. Jiang et al. [88] presented the *INAR-SSD* deep *CNN* model using the Inception module of GoogLeNet's deep CNNs and incorporated the Rainbow concatenation to distinguish the various apple leaf diseases. A lightweight *CNN* was proposed by Li et al. [89] to identify the illnesses that plague Chinese apples. The research [90] developed a mobile-friendly *CNN* model for the real-time detection of apple leaf disorders. Infected apple leaf areas were identified using the MASK *RCNN* after introducing a hybrid contrast stretching technique to enhance an image's visual impact [91]. The studies rely on a relatively small dataset, which might only capture some of the variances in apple leaf diseases. The results may need to be more generalizable due to the bias in the dataset. All the models are trained on a specific region dataset; therefore, these models fail to detect Pakistani region diseases. The studies are limited to apple leaf disease detection and failed to detect potato and guava leaf diseases.

2.2.9 Banana Leaf Diseases

In [92], Selvaraj et al. utilized different transfer learning models, i.e., ResNet50, InceptionV3, and MobileNetV1 to identify banana disease, as well as the corm weevil pest class. Researchers at the International Center for Tropical Agriculture Dataverse (*CIAT*) created a library of banana images that successfully predicted the crop's yield. The study's results may not apply to other demographics or locations due to the small sample size. The results may not apply to the system's performance in the field because the research was done in a lab. Krishnan et al. [93] utilized a hybrid fuzzy C-means technique for their segmentation and classification of banana diseases using *CIAT* dataset. The accuracy of the segmentation and classification findings can be enhanced by training the models on a big dataset. All the models are trained on a specific region dataset; therefore, these models fail to detect Pakistani region diseases. The studies are limited to banana leaf disease detection and failed to detect potato and guava leaf diseases.

2.2.10 Apple and Banana Leaf Diseases

In [94], Khan et al. proposed a technique for automatic disease classification and segmentation in food crops usingCorrelation Coefficient & Deep Features (*CCDF*). Spots on fruit are studied to learn more about illnesses such as apple rot and apple scab, as well as banana cordial leaf spot, Sigatoka, banana deightoniella leaf, and banana diamond leaf spot. Banana and apple disease features are extracted using pre-trained techniques from Caffe Alex Net and VGG16. They got a classification accuracy of 98.60% using the publicly available datasets *CASC-IFW* [95] and PlantVillage. The proposed system's computational needs and resource constraints should be discussed in the study. Environmental elements such as changes in lighting and weather can impact the system's ability to provide high-quality images. The model is trained on a specific region dataset; therefore, it fails to detect Pakistani region diseases. The study is limited to apple and banana leaf disease detection and fails to detect potato and guava leaf diseases.

2.2.11 Tomato Leaf Diseases

Many researchers [96–112] developed different deep-learning models to identify tomato leaf diseases using the PlantVillage dataset. In [96], Rangarajan et al. used VGG16 and AlexNet models that had already been trained to identify tomato leaf diseases using the PlantVillage dataset. Gadekallu et al. [97] created a new PCA–whale optimization-based *DNN* to detect tomato leaf disorders using the PlantVillage dataset. Based on the MobileNet as a feature extractor and the Extreme Learning Machine (*ELM*) as a learning algorithm, Ashwinkuma et al. [98] suggested an Optimal Mobile Network-based *CNN* named *OMNCNN*. Thangaraj et al. [99] established a *DL*-based modified-Xception technique to distinguish the different tomato diseases using the PlantVillage dataset. The research introduced by Chowdhury et al. [100] in which different U-Net architecture variants were investigated to classify the different diseases of tomatoes using the PlantVillage dataset. Altuntaş and KOCAMAZ [101] used ResNet50, AlexNet, and GoogLeNet pre-trained deep learning models as feature extractors using the PlantVillage dataset. After concatenating the three feature maps produced by deep learning models, this feature map was fed into *SVM* to identify tomato disorders. Sembiring et al. [102] proposed a *CNN* to identify tomato leaf diseases using the PlantVillage dataset. Gomaa and El-Latif [103] presented a real-time *CNN* method prediction of severe disease symptoms of Tomato Mosaic Virus (*TMV*) in different stages. Muhammad et al. [104] introduced an expert system based on a *DCNN* to predict tomato leaf disorders using the PlantVillage dataset. Nandhini and Ashokkumar [105] presented a novel binary solution encoding scheme based on the Improved Crossover Based Monarch Butterfly Optimization (*ICRMBO*) algorithm to minimize the optimization and complexity of the *CNN* model parameters. The VGG16 and InceptionV3 *DL* methods were optimized using the *ICRMBO* technique. The *DL* methods were trained on the PlantVillage dataset to detect tomato leaf diseases.

With the help of the *C-GAN*, Abbas et al. [106] introduced a *DL* DenseNet121 technique to identify tomato leaf disorders using the *TL* method. The researchers utilized the PlantVillage dataset. To enhance the efficiency of the models, a lightweight *CNN* was built in research [107] by adding several attention modules. The researcher enhanced the PlantVillage dataset by capturing the Fusarium wilt disease images from South Korea, and the total size of the dataset was 19510 images. In another study introduced by Chen et al. [108], the researchers made a mobile phone Android application that modified the AlexNet *DCNN* to classify the tomato leaf diseases using the PlantVillage dataset. Vadivel and Suguna [109] developed a *DCNN* to detect tomato leaf disorders using the PlantVillage dataset. Cengil and Çınar [110] presented a hybrid *DL* technique based on VGG16, ResNet50, and AlexNet pre-trained models to predict the tomato leaf diseases using the PlantVillage dataset. MobileNetV1, MobileNetV2, MobileNetV3, AlexNet, InceptionV3, and ResNet50 were the *DL* methods pre-trained on the ImageNet dataset in a study [111]. The methods were also installed on a Raspberry Pi 4 to create an Internet of Things (*IoT*) device to detect tomato leaf disease using the PlantVillage dataset. The authors [112] suggested a method for classifying tomato leaf diseases using transfer learning. The researchers used NASNetMobile and MobileNetV2 models to extract the features and then fed these features into different classifiers such as *MLR*, *SVM*, and *RF*. The researchers used the PlantVillage dataset. The validity and reliability of the results are improved by testing the suggested methods on a large dataset of tomato leaf images. Almost all the models are trained using the PlantVillage dataset, which is limited to a single geographic region. It meant that the diseases affecting tomatoes in the Pakistani region were unnoticed. The studies are limited to rice leaf disease detection and failed to detect potato and guava leaf diseases. These models only detect a single disease in a single leaf.

2.2.12 Tomato and Potato Leaf Disorders

Gokulnath and others [113] suggested an effective loss-fused *CNN* method for identifying plants affected by the disease. A Decision Tree (*DT*) and the K-Nearest Neighbour (*KNN*) algorithm could be used to identify disease based on spectral reflectance. The proposed method detected the tomato and potato leaves diseases utilizing the PlantVillage dataset. As a common issue in plant disease identification, inconsistent lighting is tackled head-on by the resilient lf-cnn method. There may be limitations to the suggested strategy's applicability because the dataset utilized in the experiments may differ from the diversity of plant diseases in the real world. Although this study belongs to potato leaf disease detection, it failed to detect potato and guava leaf diseases because it is trained on a specific region dataset. These models only detect a single disease in a single leaf.

2.2.13 Tomato and Grape Leaf Diseases

Zhao et al. [114] suggested a *DCNN* that integrated an attention module and residual blocks based on ResNet50 (SE-ResNet50) to diagnose the tomato and grape leaf diseases. For this purpose, the PlantVilage dataset and Grape Leaf Diseases dataset were used to train the proposed method. By training the model to zero in on the most informative parts of the input image for disease classification, the attention module could boost the model's discriminatory power. The model is trained using the PlantVillage dataset, which is limited to a single geographic region. It meant that the diseases affecting tomato and grape in the Pakistani region were unnoticed. The study is limited to tomato and grape leaf disease detection and failed to detect potato and guava leaf diseases. The model only detect a single disease in a single leaf.

2.2.14 Tea Leaf Diseases

Hu et Al. in [115] introduced a low shot learning technique for identification of tea leaf disorders. *SVM* was used for segmentation. A pre-trained VGG16 was trained with the augmented images to identify the diseased leaf accurately. An improved Faster *RCNN* was used by Hu et al. [116] to classify the tea leaf disease severity from images. The researcher used the VGG16 model to predict the severity of tea leaf diseases. Shadow and light variations of images were reduced by using the Retinex algorithm. The introduced technique can detect various disorders from a single leaf. Both the studies' experimental results are the limited sample size of the dataset used. The models used in the studies are limited to tea leaf diseases and specific region datasets; therefore, they failed to detect the tea leaf diseases of the Pakistani region and potato and guava leaf diseases.

2.2.15 Coffee Leaf Diseases

Sorte et al. in [117] developed an expert system to detect Cercospora, a disease of coffee, by using the Grey Level Co-Occurrence Matrix (*GLCM*) [118] and Local Binary Pattern (*LBP*) [119]. A segmentation and classification technique based on CNN was introduced by Esgario et al. [120] to estimate coffee leaf disease severity and classify the different diseases of coffee. The software relies on high-quality photos of coffee leaves, which may take much work in some coffee-growing regions. The study should have considered the costs of creating and maintaining the app, which may prevent farmers from using it. Both studies are limited to specific geographic regions; therefore, these models fail to detect the Pakistan region coffee leaf diseases. The scope of both studies categorized the coffee leaf diseases and failed to classify the potato and guava leaf diseases.

2.2.16 Coffee and Apple Leaf Diseases

A research was suggested by Hasan et al. [121] in which a segmentation model based on color analysis process and graph-cut technique to detect the coffee and apple leaf diseases. The introduced technique was trained on three different datasets: *RoCoLe*, coffee, and Apple. There is a strong correlation between the suggested method and high accuracy rates in detecting single and multiple diseases in coffee and apple plants, suggesting that it may have practical applications in the field. Possible limitations on the proposed method's applicability result from the article's omission of information on the time and money needed to implement it. The study is limited to specific geographic regions; therefore, it fails to detect coffee and apple leaf diseases in the Pakistan region. The scope of both studies categorized the coffee and apple leaf diseases and failed to classify the potato and guava leaf diseases.

2.2.17 Cotton Leaf Diseases

A self-created dataset from Pakistan developed 1,711 images of fusarium wilt, bacterial blight, and curl virus cotton leaf disease. A transfer learning technique based on EfficientNet-B0 was introduced by Noon et al. [122] to classify the cotton leaf diseases. This research proposes a computationally lightweight deep learning system for identifying cotton leaf diseases; this has potential utility in settings with limited computing resources. The proposed framework was only tested in a lab setting, so its results may vary when applied to the actual world. The scope of the study is to categorize the cotton leaf diseases and fails to classify the potato and guava leaf diseases. The dataset used in the study is limited in size.

2.2.18 Citrus Leaf Diseases

Luaibi et al. [123] proposed ResNet and AlexNet convolutional neural networks to predict the citrus leaf diseases. A self-created dataset was developed of 200 images. The sample size is rather small; thus, the results may not apply to all cases of citrus leaf diseases. Because of this, the results cannot be applied to a wider population. The study does not consider environmental factors that may affect disease detection accuracy. These factors include but are not limited to, lighting conditions and fluctuations in plant development. As this study was conducted in a laboratory context, the outcomes may change when applied to a more realistic agricultural setting with a wider range of variables. To see which technique performs better in disease identification, Sujatha et al. [124] used both *ML* (*SVM*, *RF*, *SGD*) and *DL* (Inception-v3, VGG-16, VGG-19,) to categorize citrus leaf disease. They used a self-created dataset from the Pakistan region to identify the melanose, greening, canker, black spot diseases, and healthy citrus leaves. Researchers and practitioners in the area will benefit from the study's thorough comparison of deep learning and machine learning approaches to plant leaf disease identification. The study's conclusions are bolstered by using a sizable dataset of plant leaf photographs, enhancing the study's generalizability to real-world situations.

Both the studies are trained on a small scale dataset. The scope of the study is to categorize the citrus leaf diseases and fails to classify the potato and guava leaf diseases.

2.2.19 Grapes Leaf Diseases

Many researchers [125–127] proposed different deep-learning models to classify grape leaf diseases. An integrated version of GoogLeNet (InceptionV3) and ResNet50 named UnitedModel was proposed by Ji et al. in [125] to learn more discriminative features for the classification of grape leaf diseases. Xie et al. [126] presented a Faster DR-IACNN based on Faster *RCNN* using the InceptionV1, Inception-ResNetV2 module, and SE-blocks to identify the grape diseases. Another study was introduced by Kaur et al. [127] in which a transfer learning-based model was developed to identify grapevine leaf disorders. The studies do not consider environmental conditions and soil quality, which might affect disease development in grapes. The studies are limited to grapes and failed to detect potato and guava leaf diseases. The scope of all the studies is limited to grapes; therefore, these studies fail to classify the potato and guava leaf diseases.

2.2.20 Grapes and Tomato Leaf Diseases

Paymode and Malode [128] conducted research based on transfer learning of Visual Geometry Group (*VGG*) *CNN* to classify the tomato and grapes leaf disorders using PlantVillage dataset. Because of the modest size of the dataset utilized in the paper, the findings cannot be applied to other datasets or different types of crops such as potato and guava. The generalizability and accuracy of the results may be compromised due to the article's failure to address possible biases in the dataset. The dataset is limited to a specific geographic area, so the study fails to detect the Pakistani region's diseases.

2.2.21 Grapes and Cucumber Leaf Diseases

Chen et al. [129] presented a *DL*-based segmentation and classification technique (*SegCNN*) that predicted the cucumber and grape diseases. The researcher created a 500 images dataset of cucumbers with slight, moderate, and severe severity diseases with normal cucumber leaves. The PlantVillage dataset was used for grape disease detection to classify the black measles, leaf blight, grape black rot diseases, and healthy leaves. Accurately identifying disease signs in plant photos is the goal of the proposed method, which includes color-based characteristics, texture-based features, and shape-based features. The scope of the study is to categorize the grape and cucumber leaf diseases and fails to classify the potato and guava leaf diseases. The dataset used in the study is limited in size and for a specific geographic region; therefore, it fails to classify the Pakistani region plant leaf diseases.

2.2.22 Peanut Leaf Diseases

Qi et al. [130] developed a logistic regression stack ensemble technique based on InceptionV3, DenseNet121, ResNet50, VGG16, and AlexNet for peanut diseases classification. They also used *LR*, *SVM*, and *RF* were chosen as the meta-model. The methods were trained on a self-created dataset of 6029 images, including scorch, leaf spot, and rust diseases developed from China. The proposed model can detect multiple diseases on a single leaf. This study introduces a unique approach to disease detection utilizing a stacked ensemble of different classifiers to enhance the accuracy and reliability of illness identification in peanut-leaf samples. The suggested approach integrates color, texture, and form features to correctly identify illness symptoms in photos of peanut leaves. The downsides of the suggested strategy, such as its dependence on image quality or the requirement for large amounts of training data, should be addressed in the work. The study is limited to peanut leaf diseases and failed to classify the potato and guava leaf diseases.

2.2.23 Olive Leaf Diseases

Uğuz and Uysal [131] proposed the transfer learning techniques based on VGG19 and VGG16 to classify the olive leaf diseases from Turkey. The researchers trained the models on data augmentation techniques used on a self-created dataset and without data augmentation techniques. The dataset consisted of 3400 images. The suggested method is thoroughly tested on a large dataset of olive leaf photos, and its success in accurately classifying multiple olive leaf diseases is demonstrated in the publication. The dataset used to test the approach was obtained in a lab, which may differ from the complexity and variation seen in actual photographs of olive leaves. Because of the high computing requirements of the suggested method, it may not apply to low-power devices or situations with limited resources. The study used a specific region dataset, thus failing to classify the Pakistani region's olive leaf disease. The scope of the study is also limited to olive leaf diseases; therefore, it fails to classify the potato and guava leaf diseases. The study is limited to finding a single disease on a single leaf.

2.2.24 Mango Leaf Diseases

Singh et al. in [132] developed a Multilayer Convolutional Neural Network (*MCNN*) to detect the anthracnose fungal disease of mango leaves. They used a PlantVillage dataset, and real-time healthy and diseased leaves images of mango. The researchers applied histograms to equalize the images and then resized the images by the central square crop method. The proposed approach was tested on a small subset of mango leaf diseases; its applicability to other plants and pathogens cannot be guaranteed. Mango leaf photos captured in the wild tend to be more complicated and variable than the dataset used to evaluate the approach. The scope of the study is limited to mango leaf disease; therefore, it fails to classify the potato and guava leaf diseases. The study can find a single disease on a single leaf.

2.2.25 Kiwifruit Leaf Diseases

Yao et al. [133] proposed a two-stage deep learning model to strip the kiwifruit leaves from the complex natural environment using the YOLOX model. The researchers developed the dataset of 2000 kiwifruit disease images from China for bacterial canker and brown spot disease. Then DeepLabV3+ model was employed to extract the color and texture feature, and the ResNet101 model was used to detect the diseases of kiwifruit on leaves. First, a pre-trained CNN is used for coarse classification, and then, in the second stage, a CNN is modified to identify the disease type accurately. Due to its high computational cost, the suggested technique may need to be more practical for application on low-power devices or in contexts with limited resources. The scope of the study is limited to kiwifruit leaf disease; therefore, it fails to classify the potato and guava leaf diseases. The study can find multiple diseases on a single leaf.

2.2.26 Cassava Leaf Diseases

For the classification of Cassava leaf disease, an attention mechanism with pre-trained CNN-based models was used by Ravi et al. [134]. The model added an attention layer to the EfficientNet model, and features from EfficientNetB5, EfficientNetB6, and EfficientNetB4 were extracted from the penultimate layer. KPCA was used to minimize the extracted feature dimensions, and the feature was fused. For classification, meta-classifiers were utilized. In the meta classifier, the first stage employed RF and SVM for predictions, and the second stage used logistic regression for classification. The researcher trained their model with the publicly available Cassava Leaf Dataset to classify the Cassava Mosaic Disease (*CMD*), Cassava Green Mite (*CGM*), Cassava Brown Streak Disease (*CBSD*), and Cassava Bacterial Blight (*CBB*) with healthy leaves of cassava. By training on a huge dataset (over 12,000 photos), the classifier can better withstand variations in both the environment and disease progression. In reality, it may be crucial to consider the constraints of the proposed method and potential causes of mistakes, neither of which are discussed in the article. Results may not apply to other regions with varying climatic circumstances and illness frequency because the dataset utilized in the study is restricted to a single country (Uganda).

The *CMD*, *CGM*, *CBSD*, and *CBB* with healthy leaves of Cassava Leaf Dataset. Anitha and Saranya [135] developed the Convolutional Neural Network (*CNN*) to detect the cassava leaf diseases. Because the study only used photos of cassava plants from a particular site, the results may need to be more generalizable to the wide variety of cassava diseases in other parts of the world. The study should have included the logistical obstacles to deploying the deep learning method in practice, such as the need for specific hardware or personnel.

A comprehensive deep learning technique based on Enhanced Convolutional Neural Network (*ECNN*) models for real-time Cassava leaf disease detection was developed by Lilhore et al. [136]. The *CMD*, *CGM*, *CBSD*, and *CBB* with healthy leaves of Cassava Leaf Dataset. In the suggested Enhanced Convolutional Neural Network (*ECNN*) model, *CNN* difficulties were resolved using a depth-wise separable convolution layer. The study's dataset, including its collection, preprocessing, and augmentation methods, is described in depth. The study's findings may not apply across all cases of cassava leaf disease or all locations because of the small sample size. Environmental factors, such as climate and soil conditions, have been ignored in this study despite their potential influence on the onset and severity of diseases affecting cassava leaves.

Because of the modest size of the dataset utilized in all studies, the findings cannot be applied to other datasets or different types of crops such as potato and guava. The generalizability and accuracy of the results may be compromised due to the studies' failure to address possible biases in the dataset. The dataset is limited to a specific geographic area, so the studies fail to detect the Pakistani region's diseases. All the approaches can identify a single disease on a single leaf.

2.2.27 Beans Leaf Diseases

A deep convolutional neural network-based pre-trained MobileNetV2 model was proposed by Elfatimi et al. [137] to classify the beans leaf diseases. The proposed method was trained on a self-created dataset of 1296 images to detect beans' rust and angular leaf spot diseases. Since only 1296 pictures were used in the analysis, the findings may only apply to some bean leaf diseases or geographic areas (Uganda). In reality, it may be crucial to consider the constraints of the proposed method and potential causes of mistakes, neither of which are discussed in the study. The scope of the study and the size of the dataset is limited. The study does not find the potato and guava leaf disease. It can detect a single disease on a single leaf.

2.2.28 Squash Leaf Diseases

Research conducted by Ganesh Babu and Chellaswamy [138] in which LPDBL-based deep learning model was developed to detect the squash leaf diseases. A squash dataset was developed for severity estimation (early, middle, and critical) of powdery mildew of squash leaf diseases and achieved 99% accuracy. High detection rates across all disease phases were achieved with the suggested method, proving the efficacy of machine learning in plant disease diagnostics. Findings may not apply to other regions with varying climatic circumstances and illness frequency because the dataset utilized in the study was restricted to a single country (India). The scope of the study is limited to squash leaf diseases, and it can identify a single disease on a single leaf.

2.2.29 Palm Leaf Diseases

Ibrahim et al. [139] proposed a deep *CNN* developed to detect palm leaf diseases. A dataset of 350 images was developed to classify the palm nutrient deficiencies such as manganese, zinc, boron, magnesium, potassium, and nitrogen with healthy leaves. As a result of the proposed method's success, deep learning is useful in plant nutrient diagnosis. The study's dataset was collected only in one country (Malaysia); hence its results may not apply to other countries with various climates and nutrient deficiency problems. Little dataset (350 photos) and no discussion of how elements like illumination and camera angle could affect categorization performance could limit the suggested approach's applicability.

2.2.30 Leaf Diseases of Multiple Crops

Various researchers [33, 140–160] developed different deep learning models to investigate the multiple crop leaf diseases. Barbedo in [140], used a pre-trained GoogLeNet to detect the different plant leaf diseases such as cassava, common bean, citrus, corn, coconut tree, kale, coffee, passion fruit, cotton, sugarcane, soybean, and wheat crops by using the individual spots and lesions instead of the entire leaf. The Plant Disease Database (*PDDB*) and expanded XDB datasets were used for training and testing. Ferentinos in [33] used pre-trained AlexNet, Overfeat, AlexNetOWTBn, *VGG*, and GoogLeNet deep learning-based architectures, to identify the normal or abnormal plant images of plant leaves. The models were trained on a publicly available dataset of plant images [161], having 87,848 plants leave images of both real agricultural and laboratory environments. The models detect the 58 distinct types of plant and disease combinations of banana, apple, cabbage, celery, cassava, cherry, cucumber, corn, grape, gourd, peach, orange, potato, pepper, soybean, pumpkin, strawberry, squash, tomato plants. Geetharamani & Pandian [141] proposed a model to differentiate between healthy and unhealthy crops based on the characteristics of leaf images. The model was trained on the PlantVillage dataset. The system identified apple, blueberry, cherry, corn, grapes, orange, peach, pepper, potato, raspberry, soybean, squash, and tomato crop diseases. Using depthwise separable convolution (DSC) rather than the convolution layer, Kamal et al. [142] constructed plant disease identification models called Modified MobileNet and Reduced MobileNet. The models can detect 55 plant leaf diseases of the PlantVillage dataset, including potatoes. A hybrid method involving CNN and autoencoders was proposed by Khamparia et al. in [143] for detecting agricultural leaf disease. Images of diseased and healthy leaves from three crops were used: potatoes with late blight and early blight, tomatoes with leaf mold and yellow leaf curl virus, and maize with rust. The PlantVillage dataset was used for training.

Ahmed and Reddy [144] developed a *CNN*-themed Android mobile application. The 14 different crops each had their unique disease profile, which was modeled using the PlantVillage dataset throughout the training process. The segmentation models proposed by Thenmozhi et al. [145] for classifying tomato, grape, and apple leaf diseases were as follows: The proposed method extracted green pixels using *HSV*, color space, and other masking schemes. The study used the PlantVillage dataset. Kushal et al. [146] built five convolutional neural networks from the ground up, including MobileNet, AlexNet, ResNet, and two additional deep-learning models. The aforementioned deep learning models were taught to recognize the many diseases that affect 58 different crops by using the PlantVillage dataset as their training ground. AR et al. [147] presented a hybrid method based on k-means clustering to analyze a leaf-based Region of Interest (*ROI*). The data set from the University of California, Irvine (UCI) was used to train the proposed approach to detect illnesses such as Cercospora, anthracnose, common rust, Alternaria diseases, and combinations of bacterial blight diseases. Wagle and others [148] compared the traditional *SVM* with modified AlexNet to detect the nine different crop diseases. The proposed method used the PlantVillage dataset. A big plant leaves image dataset from different countries was used to train a dense convolutional neural network architecture by Tiwari et al. [149]. The proposed method was trained on the rice leaf diseases, citrus leaf, bean leaf image, and PlantVillage datasets. Hassan et al. [150] developed different transfer learning models to detect multiple crop diseases using the PlantVillage dataset. Using the Grey Level Co-Occurrence Matrix (*GLCM*), the authors of [151] obtained six color features and twenty-two texture features. The *SVM* was utilized to classify one-to-one diseases. The technique above received its training on the PlantVillage dataset, and it can identify a variety of crop illnesses. Using the PlantVillage dataset, Atila et al. [152] suggested a deep learning model built on EfficientNet. Their goal was to identify several illnesses that could affect various crops. The tomato, potato, pepper, and bell crop illnesses were investigated in a study [153], which led to the development of an ensemble network. The suggested method retrieved the hybrid features using Local Binary Pattern (*LBP*), Law's mask, Scale-Invariant Feature Transform (*SIFT*), Gabor, and GLCM to enhance the classification results on the PlantVillage dataset.

Hossain et al. [154] proposed threeDepth-Wise Separable Convolutional PLD (*DSCPLD*) models to reduce the computational cost and model size. These models include segmented extended DSCPLD (S-extended MobileNet), reduced segmented DSCPLD (S-reduced MobileNet), and modified segmented DSCPLD (S-modified MobileNet). The suggested method made use of the PlantVillage dataset as well as the Rice disease picture dataset. Additionally, a new dataset consisting of 6580 photos was produced to detect the various illnesses that affect crops such as cherry, mango, apple, potato, rice, grape, and pepper. Research carried out by Li and Chao [155] in which a semi-supervised few-shot deep learning-based approach was developed to classify the various diseases that can affect a wide variety of plant species. They also made use of the PlantVillage information to identify a variety of diseases affecting multiple crops. He and his colleagues [156] proposed a Disease Image Recognition Method based on Bilinear Residual (*DIR-BiRN*). The technique was trained on the PlantVillage dataset to identify the various diseases affecting tomato, potato, grape, maize, apple, and combination disorders. Albattah et al. [157] introduced a customized CenterNet model with DenseNet77 for feature computation to improve plant disease recognition and classification accuracy while reducing training and testing time complexity. They used the PlantVillage dataset to categorize the different crop diseases. The proposed method can detect diseases in apples, custard apples, and guavas; specifically, the technique distinguished between healthy apple leaves and those with diseases such as black rot, rust, and scab. In India, Gaikwad et al. [158] created the apple dataset, the custard apple dataset, and the guava dataset to train deep learning-based CNN models such as SqueezeNet and AlexNet with the same hyperparameters. Singh et al. [159] employed five distinct convolutional neural networks (*CNN*s), such as MobileNet, EffNet, AlexNet, ShuffleNet, and LeNet, to differentiate between the several diseases that can affect various types of crops. The model mentioned above was trained using the PlantVillage dataset, which was used to train the model to identify the various illnesses affecting rice, tomato, potato, maize, and apple plants. Another research was conducted by Wang [160] to develop an improved AlexNet model to detect different plant diseases. The proposed method can detect diseases and pests from peach, potato, corn, rice, and pear plants. The researcher used the PlantVillage dataset and developed a 1200 images dataset from China.

Almost all the studies detect different plant leaf diseases, including potatoes, but these studies have different problems. Almost all the models are trained using the PlantVillage dataset, which is limited to a single geographic region. Therefore, these techniques have a high false rate of detecting potato diseases in the Pakistani region. The dataset, including most studies, has imbalanced classes, which can cause overfitting. All the models can detect a single disease on a single leaf, but these methods fail to identify multiple diseases on a single guava leaf. Almost all the models have high computational costs. The impact of environmental factors, such as climate and soil conditions, on plant leaf diseases, which might alter disease development and severity, needs to be considered.

2.2.31 Potato Leaf Diseases

A large number of researchers researched diseases that could affect potato crops, and they also trained algorithms on the dataset known as PlanVillage. A rundown of the illnesses that can affect potato leaves is presented in Table 2.2.

A CNN model was proposed by Khalifa et al. [162] to detect early blight and late blight diseases in addition to a healthy class. The researchers used the PlantVillage dataset to train their algorithm. This dataset only contains information on the crops grown in certain places. A CNN model was proposed by Rozaqi and Sunyoto [163] to detect the early blight illness of potatoes, as well as the late blight disease of potatoes, and a healthy class. They trained the model on the PlantVillage dataset to identify the prevalent diseases in a particular area. Sanjeev et al. [164] suggested a Feed-Forward Neural Network (*FFNN*) for early and late blight detection to identify both diseased and healthy leaves. The PlantVillage dataset was used for both training and testing the suggested approach. Barman et al. [165] presented a Self-Build CNN (*SBCNN*) model to identify the early blight, late blight, and healthy class in potato leaf samples. The model, tailored to a specific area, was trained using data from the PlantVillage dataset. The proposed model was also not verified by predicting the unseen data (test data).

Using a pre-trained model VGG19, Tiwari et al. [166] extracted features and classified them using KNN, SVM, and a neural network. They didn't put the model through a blind test on new information. The program learned to identify early and late blight on potato leaves by analyzing data from the PlantVillage dataset. Using a *CNN* model, Lee et al. [167] could identify potato leaves that were either healthy or affected by early and late blight. The researchers also used a regional PlantVillage dataset. We did not put the model through a blind test on new information.

To identify potato diseases like early blight, late blight, and healthy leaves, Islam et al. [168] proposed a segment-based and multi-SVM-based model. Their approach was similarly inaccurate despite making use of the PlantVillage dataset. Based on feature fusion and Principal Component Analysis-Linear Discriminant Analysis (*PCA-LDA*) feature extraction, Feature Fusion Based PCA-LDA (FF-PCA-LDA) was proposed as a classification technique for identifying potato leaf diseases by Ali et al. [169]. The researchers used the PlantVillage dataset, which had information on illnesses like late blight and early blight that can affect potatoes, as well as information on healthy leaves. An accuracy of 98.20% was achieved using the proposed strategy.

2.2.32 Guava Leaf Diseases

Previously much work has been conducted on guava leaf diseases, but the methodology exercised was only focused on detecting a single disease from a single leaf or fruit as shown in Table 2.3.

Howlader et al. [172] proposed a Deep Convolutional Neural Network (*DCNN*) consisted of eleven

Table 2.2: Summary of Potato Leaf Diseases Related Work.

Reference	Methodology	Plant Name	Disease	Dataset	Accuracy
[141]	Deep CNN	Multiple (Potato)	Multiple	PlantVillage	96.46%
[142]	Modified MobileNet	Multiple (Potato)	Multiple	PlantVillage	98.34%
[143]	CNN and Autoencoders	Potato, Maize, Tomato	Multiple	PlantVillage	97.50% 100%
[170]	ResNet50	Multiple (Potato)	Multiple	PlantVillage	98%
[171]	AlexNet, Overfeat AlexNetOWTBn, VGG and GoogLeNet	Multiple (Potato)	Multiple	PlantVillage	99.53%
[162]	CNN	Potato	Early Blight, Late Blight	PlantVillage	98%
[163]	CNN	Potato	Early Blight, Late Blight	PlantVillage	92%
[164]	FFNN	Potato	Early Blight, Late Blight	PlantVillage	96.5%
[165]	SBCNN	Potato	Early Blight, Late Blight	PlantVillage	96.75%
[166]	SVM, KNN and Neural Net	Potato	Early Blight, Late Blight	PlantVillage	97.8%
[167]	CNN	Potato	Early Blight, Late Blight	PlantVillage	99%
[168]	Segment and Multi SVM	Potato	Early Blight, Late Blight	PlantVillage	95%
[169]	FF-PCA-LDA	Potato	Early Blight, Late Blight	PlantVillage	98.20%

layers in which four convolution layers followed by four max-pooling with ReLU and three fully connected layers to detect the algal leaf spot, whitefly and rust diseases of guava along with healthy leaves. They developed their own BU Guava Leaf 2018 (*BUGL2018*) dataset, which contains 2705 images from Bangladesh. Their proposed method achieved 98.74% accuracy on the test dataset.

Al Haque et al. [173] proposed a deep learning method based on *CNN* to detect the fruit canker, anthracnose and fruit rot. The research was conducted on guava fruit in Bangladesh and achieved 95.61% accuracy but it did not detect the multiple diseases on a single leaf.

Almadhor et al. [174] proposed an AI-Driven framework to detect the rust, canker, mummification and dot diseases on guava leaves and fruit, based on high imagery resolution sensor Digital Single-Lens Reflex (*DSLR*) camera. The dataset contains only 393 sample images. *RGB*, *HSV* colour

Table 2.3: Summary of Guava Leaf Diseases Related Work.

Ref.	Methodology	Plant Name	Disease	Single Leaf Single Disease (SLSD) or Single Leaf Multiple Disease (SLMD) Detection	Localization	Accuracy
[173]	CNN	Guava	Fruit Canker, Anthracnose, Fruit Rot	SLSD	No	95.61%
[172]	CNN	Guava	Rust, Algal Leaf Spot, White Fly	SLSD	No	98.74%
[174]	Complex Tree, Fine KNN, Bagged Tree, Boosted Tree. & Cubic SVM	Guava	Dot, Mummification, Rust, Canker	SLSD	No	99%
[175]	SVM	Guava	Bacterial Blight, Anthracnose	SLSD	No	98.17%

histogram and *LBP* texture features were extracted and Decision Tree (*DT*), Fine *KNN*, Bagged Tree (*BT*), Boosted Tree, and Cubic *SVM* classifiers were used to catalog the guava leaf and fruit diseases. Bagged Tree classifier achieved the 99% accuracy.

Perumal et al. [175] proposed a single disease detection technique based on Support Vector Machine (*SVM*) to classify the healthy, bacterial blight and anthracnose diseases on a single guava leaf only and the proposed method achieved 98.17% accuracy.

2.3 Plant Leaf Segmentation Techniques

Leaf segmentation with complex background was also a challenging and complex task. Few researchers worked on different plant leaves segmentation such as [176–191].

To segment leaf images with intricate backgrounds based on past shape knowledge, a method known as automatic marker-controlled watershed segmentation was first introduced by Wang et al. [176], which combines pre-segmentation and morphological operations. After leafstalk removal, seven Hu geometric moments and sixteen Zernike moments are retrieved as shape characteristics from segmented binary pictures. A self-collected dataset of 1200 images was used, with 20 classes including 60 leaf samples for each class with complex background. They segmented the willow, plum, London plane tree, China redbud, chestnut, laurel, rose bush, hazel, panicled goldrain tree, sweetgum, honeysuckle, donglas fir, maple, arrowwood, tulip tree, ginkgo, photinia, camphor tree, seating, and Chinese allspice and got 92.6% accuracy.

Research conducted by Itakura and Hosoi [181] in which automatic leaf segmentation and parameters of plant structure were retrieved using point-cloud 3D images. The leaves were segmented automatically with 3D models combined with 2D and 3D point-cloud processing techniques. The researchers used small plants such as Japanese sacandra, umbellate, kangaroo vine, council tree, dwarf schefflera, hydrangea, and pothos. The proposed 3D model achieved 86.9% accuracy.

Ward et al. [182] trained a state-of-the-art pre-trained Mask *RCNN* model to segment the synthetic and real images of the arabidopsis plant. The researchers used the ADE20K and *CVPPP LSC* datasets to train the model. The proposed model got 90% leaf segmentation accuracy on the A1 test set of the *CVPPP* Leaf Segmentation Challenge (*LSC*) and obtained 81% mean accuracy over all five test sets.

Kumar and Domnic [184] conducted research in which plant leaf region was extracted and then counted the rosette plant leaves. Three steps were involved. The first step image enhancement method was developed using a novel statistical-based method—the second step involved segmentation of the plant's region using a graph-based technique. In the third step, leaves were counted using Circular Hough Transform (*CHT*). They used the Leaf Segmentation Challenge (*LSC*) dataset, which consisted of A1, A2, and A3 sets. The proposed method achieved 95.4% segmentation accuracy and obtained 0.7 and 2.3 dice of leaves counting for A1, A2, and A3 sets.

Gimenez-Gallego et al. [186] segment the multiple tree leaves like lemon, orange, almond, olive, loquat, fig, cherry and walnut trees from the natural background using the Support Vector Machine

(*SVM*) and SegNet deep learning model. The proposed method was trained on a self-created dataset of 251 images. SVM achieved 83.11% accuracy, and SegNet gained 86.27% accuracy.

To carry out leaf phenotyping on two greenhouse ornamentals—Maranta arundinacea and Dieffenbachia picta—Li et al. [187] proposed a novel five-stage framework. It included multiview stereo point cloud reconstruction, preprocessing, stem removal in the canopy, leaf segmentation, and leaf phenotypic feature extraction. For each leaf, phenotypic features such as leaf area, length, width, and inclination angle were determined and compared to ground realities. For the two species, 96.8% and 97.8% of the predicted leaf area were accurate.

According to the study, a new orthogonal transform-based plant region segmentation scheme was proposed [188]. First, orthogonal transform coefficients were analyzed regarding orthogonal basis vectors' response to extracting the plant region from the orthogonal basis vectors. L*a*b and the Cyan, Magenta, Yellow, Key (*CMYK*) colour spaces were utilized for noise removal after extracting the plant region. Finally, deep convolutional neural network models were used to count the leaves. They used the Computer Vision Problems in Plant Phenotyping (*CVPPP*) dataset, which included A1, A2, and A3 sets. Datasets A1, A2 and A3 had an Foreground-Background Dice (*FBD*) of 94.7%, and a dice score of 93.72% was attained using the suggested method.

Research conducted by Yang et al. [189] in which 15 species segmentation was performed using Mask *RCNN* deep learning model used VGG16 deep learning model. They used 2500 images dataset of 15 species, including gardenia jasminoides, callisia fragrans, psidium littorale, Osmanthus fragrans, bixa Orellana, ficus macrocarpa, calathea makoyana, rauvolfia verticillate, Ardisia quinquegona, baccaurea ramiflora, synesepalum dulcificum, Hydnocarpus anthelminthic, daphne odora, dracaena surculose, and mussaenda pubesens. The proposed method achieved 91.5% accuracy.

Plant leaf extraction was proposed by Amean et al. [191] using depth information from a stereo vision sensor. The system used synergistic features such as depth, shape, and colour to deal with multiple leaf segmentation and overlapping leaf separation. The disparity maps employed depth to measure discontinuities in its gradient. The algorithm was tested on 272 images of hibiscus and cotton plants, with the results showing that depth attributes were successful in distinguishing between occluded and overlapped leaves, with a separation rate of 84%. Over a variety of diverse backgrounds and changing plant canopies, the system could distinguish individual plant leaves at a rate of 78%.

Table 2.4: Summary of Plant Leaf Segmentation Related Work.

Reference	Methodology	Crop	Dataset	Accuracy
[176]	automatic marker-controlled watershed segmentation	Multiple	Self-Created	92.6%
[181]	point-cloud 3D	Multiple	Self-Created	86.9%
[182]	RCNN	Multiple	ADE20K, CVPPP LSC	90%
[184]	Graph-based Method	Rosette	Rosette, LSC	95.4%
[186]	SVM, SegNet	Multiple	Self-Created	SVM: 83.11% SegNet: 86.27%
[187]	Five-stage Framework	Multiple	Self-Created	97.8%
[188]	DCNN	Rosette	CVPPP	94.7%
[189]	Mask RCNN	Multiple	Self-Created	91.5%
[191]	Disparity Maps	Hibiscus, Cotton	Self-Created	84%

2.4 Publicly Available Datasets

Since deep learning-based plant disease identification is a relatively new field, only a few benchmarking datasets exist for evaluating the deep learning-based plant disease identification method. The potato and guava publicly available datasets are described below.

2.4.1 Potato Leaf Disease Datasets

In literature, all the potato leaf disease datasets consisted of a single disease on a leaf. The researchers used the following potato leaf disease datasets:

2.4.1.1 Potato Leaf Disease Dataset

The Potato Leaf Disease Dataset was publicly available on Kaggle [192]. It contained 1500 images of three classes: late blight, early blight diseases, and healthy leaves. The dataset contained only a single disease on a leaf, and each class consisted of 500 jpg images of size 256 × 256.

2.4.1.2 Potato Leaf (Healthy and Late Blight)

The potato Leaf dataset was developed by Addis Ababa Science and Technology University [193]. It consisted of only two classes late blight and healthy. The healthy class comprised 363 jpg images, and the late blight class comprised only 63 images. The data was collected with the help of a smartphone and digital camera in an uncontrolled setting on a working potato field in Holeta, Ethiopia.

2.4.1.3 PlantVillage Dataset

The PlantVillage dataset [161] was developed by Penn State University (US) and EPFL (Switzerland), which is a non-profit project. The database consists of JPG colour images with 256 × 256 dimensions. It has 38 classes of diseased and healthy leaves of 14 plants. In the PlantVillage dataset, the potato crop has three types: early blight, late blight, and healthy. It consisted of 1000 leaves for late blight, 1000 for early blight, and 152 images of healthy leaves.

2.4.2 Guava Leaf Disease Datasets

We found only two publicly available datasets of guava which focus only on a single disease on a leaf. The publicly available datasets are presented below.

2.4.2.1 Guava Fruits and Leaves Dataset

Guava fruits and leaves dataset [194] was developed by Rauf and Lali from tropical areas of Pakistan. The dataset was developed to detect a single disease on a leaf and fruit. It consisted of rust, mummification, canker, and dot diseases of both leaves and fruit. It consisted of 306 images having 300 dpi resolution and 6000 × 4000 sizes. The dataset contained 70 images of rust, 83 images of mummification, 77 images of canker, and 76 images of dot diseases.

2.4.2.2 Guava Disease Dataset

The Guava disease dataset was found on Kaggle [195], which consisted of 2300 images. It consisted of five classes: rust, mummification, canker, dot diseases, and healthy guava leaves. The dot class consisted of 178 images, canker had 278, mummification class attained 275 images, rust class contained 279 images, while the healthy class contained 1289 jpg images. The guava disease dataset also included a single disease on a single guava leaf.

2.4.3 Plant Leaf Species Datasets

Regarding computer vision, investigations of plant categorization using image processing have recently emerged as a hot area. Numerous datasets in the literature can be used to evaluate plant classification systems. These datasets include: ICL [196], MalayaKew [197], PlantCLEF [198], LeafSnap [199], Swedish [200], Foliage [201], and Flavia [202]. A wide range of difficulties, including fine-grained complexity, imbalanced distribution, substantial intraclass variability, modest interclass variability, and noisy pictures, are well-illustrated by these datasets, which reflect the issue area. In the literature, no dataset included potato and guava leaf species.

2.5 Convolutional Neural Network (CNN) Architecture

The development of powerful processing devices like the *GPU* has led to the emergence of new fields of application connected to *DL*. Traditional artificial neural networks inspired the idea of DL. To extract the essential information, CNN played a crucial role in deep learning by stacking several preprocessing layers. These features were sent to fully linked levels to make a call. After Krizhevsky et al. [203] demonstrated remarkable success at image classification using *CNN* in 2012, *DL* models expanded rapidly. Since then, *CNN* has found use in a wide variety of *DL* applications, including but not limited to pattern identification, image classification, object detection, voice recognition, and many more [204, 205]. Instead of traditional neural network layers, CNNs use convolution techniques [206, 207]. Traditional feature extraction handcrafted methods often depend on expert knowledge and need expensive human labor, while CNN provides automatic feature extraction. A convolutional neural network (CNN) is an architecture of many layers, each responsible for a different feature extraction and transformation aspect. CNNs are built with several layers, the most important being convolutional, pooling, and fully connected layers, as shown in Figure 2.1. Each CNN consists of two parts:

1. Feature Learning
2. Classification

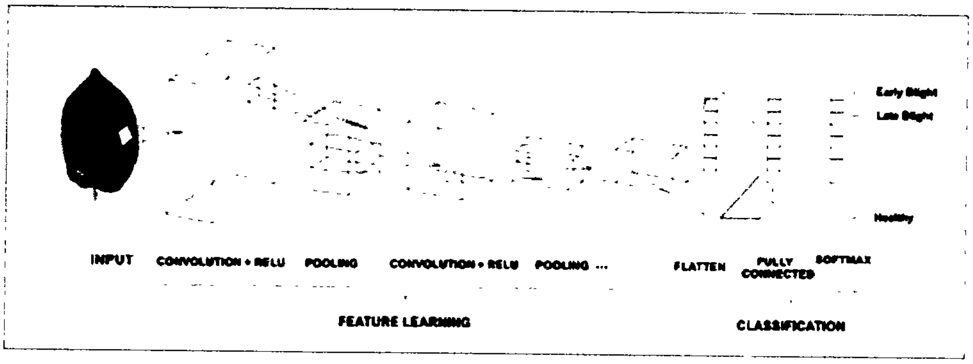


Figure 2.1: Convolutional Neural Network (CNN) Architecture.

2.5.1 Feature Learning

The automatic feature learning task of the CNN model is the responsibility of the following layers:

1. Convolutional Layer
2. Pooling Layer

2.5.1.1 Convolutional Layer

CNNs rely on their convolutional layers to function. The input image is convolved with a set of learnable filters (kernels) to generate feature maps. Each filter is optimized for discovering unique characteristics or patterns in the raw data. Filters in lower convolutional layers extract the small-scale features; conversely, filters in the upper convolutional layers will extract the large-scale features, as shown in Figure 2.2 [208].

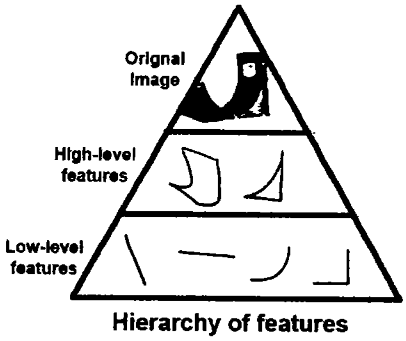


Figure 2.2: Feature Hierarchy.

The mathematical definition of convolution layers, where the 2D image matrix (I) is convolved with the smaller 2D kernel matrix (K), is provided by Equation 2.1. Convolution layers are widely employed in digital image processing.

$$S_{ij} = (I * K)_{ij} = \sum_{m}\sum_{n} I_{ij} \cdot K_{i-m,j-n} \tag{2.1}$$

When using a filter bank or kernels, the input from a layer designed to detect characteristics will be spatially translated without modification to the output [209]. Each filter or kernel is designed to detect a specific feature at each input point. As it is defined by LeCun [209], there is a bank of m_1 filters in each convolutional layer, and the output $Y_i^{(l)}$ of the lth layer consists of $m_1^{(l)}$ feature maps of size $m_2^{(l)} \times m_3^{(l)}$. The ith feature map is computed as follows:

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{ij}^{(l)} \times Y_j^{(l-1)} \tag{2.2}$$

$K_{ij}^{(l)}$ is the filter with dimensions $(2h_1^{(l+1)} \times 2h_2^{(l+1)})$ that connects the jth feature map of the $(l-1)$ layer to the ith feature map of the lth layer, and $(\times)$ is the 2D discrete convolution operator. B_i^l is the trainable bias parameters matrix.

In convolution, a tiny filter is vertically slid from left to right across the entire image. Figure 2.3 shows a convolution operation with a 3x3 convolution kernel applied to a 4x4 input image. Each node's output is the sum of the products of the kernel elements at that node times their respective input values. This is repeated with different kernels until enough feature maps have been generated [210].

The dimensions of the output characteristics map are less than those of the input maps. A padding strategy [210, 211] can preserve the same in-plane dimension by inserting zeros around the input and fitting the kernel's center to its peripheral parts. Additionally, a stride is the distance between two consecutive kernel nuclei. Before subsampling, feature maps are typically down sampled to a resolution of 1 stride. However, occasionally longer strides will be required.

The combination of different parameters in convolutional layers affects the overall performance of the proposed model. The following are the important building blocks or parameters that affect the performance of a convolutional layer:

- **Data Format:** An input image's parameter height, width, and depth (channel) is defined in

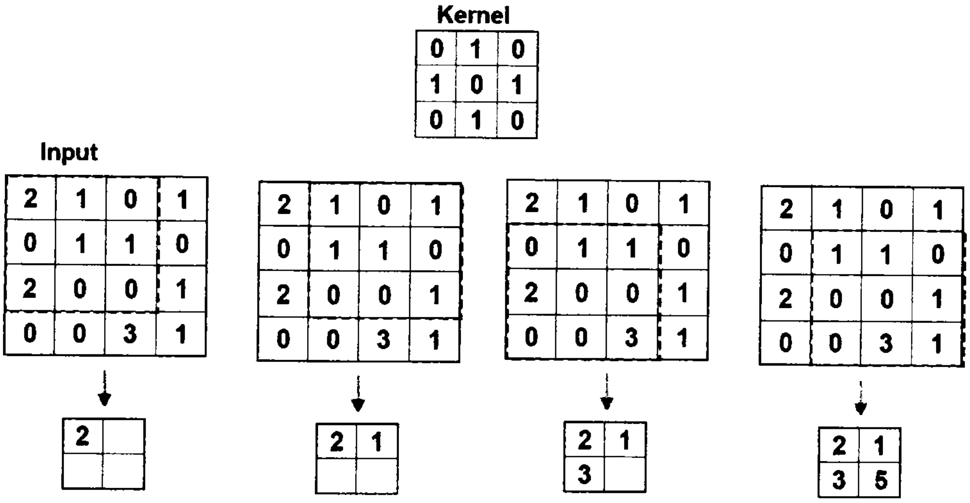
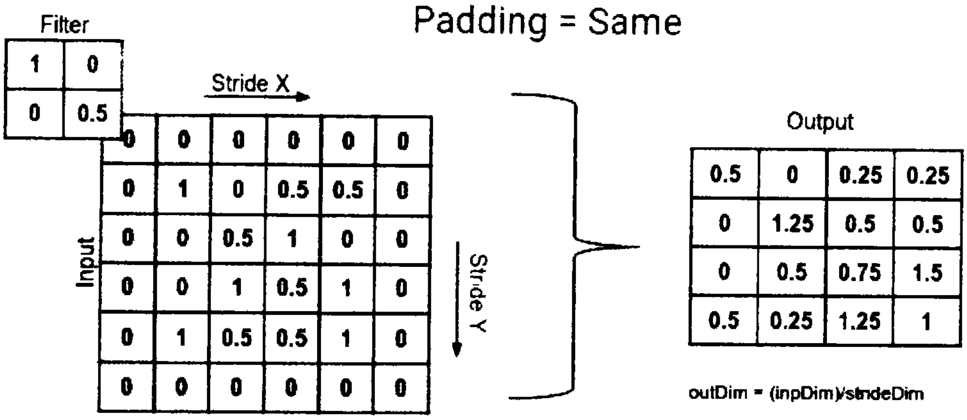


Figure 2.3: Feature Hierarchy.

data format. For example, in our case data format values (64, 64, 3) mean input image has the dimensions 64x64, and 3 denotes the colour channel (RGB); the number of channels and the number of filters should be the same. If a grayscale image is passed to the convolution layer, the depth value will be 1. Larger dimensions will increase the computational cost.

- **Number of Filters:** The first required parameter of the convolutional layer is the number of filters. The exact number of filters is decided after tuning the model depending upon the depth of the developed neural model and dataset complexity.
- **Filters:** The second important and required parameter is the filter size. Typically filter sizes maybe 1x1, 3x3, 5x5, 7x7 and more. Kernel size calculates after the fine-tuning of the model.
- **Stride:** The fourth important parameter is stride, which is used to move or step the convolution along with the x-axis and y-axis on the input image. The size of the convolved feature matrix depends on the value of the stride. The image size can be reduced with the help of stride. The value of the stride chooses after the fine-tuning of the CNN model.
- **Padding:** Another required parameter is image padding. If an image 5x5 is convolved into a 3x3 filter, it produces 3x3 output. There are two disadvantages; in this case, each time we perform a convolutional operation, the image size will shrink, and the other disadvantage is that pixel present if we want to move or find the features from the corners of the input image, then we can use the zero-padding on the corner of the image are used only a few numbers of times as compared to the central image. It means there is an information loss on the image's

borders. To overcome this issue, an additional border of zeros on the borders of the original image will be added, making a 4x4 matrix instead of a 2x2 matrix. It will give the result that is the original matrix 4x4, as shown in Figure 2.4. Padding also depends on the size of the filter. The value of padding may be "Valid" or "Same". We choose the padding after the fine tuning of the model.



Filter

1	0
0	0.5

Input

0	0	0	0	0	0
0	1	0	0.5	0.5	0
0	0	0.5	1	0	0
0	0	1	0.5	1	0
0	1	0.5	0.5	1	0
0	0	0	0	0	0

Output

0.5	0	0.25	0.25
0	1.25	0.5	0.5
0	0.5	0.75	1.5
0.5	0.25	1.25	1

Figure 2.4: Same Padding Process.

- **Activation:** An artificial neuron calculates the weighted sum of its input and then adds the bias, and after that, it decides whether it should be fired or not.

$$Y = \sum(\text{weight} \times \text{input}) + \text{bias} \tag{2.3}$$

Y's value can range from -ve to +ve, and it does not know the bounds. So, how will it be decided whether the neuron should fire or not? For this purpose, an activation function is added to any CNN model to check the value of Y and decide whether the outside connections consider this neuron as fired (activated) or not. Nonlinear activation functions (Equation 2.4) are applied to the output of the filter bank to generate activation maps, from which only active features are promoted to the subsequent layer. The output of neurons is governed by this function. Then, $f(.)$ is an example of an activation function, and its behavior looks like this:

$$\Phi(Y_i^{(l)}) = f(B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{ij}^{(l)} \times Y_j^{(l-1)}) \tag{2.4}$$

Most convolution layers use the Rectified Linear Unit (ReLU) activation function. Semi-rectification of the function is depicted in Figure 2.5(a) [207, 212]. Figure 2.5(b) depicts the hyperbolic tangent (tanh) with domain [1, 1], which has a similar shape to the Sigmoid function [207, 213]. The mapping between zero and negative numbers is nearly exact, which is a plus. Figure 2.5(c) displays the hyperbolic tangent (tanh) with domain [1, 1], which has a similar shape to the Sigmoid function [207, 213]. The mapping between zero and negative numbers is nearly exact, which is +ve.

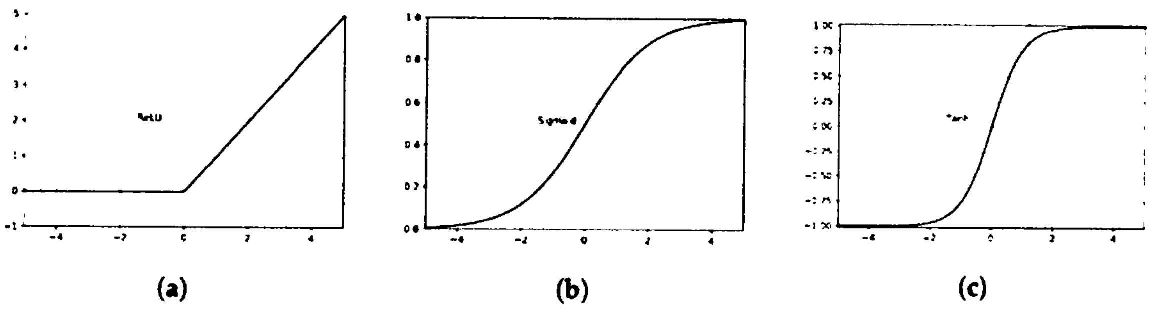


Figure 2.5: (a) ReLU, (b) Sigmoid, and (c) Hyperbolic Tangent.

2.5.1.2 Pooling Layer

We can reduce the spatial size of the convolved feature matrix by using the pooling layer like a convolutional layer, which helps decrease the computational power by reducing the dimensions of the data. The pooling layer also extracts the dominant features used in positional and rotational invariants. Every CNN model can usually have more than one convolutional and pooling layer. The pooling layer has the following advantages [208]:

- Computational load, both the training and run time, is reduced.
- Reduces the usage of memory.
- Reduces the trainable weights, especially in images that have many pixels.
- Tolerant of the network and encourages location invariance.
- Reduces overfitting risk.

The hyperparameters used in the pooling layer are:

- Filter Size

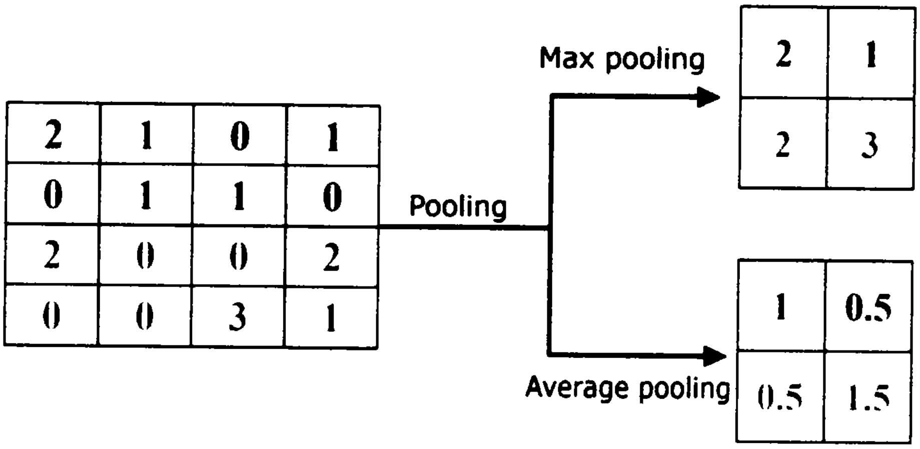
- Stride
- Pooling

The two basic pooling operations are described below, and a 2x2 filter is used as an example in Figure 2.6. Maximum pooling (or "max pooling") finds the maximum value for each input patch [214]. The max-pooling layer preserves the highest value in each patch by iteratively sliding the filter across the feature map. In mathematical notation, it looks like this:

$$f_{\max}(A) = \max_{n \times m}(A_{n \times m}) \tag{2.5}$$

The max pooling layer frequently uses a 2x2 filter with a stride of 2. The input is down sampled by a factor of 2 in each dimension, and 75% of the convolutional outputs are thrown away. The average value of each input patch is calculated, and the results are pooled together [214]. The average pooling layer is responsible for down sampling the convolutional activation by averaging over pooling regions of the input. These are the mathematical definitions of it:

$$f_{\text{avg}}(A) = \frac{1}{(n+m)} \left(\sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} \right) \tag{2.6}$$



2	1	0	1
0	1	1	0
2	0	0	2
0	0	3	1

2	1
2	3

1	0.5
0.5	1.5

Figure 2.6: Examples of Pooling Operations.

2.5.1.3 Dropout

A dropout layer can be used as a regularization layer to prevent the network's neuron units from over-adapting to one another. The dropout method enhances network performance while allowing the overfitting issue to be tackled. It applies to the network at any layer.

2.5.2 Classification

Automatic feature extraction is performed by all the convolutional and pooling layers, which build many hierarchies of features. The output of the topmost convolutional layer is very high-level features. When these high-level features are passed to fully connected layers, the classification zone starts. Classification usually consists of a few fully connected layers that help the network classify the different classes by learning the high-level feature [208].

2.5.2.1 Fully Connected Layers

Based on feature extraction or learning, the data is then classified into various classes using the fully connected layers. We can generally use the fully connected layers after the feature learning to train the model end-to-end. The basic purpose of a fully connected layer is to take the extracted features from preceded layers to predict the class label of a given sample. Practically full connection process works as under:

- The fully connected layer neuron detects certain features like the nose etc.
- It stores its probability value.
- Then, this value is communicated to the classes.
- After checking the feature, classes decide whether it is relevant to them.

There are three types of fully connected layers on CNN, as shown in Figure 2.7.

- **Fully Connected Input Layer (Flatten):** The purpose of this layer is to receive the extracted features from preceded layers, "flattens" the convolved matrix into a single vector, as shown in Figure 2.8, and then passes it to the next stage as an input.
- **Fully Connected Hidden Layer:** This layer receives the input vector (feature analysis) from the Flatten layer and learns the features to predict the class labels of training samples. More than one fully connected hidden layer may be used in CNN, depending on the complexity of the dataset. We will choose the number of hidden layers during the fine-tuning of the proposed method. The following important parameters of the fully connected hidden layer are chosen after the fine-tuning of the proposed method:

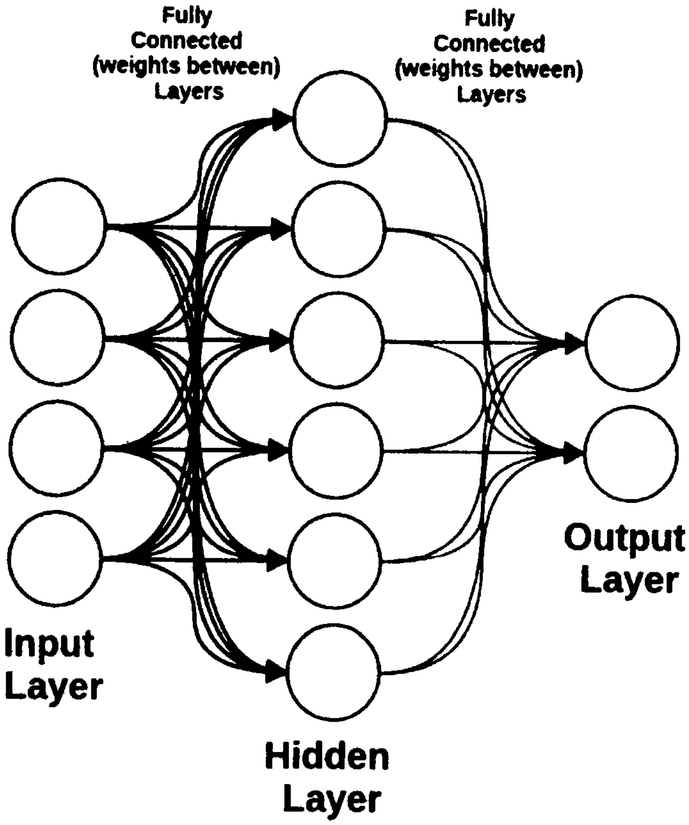

Fully Connected (weights between) Layers
Fully Connected (weights between) Layers
Input Layer
Hidden Layer
Output Layer

Figure 2.7: Fully Connected Layer.

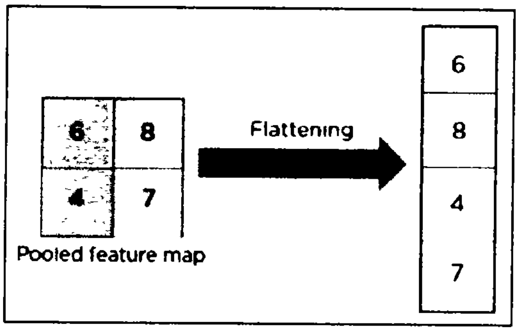


Figure 2.8: Fully Connected Layer Flatten.

- **Units or Neurons:** The fully connected layer neuron detects certain features like the nose etc. All the input received from the fully connected input layer (flatten) is directly connected with the neuron of the first fully connected hidden layer, and the neuron of the hidden layer is directly connected with the fully connected output layer, as shown in Figure 2.7. We decide the number of neurons or units after the fine-tuning of the CNN model.

- **Activation Function:** Many activation functions exist, such as linear, step, Tanh, Sigmoid, and ReLu functions. We decide which activation function will be based on the best serving function for the faster training process of the CNN model after the fine-tuning phase.

- **Fully Connected Output Layer:** Finally, the fully connected output layer predicts the final probabilities for the presence of a class. A fully connected layer uses the backpropagation process to feature extracted by preceded layers. Each neuron of the fully connected layer receives the feature vector and prioritizes the most accurate label. For the final decision, the output of all neurons in a fully connected layer is combined using a majority vote technique.

  The following parameters are chosen after the fine-tuning of the CNN model:

  - **Units or Neurons:** The fully connected layer neuron detects certain features like the nose etc. All the input received from the fully connected input layer (flatten) is directly connected with the neuron of the first fully connected hidden layer, and the neuron of the hidden layer is directly connected with the fully connected output layer, as shown in Figure 2.7. We decide the number of neurons or units after the fine-tuning of the CNN model.

  - **Activation Function:** The number of units or neurons of a fully connected output layer should be the same as several classes the network has.

  - **Activation Function:** Many activation functions exist, such as step function, linear function, Sigmoid function, Softmax function, Tanh function, and ReLu function. The sigmoid function is used for binary classification, and the Softmax function is used for multi-class classification. Therefore, we use the Softmax activation function because our problem is a multi-class problem.

2.5.3 Training and Optimization

Backpropagation, a form of stochastic gradient descent (SGD), is used to teach CNNs. During training, the model learns to optimize its weights and biases to reduce the loss function from its initial values, which are chosen at random. Cross-entropy loss and mean squared error are two common loss functions used in CNNs for classification and regression. Training efficacy and generalization can be enhanced by applying optimization strategies like learning rate scheduling, momentum, and weight decay.

2.6 YOLOv5 Architecture

Object detection using deep neural networks was accomplished by employing the *YOLO* model [215]. As an input image is processed, *YOLO* flags the location and class of the desired object. In order to determine its location and category, it places a little rectangle around the object. From version 1, *YOLO* advanced to version 5. The *YOLO*v5 network improved upon the previous four versions by being significantly faster, smaller, and more accurate. Fast *RCNN* [216], Faster *RCNN* [217], *SSD* [218], and *YOLO* [215] have all been introduced to the field of object identification. The *YOLO* technique was tied into the whole end-to-end detection process. *YOLO* saw object identification as a way to overcome regression issues by simultaneously achieving input from the real picture and output at the desired position. With *YOLO*'s detecting features, it was possible to acquire real-time processing, which was particularly useful when dealing with large quantities of image data [219]. The weight file size for the YOLOv5 target identification network model is roughly 90% smaller than YOLOv4, proving its viability for use in embedded devices for real-time detection. Hence, the YOLOv5 network's portability, precision, and quickness in identifying objects make it preferable. There are four distinct YOLOv5 designs, each with its name: YOLOv5l [220], YOLOv5x [220], YOLOv5m [220] and YOLOv5s [220]. These versions are distinguished by their convolution kernels and the number of feature extraction modules they employ at each node. Model complexity and the number of parameters used in the four designs increase. Figure 2.9 depicts the YOLOv5s architecture that was used in this research.

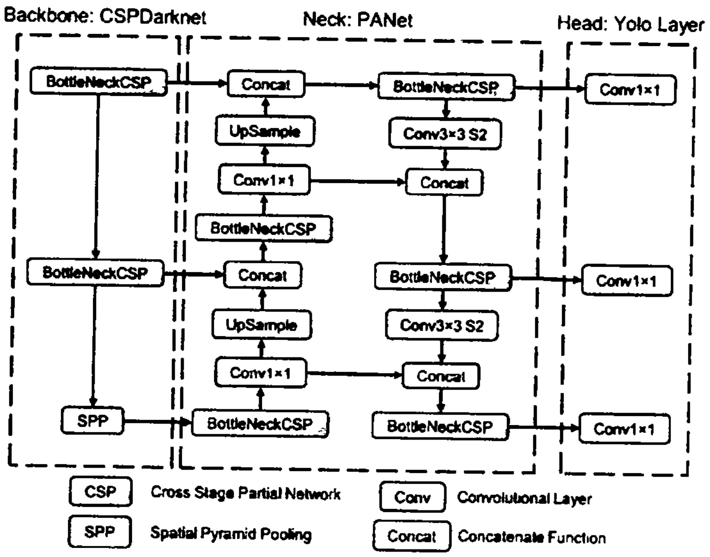


Figure 2.9: YOLOv5s Model Architecture.

When discussing *YOLO*v5, "real-time detection" refers to the algorithm's capacity to carry out object identification tasks in real-time or near real-time. The algorithm must be able to process multiple frames of input video or photos per second in order to offer timely detection results. With its highly streamlined design and efficient implementation methodologies, *YOLO*v5 can provide real-time detection. Its real-time performance is due, in part, to the following factors:

First, the network architecture: YOLOv5 uses a simplified one that strikes a good compromise between precision and speed. It employs a sequence of convolutional layers to efficiently extract characteristics from input images, typically with smaller kernel sizes.

Second, *YOLO*v5 uses an efficient backbone network, like CSPDarknet53 or EfficientNet, which keeps computational complexity low without sacrificing detection accuracy. These infrastructure networks are optimized for rapid and reliable picture processing.

Third, *YOLO*v5 uses a Feature Pyramid Network (FPN) to help the model extract multi-scale characteristics from various nodes in the network. It strengthens the model's capacity for accurately detecting objects across various sizes.

Fourthly, model quantization, network pruning, and model acceleration libraries like NVIDIA's TensorRT are just a few of the optimization strategies used by *YOLO*v5. These methods aid in maximizing the pace at which the model can conclude without sacrificing too much precision.

Using these methods, *YOLO*v5 can detect numerous objects (disease regions) in each frame of an input stream, making it suitable for applications such as plant leaf disease diagnosis, which requires real-time object detection. Applications that benefit from real-time detection include real-time monitoring, surveillance, and automation systems, where rapid or near-instantaneous findings are essential.

The Detect, Backbone, and Neck networks are the three main parts of the YOLOv5s [220] system. The primary network is a *CNN* that generates visual features by splicing together many high-resolution images. Specifically, the first backbone layer is meant to speed up the model's training and reduce computing time. A 12 × 320 × 320 pixel feature map was created by first slicing the input 3-channel image (3 × 640 × 640 in YOLOv5's architecture) into four 3 × 320 × 320 pieces. After being fed into a convolutional layer with 32 convolution kernels, this map yielded a 32 × 320 × 320 feature map. The output was processed using the BN (batch normalization) layer and the Hardswish activation function before being sent to the final layer. The BottleneckCSP module, located at the third tier of the backbone network, is meant to improve deep feature extraction in an image. The simplest form of the BottleneckCSP is a convolutional layer (Conv2d + BN + Hardswish activation function) with a 1 × 1 kernel size and a 3 × 3 kernel size linked by a Bottleneck module. Its output and input via the residual structure make up the bottleneck module's output. Each half of the input to the BottleneckCSP module undergoes a convolution operation that halves the volume of channels of feature maps, effectively doubling the processing speed. The Conv2d layer and Bottleneck module in the second branch is then used to merge the final feature maps from both branches. After passing through the Conv2d and *BN* layers in sequence, the final feature map is almost the same size as the input to the BottleneckCSP component.

The ninth layer of the Backbone network is the Spatial Pyramid Pooling (*SPP*) module, which is designed to recover the network's receptive field by transforming a feature map of arbitrary size into a feature vector of fixed size. The YOLOv5s SPP module's input feature map has the dimensions (512 × 20 × 20). The initial output of the convolutional layer is the 2562020 feature map, with a convolution kernel size of 11. This feature map was intimately linked to the output feature map, which is now subsampled via three concurrent max-pooling layers. The final feature map has dimensions of 1024 × 20 × 20. When everything has been processed via the convolutional layer using a 512 convolution kernel, the final output feature map has dimensions of 512 × 20 × 20. To create Feature Pyramid Networks (*FPN*), a series of feature aggregation layers of mixed and merged image features is used, and the neck network is the primary component of this series. A detect network is then fed the resulting feature map (prediction network). In addition, this network's feature extractor uses a novel *FPN* structure, which aids in bottom-up learning, feature transmission, and multi-scale object recognition. The same target object can be accurately detected across various scales and sizes.

The recognition network is used mainly in the model's final recognition stage, which correlates anchor boxes on the feature map output by the last layer. It spits out a vector that contains data about the object's score, bounding box position, and the likelihood that it belongs to the category of interest. In order to properly recognize objects across a wide range of sizes in images, YOLOv5's identification network uses a three-layer detect architecture with 80 × 80, 40 × 40, and 20 × 20 detect layers, each of which takes as input a feature map with those specific dimensions. By the end of the process, each detects layer will have produced a 21-channel vector ((2 classes + 1 class probability + 4 surrounding box position coordinates) × 3 anchor boxes). Following this, the image's leaves were recognized, and other bounding boxes and target class labels were generated and assigned accordingly.

2.7 Open Issues

There are several research issues in this research area. The first issue is that different varieties of potato and guava exist in other regions. Therefore, variations in potato and guava diseases prevail worldwide for several reasons, including the shape of the disease, symptoms, colour of leaves, varieties of potato and guava and environmental factors. No study available in the literature can detect the potato and guava species. Suppose we can research the Pakistani region's potato and guava varieties' species and diseases. In that case, it will be helpful for the Pakistani farmers to detect the different diseases of potatoes and guava in their early stages. The second issue is the infected patches detection. The other issue in the literature is that all research is done on a single disease on a single leaf, but the earlier works failed to detect the multiple diseases on a single leaf. Most crops or leaves were attacked with various diseases on a single leaf. To the best of our knowledge, no work has been reported on the detection of multiple diseases of guava on a single leaf. As reported in the literature, real-time disease detection and localization is another issue. Also, no work has been done on guava disease spot localization. The fifth biggest issue in the literature is the non-availability of different varieties of potato and guava diseases datasets in Pakistan. However, the PlantVillage dataset was available in the literature, which consisted of potato diseases developed in Switzerland and the USA. Almost all the research has been done on the PlantVillage dataset. Therefore, there is a dire need to develop a dataset consisting of different varieties of potato leaf diseases in Pakistan. In literature, only a dataset of a single disease on a single leaf exists for the guava leaf. Still, the real-time dataset of guava for multiple leaf diseases is also unavailable. So, there is a need to develop a dataset which can detect the various guava diseases on a single leaf. Segmentation of guava leaf is another crucial issue. In literature, no study exists which can segment the guava leaves.

Chapter 3

A Hybrid Deep Learning Approach to Classify the Plant Species

3.1 Overview

Plants are essential to the survival of all life on Earth. Among other things, they provide us with oxygen to breathe, food, and medicine. Every living thing depends on them [221]. Plants are a vital part of our planet's ecosystem, and there are around 391,000 vascular plant species worldwide [222]. Detecting and eliminating weeds, for example, necessitates accurate identification of plant species utilizing automated methods that rely on human expertise [57]. To examine all of the planet's plant species and discover certain features that allow botanists to distinguish between them is an impossible task [49]. Hand-species identification can be time intensive and error-prone even when performed by experts in a given plant taxon. making it difficult to scale up to high-throughput needs.

A biologist has difficulty categorizing plant species that appear to share many traits but are actually distinct from each other. When botanists utilize manual identification, they rely on the specifically defined traits of a plant as a key to identifying plant species. Each step in the identification process necessitates answering a question regarding the plant's characteristics. A polytomous or dichotomous key is used to identify the next step in the identification process [54]. They put much effort into researching and identifying unique traits in numerous plant species [223]. An alien plant's shape, texture, colour, and venation' are some of the keys to its identification. These traits finally lead to the target species when thoroughly analyzed. Taxonomic skill required to identify a plant species from its natural habitat is beyond the capabilities of the average individual. As a result, traditional methods for identifying plant species are difficult for laypeople and professional taxonomists. Even for the most experienced botanists, determining the species can be time-consuming. The identification and categorization of plants should be computerized or automated.

In the past, researchers have only utilized healthy leaves to classify and divide plants. No research on the leaves of guava, potatoes, or java plums was located in the literature review. Therefore, it was also challenging to separate guava, java plum, and potato species. The availability of a dataset for guava, potato, and java plum species was also challenging. There was a need to develop a dataset for potato, guava, and java plum plant leaf species. To resolve the issues mentioned earlier, the current research was carried out. For this purpose, two datasets, Plant Leaf Species Dataset (*PLSD*) and Plant Leaf Species Segmentation Dataset (*PLSSD*), were developed. The Plant Leaf Species Dataset (*PLSD*), which included healthy and diseased leaves of guava, potato, and java plum for classification, and *PLSSD* generated for semantic segmentation of the above classes, were used to tackle the above problems. The leaves of the guava, java plum, and potato plants were used to create a hybrid deep-learning model. In the first stage, a unique segmentation model based on MobileNetV2-UNet was developed to segment the plant leaf species. The Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*) was created to classify the guava, potato, and java plum plant leaf species in the second step. These are the main things this chapter contributes:

1. A Plant Leaf Segmentation based on MobileNetV2-UNet based technique is developed to segment the guava, java plum, and potato healthy and diseased leaves. The MobileNetV2 is used as an encoder part and UNet as a decoder part.

2. A Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*) is developed based on MobileNetV2, InceptionV3, and ResNet50 models to classify the guava, java plum, and potato leaves. The proposed approach has been thoroughly researched and tested on healthy and damaged plant leaf species to guarantee the system's diversity.

3. Two first-ever plant species datasets are developed to segment and classify the guava, java plum, and potato plant leaf species from the central Punjab, Pakistan.

3.2 The Proposed Methodology

This study suggests a hybrid deep-learning method for plant leaf species identification. A segmentation model based on MobileNetV2 and UNET is first used to segment the leaves of guava, java plum, and potato. Then the plant species detection stacking ensemble deep learning model (PSD-SE-DLM) is used to classify the species of those leaves. In total, there are two stages to the proposed method. Figure 3.1 is a flowchart depicting the suggested procedure.

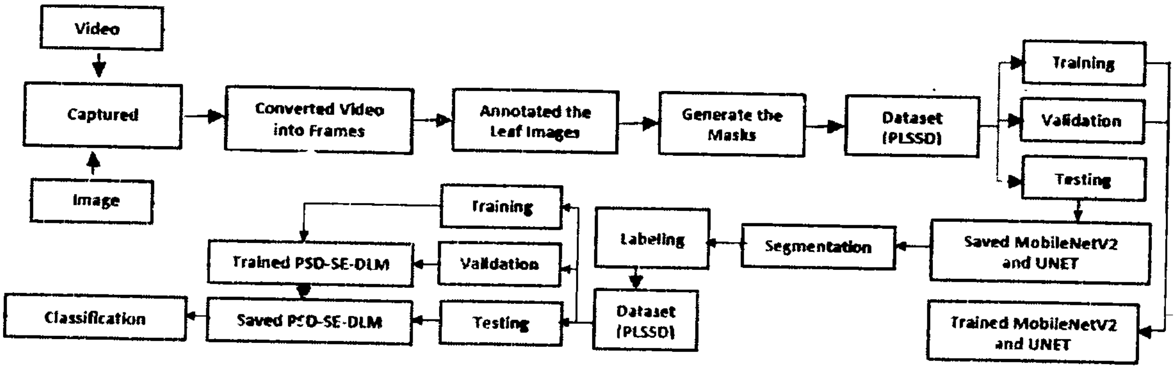


Figure 3.1: Flowchart of the Proposed Hybrid Deep Learning Model.

3.2.1 Dataset Preperation

This study proposes a hybrid approach trained and tested on two datasets. We created two datasets using PLSSD for MobileNetV2-UNet training and Plant Species Dataset (PLSD) for PSD-SE-DLM training. In the next part, we present a comprehensive overview of the datasets:

3.2.1.1 The Plant Leaf Species Dataset (*PLSD*)

A Plant Leaf Species Dataset (*PLSD*) was created in central Punjab, Pakistan. The *PLSD* dataset included healthy and infected guava, potato, and java plum leaves, as depicted in Figure 3.2. We constructed our real-time dataset using images and videos. The variations in the real-time dataset were made using a wide variety of capture equipment, including cell phone cameras, digital cameras, and drones. Mobile phones and digital cameras had a capture distance of 1 to 2 feet, whereas drones had a capture distance of 5 to 10 feet. The drone's fanning caused visual and video distortion due to the fluttering of the plants' leaves. Therefore we kept as much space between them as feasible. Okara was chosen because it is the most productive area in Pakistan for growing potatoes, guava, and java plum. We zero down on the Okara district's Coroda, Mozika, and Sante potatoes.

In November 2020, farmers planted several varieties of potatoes in the open fields once they had been acclimatized to their new surroundings. The leaves of the guava and java plum cultivars Choti Surahi, Bari Surahi, Gola, Golden, and Sadabahar were chosen. Images and videos were taken at different times of day, in varied lighting settings, during different seasons (summer, winter, spring, and fall for guava and java plum), and under diverse weather, circumstances to track how these factors affected disease prevalence and severity. To account for differences in the quality of the dataset, the pictures were captured at various sizes. Following this, plant pathologists classified the photographs according to whether or not they depicted guava, potato, or guava leaf. We used 5,680 images of guava, potato, and java plum leaves, both healthy and diseased, to create the *PLSD* dataset. The plant leaf species collection includes 1900 photos of guava leaves, 1900 photographs of potato leaves, and 1880 images of java plum leaves (Table 3.1).

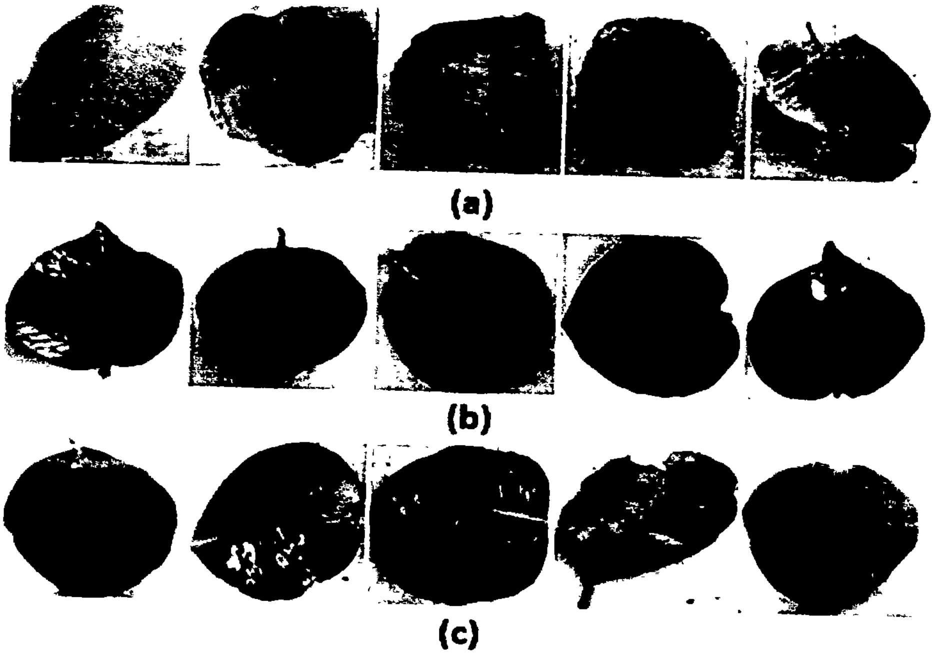


Figure 3.2: (a) Guava Leaf (b) Potato Leaf (c) Java Plum Leaf Species.

3.2.1.2 The Plant Leaf Species Segmentation Dataset (PLSSD)

As can be seen in Figure 3.3, we used the Plant Leaf Species Dataset (*PLSD*) to select both healthy and diseased leaves from guava, java plum, and potato plants. Then, using the Labelme semantic tool, these photos were annotated. Various guava, java plum, and potato leaf species were used to create a diverse dataset. Annotations made with python code were then utilized to construct the

Table 3.1: Exposition of the Plant Leaf Species Dataset (*PLSD*).

Class Labels	**Samples**
Guava	1900
Potato	1900
Java Plum	1880
Total Samples	**5680**

masks of multi-class photos. The Plant Leaf Species Segmentation Dataset (PLSSD) was created using the original images and masks, as depicted in Figure 3.3. Each dataset class consisted of 1329 images and the same number of masks. The PLSSD dataset consisted of 3987 images as shown in Table 3.2.

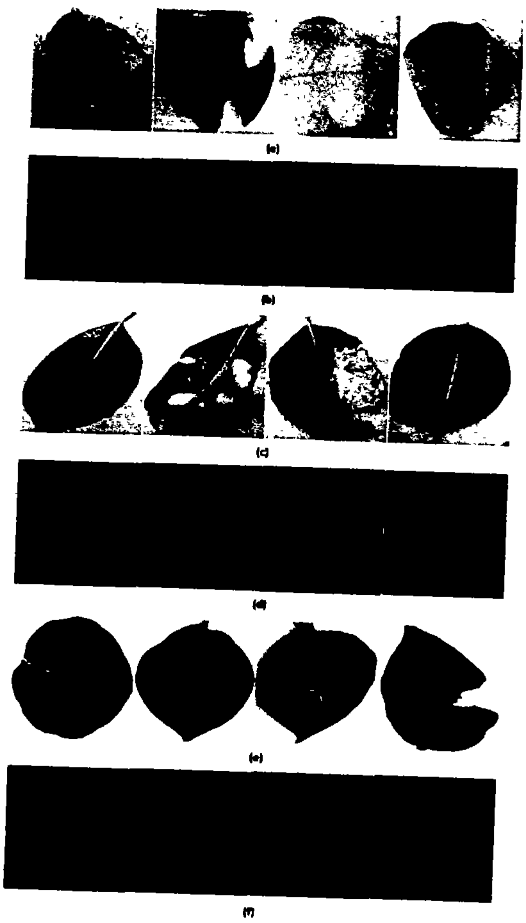


Figure 3.3: (a) Guava Leaf (b) Guava Leaf Mask (c) Java Plum Leaf (d) Java Plum Mask (e) Potato Leaf (f) Potato Mask.

Table 3.2: Summary of the Plant Species Segmentation Dataset (PLSSD) Splitting.

	Class	Samples	Total Samples
Training	Guava	966	2,898
	Java Plum	966	
	Potato	966	
Validation	Guava	242	726
	Java Plum	242	
	Potato	242	
Testing	Guava	121	363
	Java Plum	121	
	Potato	121	
Total Samples			**3,987**

3.2.2 Image Pre-Processing

The *PLSD* and *PLSSD* image datasets were pre-processed to facilitate more reliable classification outcomes and enhanced feature extraction. The python code was used to extract still images from the plant videos. The photos were then resized to 224 x 224 pixels using python scripts that clipped out the excess background.

3.2.3 Dataset Splitting

Inside the *PLSD* are three distinct data sets: a training set, a validation set, and a test set. After the *PSD-SE-DLM* model had been trained with the training dataset's help, its efficacy was evaluated with the help of the validation and test datasets. Because of this, we segmented the data sets into a training set (79.23%), a validation set (10.566%), and a test set (20.37%) (10.21%). The *PLSD* dataset contains 4,500 pictures; 600 for training and validation; and 580 for testing. The same image ratios of 79.23% were employed to train the guava, potato, and java plum-labeled images. Regarding *PLSD*, the remaining 20.77% of fresh photos were divided into a validation and testing group with 10.55% and 10.21%.

Training, validation, and testing sets make up the Plant Leaf Species Segmentation Dataset (*PLSSD*). The acrshortplssd training set consisted of 2,898 photos, with 966 pictures devoted to each of the three classes (guava, java plum, and potato). In addition, the validation set included 242 samples from each class and 726 photos altogether. The testing set, on the other hand, contained 363 samples for inference, with 121 examples for each category. For all classes, there were 3987 samples in the PLSSD, as shown in Table 3.2.

Table 3.3: Summary of the Plant Leaf Species Dataset (*PLSD*) Splitting.

	Guava	**Potato**	**Java Plum**	
Training	1500	1500	1500	4500
Validation	200	200	200	600
Testing	200	200	180	580
Total Samples	**1900**	**1900**	**1880**	**5680**

3.2.4 The MobileNetV2-UNet Segmentation Model Architecture For Plant Leaf Species

Images are classified by the network by assigning them a category (or label). It's possible to know the shape of an object, which pixel corresponds to which object, and so on, but it's not always possible. We would give each image pixel a specific label to help us sort it out. Segmentation is breaking down a problem into smaller, more manageable pieces. A segmentation model provides more information about the image than a more generalized model.

For example, in the traditional encoding-decoding framework used by the U-Net model of semantic segmentation networks, the VGG-Net [224] network is the principal feature extraction network. However, the model's complex network topology and heavy-weight parameters make it impractical for usage in embedded devices. MobileNetV2 was developed by Mark Sandler [225] as a acrshort-cnn specifically for cellular phones. The MobileNet network is used as a basis for the improvement of MobileNetV2.

When combined with depthwise separable convolution, the inverse residual structure reduces the number of network parameters while significantly minimizing the loss of low-dimensional spatial information, making the network more amenable to the real-time requirements of the embedded platform. The MobileNetV2 network is combined with the U-Net semantic segmentation model to produce the MobileNetV2-UNet model. Figure 3.4 depicts the overall model structure as shown in Figure 3.4 (B1-B17). MobileNetV2's 17-layer inverted residual block was used in place of VGG's standard convolutional network as the model's feature extraction backbone. As can be seen in Figure 3.4, to obtain more basic features, only the first two convolutional layers of U-Net were retained (C1–C2). The model feature extraction efficiency is enhanced, and information loss due to picture compression is diminished when the number of parameters is decreased. The right side of Figure 3.4 depicts the decoding process, which is a feature recovery procedure in which the feature recovery layers concatenate with the extracted features from the coding section

(CN1-CN5). A feature recovery procedure is implemented.

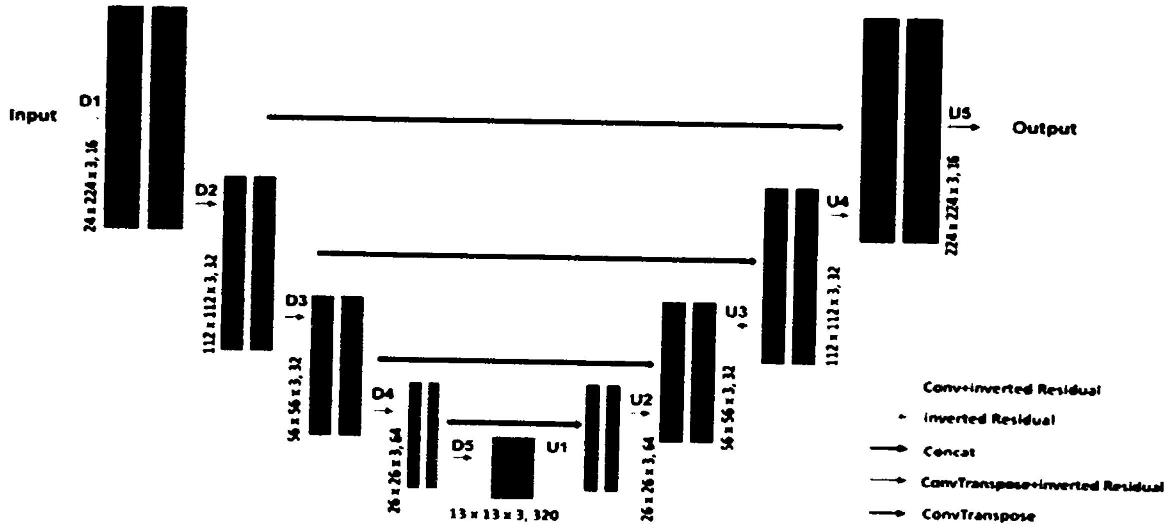


Figure 3.4: MobileNetV2-UNet Architecture.

3.2.5 The Proposed Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*) Methodology

Ensemble learning [226] pools the predictions from different models into a single set to reduce the possibility of making a wrong decision. For instance, in the gradient boosting ensemble technique [227], model development occurs through a continuous process of reflecting on and gaining wisdom from previous errors. Suppose one of the models produces inaccurate predictions. In that case, the subsequent models will attempt to make up for it by doing relatively well on the dataset, thereby boosting the overall performance of the ensemble. The two most essential characteristics that are anticipated to be possessed by a method are bias and variance [228]. The ensemble method tends to reduce both of these characteristics by combining individual models, resulting in a robust learner that is much more responsive and less data-sensitive.

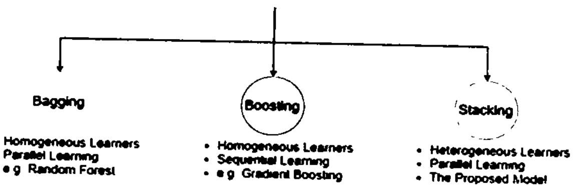


Figure 3.5: Ensemble learning techniques

Bagging, boosting, and stacking are alternative strategies for merging different types of learners, as shown in Figure 3.5 [229]. Instead of bagging and boosting, Stacking trains the tier-2 (meta classifier) learner by integrating the predictions from many independent models trained as base/tier-1 learners simultaneously. Stacking can accomplish [229] independence amongst varied learners by simultaneously merging base models and achieving [229] dependence between learners by progressively introducing the meta learner. As a consequence, it results in improved precision of the forecast and a reduced likelihood of overfitting. The stacking approach is a meta-classifier that aggregates the base learners' projections and is used for model combinations. The base learner must be chosen to construct an effective model when using the stacking process. Several deep learning algorithms, such as MobileNetV2 [225], InceptionV3 [230], and ResNet50 [231], key-value networks are utilized to select the base learner. These algorithms provide the highest possible accuracy in classification while doing so. Incorporating the best features of MobileNetV2, InceptionV3, and ResNet50 into a single model allowed us to obtain the highest accuracy. When it comes to combining models, the stacking approach refers to a meta-classifier that integrates the predictions made by several basic learners, as depicted in Figure 3.6.

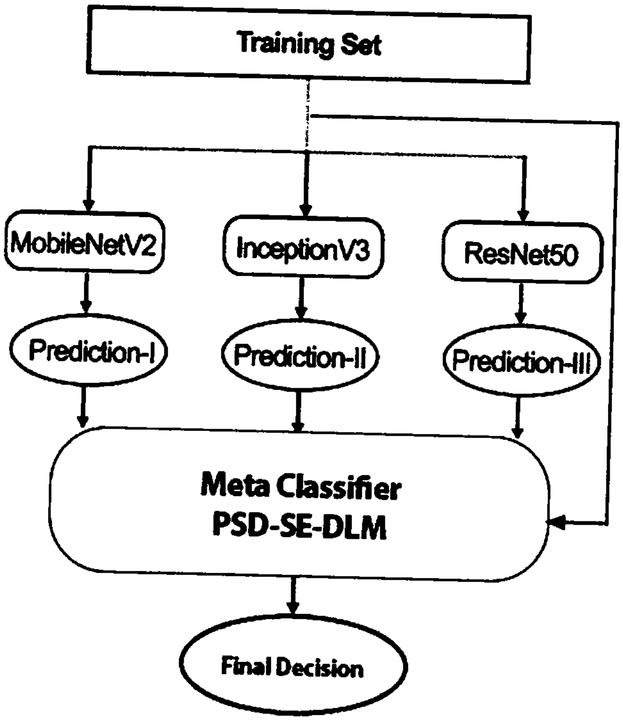


Figure 3.6: The Proposed Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*)

Transfer learning models are models that are trained on huge datasets that contain millions of images. These models are taught using transfer learning. Since the models were trained using such a vast dataset, they can easily generalize to new situations. The features learned from the larger datasets help tackle a problem that consists of either fewer or smaller data. This helps eliminate the requirement of training a model from the ground up. We used the transfer learning models as the base models. The description of the basic learners are as an under:

3.2.5.1 The First Base Model: MobileNetV2

MobileNetV2 is an architecture built by Google based on 1.4 million photos from 1000 different classes [225]. This architecture is a sophisticated type of *DCNN* that works quite well on mobile devices. With MobileNetV2, there is no need to start the training process from scratch; all that has to be done is a modification to the model's output layers at the very end. The architecture of MobileNetV2 is based on the architecture of its earlier iteration (i.e., MobilenetV1). It introduced a new ”inverted residual” structure to store the information. The issue of information being lost in convolution blocks due to the presence of a nonlinear layer is solved by employing the Depthwise Separable Convolution (*DSC*) method, which uses a linear bottleneck layer [225]. The fundamental components of MobileNetV2 are laid out in Figure 3.7.

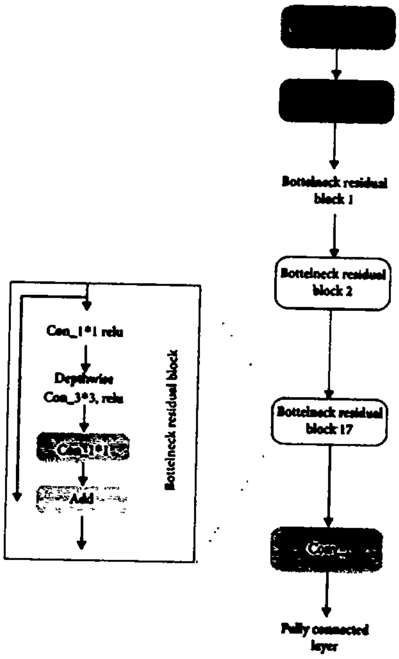

Bottelneck residual block 1
Bottelneck residual block 2
Bottelneck residual block 17
Con_
Fully connected layer
Con_1*1 relu
Depthwise Con_3*3, relu
Con_1*1
Add
Bottelneck residual block

Figure 3.7: MobileNetV2 Architecture

3.2.5.2 The Second Base Model: InceptionV3

There are three inception blocks in the Szegedy et al. [230]'s InceptionV3 model, each of which has parallel convolutions. With such modules, the deep architecture's computations become more efficient, and the overfitting problem is alleviated. An annual event, the ImageNet Large Scale Visual Recognition Challenge (*ILSVRC*) features 1.4 million photos from 1,000 different product categories [232]. This challenge is significant in the fields of picture classification and recognition. Krizhevsky et al. [233] propose employing the AlexNet model and report considerable gains in their object recognition and classification investigation. When that has been done, multiple convolutional models are created to bring the Top-5 error rate of object identification and classification down to an acceptable level. Figure 3.8 shows the top five error rates from item identification findings on ImageNet, and the most fantastic recognition results are observed for GoogleNet (Inception-v1). The results indicate that boosting the model layer's depth leads to improved recognition performance.

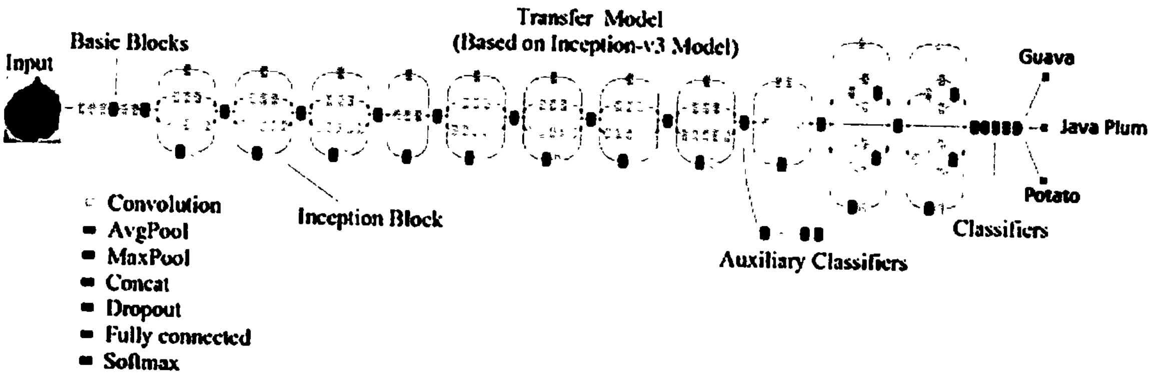


Figure 3.8: InceptionV3 Architecture

Compared to its predecessor, GoogleNet, the Inception-v3 model performs exceptionally well in object recognition (Inception-v1). The Inception-v3 architecture comprises three pieces: the enhanced Inception unit, the regular convolutional block, and the classifier. Feature extraction uses the fundamental convolutional block, consisting of layers that alternate between convolutional and max-pooling. The enhanced Inception module was developed based on NetworkIn-Network [234], a method in which multi-scale convolutions are carried out in parallel, and the convolutional results of each branch are further concatenated. When auxiliary classifiers are used, training results are more consistent, greater gradient convergence is achieved, and problems with concurrently disappearing gradients and overfitting are relieved. All of these benefits are made possible by the utilization of auxiliary classifiers. To decrease the number of feature channels and shorten the training time, Inception-v3 extensively uses the 1 x 1 convolutional kernel. In addition, the huge convolution is broken down into a series of smaller convolutions, bringing the total computation cost and the number of parameters under control. In conclusion, Inception-superior v3's performance in object identification places it at the forefront of the present state-of-the-art because of its unique architecture. This model is, therefore, widely used for the function of transfer learning.

3.2.5.3 The Third Base Model: ResNet50

The ResNet [231] model is one of the more well-known ones, and it has an excellent track record of success in various computer vision challenges. Other examples include Inception v3 [230], MobileNet [225], and GoogleNet [235], amongst many others. These models are trained with data from numerous datasets that include many different kinds of images. Such pre-trained model weights can be used by transfer learning techniques to efficiently address various computer vision challenges (dataset and computing resources). The ResNet50 model, a Convolutional Neural Network (*CNN*), comprises fifty layers. Using the ResNet50 model's pre-trained weights, we performed transfer learning on a small set of photos of plant species. The internals of the ResNet50 model and the many pre-trained weights it makes use of are discussed below. It can see that the overall structure of the ResNet50 model, including the fine-tuning setup for ResNet50, is in Figure 3.9.

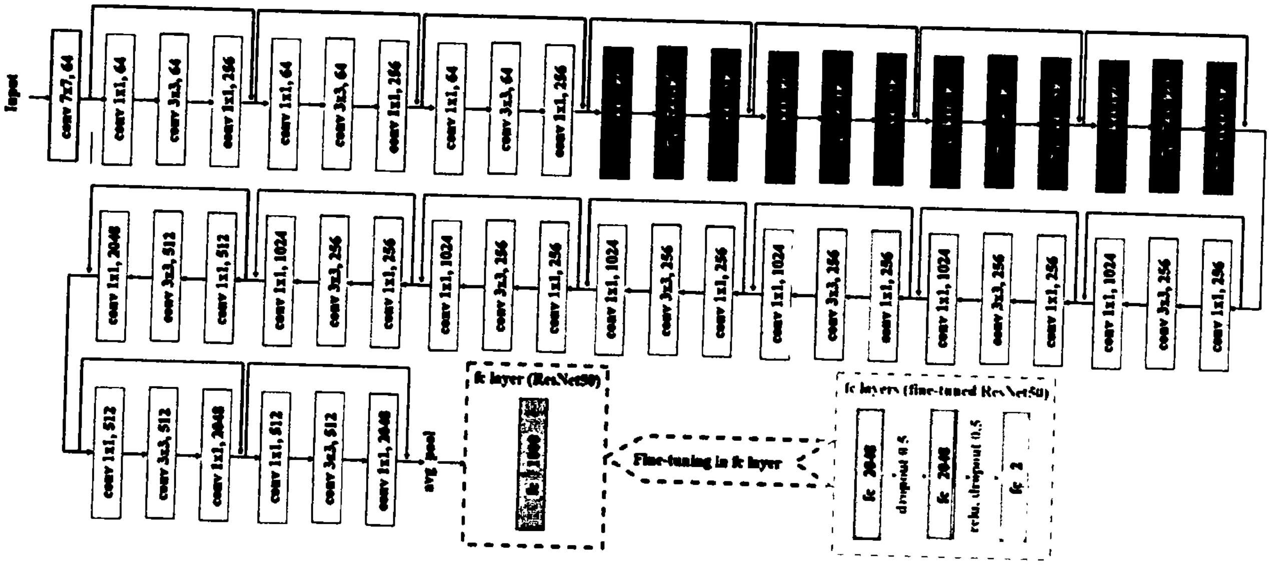


Figure 3.9: The ResNet50 Architecture

3.2.6 Experimental Setup

The proposed techniques were trained and tested using a Google Colab Pro account with powerful Graphical Processing Units (GPUs) without configuration requirements. The transfer learning deep learning models were used for training. We conducted four experiments in which three base models were used then a stacking ensemble model was developed using the output of the base models. All experiments of proposed *PSD-SE-DLM* used *Adam* optimizer with a learning rate of 0.0001, and SparseCategoricalCrossentropy loss functions were utilized for compiling all the models. We used 16 batch size, early stopping, saved the best val_loss model, and 5 epochs to train all the models. The proposed MobileNetV2-Unet model used 4 batch size, 50 epochs, early stopping, and saved the best val_loss model. The configuration details of all the models depicted in Table 3.4.

Table 3.4: *PSD-SE-DLM* Experimental Setup Configurations.

***PSD-SE-DLM* Configurations**	
Platform Used	Google Colab Pro account
Base Models Used	MobileNetV2, InceptionV3, ResNet50
Meta Classifier	*PSD-SE-DLM*
Optimizer Used	*Adam*
Learning Rates Used	0.0001
Loss Function Used	SparseCategoricalCrossentropy
Batch Size Used	16
Batch Size	5
MobileNetV2-UNet Configurations	
Batch Size	4
Epochs	50
Training Stopping Criteria	Early Stoping
Saved Model	Best Val Loss

3.2.7 Evaluation Measures

3.2.7.1 Classification Accuracy

Take the total number of correct classifications made by the classifier and divide it by that number to get a sense of its accuracy.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{3.1}$$

3.2.7.2 Precision

When evaluating a model's performance, it's important to remember that classification accuracy isn't always an accurate indicator. Expecting a high accuracy rate is unreasonable because the model is not learning anything, and all samples are assumed to be of the best quality. It may happen if there is an imbalance in the number of samples in each class. Following on the heels of this definition, precision is the proportion of True Positive (*TP*) (accurately identified samples) relative to the total number of positive samples identified (either incorrectly or correctly), which is stated as:

$$Precision = \frac{TP}{(TP + FP)} \tag{3.2}$$

3.2.7.3 Recall

Recall the fraction of input samples from a class that the model correctly predicts is another critical metric. The formula for determining the recall is as follows:

$$Recall = \frac{TP}{(TP + FN)} \tag{3.3}$$

3.2.7.4 F1 Score

Recall and precision can be measured by using a statistic known as the f1 score:

$$F1Score = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{3.4}$$

3.2.7.5 ROC Curve

Cutoff limits for classifiers generate a Receiver Operating Characteristic (*ROC*) curve. The cutoff threshold of a superior model can be calculated using the widely used *ROC* curve. The *TPR* contrasts favorably with the *FPR* at several different cutoff points.

3.2.7.6 Dice Score

In image segmentation and binary classification tasks like those found in medical image analysis or natural language processing, the Dice score (also known as the Dice coefficient or Dice similarity coefficient) is a statistical metric used to measure the similarity or overlap between two sets or groups.

The formula for determining a Dice score is as follows:

$$\text{Dice Score} = \frac{2 \cdot |A \cap B|}{|A| + |B|} \tag{3.5}$$

Where:

- The cardinality of the union of two sets is denoted by the expression $|A \cap B|$.
- The value $|A|$ indicates how big the set A is.
- $|B|$ represents the size of set B.

There can be no overlap or similarity between the sets if the Dice score is zero, and there can be 100% overlap or 100% similarity if the score is one. When comparing two sets, a higher Dice score indicates a higher level of resemblance or agreement.

In applications such as image segmentation, a high Dice score indicates good agreement or accuracy in the predicted segmentation (set A) relative to the ground truth segmentation (set B).

3.3 Results and Discussion

The experimental results focused on the following:

1. The performance of the MobileNetV2-UNet segmentation model was analyzed on Plant Leaf Species Segmentation Dataset (*PLSSD*).
2. Plant Species Detection (PSD) was used to evaluate the effectiveness of the proposed Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*).
3. Compare the suggested *PSD-SE-DLM* model to the best existing models.

3.3.1 The Performance Analysis of the MobileNetV2-UNet Plant Species Segmentation

Figure 3.10 (a) is a performance graph depicting the proposed model's loss and dice score performance on both the training and validation sets. The training loss decreased from 10.30% to 5.91% after 28 epochs. The initial rate of validation failures is 6.45%, falling to 5.95% by the epoch's conclusion. Dice Score during training and validation of the suggested MobileNetV2-UNet is displayed in Figure 3.10 (b). Validation dice score started at 96.53% and increased to 96.74% by the period's conclusion, whereas training dice score was 89.47% over that time. The test set prediction results from the MobileNetV2-UNet segmentation model are shown in Figure 3.11. The suggested method obtained a test set (unseen data) dice score of 96.38%. The complicated and damaged leaves of guava, java plum, and potato were successfully segmented using the proposed segmentation method. MobileNetV2-UNet works remarkably well as a plant leaf segmentation system, according to all parameters.

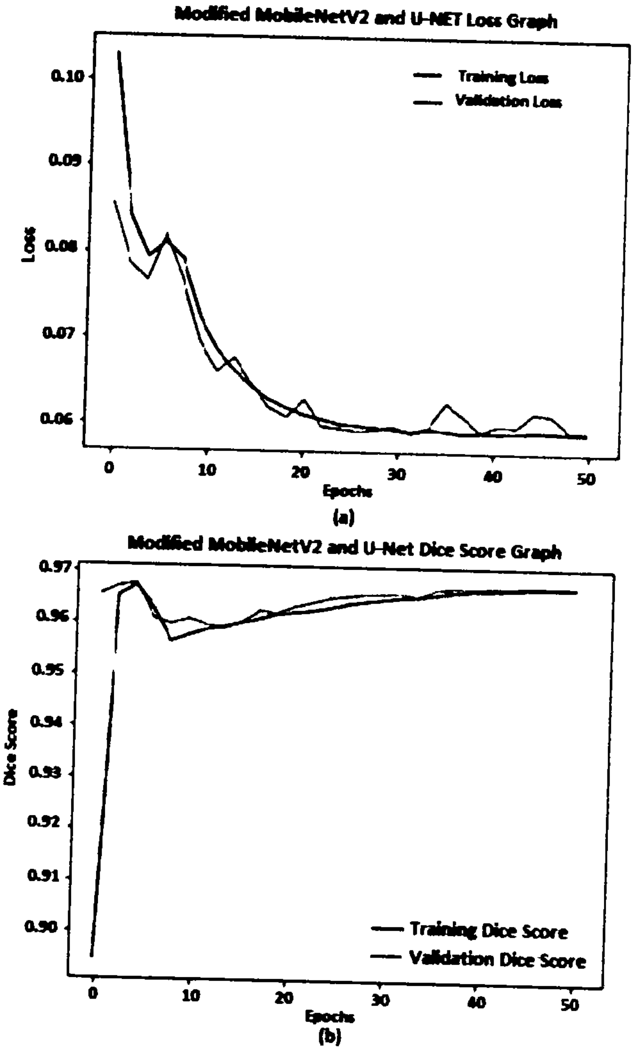


Figure 3.10: The MobileNetV2-UNet Plant Leaf Segmentation Model Dice Score and Loss Graph.

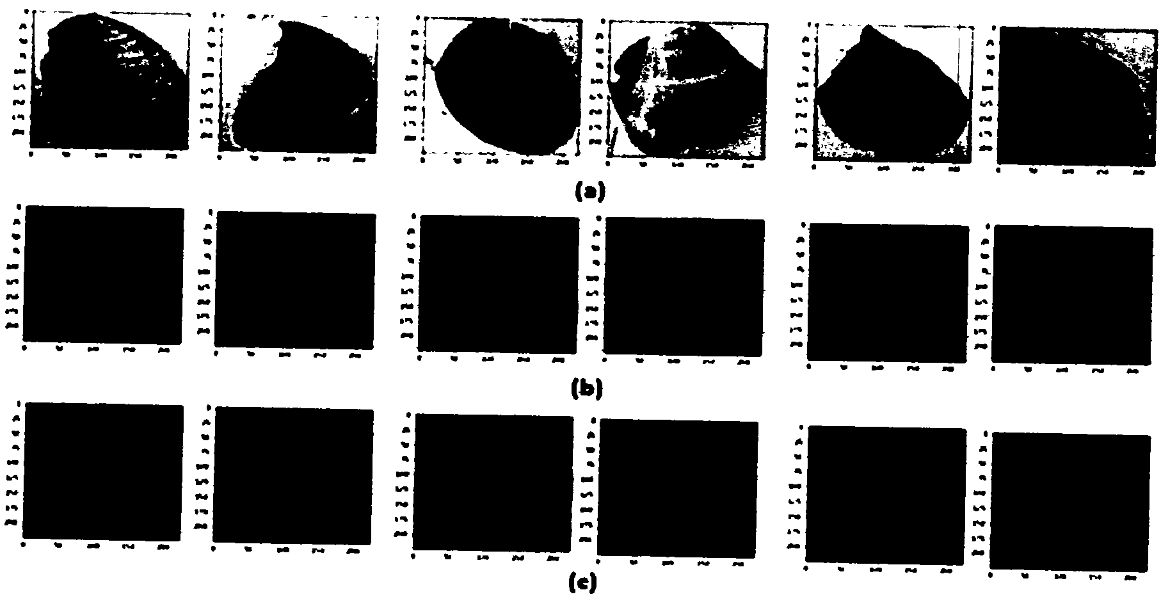


Figure 3.11: The MobileNetV2-UNet Plant Leaf Segmentation Model Prediction on Test Set.

3.3.2 The Performance Analysis of the Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*)

The proposed Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*) performance for detecting plant species was also observed. As shown in Figure 3.12, the training accuracy began at 94.97% after the first epoch and achieved its maximum (100%) following the final epoch of the algorithm. During the start to final epoch, *PSD-SE-DLM* training loss dropped from 39.18 to 0.52, and *PSD-SE-DLM* validation loss dropped from 14.0 to 2.40. The validation accuracy decreased from 99.32% at the beginning of the previous epoch to 99.16% at the end of that era. Training loss decreases dramatically from epoch 1 to 5 (see Figure 3.12), demonstrating that the model is improving and learning. An increase in generalization performance is also shown in the validation loss from epoch 1 to epoch 2. Nonetheless, there is a minor rise in validation loss during epoch 3, followed by a drop during epochs 4 and 5. During epoch 1, the training loss for the model is 39.18, which is the maximum it will get. It may imply that the model has room for improvement after the initial training. Training loss is minimized to 0.52 on epoch 5, the best performance of the model. During epoch 1, the validation loss is 14%, the maximum it will become for this model. During epoch 5, the validation loss is 2.40 percentage points lower than at any other time in the model's history. It may indicate that the model was not initially tuned to provide the best possible generalization and needs to be adjusted. Training and validation loss values decrease, indicating that the *PSD-SE-DLM* model is improving. However, the model may require additional optimization and fine-tuning to increase performance, as evidenced by the high values of training and validation loss at the outset. The *PSD-SE-DLM* model generally does well on both the training and validation data, as measured by the accuracy values. Figure 3.12 shows that at the end of the second epoch, the model has learned to identify all training data correctly. The model needs work on its generalization to unseen data if the validation accuracy stays at 99.16% as it does in the remaining epochs.

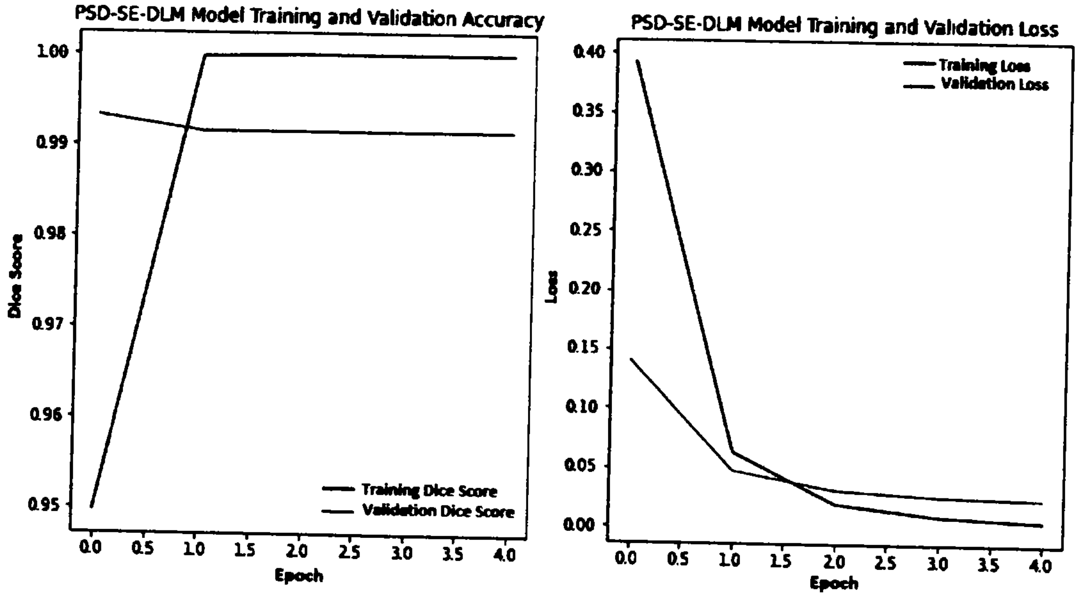


Figure 3.12: The Proposed *PSD-SE-DLM* Model Accuracy and Loss Graph

Accuracy, recall, precision, and f1 score for the *PSD-SE-DLM* method on the test set of unseen data are detailed in Table 3.8. As can be viewed in Table 3.5, the *PSD-SE-DLM* technique did quite well on the test data. Precision and recall for the guava plant leaf species were both 99%, and recall was 100%. Java plum class obtained a 100% f1 score, recall rate, and precision, respectively while the potato class achieved the f1 score of 99%, 98%, and 100%, respectively. High scores for precision and recall show that the model is effective at distinguishing between plant species. As shown in Table 3.5, the model has a perfect score for the guava and java plum species and a 98.5% score for the Potato species. The model's accuracy and recall are substantial, as measured by the f1 score (the harmonic mean of these two metrics) for all three plant species. On average, the suggested *PSD-SE-DLM* model has a 99.48% success rate in classifying all three plant species.

An intuitive visual indication of a model's classification accuracy was obtained using a confusion

Table 3.5: Classification Accuracies, Precision, Recall & F1 Score of the Suggested *PSD-SE-DLM* Technique.

Performance Measures	Guava	Java Plum	Potato	Average
Accuracy	100%	100%	98.5%	99.48%
Precision	99%	100%	100%	-
Recall	100%	100%	98%	-
F1 Score	99%	100%	99%	-

matrix *CM*. The *CM* displayed the accurate guesses along the diagonal and the wrong ones off to the side. A darker colour indicated that the proposed model of the relevant class had a greater classification accuracy, whereas a lighter colour indicated misclassified samples were present. Figure 3.13 depicts the *CM* utilized to assess the proposed model's efficiency on the test set. Overall, the *PSD-SE-DLM* model performed extremely well in classifying plant leaf species, with an accuracy of 99.48%, as depicted in Figure 3.13. All photos of guava and java plum leaves were correctly categorized, demonstrating the model's proficiency in identifying these plant species. The model also performed very well when classifying photos of potatoes, with only three out of 200 cases being misclassified, for an accuracy of 98.50%. Results from the confusion matrix showed that the suggested *PSD-SE-DLM* model performed exceptionally well on the experimental dataset, correctly classifying all of the plant leaf species tested.

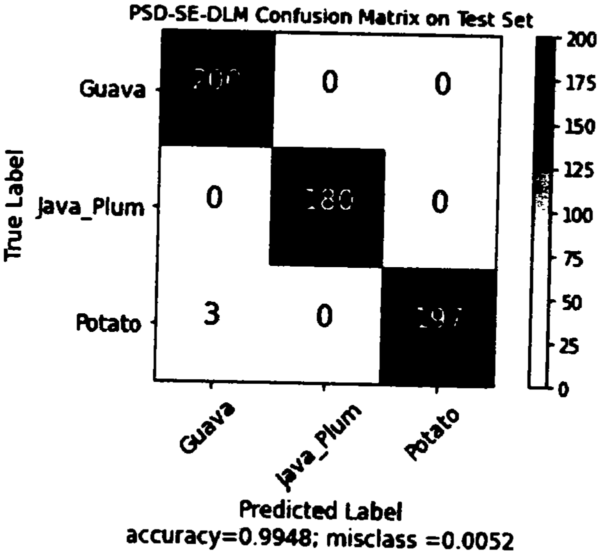


Figure 3.13: The Proposed *PSD-SE-DLM* Model Confusion Matrix on Test Set

Figure 3.14 demonstrates how the *ROC* curve can be used to assess the proposed *PSD-SE-DLM* model's efficacy. The potato class is depicted in a pale green colour, while the element of chance is shown in blue. The light blue tint signifies the guava. The orange colour indicates the java plum class. The *PSD-SE-DLM* model outperformed its competitors, with classes like guava and java plum obtaining around 100% and the potato class getting an area under the curve of 99%. The area under the curve (AUC) values between 80 and 90 in a ROC plot are considered good, and AUC values between 90 and 100 are considered excellent in classification issues, as stated in [236].

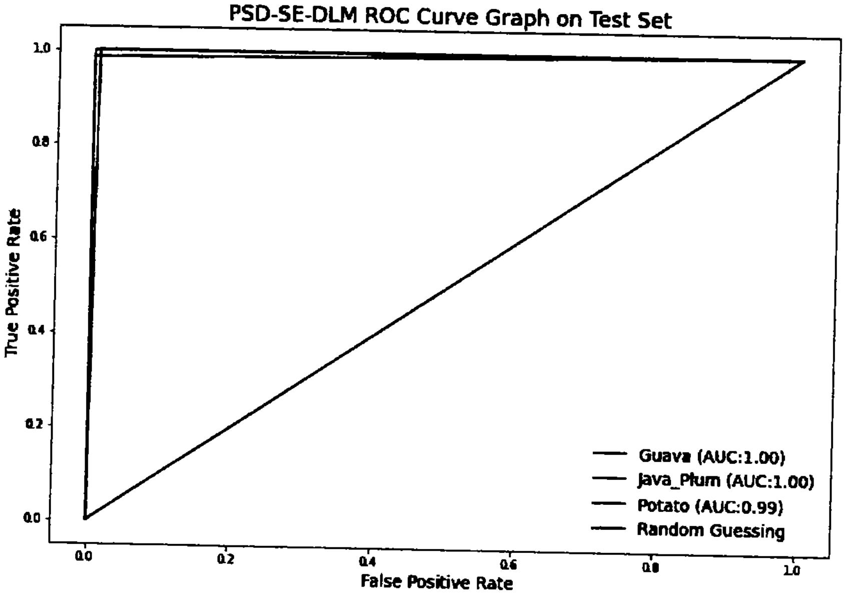


Figure 3.14: The Proposed *PSD-SE-DLM* Model *ROC* Graph on Test Set

All the evaluation measures such as, precision, recall, f1 score, accuracy, confusion matrix, and ROC curve showed that the proposed *PSD-SE-DLM* model has achieved excellent performance to classify the guava, java plum, and potat plant leaf species. The proposed *PSD-SE-DLM* model has the highest classification accuracy (99.48%) on the experimental dataset. Using a stack ensemble technique, the proposed *PSD-SE-DLM* model mixes many base models to make use of the best features of each and boost classification accuracy.

3.3.3 Comparison Analysis of the Proposed *PSD-SE-DLM* Model with Existing Techniques

In literature no study has been found that worked on guava or potato or java plum or combination of three plant species. Therefore, comparison of this research is not possible. We compare our model with the other plant leaf species. The study presented in [46] proposed an ensemble model which can detect the multiple plant leaf species and got the 92.60% accuracy as presented in the Table 3.6. Xio et al. [44] got 92.8% accuracy on InceptionV3 and ResNet50 model obtained 92.4% accuracy. Pereira et al. [45] investigated the AlexNet and achieved 89.75% accuracy. Pearline and Kumar [58] investigated CART, KNN, NB, MLR, and LDA classifier and obtained 98.71% accuracy using MLR classifier. The proposed ensemble model used the stack ensemble technique to find the guava, potato and java plum plant leaf species and obtained 99.48% accuracy. The suggested ensemble model used the stack ensemble technique for the guava, potato, and java plum leaf species and achieved 99.48% accuracy, as depicted in Table 3.6. The results showed that the suggested PSD-SE-DLM technique attained higher accuracy than the existing studies.

Table 3.6: Comparison of the Proposed *PSD-SE-DLM* Model with other Existing Models.

Ref.	Year	Methodology	Dataset	Accuracy
[46]	2017	Ensemble Model	LifCLEF2017	92.60%
[44]	2018	InceptionV3, ResNet50	PlantCLEF, Oxford Flower	InceptionV3 (92.8%), ResNet50 (92.4%)
[45]	2019	AlexNet	DRGV2018, Flavia	89.75%
[58]	2022	CART, KNN, NB, MLR, and LDA	Flavia	98.71%
Proposed Model		*PSD-SE-DLM*	PLSD	99.48%

3.3.4 Comparison Analysis of the Proposed *PSD-SE-DLM* Model with State-of-the-Art Techniques

The proposed PSD-SE-DLM model was put to the test on the PLSD dataset by combining it with the MobileNetV2 [225], ResNet50 [231], GoogLeNet [235], and AlexNet [237] models using transfer learning. Thus, there was necessary to be consistent between studies concerning the environment and the methods used to enhance the data. The accuracy of modern deep learning methods is displayed in Table 3.7. As shown in Table 3.7, the accuracy of the MobileNetV2 model was 95.52%, ResNet50 was 96.10%, GoogLeNet was 92.35%, and AlexNet was 95.87%, and that of the suggested PSD-SE-DLM model was 99.48%. As shown in Table 3.7, the findings demonstrated that the proposed PDDCNN model achieved the best accuracy (99.75%) compared to other state-of-the-art models, as depicted in Figure 3.15.

Table 3.7: Comparison of the Proposed *PSD-SE-DLM* Model with other State-of-the-Art Models.

Ref.	**Methodology**	**Dataset**	**Accuracy**
[225]	MobileNetV2	PLSD	95.52%
[231]	ResNet50	PLSD	96.10%
[235]	GoogLeNet	PLSD	92.35%
[237]	AlexNet	PLSD	95.87%
Proposed Model	*PSD-SE-DLM*	PLSD	99.48%

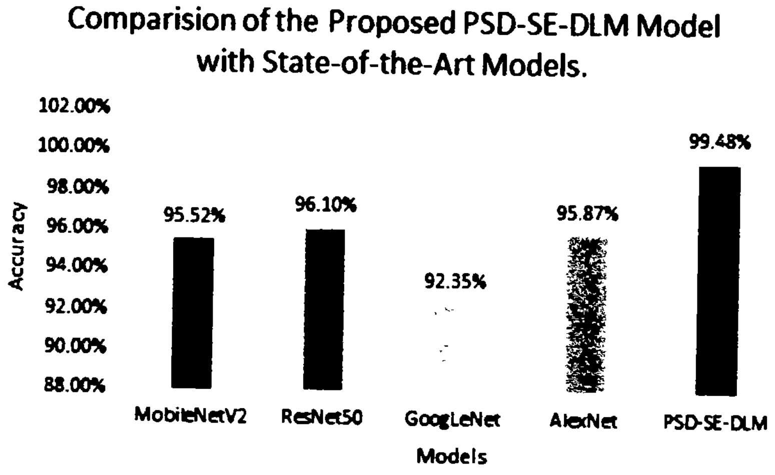


Figure 3.15: The Proposed *PSD-SE-DLM* Model's Comparison with State-of-the-Art Models.

3.4 Chapter Summary

When classifying and segmenting plant species using deep learning, detecting ones that have not been discovered before can be challenging. It is difficult to precisely separate and describes many different plant species since their outward appearances are strikingly similar. While the use of robust deep learning architectures has helped to boost classification accuracy, these structures frequently yield models that rely on a massive training dataset and hence need to be more scalable. This chapter proposes a hybrid approach to classifying the guava, potato, and java plum leaf species. The hybrid model that has been presented utilizes not one but two new methods. The guava, potato, and java plum plant species have been segmented using MobileNetV2-UNet architecture, and this is the first model to use this design. The Potato, Java Plum, and Guava Plant Species Detection using Stacking Ensemble Deep Learning Model (*PSD-SE-DLM*) is a second model that uses plant species detection stacking ensemble deep learning. To train the suggested models, two datasets called Plant Leaf Species Dataset (*PLSD*) and Plant Leaf Species Segmentation Dataset (*PLSSD*) were produced in Punjab, Pakistan. These datasets included healthy and diseased leaf samples of guava, java plum, and potato plants. The suggested model was completely automated compared to existing leaf species detection techniques. The plant leaves were photographed and manually labeled for training reasons. With the proposed model, real-time identification of plant leaves is also possible.

Chapter 4

Real–Time Multiple Guava Leaf Disease Detection from a Single Leaf Using Hybrid Deep Learning Technique

4.1 Overview

Because of its adaptability to many climates and soil types and its high medicinal and nutritional qualities, guava (Psidium guajava L.) has become commercially significant in the subtropics and tropics. The guava tree's fruit and leaves are extremely rich in minerals and nutrients [238]. Food production should be increased up to 70% by 2050 due to an increase in the population [5, 239]. Due to how they might affect the quality of a product, many methods for diagnosing plant diseases have been created [22–24]. These techniques include visual observation, spectrometer analysis [22], and molecular techniques, i.e., the polymerase chain reaction [23] and polymerase chain reaction [24]. These methods are not suitable as they are not time efficient, and sometimes they are much more expensive and need technical experts etc.

The advent of *CV*, *ML*, and *AI* has made improvements in developing computerized models, allowing for accurate and early identification of plant leaf diseases. Several methods, including traditional *ML* techniques like *KNN*, *SVM*, random forest, etc., are utilized for automating the detection of plant diseases. A significant difference prevails in the accuracy of the procedures applied. [238, 240]. Though the proposed methods achieved multiple detections, identification accuracy is comparatively low, particularly for managing massive databases containing various imperative features.

Deep Learning (*DL*) has recently become the standard for tackling complex problems in the real world. The precision and efficiency of the deep learning approaches are unmatched. The primary applications of *DL* in agriculture [241–243] have improved agricultural production through evolution, control, and sustainability. *DL* helps illuminate a few complex tasks like pattern analysis, image classification, facial expression, medical disease classification, relation extraction, sentiment and natural language processing [244–248].

Different deep learning models are trained to do different things when detecting and identifying plant leaf diseases. Let us get into these endeavors: One use of deep learning for plant disease diagnosis is determining whether or not an image of a leaf is of a healthy or sick kind. The model is educated using a collection of labeled images of both healthy and diseased leaves. Disease detection aims to create a model that can reliably differentiate between healthy and sick leaves using only the leaf images and the visual patterns and attributes collected from them. By examining new, unseen leaf images (test data), the trained model can automatically detect the presence of illnesses in plant leaves. Disease detection in plant leaves using deep learning takes it one step further by trying to identify the particular pathogen or illness that has affected the leaf. Training a deep learning model requires a dataset of annotated photos of various leaf diseases, each belonging to a predefined class. The goal of disease identification is to create a model that can reliably assign an observed symptom to one of several disease categories based on an image of a leaf. The model must be taught to recognize the specific symptoms and patterns indicative of various diseases. Once the model has been trained, it can identify the disease in a leaf image. In summary, when discussing plant leaf diseases in the context of deep learning, the distinction between healthy and sick leaves is the primary focus, while identifying the exact disease or pathogen present in a leaf image is the primary goal of plant leaf disease identification. There is a difference in the classification granularity between the two objectives, but both require training deep learning models on labeled datasets.

Guavas are native to South America, although many additional types of guava are grown in other parts of the world. Different environmental conditions, guava cultivars, leaf colours, and other factors contribute worldwide to the fantastic range of guava diseases. Diseases of various kinds attacked most crops or leaves. One disease on one leaf is the main focus of the literature, although previous studies overlooked the coexistence of many disorders on a single leaf. As reported in the literature, real-time disease diagnosis and localization is another issue [249]. We tried to find research reports on detecting numerous guava diseases on a single leaf. The localization of disease spots in guavas still needs to be explored. Another issue is that no comprehensive study exists for segmenting guava leaves. There is only one illness dataset in the literature for the guava leaf. Unfortunately, there is currently no accessible real-time dataset on numerous leaf diseases in guava. Therefore, creating a dataset that can simultaneously identify multiple guava leaf diseases is essential. As a result, there is an urgent need to develop a method for identifying and segmenting guava leaves, which may detect many illnesses simultaneously on a single leaf. Those guava illnesses and segmentation issues have been prioritized in the proposed endeavor.

The following are the chapter's contributions:

1. To separate the infected guava patches, we created a segmentation method called Guava Infected Patches Modified MobileNetV2 and U-Net (*GIP-MU-NET*). Modified MobileNetv2 is used for the encoder in the suggested *GIP-MU-NET* technique, whereas up-sampling layers of U-Net are utilized for the decoder.
2. A unique Guava Leaf Segmentation Model (*GLSM*) is designed to distinguish between healthy and diseased areas on guava leaves.
3. A real-time Guava Multiple Leaf Diseases Detection (*GMLDD*) approach is created using YOLOv5 to detect and localize several disorders from a single guava leaf. It is the first time a model has been used to detect many diseases in a single photograph of a guava leaf. The suggested method can accurately detect and localize various diseases from a single image, such as anthracnose, nutrient deficiency, wilt diseases, and insect attack.
4. A first-ever Guava Leaf Disease Dataset (*GLDD*) is developed to identify many disorders on a single leaf. The dataset consists of five classes: anthracnose, nutrient deficiency, wilt diseases, insect assault, and healthy.

4.2 Proposed Work

This study proposes a hybrid *DL* model for various disorders detection and localization on a single guava leaf in real-time. There are three stages to the suggested procedure. The first component *GIP-MU-NET* finds the infected guava patches from the guava tree pictures. Second, we present the GLSM to separate leaves from guava images. As a final step, the *GMLDD* model detects numerous illnesses of a guava leaf. Figure 4.1 depicts the suggested method's systematic flow diagram.

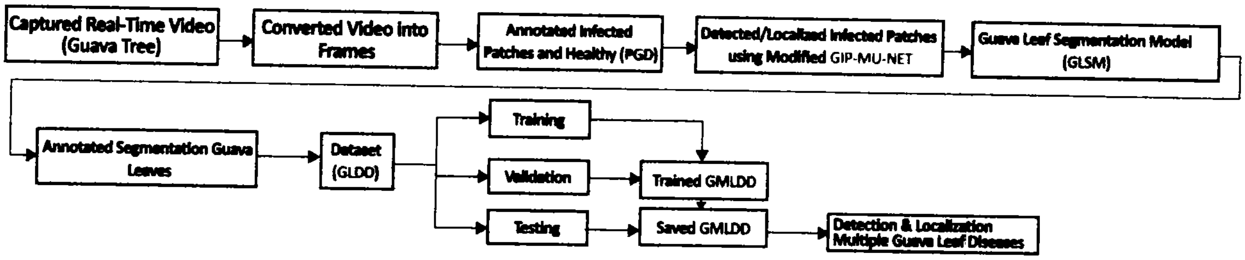

Captured Real-Time Video (Guava Tree)
Converted Video into Frames
Annotated Infected Patches and Healthy (PGD)
Detected/Localized Infected Patches using Modified GIP-MU-NET
Guava Leaf Segmentation Model (GLSM)
Annotated Segmentation Guava Leaves
Dataset (GLDD)
Training
Validation
Testing
Trained GMLDD
Saved GMLDD
Detection & Localization Multiple Guava Leaf Diseases

Figure 4.1: Diagram Illustrating the New Hybrid *DL* Approach.

4.2.1 Data Preparation

The success of a *DL* technique is highly dependent on using a high-quality dataset. We created the Guava Patches Dataset and Guava Leaf Disease Dataset, real-time datasets gathered in the Okara district of Central Punjab, Pakistan. Here is an in-depth breakdown of the various types of data sets:

4.2.1.1 Guava Patches Dataset (*GPD*)

A real-time dataset was developed in the form of videos and pictures. The dataset capturing details is already described in section 3.2.1.1. With the help of Python code, guava plant videos were converted into frames (images). Then, with the assistance of plant disease experts, patches of plant images were annotated into infected ones (Figure 4.2), ensuring the dataset's authenticity. LabelMe tool [250] was used to make the annotations, and the binary masks of the annotated images were generated with the help of python code. The Guava Patches Dataset (*GPD*) contained 1190 images of healthy and infected patches exhibited in Table 4.1. The dataset can be accessed online [251].

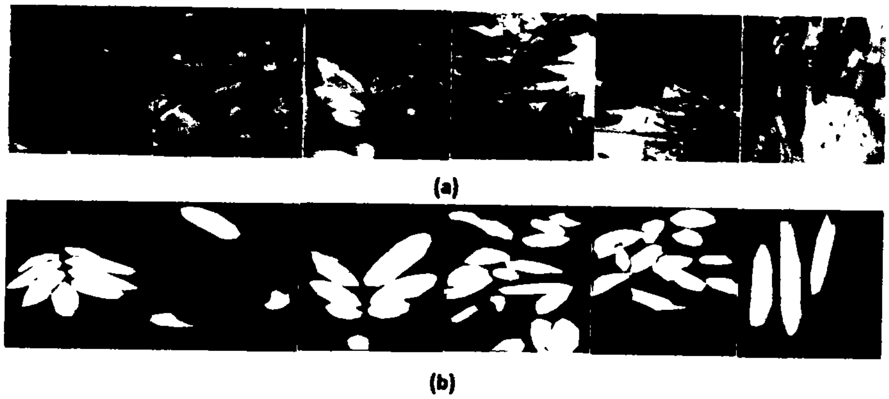


Figure 4.2: (a) Infected Patches Images (b) Infected Patches Masks.

Table 4.1: Guava Patches Dataset (GPD) Summary

Split	**Images**
Training	956
Validation	240
Testing	120
Total	**1196**

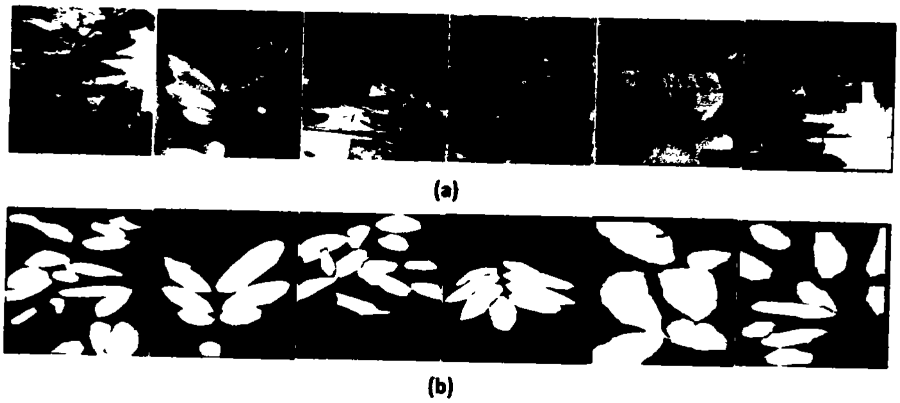


Figure 4.3: (a) Examples of Guava Leaf Images and (b) Examples of Guava Images Leaves' Masks.

4.2.1.2 Guava Leaf Disease Dataset (*GLDD*)

Using the Guava Patches Dataset, the guava leaf images were extracted using the proposed *GLSM* model. The guava leaf images and their masks were shown in Figure 4.3. The leaves were annotated with the help of plant pathologists into five classes: nutrient deficiency, insect attack, anthracnose, wilt, and healthy, as depicted in Figure 4.4. The five classes contain 2542 images with 6241 targets (labels/annotations) are depicted in Table 4.2. The dataset is available to be viewed on the web [252]. After annotating the images of *GPD* and *GLDD*, the Roboflow [253] was employed to perform the auto-orient with *EXIF*-orientation stripping and resizing the image size into 416 x 416. Deep learning models needed massive training data, which helped avoid over-fitting. As a result, the datasets were improved with the aid of data augmentation tools used on the Roboflow website. Several data augmentation methods were used for this purpose. It includes horizontal flip with 50% probability, vertical flip with 50% probability, 90-degree rotation with equal chance (upside-down, clockwise, and counter-clockwise), random rotation degree range -15 and +15, random brightness between -25 and +25%, and random shear degree range -15 to +15 vertically and -15 to +25 horizontally were all used.

Table 4.2: Guava Leaf Diseases Dataset (GLDD) Summary

Split	Images	Class	Label Samples	Total Samples
Training	2346	Anthracnose	2180	5777
		Healthy	447	
		Insect Attack	1398	
		Nutrient Deficiency	1281	
		wilt	471	
Validation	98	Anthracnose	53	232
		Healthy	24	
		Insect Attack	67	
		Nutrient Deficiency	69	
		wilt	19	
Testing	98	Anthracnose	53	232
		Healthy	24	
		Insect Attack	67	
		Nutrient Deficiency	69	
		wilt	19	
Total	**2542**		**6241**	**6241**

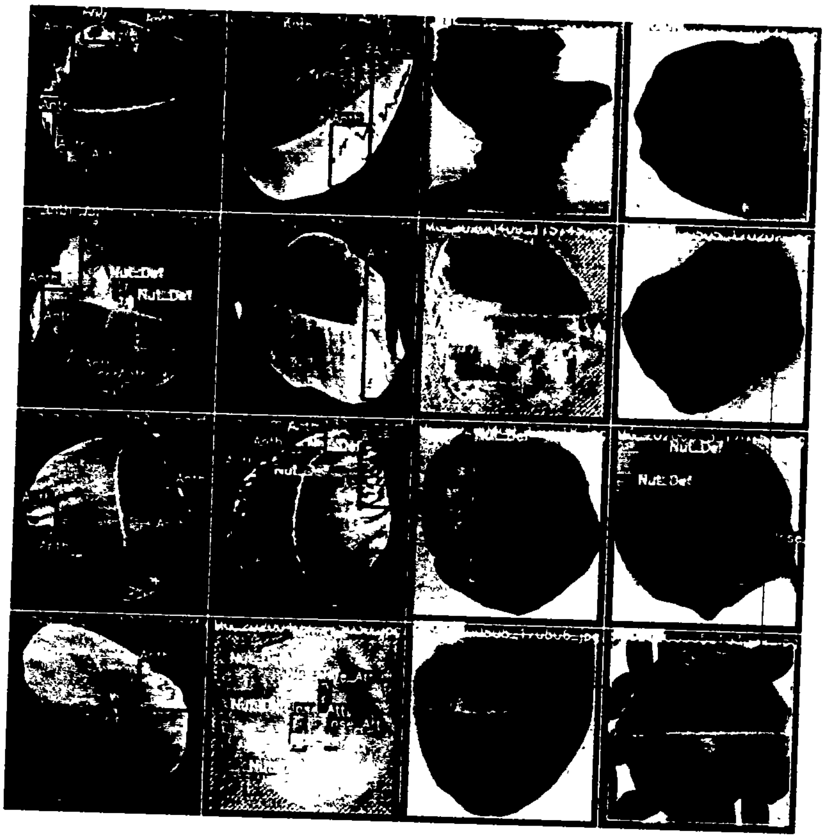


Figure 4.4: Examples of Guava Leaf Multiple Diseases Images.

4.2.2 Image Preprocessing (Data Division)

The suggested technique was trained, validated, and tested by dividing datasets into various groups. The model was then tested on unseen data to ensure it was accurate. According to Table 4.1, 1196 pictures were used as as a part of the *GPD* training set, with the same number of masks (targets/labels) for the infected classes. The *GPD* dataset's training set included 956 photographs and targets, while the validation set had 240 photos with masks. However, the test set had 120 images and 120 masks.

Each training, validation, and test set in the *GLDD* contained 2346 images and 5777 targets (see Table 4.2). The training set of *GLDD* consisted of 2180, 447, 1398, 1281 and 473 labels for anthracnose, healthy, insect attack, nutrient deficiency and wilt class. The validation set included 53 targets for anthracnose, 24 for healthy, 67 for insect attack, 69 for nutrient deficiency and 19 annotations for wilt class. At the same time, the test set contained 53, 24, 67, 69 and 19 images for anthracnose, healthy, insect attack, nutrient deficiency and wilt class.

4.2.3 The Proposed Network

4.2.3.1 Guava Infected Patches Modified MobileNetV2 and U-Net (*GIP-MU-NET*) Segmentation Technique

First, the infected guava areas must be cut into smaller pieces. In order to do this, we created a brand new deep learning model we call *GIP-MU-NET*. The proposed method implements an encoder-decoder U-Net [254] architecture. However, we replaced the encoder part with the modified MobileNetV2. The computational cost is further reduced by eliminating the block IDs 6, 9, 10, 13, and 16. The filters are also decreased from 160 to 128, 320 to 160, and 1280 to 256. As a decoder, the up-sampled component of U-Net is employed. MobileNetV2 [225] is a neural network designed for machines with low processing capability, such as mobile devices. The MobileNetV2 offers favourable results in terms of accuracy while needing less memory and processing cost. When this is done, they become a superfast network for handling images. MobileNetV2 is a lightweight CNN used in synchronous tasks due to its reduced number of trainable parameters and reduced computational requirements. Figure 4.5 depicts the *GIP-MU-NET* architecture with encoder-decoder components. Features are extracted from the input dataset using an updated version of MobileNetV2; *GIP-MU-NET* receives these features to complete the segmentation task.

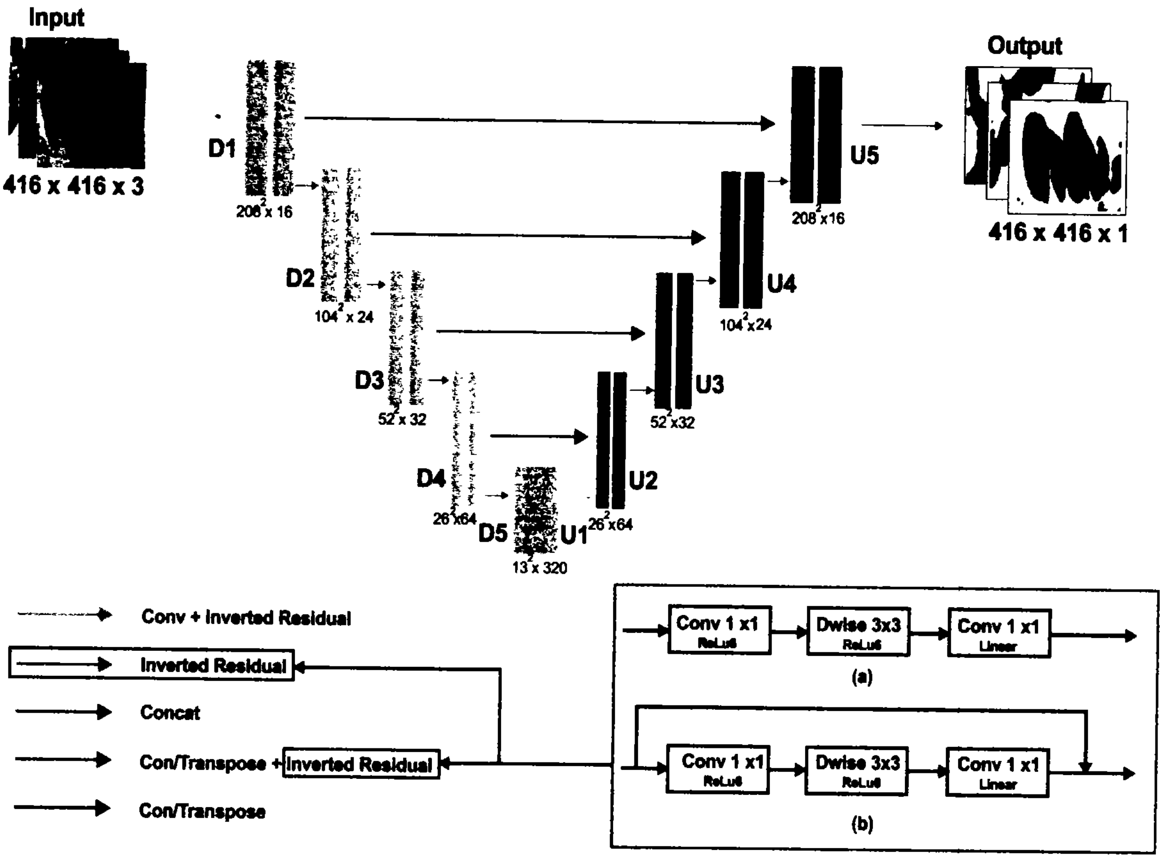


Figure 4.5: Architecture of the Proposed GIP-MU-Net Model.

4.2.3.2 Guava Leaf Segmentation Model (*GLSM*)

This study aims to make the Guava Leaf Segmentation Model (*GLSM*) from a U-Net-like model to segment guava leaves from the images. Figure 4.6 shows the proposed *GLSM* model's structure. The encoder section of this model accepts 416x416x3 dimensions as input, and the encoder part's output is then passed on to the next part of the model. As a result, the encoder part's output was processed by the decoder. Both an entering block and a residual block make up the encoder. The *BN* and *ReLu* activation function are utilized in the Conv2D encoding algorithm. The residual block consists of *ReLu*, SepereableConv2D, and batch normalization. Three residual blocks are employed in the encoder. The structure of each residual block is the same. Sixteen filters are applied in the entry block. Each of the three residual blocks has different filters: 32 for the first, 64 for the second, and 128 for the third. The decoder employs ReLu, Conv2D transpose, and batch normalization to correct the inversion of the residual blocks. The leftover blocks that were inverted have been joined together. In the decoder, we employ the utilization of four inverse residual blocks.

It is the opposite in the decoder, where the filters are utilized in the first, second, third, and fourth inverted residual blocks (128, 64, 32, and 16 filters, respectively). The examples of guava leaf images concerning respective masks are represented in Figure 4.3.

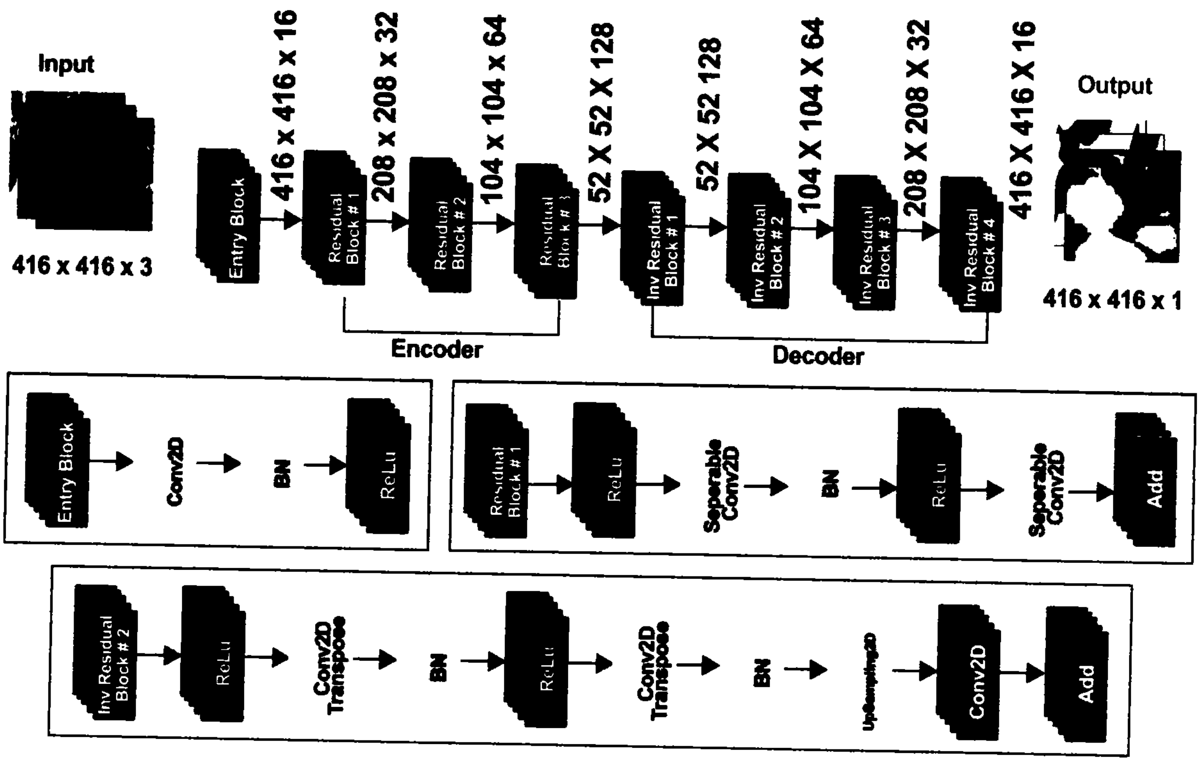


Figure 4.6: Architecture of Proposed GLSM Method.

4.2.3.3 Guava Multiple Leaf Diseases Detection (*GMLDD*) Technique on a Single Leaf

The proposed Guava Multiple Leaf Diseases Detection (*GMLDD*) Technique on a Single Leaf used YOLOv5 architecture, and it is already discussed in section 2.6. We have changed the default optimizer (*SGD*) to Adam, and it is because the Adam optimizer is one of the extensively used optimizers. It has been adopted as a standard for deep learning studies and is suggested as the default optimization approach. Compared to other optimization methods, this requires less tuning and has fewer overheads regarding development complexity, execution time, memory usage, and storage requirements [255]. Therefore, the *Adam* optimizer was employed in this study. All other configuration details of the proposed GMLDD model are described in the experimental setup.

Table 4.3: Configuration Parameters of the Proposed GMLDD Model

Model Name	Guava Multiple Leaf Diseases Detection Model
Number of Classes	5 (Anth, Healthy, Inse_Att, Nut_Def, and Wilt)
Image Size	416 x 416
Batch Size	32
Epochs	500
Training Optimizer	*Adam*

4.2.4 Experimental Setup

The approaches were trained and evaluated using powerful Graphics Processing Units *GPUs* for free on Google Colab. No special configuration was necessary. The datasets were labeled using the publicly accessible LabelMe tool [250]. Image scaling, data enhancement, and data partitioning were all accomplished with the help of the Roboflow [253] website. The models were completely retrained.

The *GIP-MU-NET* uses the following hyperparameters: (416×416) training image size, 8-batch size, 125-epochs, *Adam* optimizer, binary cross-entropy loss function, and an initial learning rate of (1e-4). In this case, the *LR* is decreased by a factor of 0.1 thanks to the ReduceLROnPlateau procedure. If the Val accuracy has not changed after ten epochs, the minimum *LR* will be set to 1e-6. The best possible Val accuracy weight was safely archived.

For training a *DL* network, the *GLSM* model uses hyperparameters (416×416 image size, 4 batch size, 75 epochs, and 1e-4). The *Adam* optimizer and the *SGD* optimizer are used in experiments. The binary cross-entropy loss function is used to construct the proposed technique. When the ReduceLROnPlateau function is used, the *LR* is decreased by 0.1. After ten epochs, the *LR* should be at least 1e-6 if the val accuracy has improved.

Table 4.3 displays the setup information for the proposed *GMLDD* model, which employs images with a resolution of 416×416. The *GMLDD* model has a 32-batch size and a 500-epoch time step. The suggested model is trained with an *Adam* optimizer. Anthracnose, healthy, insect assault, nutritional deficit, and guava wilt illness are some of the categories used.

4.2.5 Evaluation Measures

The accuracy of the suggested model was measured in various ways. Section 3.2.7 reviewed the recall, precision, and f1 score. The performance of the recommended strategy was also assessed using mean Average Precision (*mAP*), in which the harmonic average of recall and accuracy is determined.

mean Average Precision (*mAP*): We compute the mAP at various levels of IoU for each class k, and then we average these results to get the mAP for the entire set of test data.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{4.1}$$

$$AP_k \;=\; the\, AP\, of\; class\, k$$

$$n \;=\; the\, number\;\; of\;\; classes$$

4.3 Results and Discussion

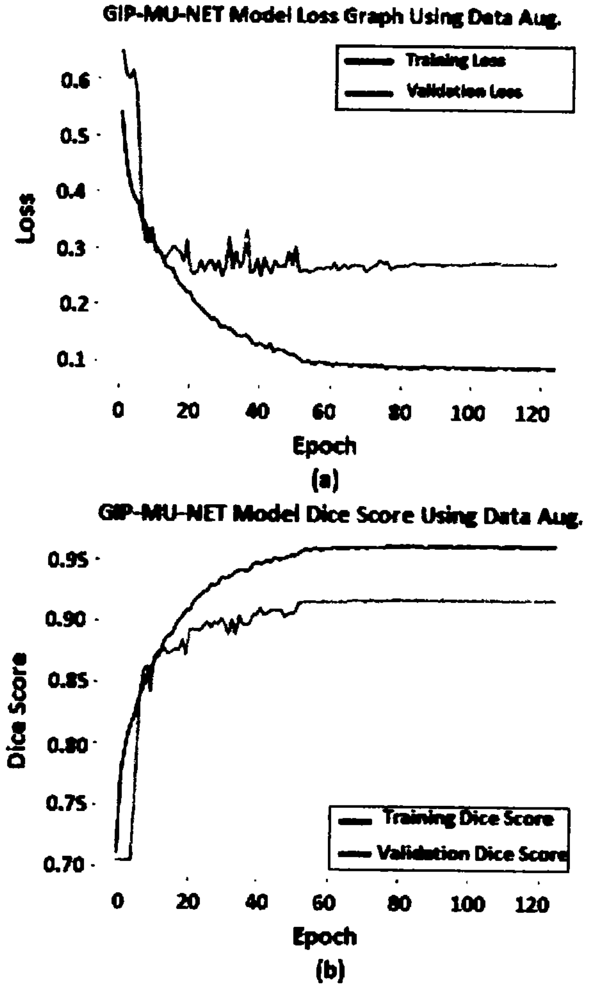
All the proposed methods' performance was evaluated using different evaluation measures such as precision, recall, f1 score, accuracy, and mAP. In this section all the experimental results focused on the following:

1. The Guava Patches Dataset (*GPD*) was used to analyze the efficiency of the suggested *GIP-MU-NET* methods.
2. We used the Guava Leaf Segmentation Model (*GLSM*) to separate individual leaves from clusters of leaves.
3. The proposed Guava Multiple Leaf Diseases Detection Model performance was measured using Guava Leaf Disease Dataset (*GLDD*) to identify and localize the guava multiple diseases on a single leaf.
4. We compared the proposed *GMLDD* model with other *YOLO* variants.

4.3.1 Proposed Guava Infected Patches Modified MobileNetV2 and U-Net (*GIP-MU-NET*) Performance

We conduct two experiments to evaluate the proposed GIP-MU-NET model performance. Data augmentation techniques are applied to the training set in the first experiments. Figure 4.7 depicts the proposed model's dice score and loss performance throughout the training and validation sets. Fast loss reduction during training is visible up to epoch 57, after which it stabilizes at less than

10%. However, after the first 15 epochs, the validation loss stabilizes at around 30%. Figure 4.7 displays the training and validation dice score of the suggested *GIP-MU-NET*. Results show that training dice score rapidly improves up to 40 epochs, reaching nearly 96.62%, and validation dice score similarly improves up to 20 epochs, reaching nearly 96.62%, while 92.38%.



(a)

(b)

Figure 4.7: GIP-MU-NET Model Dice Score Graph using Data Augmentation.

Table 4.4 displays the performance of the suggested *GIP-MU-NET* technique on data that has not been viewed previously (the test set). It demonstrates that the proposed model was 92.41% accurate on unseen data (Test Set) using data augmentation techniques applied on training set, as shown in Figure 4.8. Figure 4.9 displays the projected patch results from the suggested model. The *GIP-MU-NET* performs exceptionally well according to all criteria.

Table 4.4: The Proposed *GIP-MU-NET* Dice Score Comparison on Test Set.

	Images	Dice Score
Data Augmentation	120	92.41%
Without Data Augmentation	120	64.50%

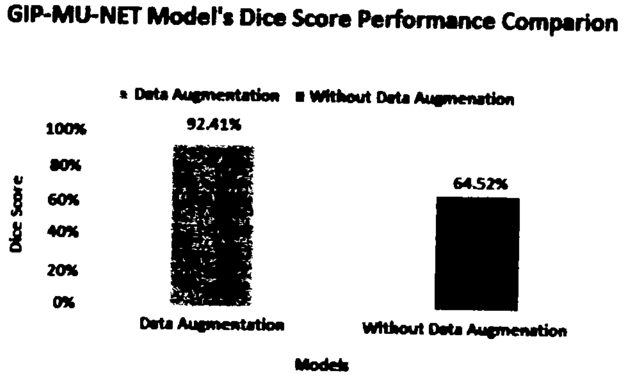


Figure 4.8: The Proposed *GIP-MU-NET* Dice Score Comparison Graph on Test Set.

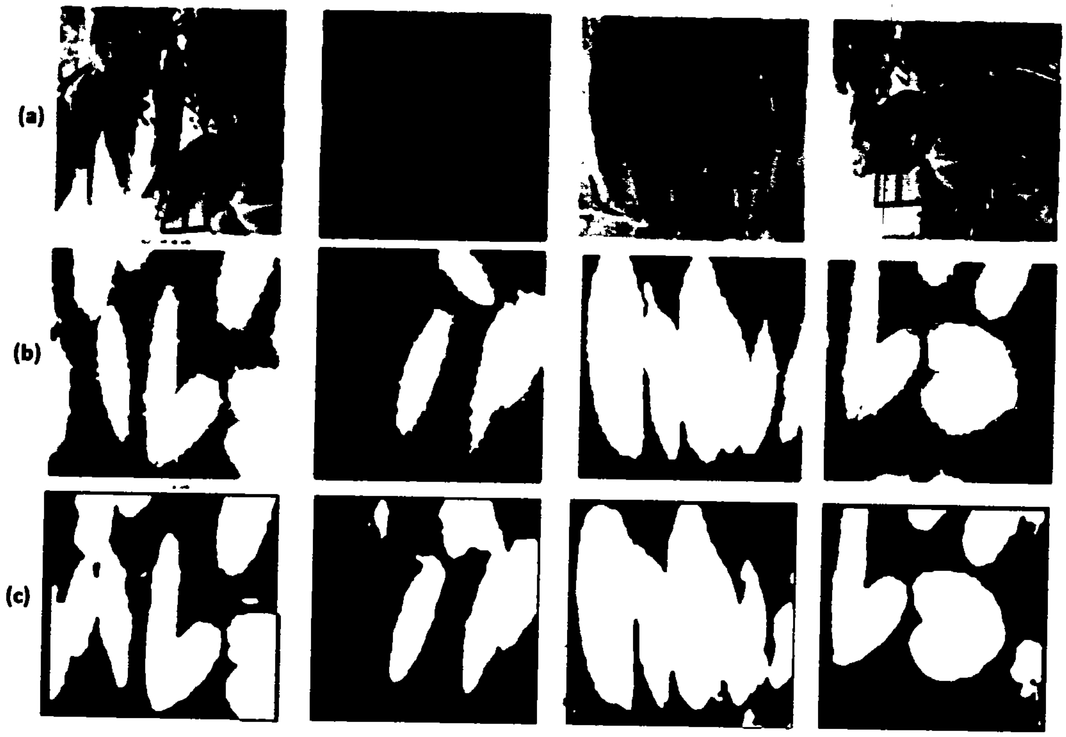


Figure 4.9: (a) Original Images, (b) Ground Truth, and (c) Infected Patches Predictions Applying Data Augmentation Techniques.

We do not use data augmentation methods in the second experiment on the training set. The proposed model's dice score and loss performance across the training and validation sets are shown in Figure 4.10. Loss suppression is performed gradually throughout training, as depicted in 4.10 (a). However, the validation loss worsens when training continues, revealing the overfitting problem. During the training process, it does not appear steady. Dice score during training and validation for the proposed *GIP-MU-NET* is shown in Figure 4.10 (b) below, without using any data augmentation methods. It is clear that as training progresses, dice score improves, although validation dice score initially appears unstable before stabilizing somewhat. Both the training and validation phases have demonstrated the overfitting issue. Without employing data augmentation techniques on the training set, the suggested *GIP-MU-NET* model can reach an dice score of 92.26% by the end of the training process and obtains 64.94% validation dice score.

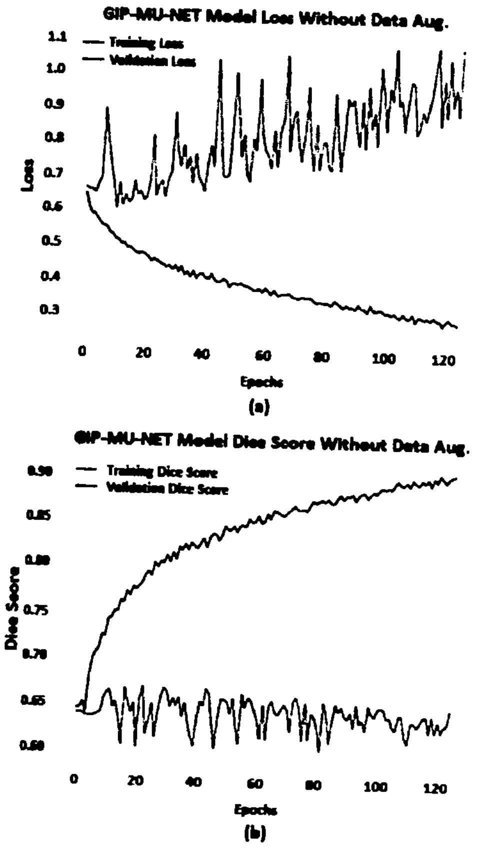


Figure 4.10: The Proposed *GIP-MU-NET* Dice Score Graph Without Data Augmentation.

Figure 4.11 displays the prediction patch results from the proposed *GIP-MU-NET* model. The results show that the proposed *GIP-MU-NET* model faces the problem of overfitting without applying data augmentation techniques. The proposed *GIP-MU-NET* model obtains 64.52% dice score on the test data when we do not apply any data augmentation technique on the training set, as shown in Table 4.4 and Figure 4.8.

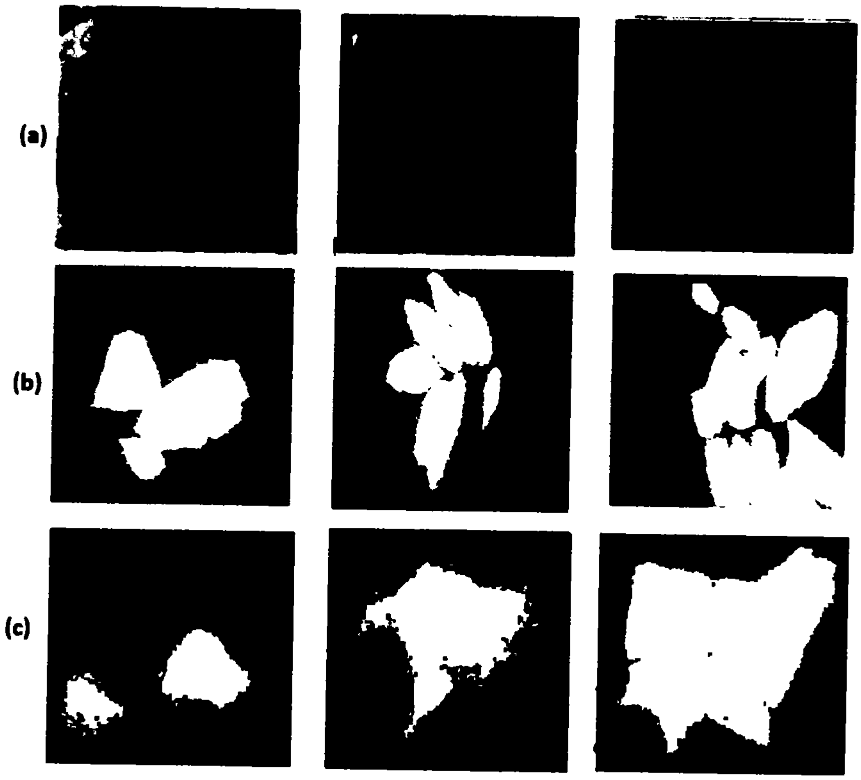


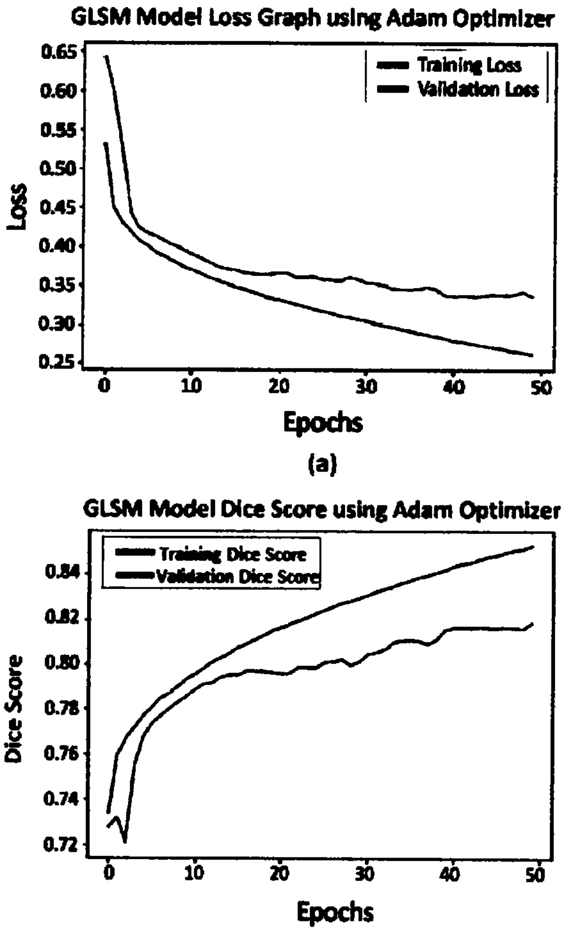
Figure 4.11: (a) Original Images, (b) Ground Truth, and (c) Infected Patches Predictions Without Data Augmentation Techniques.

All the evaluation matrix show that the proposed *GIP-MU-NET* having data augmentation techniques apply on the training set obtained the excellent performance as compared to do not apply data augmentation technique.

We propose the following changes to the *GIP-MU-NET* to increase its efficiency: 1). If the dataset were improved, it might work better. Overfitting happened if training data needed to be sufficiently large for the deep learning methods. Improving the dataset may help eliminate the overfitting issue [256]. 2). Data augmentation methods are an additional tool for improving the dataset. Only three data augmentation strategies were employed in this study. So, with the proper data augmentation techniques, we can make the dataset better.

4.3.2 Proposed Guava Leaf Segmentation Model (*GLSM*) Performance

The proposed *GLSM* model performance has been observed by conducting two experiments. In the first experiment, the *Adam* optimizer is used, and the other experiment is utilized the *SGD* optimizer. In the first experiment, the performance of the proposed *GLSM* model is discussed by training and validation process using *Adam* optimizer, as shown in Figure 4.12. Training loss for the proposed *GLSM* technique drops swiftly to 25% until 50 epochs (Figure 4.12). In comparison, validation loss drops rapidly to 37% until 20 epochs (also shown in Figure 4.12), and then loss declines slowly to 35% after 50 epochs. Early on, the proposed technique shows significant improvement in training dice score. After 50 epochs, it reaches 86%, and after 17 epochs, it reaches 82% for validation dice score. The training and validation process shows exceptional segmentation results using the *Adam* optimizer, as depicted in Figure 4.12.



(a)

(b)

Figure 4.12: GLSM Model Dice Score Graph Using Adam Optimizer.

Table 4.5: The Proposed *GLSM* Dice Score Comparison on Test Set.

Optimizer	Images	Dice Score
Adam	119	83.40%
SGD	119	73.79%

The proposed *GLSM* technique is evaluated by running test data that has yet to be analyzed. According to Table 4.5, the proposed strategy improved dice score by 83.40% on 119 test images, as depicted in Figure 4.14. Figure 4.13 displays the proposed *GLSM* segmentation predictions for guava. The proposed technique performs well even with a smaller data set. The efficiency of the proposed *GLSM* method can be improved through extensive dataset training. We need considerably larger datasets or data augmentation methods to prevent underfitting and overfitting.

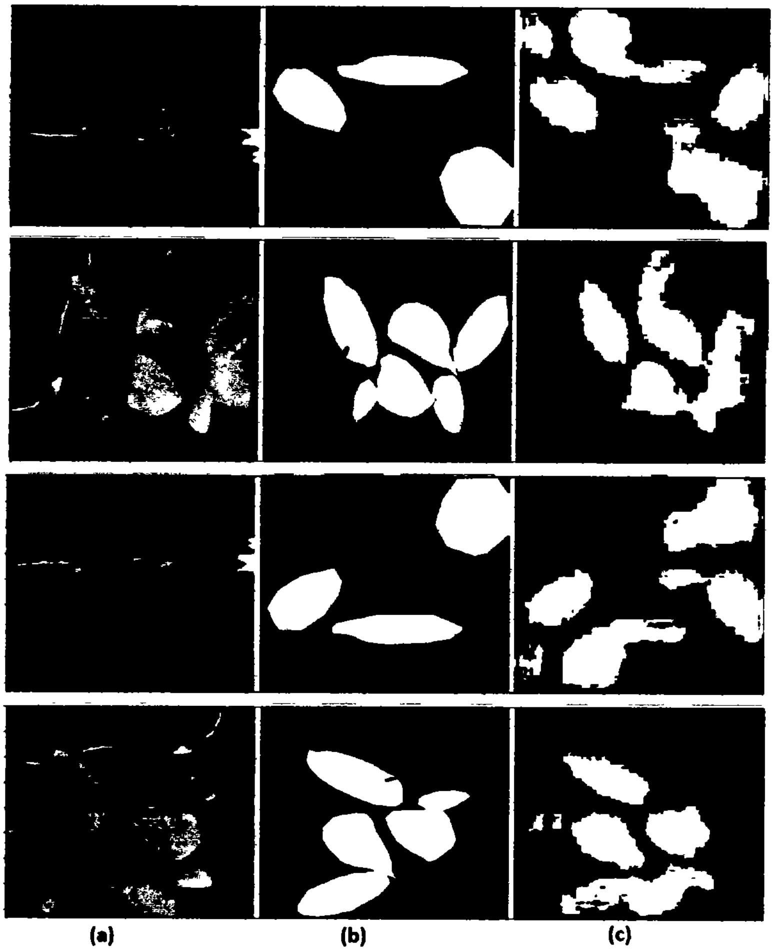


Figure 4.13: (a) Original Image (b) Generated Mask (c) Predicted Mask of the *GLSM* Using *Adam* Optimizer.

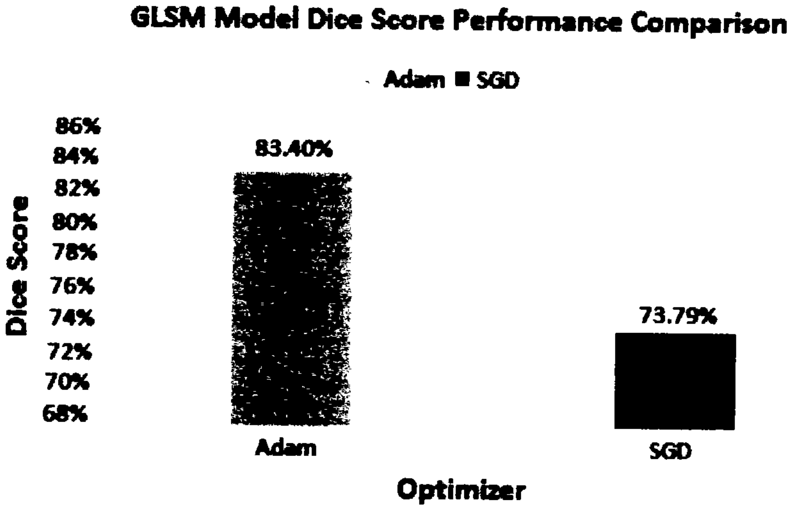
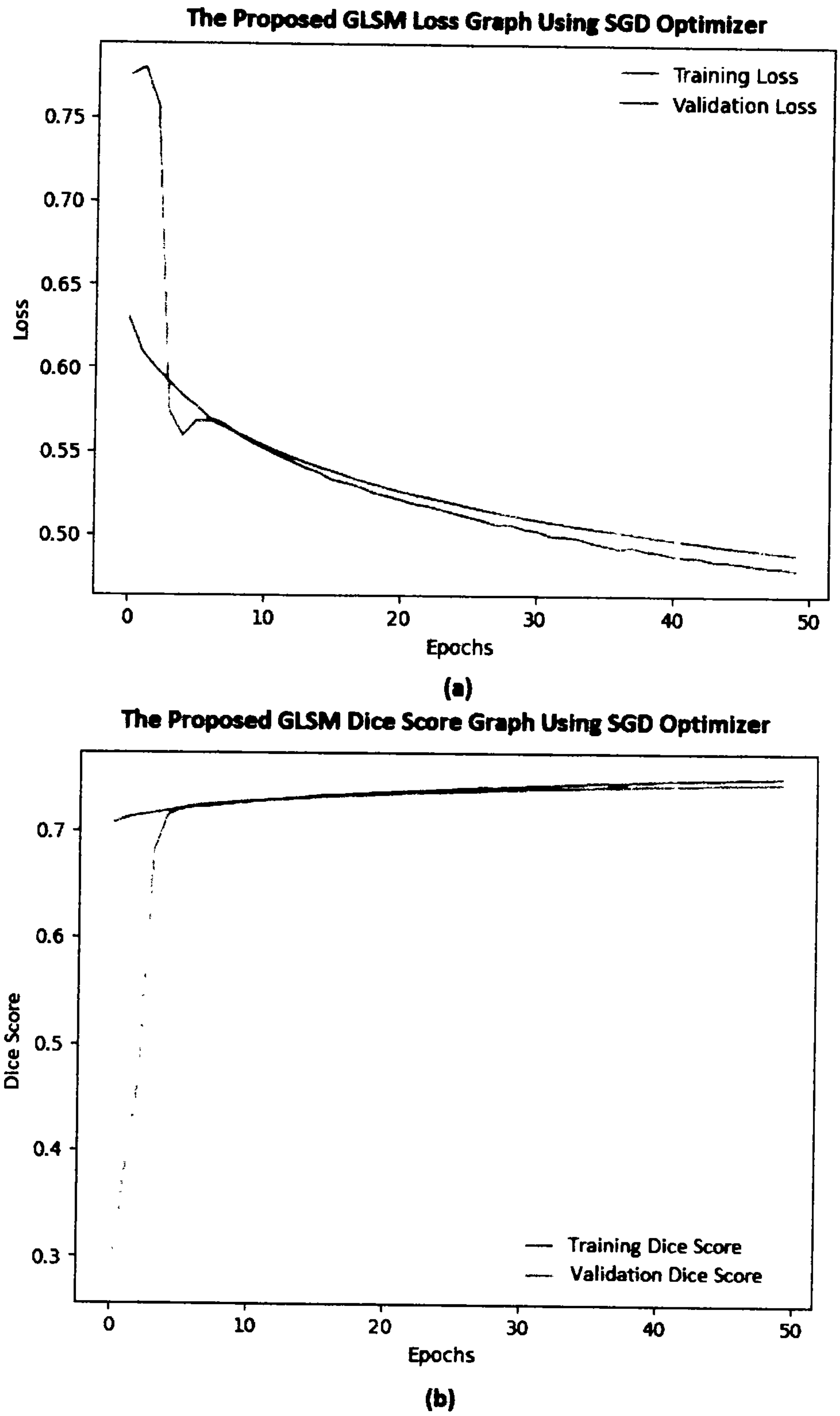


Figure 4.14: The Proposed *GLSM* Dice Score Comparison Graph on Test Set.

The second experiment is conducted using the *SGD* optimizer to train the proposed *GLSM* model. TThe model is trained with a loss of 62.98% and an accuracy of 70.67% in the first iteration of training, as shown in Figure 4.15. The validation dice score is just 28.02%, whereas the validation loss is significantly larger (77.56%). This result suggests that the model's early performance could be better. The model's effectiveness increases throughout training. The training and validation losses decrease, showing that the model improved its predictive abilities. Training loss is 47.89% by Epoch 50, whereas validation loss is 48.86%. Also, from Epoch 1 to Epoch 50, there is a rise from 70.67% to 75.03% in training dice score. At the end of the training, the validation dice score has similarly increased, though slower, to 74.45%. Notably, the model repeatedly underperforms on the validation set compared to the training set. It indicates the model may have been overfitting to the training data, which means it may have learned to recognize just the patterns in the training set. It becomes obvious throughout training when validation dice score both get up. In conclusion, the suggested *GLSM* model's dice score decrease throughout training while using the *SGD* optimizer.

The results of the proposed *GLSM* utilizing the *SGD* optimizer are shown in Table 4.5. The test images column shows the number of photos used to evaluate the model's performance. There are 119 sample photos here. The results show that the *GLSM* model trained with the *SGD* optimizer performed well on the test set, with a dice score of 73.79%, as depicts in Table 4.5 and Figure 4.14. The prediction of the proposed *GLSM* model using *SGD* is shown in Figure 4.16.



The Proposed GLSM Loss Graph Using SGD Optimizer

(a)

The Proposed GLSM Dice Score Graph Using SGD Optimizer

(b)

Figure 4.15: *GLSM* Model Dice Accuracy Graph Using *SGD*.

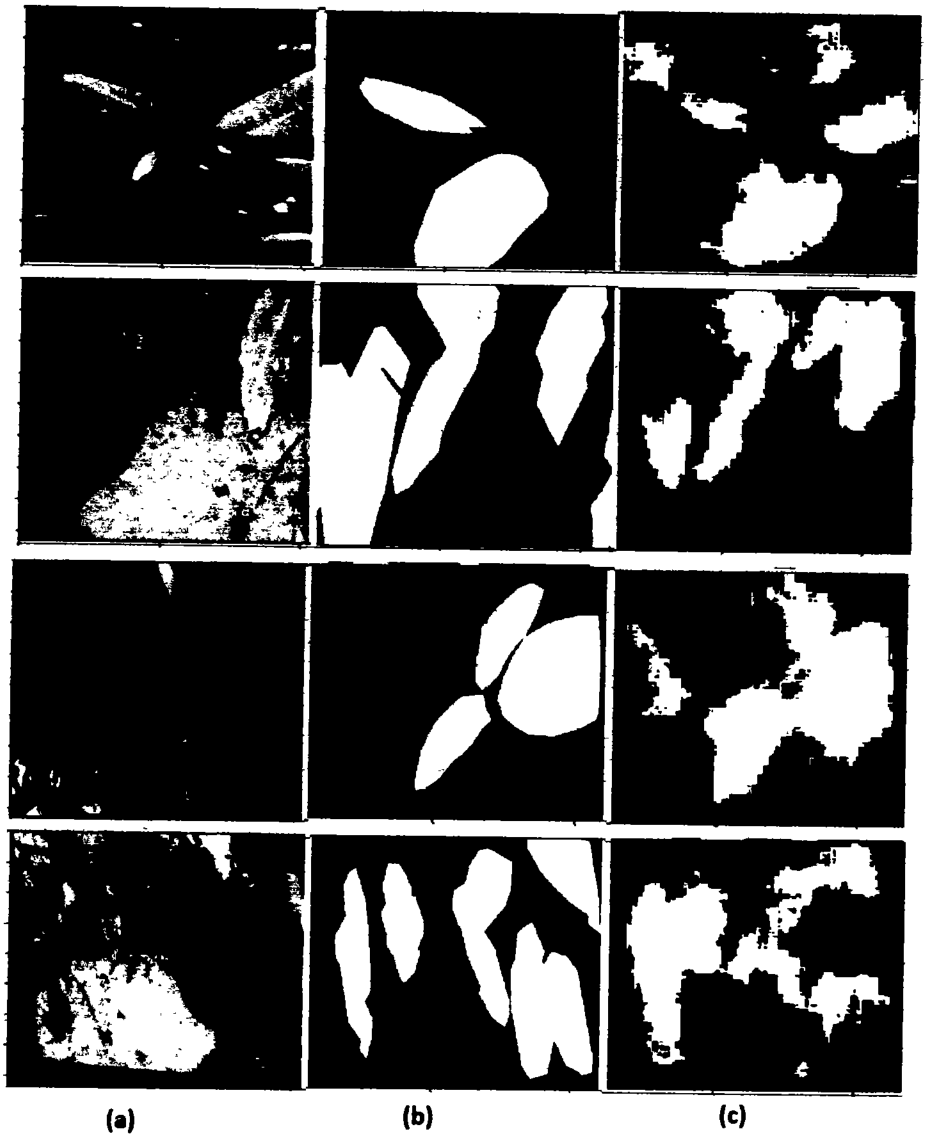


Figure 4.16: (a) Original Image (b) Generated Mask (c) Predicted Mask of the *GLSM* Using *SGD*.

The results show that the proposed *GLSM* model performs exceptionally well when using the *Adam* optimizer in the training process. The model's performance could be enhanced by employing regularization strategies (e.g., dropout, weight decay) or fine-tuning hyperparameters (e.g., learning rate, batch size). The model's generalization abilities can be better gauged if its performance is tracked over time on a distinct test set. Overfitting is obvious, though, as validation performance consistently lags behind training performance. It is recommended that additional optimization techniques and testing on a test set be undertaken to enhance the model's generalizability.

4.3.3 Proposed Guava Multiple Leaf Diseases Detection (*GMLDD*) Model Performance

The outcomes of the suggested approach on the training and validation sets are graphically presented in Figure 4.17. Figure 4.18 displays the outcomes of the recommended approach. In addition, to the effectiveness of detecting diseases under different illumination and complex backgrounds, the proposed *GMLDD* model's predictions have been shown in Figure 4.19. Using field-level data, the proposed *GMLDD* model has shown excellent disease prediction capability under different illumination and complex background.

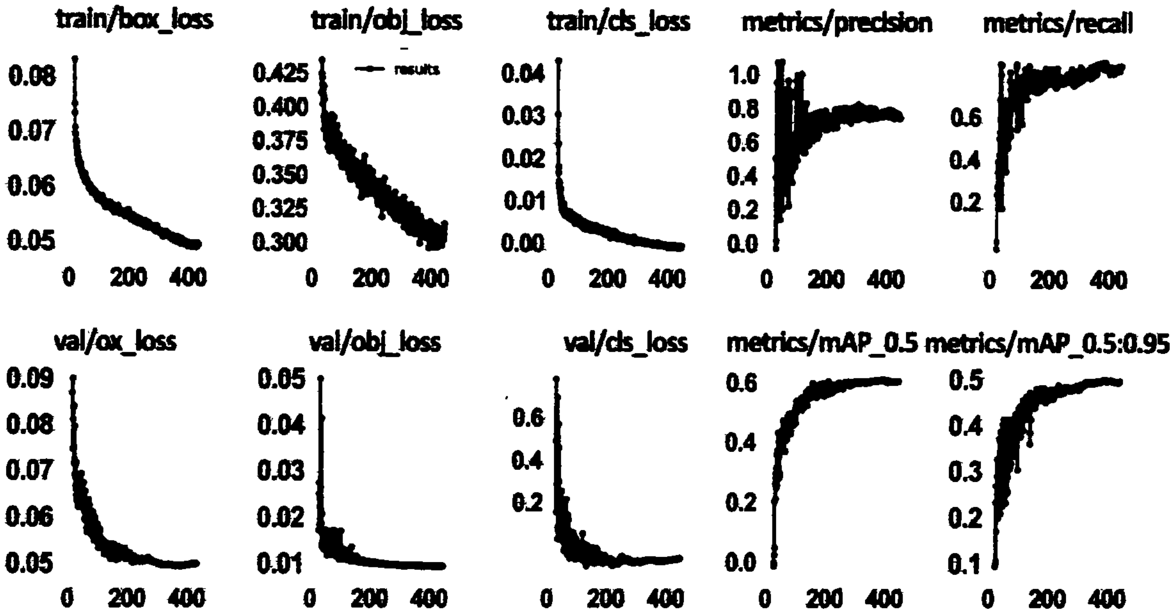


Figure 4.17: Training and Validation Graph of the GMLDD Technique.

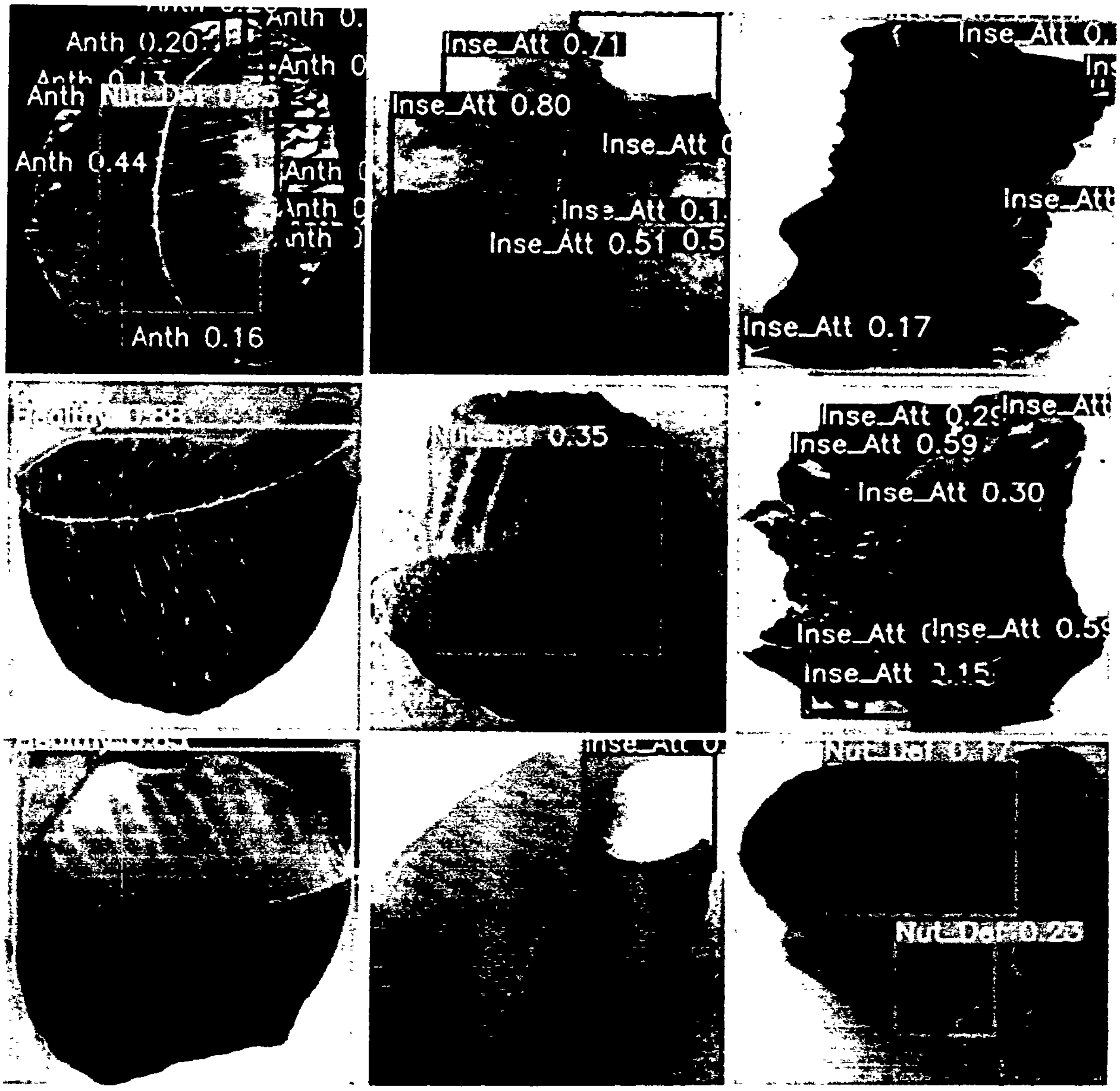


Figure 4.18: The Proposed GLMDD Model Predictions.

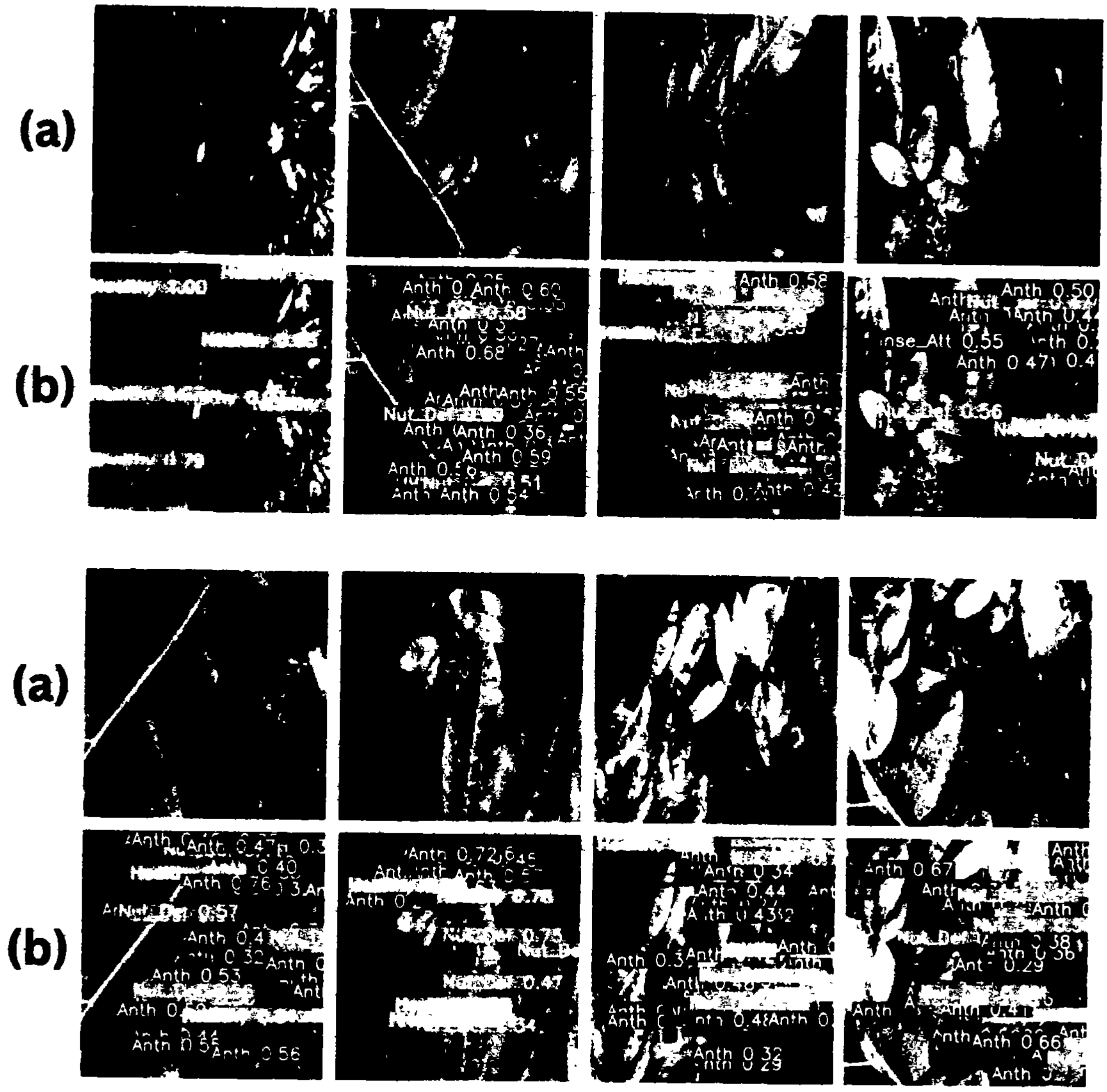


Figure 4.19: The Proposed GLMDD Model Field Level Predictions Under Different Illumination and Complex Backgrounds. (a) Original Images (b) Predictions.

Box loss, object loss, and categorization loss are all depicted in Figure 4.17. How well did the model pinpoint the object's location, and how much did its bounding box obscure it, as measured by the box loss? A loss of objectness was defined as quantifying the likelihood that an object existed inside a specific region of interest. It would be more objective if the image window usually contained a particular item. The classification loss of the procedure showed how well the model could identify an object's class [258]. After 55 epochs, there was a dramatic rise in recall, precision and *mAP*. Conversely, as depicted in Figure 4.17, box loss, objectness loss, and classification loss decreased significantly after 55 epochs.

An unknown data set (test set) containing five classes—anthracnose, healthy, insect assault, nutrient shortage, and wilt—was used to compute the inference of the suggested method. As can be illustrated in Table 4.6, the test dataset consisted of 98 images with 232 targets (labels) across all classes. The proposed model had a 45.4% precision, 54.7% recall, a 44.3% mean absolute percentage (0.5) and a 14.0% mean absolute percentage on 53 anthracnose-labeled samples. Out of 24 labels considered "healthy," 93% were accurate, 100% were remembered, 96% were mAP@0.5, and 92% were mAP@0.5:0.95 (Table 4.6). The insect attack used 67 targets and achieved 64.6% precision, 79% recall, 79% mAP@0.5, and 35.3% mAP@0.5:0.95 (Table 4.6). Nutritionally deficient students earned a 66.5% precision rate, 31.7% recall rate, 35.5% mAP@0.5, and 11.9% mAP@0.5:0.95 across 69 objectives (Table 4.6). We achieved 96.3% accuracy, 100% recall, 99.5% mean absolute precision at a significance level of 0.5, and 98.2% mean absolute precision at a significance level of 0.5:0.95 across 19 targets related to wilt disease (see Table 4.6). The outcomes of the suggested strategy in ten distinct categories are illustrated in Table 4.6. Overall, the approach did quite well, with percentages of 73.3% for precision, 73.13% for recall, 71.03% for mAP@0.5, and 50.3% for mAP@0.5:0.95.

Table 4.6: The proposed GMLDD model performance

Class	**Images**	**Targets**	**Precision**	**Recall**	**mAP@.5**	**mAP@.5:.95**
All	98	232	73.3	73.1	71.0	50.3
Anthracnose	98	53	45.4	54.7	44.3	14.0
Healthy	98	24	93.3	100	95.9	92.0
Insect Attack	98	67	64.6	79.1	79.8	35.3
Nutrient Def.	98	69	66.5	31.7	35.5	11.9
Wilt	98	19	96.3	100	99.5	98.2

Figure 4.20 depicts the confusion matrix for the suggested procedure. The proposed model achieved 70.8% correct and 29.2% wrong predictions for all categories (Figure 4.20). The anthracnose class achieved 53%, the healthy category attained 100%, the insect attack gained 75%, nutrient deficiency accomplished 26% correct predictions, and the wilt class predicted 100% accurate predictions.

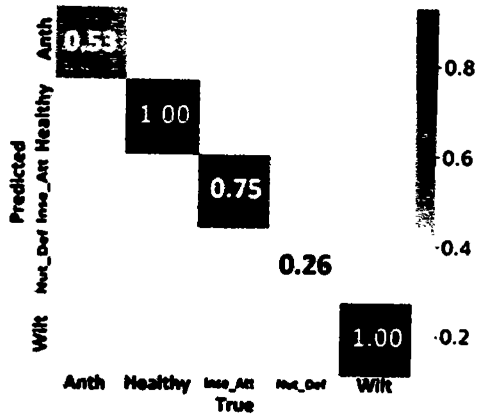


Figure 4.20: The Proposed GMLDD Model Confusion Matrix.

Figure 4.21 (a) depicts the p graph, where precision (y-axis) and confidence (x-axis) are displayed against various probability cutoffs. The p-curve demonstrates that for anthracnose disease, accuracy improved steadily from 0% at the lowest confidence level to 100% at the highest. At a confidence level of 0.05 to 78, the healthy class obtained a precision of less than 90%. The optimal level of accuracy for the healthy group was found to be 84.3%. At a confidence level of 0 to 55, the accuracy of insect class identification improved steadily, and at a confidence level of 79, accuracy approached 100% (Figure 4.21 (a)). Confidence in the nutritional deficit class rose from 0 to 50 with time. At a confidence level of 51 to 56, precision dropped precipitously to 50%, and from a confidence level of 57% onwards, it reached its utmost limit. The wilt class, on the other hand, achieved a precision of 95% or more and showed steady improvement, beginning at a confidence level of 0.03. The precision of all categories improved linearly at a confidence level below 84.3, peaking at a confidence level of 84.3. Hence, at a confidence level of 84.3, our model performed admirably in accuracy.

Figure 4.21 (b) exhibits Recall Curve (*R-Curve*) graph represented recall (y-axis) and confidence level (x-axis) on test data. The r-curve chart demonstrated that at a 0% confidence level, the proposed model had a recall of about 82% for the anthracnose class. It gradually dropped from a high of 58 at the lowest confidence level to zero at the highest. Under the proposed strategy, the recall was 100% at the 70 confidence level for the healthy type but rapidly dropped beyond that. The final result was a zero recall with an 89% certainty. Confidence levels ranged from 0 (highest) to 82 (lowest) for the insect attack class, with a maximum recall of 97% at the 0 confidence level. After a sharp drop from 1 to 51%, the recall rate for nutrient deficit began at 90% at a confidence level of 0. When it came to recalling wilt classes, however, there was a distinct plateau at a confidence level of 83, followed by a precipitous drop to zero at a confidence level greater than 88. At a 0 confidence level, the suggested method had a recall of 97% across all classes, and this recall gradually dropped to around 0% at a 88 confidence level (Figure 4.21 (b)).

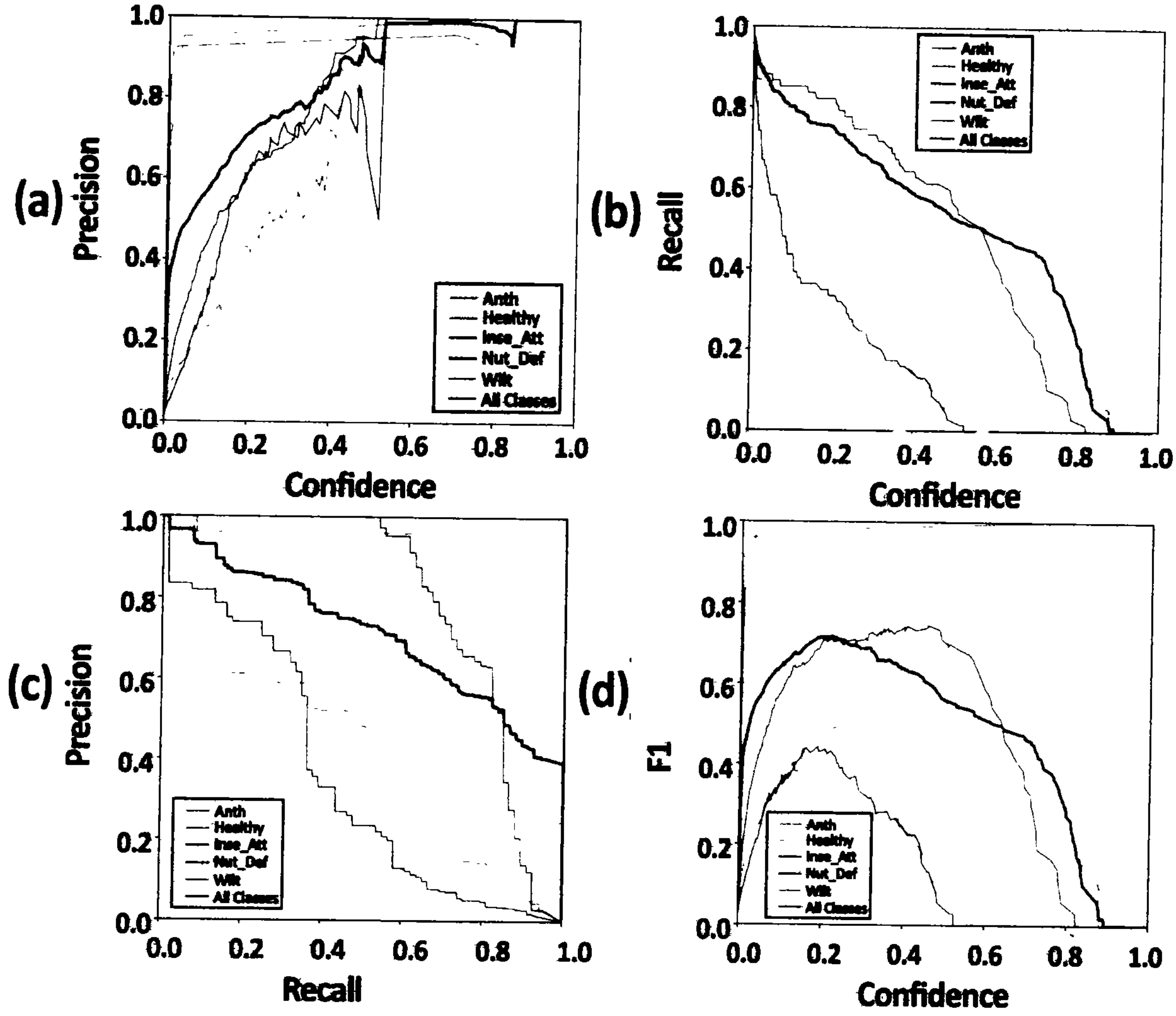


Figure 4.21: (a) Precision Curve (P Curve) (b) Recall Curve (R Curve) (c) Precision Recall Curve (PR Curve) (d) F1 Curve.

When plotted against different thresholds, the Precision-Recall Curve (*PR-Curve*) showed where accuracy and recall were optimal. If the area under the curve was significant, recall and precision were high, and the proportion of false positives was small. Our model's large area under the curve expression in the findings indicates that the proposed model performed well. Figure 4.21 (c) shows that the anthracnose, healthy, insect attack, nutrient deficit, and wilt classes all reached 71.0

mAP@0.5, whereas the other categories only reached 44.3, 95.9, 79.8, 35.5, and 99.5, respectively.

In Figure 4.21 (d), the f1 curve, also known as the f score, quantifies the accuracy of a model on a given dataset. The f score for this approach was calculated by taking the average of its recall and precision values. The model's accuracy and reliability were combined. The greater the f1 score, with 0 being the worst and 100 being the best accuracy was. The p-curve chart for anthrax disease demonstrates that the most incredible possible accuracy was achieved at a confidence level of less than 0.53. For the healthy group, the accuracies ranged from less than 90% (0.05 to 78%). The highest level of accuracy for the healthy class was obtained with a confidence level of 84.3. From a confidence level of 0 to 55, accuracy in classifying insects improved steadily, and accuracy of 100% was achieved at a confidence level of 79 (Figure 4.21 (d)). From a confidence level of 0 to 50, the nutrient deficiency class steadily improved accuracy. From a confidence level of 51% to 56%, precision rapidly dropped to 50%, reaching a maximum of 57% from that point on. The wilt class, on the other hand, achieved less than 95% accuracy and steadily grew from a confidence level of 0.03 onwards. Maximum accuracy was attained across the board with a degree of confidence of 84.3, and accuracy improved as confidence decreased. As a result, the accuracy of our model was satisfactory at the 84.3% confidence level. The findings (shown in Figure 4.21 (d)) indicated that the proposed model obtained the highest f1 score at the 22.2 confidence level.

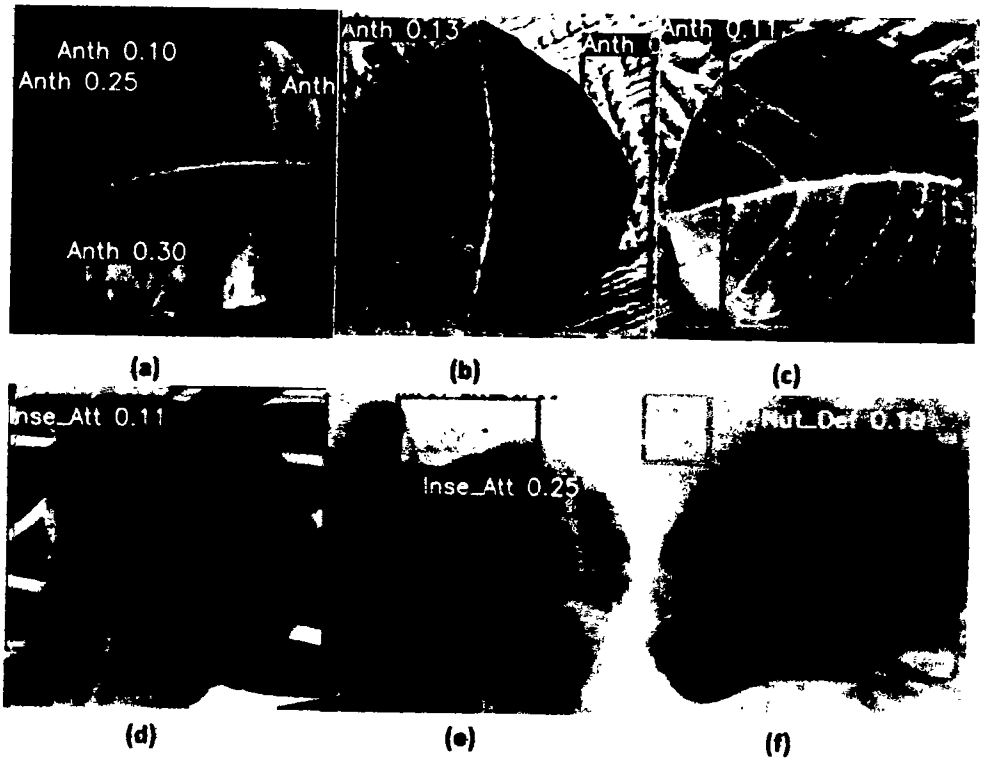
The results showed that the healthy and wilt groups achieved over 93% precision, recall, and mAP@0.5. However, among all the categories, anthracnose disease had the lowest precision, recall, and mAP@0.5 (less than 46% precision). On the other hand, the insect attack and nutrient deficiency class achieved more than 64% precision, recall and mAP@0.5.

The proposed model for detecting multiple-leaf illnesses in guava needs specific tweaks, which we provide below:

- The five groups represented in the *GLDD* are extremely unbalanced. There is a severe lack of data for the affected group. By applying the balanced dataset classes [256, 257], we can boost the suggested model's efficiency.
- The effectiveness of a dataset may be enhanced by increasing its size. The deep learning methods require massive data samples for training; otherwise, overfitting occurs. Therefore, we can resolve the problem of overfitting by enhancing the dataset [256, 257].
- The complex background is another reason for misclassification. In the dataset, we used complex backgrounds, such as background colour, resembling leaf diseases, as shown in Figure 4.22. For example, brown background misleads to the anthracnose, as shown in Fig-

ure 4.22 (a-c), and the disease light pink deceives the nutrient deficiency class, as depicted in Figure 4.22 (f). Using a transparent or plain background and avoiding complex backgrounds can boost the effectiveness of the proposed strategy.

- Taking off the curved or folded leaves from the training is another option to boost the proposed method's efficiency. The curly or folded leaf can mislead the insect eaten class, as shown in Figure 4.22 (d,e). We can enhance the proposed method's efficiency by improving the curly or folded leaf training.
- The small object sizes and annotation scheme are another reason for misclassification. In YOLOv5, only a rectangle is used to annotate the object. In guava leaf diseases, leaves are not in the proper shapes from the edges. And when we use a rectangle to annotate the object or disease area, the unnecessary information or area is also included, as shown in Figure 4.22 (a-f). This irrelevant inclusion ultimately misleads the classification. In order to make the suggested technique more effective, we can eliminate the extraneous data that could confuse the annotation process and throw off the categorization [256].



(a) (b) (c)

(d) (e) (f)

Figure 4.22: The Proposed *GMLDD* Model Misclassification Examples.

4.3.4 Comparison of Proposed *GMLDD* Model with Other *YOLO* Variants

There currently needs to be literature on single-leaf detection of various guava leaf diseases, as far as we know. For this reason, we cannot directly compare our approach to the current standard. On the *GLDD* dataset, however, we evaluate how well the suggested method stacks up against other *YOLO* variations. There have been many distinct versions of YOLO produced throughout time, each with its own set of improvements and modifications for certain use cases; these versions include YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv4 Tiny, YOLOv5, YOLOX, and so on. However, the community has developed other enhancements and modifications to the YOLO framework. Which YOLO variation is best depends on the task, the available hardware, and the tradeoffs we are willing to make between speed and accuracy. The overall performances of the suggested detection method are compared in Table 4.7 to those of YOLOv3 [258], YOLOv4 [259], and YOLOv4 Tiny [260]. The recommended model outperformed the other models in terms of precision (73.3%), recall (73.1%), and mAP0.5 (71%). When the average detection times of these three techniques were evaluated, it was found that YOLOv4 detected events on average 45.301 milliseconds faster than the other two models. That is why it is clear that the suggested detection method is the best option for pinpointing boundaries among the four methods considered. The proposed model's detection time was faster than the YOLOv3, and YOLOv4 Tiny models, which had a detection time of 46.360 ms. Despite this, it can detect high-resolution images in real-time with greater performance and speed than the other two existing models.

Table 4.7: The Proposed GMLDD comparison with YOLOv3, YOLOv4, and YOLOv4-Tiny

Model	Precision (%)	Recall (%)	mAP@0.5	Time (ms)
YOLOv3	66.2	65.0	65.1	48.205
YOLOv4	70.1	69.5	69.8	45.301
YOLOv4 Tiny	-	-	69.7	50.402
Proposed Model	73.3	73.1	71.0	46.360

4.4 Chapter Summary

No study was presented in the literature to detect the infected patches from the guava plant, and it is mainly for the unavailability of the dataset. For this purpose, we developed a Guava Patches Dataset (GPD) from Okara, Central Punjab, Pakistan. The dataset consisted of healthy and infected patches. Data augmentation techniques were used to increase the dataset's size to a reasonable level. Different segmentation approaches were employed to highlight the region of interest to learn from the data successfully. Several segmentation techniques, such as UNET and MobileNetV2-UNET, were presented in the literature. However, these techniques posed several issues, such as high computational costs and limited accuracy. MobileNetV2 is a lightweight convolutional neural network employed in synchronous functions for two reasons: first, it has fewer trainable parameters than classic convolution; second, it has fewer training parameters and lower computing costs. A segmentation method called Guava Infected Patches Modified MobileNetV2 and UNET (GIP-MU-NET) was created to address these problems. The proposed GIP-MU-NET consisted of modified MobileNetV2 as an encoder, and UNET was used as a decoder. The reason for using the modified MobileNetV2 as an encoder part was to reduce the computational cost. The computational cost was further reduced by eliminating the block IDs 6, 9, 10, 13, and 16. The filters were also decreased from 160 to 128, 320 to 160, and 1280 to 256.

To our knowledge, no study intended to segment guava leaves was ever documented in the literature. Therefore, a novel Guava Leaf Segmentation Model (GLSM) was developed from a U-Net-like model to segment guava leaves from the images of healthy and infected patches. The proposed GLSM model accurately segmented the healthy and diseased leaves from the infected and healthy patches. The encoder comprised two components: the entry block and the residual block. Conv2D, batch normalization, and ReLu activation functions were used in the encoder. The residual block consisted of ReLu, SepereableConv2D, and batch normalization. Three residual blocks were employed in the encoder. The decoder used inverted residual blocks containing ReLu, Conv2Dtranspose, and batch normalization.

The most critical task was detecting and classifying multiple guava diseases on a single leaf, as no research could detect multiple diseases on a single guava leaf. The main hurdle was the unavailability of the dataset. We developed the first-ever Guava Leaf Diseases Dataset (GLDD) to address this issue to detect various diseases on a single leaf. The dataset comprised five classes: anthracnose, nutrient deficiency, wilt diseases, insect assault, and healthy. It was developed in the district of Okara, Central Punjab, Pakistan. Then, a real-time Guava Multiple Leaf Disease Detection (GMLDD) model based on YOLOv5 was devised to detect and localize multiple diseases from a single guava leaf. The YOLOv5 model was known for its efficient capability of real-time detection of objects. The YOLO algorithm used convolutional neural network (CNN) models to detect objects in an image. The proposed model was totally automated compared to the most advanced methods currently used to detect leaf diseases. In order to detect leaf disorders in plants in real-time, the proposed approach could be used. The ability to detect several illnesses on a single leaf in its early stages, before they cause severe damage and economic hardship, is a massive boon to farmers.

Chapter 5

Multi-Level Deep Learning Model for Potato Leaf Disease Recognition

5.1 Overview

Deep learning approaches already exist in the literature and are misfits for diagnosing regional potato leaf diseases in Pakistan. Varieties of potatoes, climate change, disease symptoms, and other external factors are significant to incorrect identification. So, the existing techniques failed to achieve high accuracy for the other areas, such as Pakistan. Since the dataset was small, to begin with, and any model's latest version can be certified as good if tested on unseen data, most approaches did not assess their efficacy on unseen images. The low convergence speed due to the vast number of trainable parameters is also the main hurdle, and accuracy needs to be improved. The last problem in the literature is the non- availability of the potato leaf segmentation technique. Therefore, there is a dire need to develop a new dataset to detect the Pakistani region potato leaves' diseases so farmers in Pakistan can determine the diseases of potatoes in their early stage, enhance their income, and boost the country's economy. This research is conducted to resolve the above research gaps.

This part of the thesis investigates a multi-stage deep learning model for potato leaf disease detection. Initially, the picture of the potato plant is segmented using the *YOLOv*5 method to remove the leaves. We created a unique Potato Leaf Disease Detection Convolutional Neural Network (*PDD-CNN*) for the second layer to classify late and early blight in potato leaves. The recommended *PDDCNN* was then evaluated using the Potato Leaf Dataset (*PLD*). In order to compile the *PLD* dataset, researchers in the Central Punjab area of Pakistan took photographs of potato leaves from several places. We employ a cropping and labeling tool to help plant pathologists crop and identify the images. The most significant contributions include:

The following are the main contributions:

- A method based on *YOLO*v5 has been developed for real-time potato leaf segmentation and extraction, which may be used to remove the leaves from potato images.
- For late and early blight recognition in potato leaves, a novel deep learning system known as Potato Leaf Disease Detection Convolutional Neural Network (*PDDCNN*) has been created.
- The proposed method uses the minimum possible number of parameters compared to state-of-the-art models.
- The development of a Potato Leaf Dataset (*PLD*) from the Central Punjab region of Pakistan by capturing three types of potato leaf images: early, late blight diseases and healthy.

5.2 Proposed Methodology

Many problems exist in the literature using deep learning approaches, including incorrect identification of potato leaf diseases, variation in potato diseases, varieties and environmental factors. The existing systems have a high false rate to recognize potato diseases in the Pakistani region. The existing potato leaves dis- ease datasets contain inadequate training samples with imbalanced class samples. Another issue is that current approaches are inaccurate and have a slow convergence time because of the huge number of trainable parameters. In this research, we propose classifying diseases affecting potato leaves using a multi-layered deep learning network. As a first step, it segments the image of the potato plant using the YOLOv5 algorithm to get rid of the leaves. A novel Potato Leaf Disease Detection Convolutional Neural Network (*PDDCNN*) was built at the second layer to recognize late and early blight in potato plants from photos of their leaves. The proposed method is represented by the flow diagram in Figure 5.1 and the overall design in Figure 5.2.

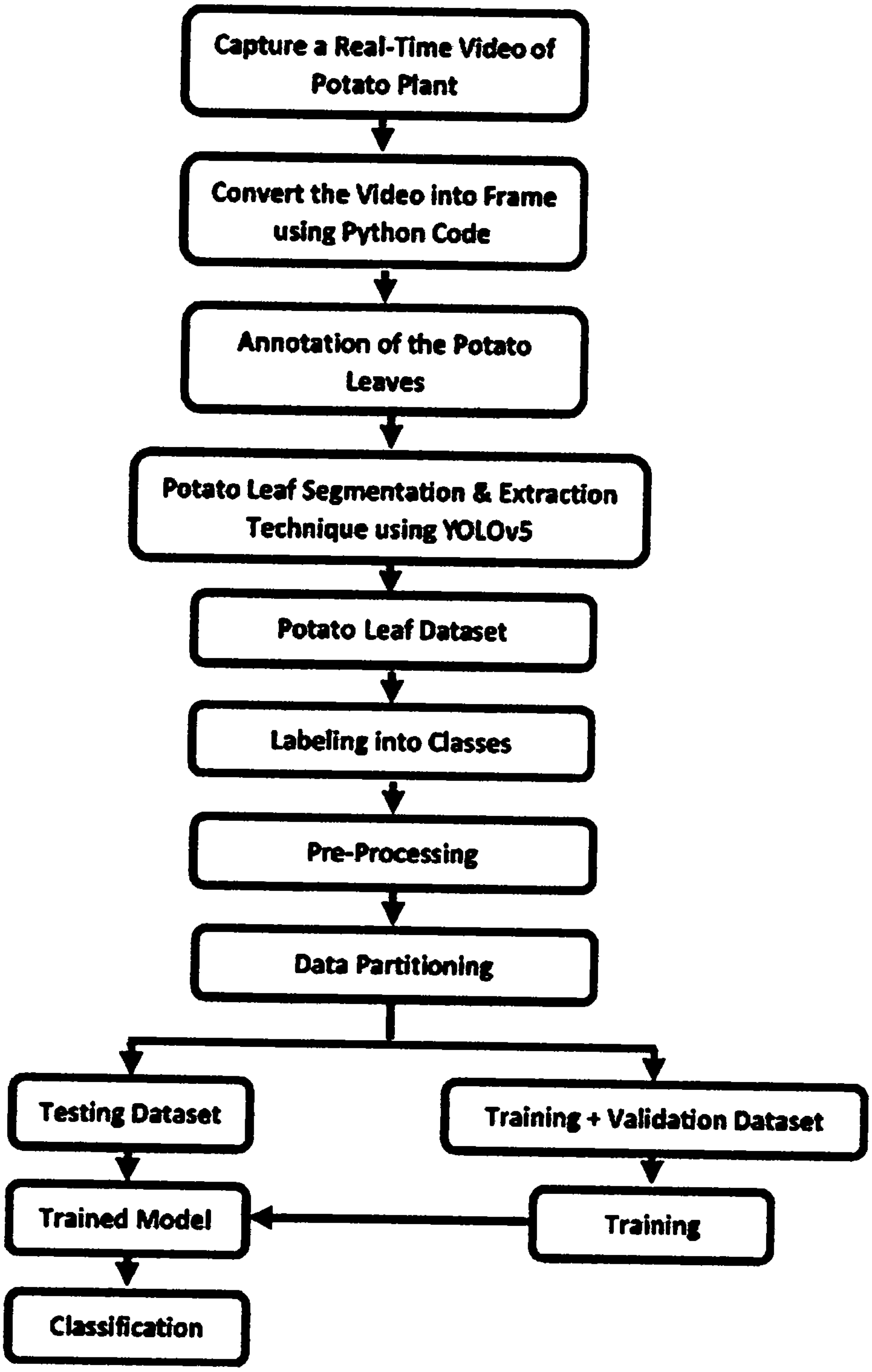

Capture a Real-Time Video of Potato Plant
Convert the Video into Frame using Python Code
Annotation of the Potato Leaves
Potato Leaf Segmentation & Extraction Technique using YOLOv5
Potato Leaf Dataset
Labeling into Classes
Pre-Processing
Data Partitioning
Testing Dataset
Training + Validation Dataset
Trained Model
Training
Classification

Figure 5.1: The System Flow Diagram of the Proposed *PDDCNN* Technique.

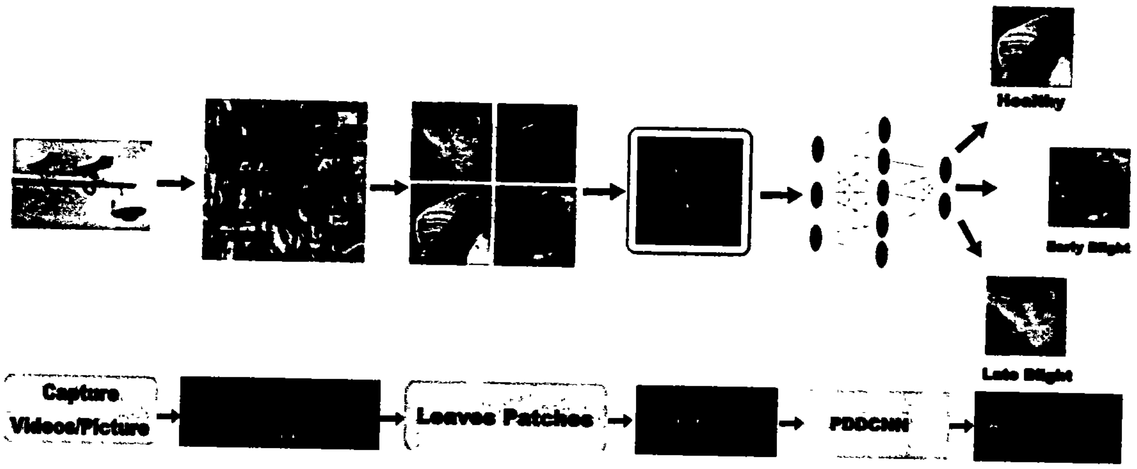


Figure 5.2: The Block Diagram of the Proposed *PDDCNN* Technique.

5.2.1 Dataset

The use of a high-quality dataset is critical to the efficiency of a *DL* model. This study makes use of the following data sets.

5.2.1.1 PlantVillage Dataset

The details of the PlantVillage dataset have already been described in section 2.4.1.3. According to Table 5.1, the following quantities of leaves were chosen for the experiments: images of 1000 leaves with late blight, 1000 leaves with early blight, and 152 healthy leaves.

Table 5.1: Summary of the PlantVillage Dataset.

Class Labels	Samples
Early Blight	1000
Late Blight	1000
Healthy	152
Total Samples	**2152**

5.2.1.2 Potato Leaf Dataset (*PLD*)

For potato leaf diseases, only the PlantVillage dataset is currently available for public use; hence this is the only dataset used to construct models in the literature. All the researchers used the PlantVillage dataset in their research, but there are many research gaps found in the literature. The

PlantVillage dataset has been developed from a specific region under certain geographic and environmental factors. Variations in form, genetic diversity, and local climate contribute to regional differences in potato leaf diseases. Therefore, the existing systems have a high false rate to recognize potato disease detection in the Pakistani region potato leaf images, as shown in Table 5.2. The PlantVillage dataset also has fewer images and an imbalanced class distribution. Therefore, there is a dire need to develop a new potato leaves dataset collecting from the Pakistani areas. Scientists may use this data to fine-tune their algorithms for spotting diseases in potato leaves, which will benefit farmers in Pakistan by allowing them to identify problems early.

Table 5.2: Classification Accuracies of the Proposed PDDCNN Model Training on PlantVillage and Testing on the PLD Dataset.

Training Dataset	Testing Dataset	Early Blight Accuracy	Healthy Accuracy	Late Blight Accuracy	Total Accuracy	Overall Testing Accuracy
PlantVillage	PLD	95.71%	08.82%	23.94%	807	48.89%

Thus, a new Potato Leaf Dataset (*PLD*) has been developed from Pakistan's Central Punjab region. The dataset capturing details has already been described in section 3.2.1.1. Potatoes were grown in rows and segregated at a distance of 3 feet apart from each other. The seeds of the plants were grown by digging a 6 to 8–inch deep, 5–inch wide hole in the soil. After placing seeds in a pit hole that had been prepared, the soil was amended with manure and afterward watered with canal water. Using LabelMe, we exported YOLOv5 PyTorch and XML annotations of both diseased and healthy leaves. The YOLOv5s method is trained entirely from scratch for segmentation and leaf extraction. The Python algorithm used to extract the leaves was trained with the results and annotations from the YOLOv5s model. The PLD dataset includes 4062 images of diseased and healthy potato leaves, chosen with the assistance of plant pathologists. Plant pathologists then classified the pictures into early blight, late blight, and normal leaves. The plant leaf dataset captured 1628 photos of early blight leaves, 1414 photos with late blight leaves, and 1020 images with healthy potato leaves, as shown in Table 5.3. Figure 5.3 displays a selection of photos from the PLD dataset. The PLD dataset can be accessed from [261].

Table 5.3: Summary of PLD Dataset.

Class Labels	Samples
Early Blight	1628
Late Blight	1414
Healthy	1020
Total Samples	**4062**

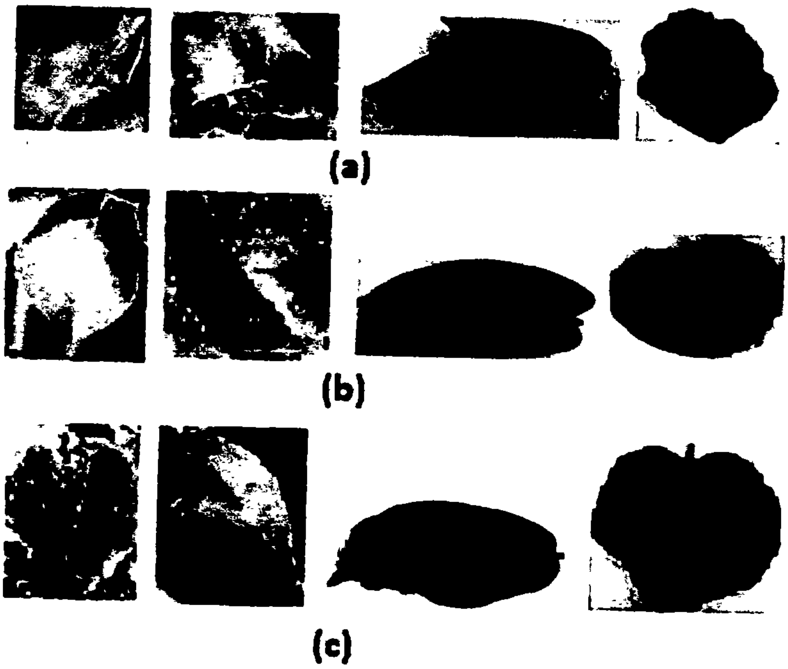


Figure 5.3: Examples of Potato Leaf Images: (a) Early Blight, (b) Healthy and (c) Late Blight.

5.2.2 Image Pre-Processing

Pre-processing was used on the final *PLD* pictures to provide more consistent classification results and improved feature extraction. The *CNN* approach required extensive retraining, and a large image dataset was necessary to reduce the likelihood of overfitting.

5.2.2.1 Data Augmentation

Several data augmentation methods were performed to the training set to combat overfitting and enlarge the variety of the dataset using the Image Data Generator method of the Keras library in Python. The computational cost was decreased using scale transformation while maintaining the same range and smaller pixel values. Therefore, every pixel value was ranged from 0 to 1 using the parameter value (1./255). Images were rotated to a specific angle using the rotation transformation; therefore, 25° was employed to rotate the images. Images can shift randomly either towards the right or left by using the width shift range transformation; selected a 0.1 value of the width shift parameter. Training images moved vertically using the height shift range parameter with a 0.1 range value. In the shear transformation, one picture axis is held constant while an angle stretches the other, called the shear angle; in this case, the shear angle was set at 0.2. The zoom range argument was utilized to realize the random zoom transformation; >1.0 means magnifying the images, and <1.0 was utilized to zoom out the photo; therefore, 0.2 zoom_range was employed to magnify the image. Flip was applied to flip the image horizontally. We also included a brightness adjustment (where 0.0 means no brightness and 1.0 signifies maximum brightness), so we chose a zoom range of 0.5-1.0. In order to implement this transformation, we randomly shifted the channel values within a 0.05-point range using the fill mode closest.

5.2.3 Dataset Split

The *PLD* dataset had three distinct sections: training, validation, and testing. We utilized the training set to teach the *PDDCNN* method and the validation and test sets to test its efficacy. As a result, we used an 80-10-10% split for our training, validation, and testing sets. A total of 3257 photos were utilized for training, validation, and testing on the *PLD* database.

The *CNN* model was trained by passing training samples from the input layer to the output layer, where predictions were made and where the causes of any discrepancies or failures were discovered. When a forecast went wrong, back-propagation was used to correct the problem. This study used the back-propagation algorithm to fine-tune the model's parameters for more accurate predictions. It was considered one "epoch" once the forward and backward propagation processes were finished. The Adam optimization algorithm was employed in the model. The current investigation maintained the 80% image ratio by using training photos labeled as early blight, healthy, and late blight. Only 20% of the original photos were used, and the remaining 80% were divided equally (10% each for validation and testing) across the two datasets. As such, the proposed *PDDCNN* model was trained using a dataset of labeled training images to predict the labels for those images.

5.2.4 Potato Leaf Segmentation and Extraction Technique Using YOLOv5

Since we have already explained the proposed method of using YOLOv5 to segment and extract information from potato leaves in section 2.6, we will merely present the model's architecture here. The YOLOv5s framework employed in this study is shown in Figure 5.4. All the specifics of the suggested model's configuration are laid forth in the experiments.

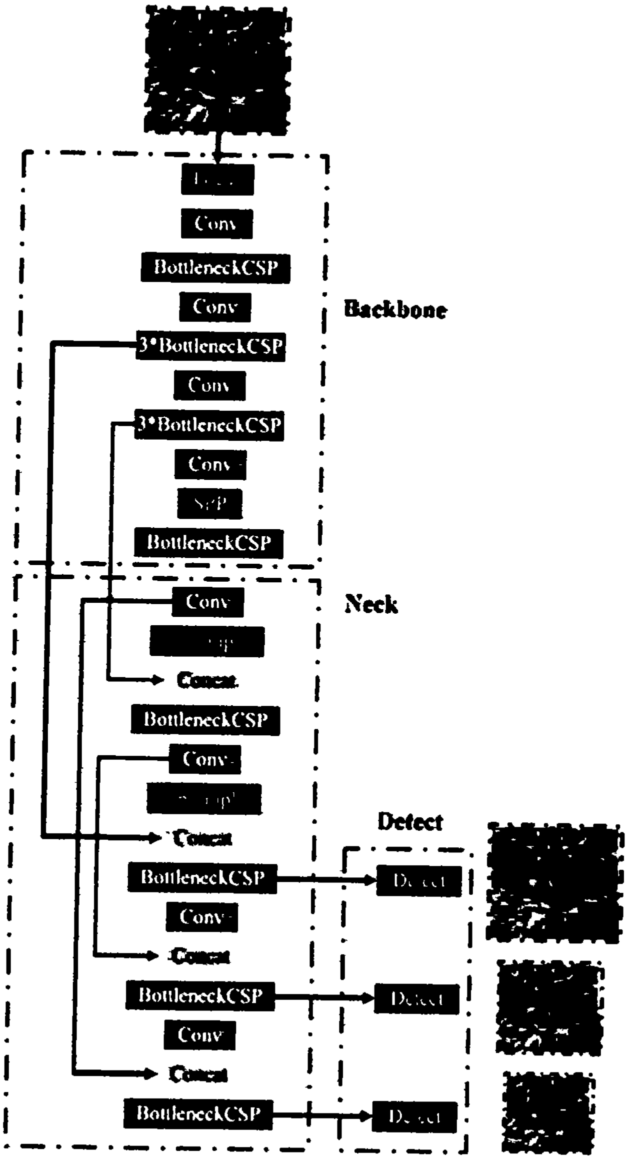


Figure 5.4: The Proposed Potato Leaf Segmentation and Extraction Technique Using YOLOv5 Model Architecture.

5.2.5 Potato Leaf Disease Detection Convolutional Neural Network (*PDD-CNN*)

The proposed *PDDCNN* is based on the *CNN*, it has already discussed in section 2.5. The design of the suggested *PDDCNN*, which was used to distinguish between diseased and healthy potato leaves (see Figure 5.5), was displayed.

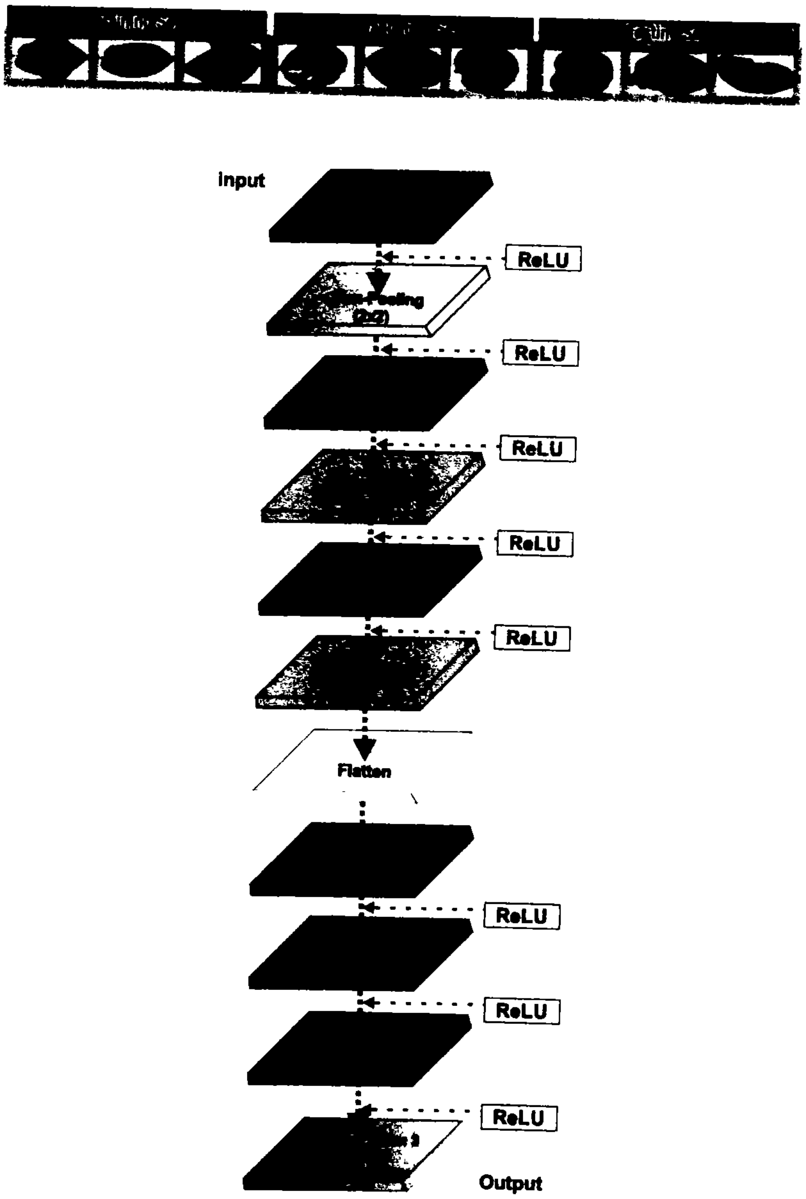


Figure 5.5: The Architecture of the Proposed PDDCNN Model.

The *PDDCNN* was used the following building blocks:

1. To prepare for future analysis, the sequential model extracted critical features from the input image using a series of layers.
2. The first convolution layer has a 256 × 256 × 3 input image structure, 16 filters, a 3 × 3 kernel, 1 × 1 padding, and ReLU activation.
3. After the initial convolutional layer, the image size was decreased by employing a max-pooling layer with a pool size of (2,2).
4. The second convolutional layer used 32 filters with kernel size 3 × 3, stride value 1 and the ReLU nonlinear activation function.
5. After the second convolutional layer, max-pooling was applied with pool size (2,2).
6. Regarding the final convolutional layer, 64 filters with a 3 × 3 kernel size were chosen. The padding and activation function *ReLu* was used once more for a stride of 1.
7. Then, the flatten layer was used to turn the convolved matrix into a vector in a single dimension.
8. The generated features were used for classification or decision making at one of the four concealed or fully linked layers.
9. The first fully connected layer (sometimes called a dense layer or a buried layer) was used with 512 neurons, followed by ReLU activation functions.
10. The second fully connected layer was used with 256 neurons, followed by ReLU activation functions.
11. To create the third hidden layer, 128 neurons and the ReLu activation function were employed.
12. The number of classes always determines the neuronal output. Since this study involved multiple classes, the softmax activation function was implemented on a layer of three neurons serving as the final hidden or output layer.
13. Predictions made in the model's output layer class label evaluation layer were used to assess the model's overall accuracy.

The configuration details and various parameters of the proposed *PDDCNN* are given in Table 5.4.

Table 5.4: The Proposed *PDDCNN* Model Configuration Details of Various Parameters.

Convolution Layers	3 (with 3 × 3 filters/kernels each)
Max-Pooling Layers	3 (with (2,2) pool size each)
Hidden Layer Neurons	512 (1st), 256 (2nd), 128 (3rd)
Output Layer Activation Function	Softmax
Batch size	32
Epochs	100
Training Optimizer	Adam
Loss Function	Categorial Cross Entropy

5.2.6 Experimental Setup

Experiments with the proposed *PDDCNN* models were run with identical settings, including an Adam optimizer, a Categorical Cross Entropy loss function, a 32-batch size, 100 epochs with data augmentation techniques used, and 20 epochs with no data augmentation. The models were trained and evaluated using a Google Colab Pro account equipped with high-powered *GPU* that did not require special settings. TensorFlow [262], the Keras open-source libraries, and the Python programming language were utilized in the experimental implementation of the proposed *PDDCNN* method. The best validation accuracy was kept, and an early termination criterion of 0.0001 was utilized for learning. On the other hand, a YOLOv5s model was trained using 100 epochs, 416 × 416 images, and a batch size of 32 for the Potato Leaf Segmentation and Extraction Method. The hyperparameters were left at their default values. First, the output of the trained model is stored into a YOLOv5 format file, then this text file and annotation stored in files were stored in a CSV file. Then using the python code, the annotations of the leaves were cropped and stored in a folder in jpg image format. The evaluation measures used in this chapter are described in section 3.2.7.

5.3 Results and Discussion

This section discusses the proposed *PDDCNN* model' performance in detail on *PLD* and PlantVillage datasets using and without data augmentation techniques.

The results of the proposed *PDDCNN* model focused on:

1. Identifying the stages of blight on potato leaves and classifying them as early, late, or healthy.
2. Test the proposed *PDDCNN* model on the PLD dataset with and without data augmentation methods for the training set.

3. Analyzing how well the proposed *PDDCNN* model does with and without data augmentation on the publicly available dataset PlantVillage.
4. On the cross-dataset, we quantified the efficiency of the suggested model.
5. In order to make a fair comparison to other top-tier networks, such as VGG16 [224], InceptionResNetV2 [263], DenseNet_121 [28], DenseNet169 [28] and Xception [264], using the transfer learning.
6. In order to compare the efficacy of deep learning for disease detection in potato leaves to that of previous studies.

5.3.1 Proposed *PDDCNN* Model Performance on *PLD* Dataset

To test the efficacy of the proposed *PDDCNN* model, two groups of experiments were carried out. To begin, we conducted a series of experiments in which we added new data to the training set of the *PLD* dataset using one of four sets of data augmentation methods. The second trial involved training without the help of any data augmentation methods. Table 5.5 displays the results of comparing four groups utilizing the parameters described in Section 5.2.2.1 to supplement the *PLD* dataset. Set #1 utilized a single data augmentation method and got 97.56% accuracy; set #2 used a pair of methods and got 98.28%. Set # 3 achieved 99.02% accuracy with five data augmentation techniques and set # 4 had 99.75% accuracy using seven data augmentation techniques. It confirmed that as we increased the training samples using more data augmentation techniques, the accuracy also increased. The set # 4 achieved the highest accuracy because we increased the training samples using the seven data augmentation techniques. The results showed that the *PDDCNN* required a vast amount of training samples for training.

In Figure 5.6, we can see the outcomes of set #4's data augmentation strategies. Based on the experimental results, the suggested approach successfully identified early blight 99.38% of the time, healthy plants 100% of the time, and late blight 100% of the time. The average accuracy on the PLD dataset was 99.75%, as shown in Figure 5.6. Figure 5.7 depicts the training, validation accuracy, and losses in each epoch for the proposed *PDDCNN* approach on the *PLD* dataset with data augmentation approaches applied to the training set. It was discovered that by applying data augmentation techniques to the training set, the suggested method produced very high identification rates on the *PLD* dataset.

Table 5.5: Classification Accuracies on Different Sets of Data Augmentations of the Proposed *PDDCNN* Model on the *PLD* Dataset.

Set #	Data Augmentation Used	Early Blight	Healthy	Late Blight	Average
1	rotation_range	96.32%	97.06%	96.48%	97.56%
2	rotation_range, width_shift_range, height_shift_range	98.77%	98.04%	97.89%	98.28%
3	width_shift_range, height_shift_range, shear_range, zoom_range, horizontal_flip	99.39%	99.02%	98.59%	99.02%
4	rotation_range, width_shift_range, height_shift_range, shear_range, zoom_range, horizontal_flip, brightness_range, channel_shift_range, fill_mode = nearest	99.38%	100%	100%	99.75%

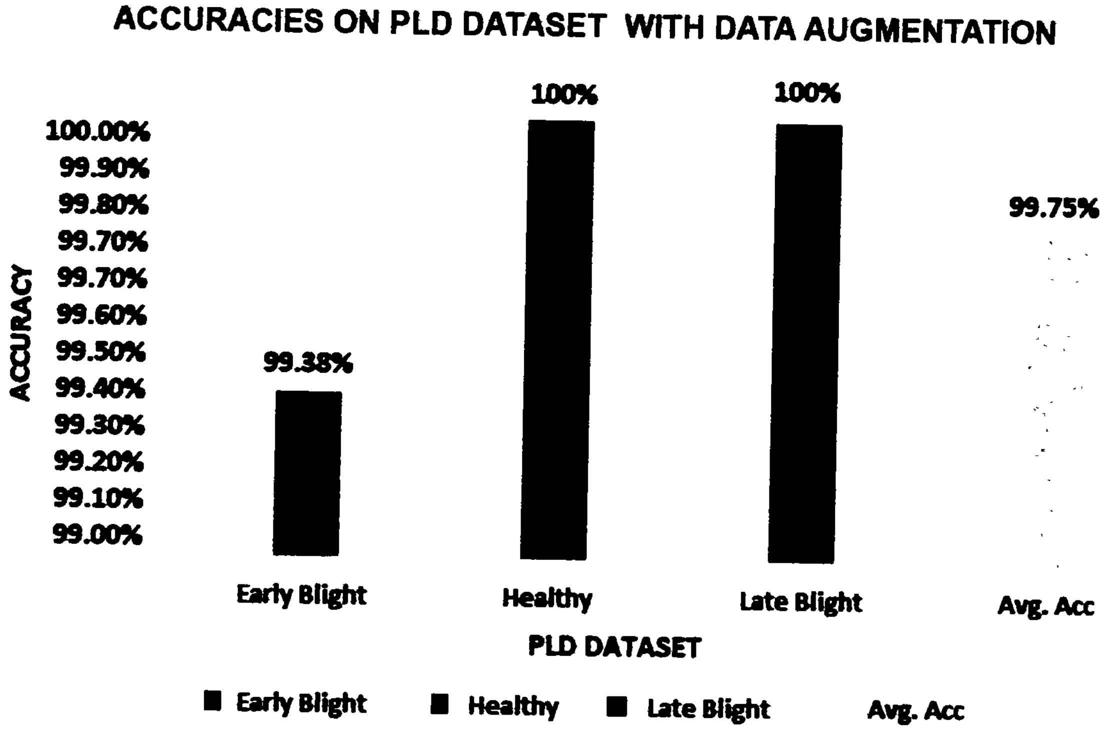
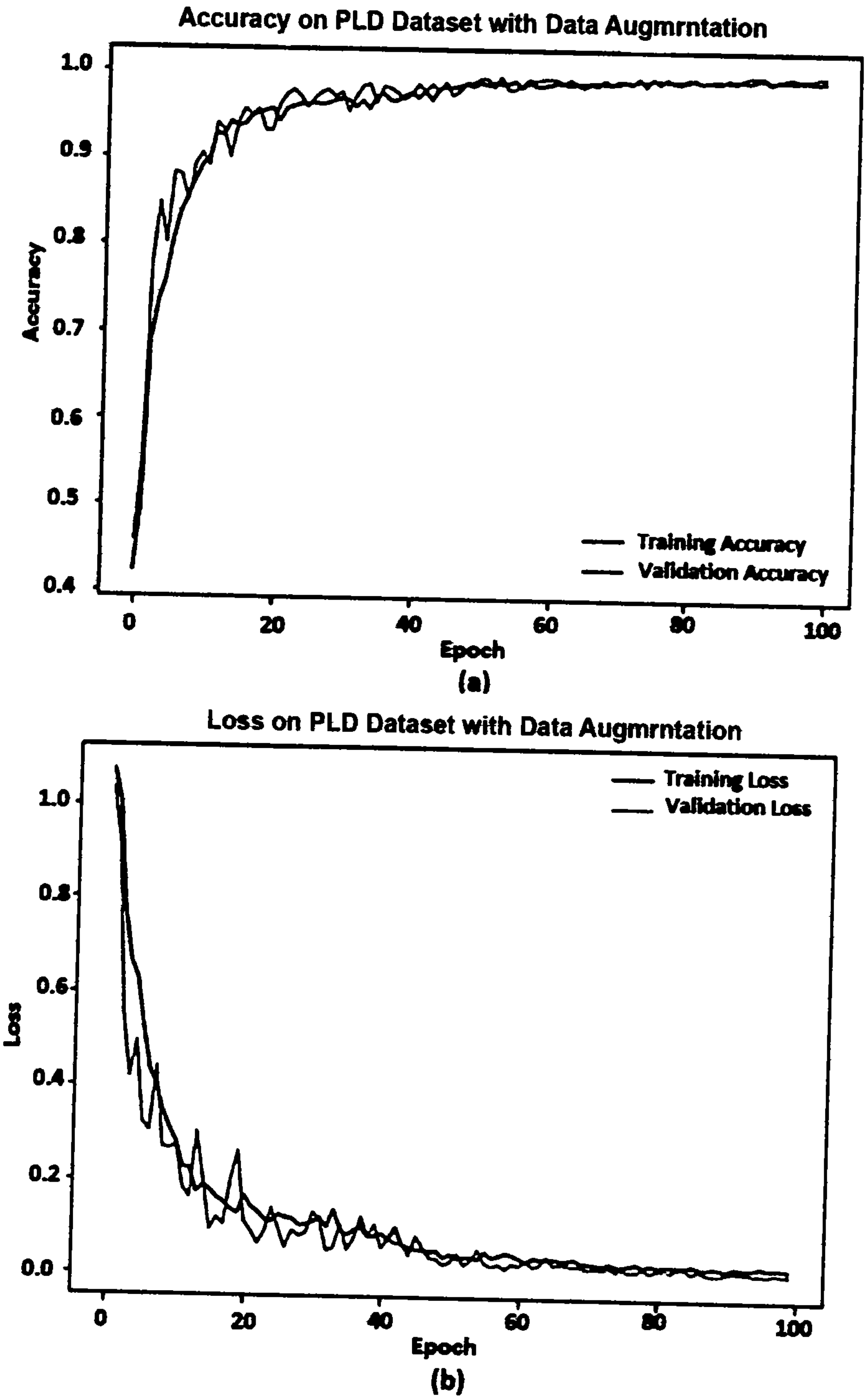


Figure 5.6: Accuracies Graph of *PDDCNN* on *PLD* with Data Augmentation.



Accuracy on PLD Dataset with Data Augmrntation

(a)

Loss on PLD Dataset with Data Augmrntation

(b)

Figure 5.7: a) Accuracy Graph of *PDDCNN* on *PLD* using Data Augmentation. (b) Loss Graph of *PDDCNN* on *PLD* using Data Augmentation.

Table 5.6: Classification Accuracies, Precision, Recall and F1-Score of the Proposed *PDDCNN* Model on the *PLD* Dataset Using Data Augmentation Techniques.

	Performance Measures	Early Blight	Healthy	Late Blight	Average
With Data Augmentation	Accuracy	99.38%	100%	100%	99.75%
	Precision	100%	99%	100%	-
	Recall	99%	100%	100%	-
	F1-Score	100%	100%	99%	-

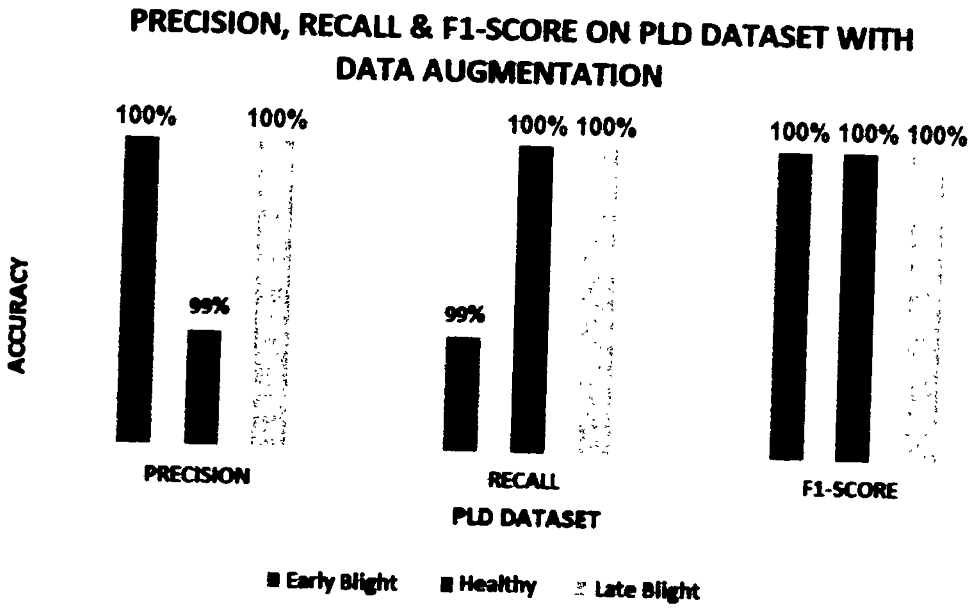


Figure 5.8: *PDDCNN* Precision, Recall and F1-Score on the *PLD* Dataset with Data Augmentation.

The Table 5.6 displays the results of applying data augmentation approaches to the proposed *PDDCNN* model on the *PLD* dataset. Three classes—early blight, healthy, and late blight—were used to test the model. The model's accuracy for all three classes is relatively high, with healthy and late blight receiving perfect scores of 100% and 99.38% for early blight. Since the model's accuracy for early blight and late blight is 100%, its successful predictions for these classes were realized. Just 1% of the positive predictions provided by the model for this class were inaccurate for healthy, where the precision is 99%. The model also has an extremely high recall rate, scoring 100% for early blight and Healthy, proving that it correctly identified every instance of both classes. The recall for late blight is 100%, indicating that the model accurately detected all instances. The harmonic mean of recall and precision, known as the F1 score, strikes a compromise between the two metrics. Early blight and healthy have an f1 score of 100%, demonstrating an ideal harmony between precision and recall. The model is doing exceptionally well for all three classes; however, the precision and recall metrics are slightly out of balance for late blight, as indicated by the f1 score of 99%, as depicted in Figure 5.8.

The performance of the proposed *PDDCNN* model on the 5.9 dataset utilizing data augmentation approaches is presented in the confusion matrix illustrated in Figure 5.9. The matrix shows that 162 of the 163 early blight samples were classified correctly, while just one was incorrectly labeled as healthy. None of the 102 healthy samples were misclassified; all 102 were appropriately categorized. Similarly, none of the 142 Late Blight samples were misclassified; all were accurately identified. The model's accuracy is 99.75%, meaning that 99.75% of the samples were correctly identified, with only 0.25% misclassified. In conclusion, the confusion matrix shows that the Proposed *PDDCNN* Model used data augmentation strategies to obtain high accuracy and perform admirably on the PLD dataset for all three classes: early blight, healthy, and late blight.

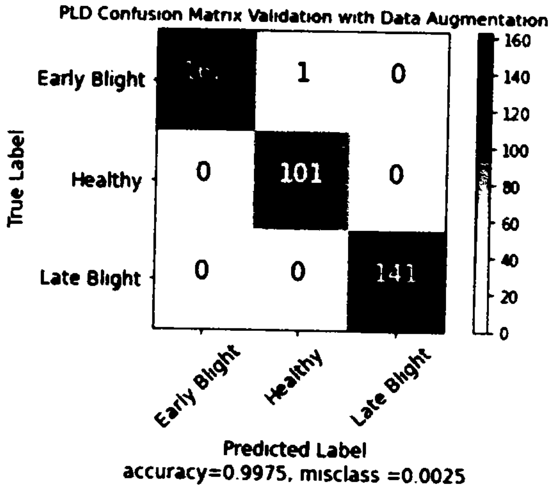


Figure 5.9: The Confusion Matrix of *PDDCNN* on *PLD* with Data Augmentation.

Figure 5.10 displays, using data augmentation approaches, the *ROC* curve of the *PDDCNN* model on the *PLD* dataset. Three separate classes—early blight, healthy, and late blight—were used to evaluate the model. In Figure 5.10, the Early Blight, Healthy, and Late Blight AUC values are 100%, demonstrating that the model performed flawlessly for all three classes. The model successfully balanced sensitivity (true positive rate) and specificity (true negative rate) and accurately identified all positive and negative samples for each class according to an AUC of 100%. The ROC curve reveals that the *PDDCNN* model used data augmentation techniques to obtain a excellent classification performance on the *PLD* dataset, as seen by the excellent AUC values for all three classes.

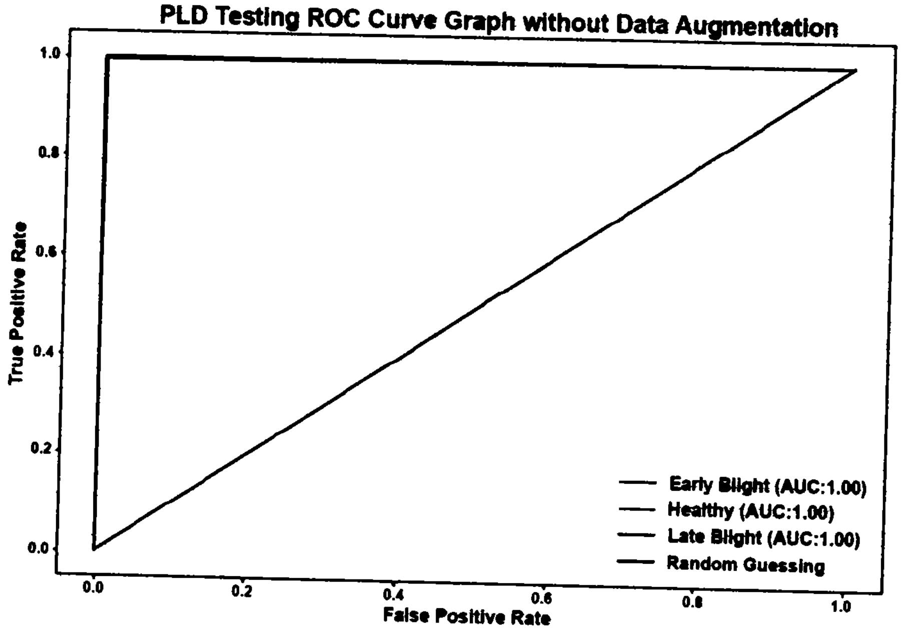


Figure 5.10: *ROC* Curve on the *PLD* Dataset with Augmentation Techniques.

When incorporating data augmentation strategies into the training set, the suggested method demonstrated outstanding performance on the *PLD* dataset in all assessment metrics.

In the second experiment, the performance of the proposed *PDDCNN* model was measured without using data augmentation techniques on the *PLD* dataset's training set. In this second experiment, we utilized the identical model and set of parameters as the first, but we simulated only 20 epochs instead of 100. As shown in Figure 5.11, the suggested technique averaged 91.15% accuracy on the *PLD* dataset and reached a precision of 93.87% for identifying early blight, 84.47% for identifying healthy plants, and 92.25% for identifying late blight. Accuracy during training and validation, as well as losses during each period, were all displayed in Figure 5.12. The proposed strategy produced lower identification rates without using data augmentation techniques in the training set. As a result, it is recommended to be trained on a massive dataset to improve its classification accuracy.

The proposed PDDCNN model performance was evaluated without applying the data augmentation techniques on the *PLD* dataset's training set in the second experiment. The same model and parameters were used as in the first experiment, but the number of epochs was reduced to 20. The proposed method achieved 93.87%, 84.47% and 92.25% accuracy for early blight, healthy and late blight, respectively, and gained 91.1% average accuracy on the PLD dataset, as shown in Table 5.7 and Figure 5.12. The complete training and validation accuracy and losses in each epoch were shown in Figure 5.11. The proposed method achieved lower identification rates without using data augmentation techniques compared to using data augmentation techniques applied to the training set. Therefore, for better classification accuracy, it should train on a large-scale dataset.

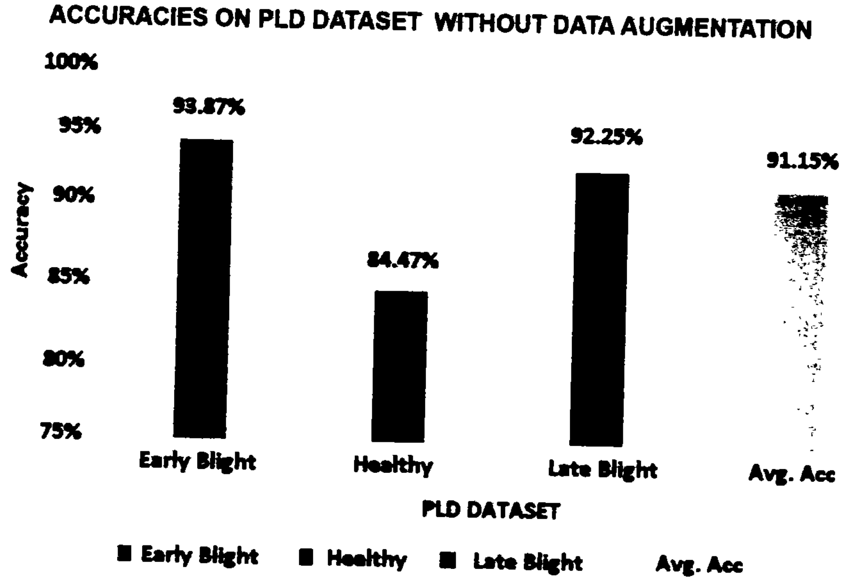


Figure 5.11: Accuracies Graph of *PDDCNN* on *PLD* without Data Augmentation.

Table 5.7: Classification Accuracies, Precision, Recall and F1-Score of the Proposed *PDDCNN* Model on the *PLD* Dataset without Applying Data Augmentation Techniques.

	Performance Measures	Early Blight	Healthy	Late Blight	Average
Without Data Augmentation	Accuracy	93.87%	84.47%	92.25%	91.15%
	Precision	87%	94%	90%	-
	Recall	92%	85%	88%	-
	F1-Score	96%	92%	94%	-

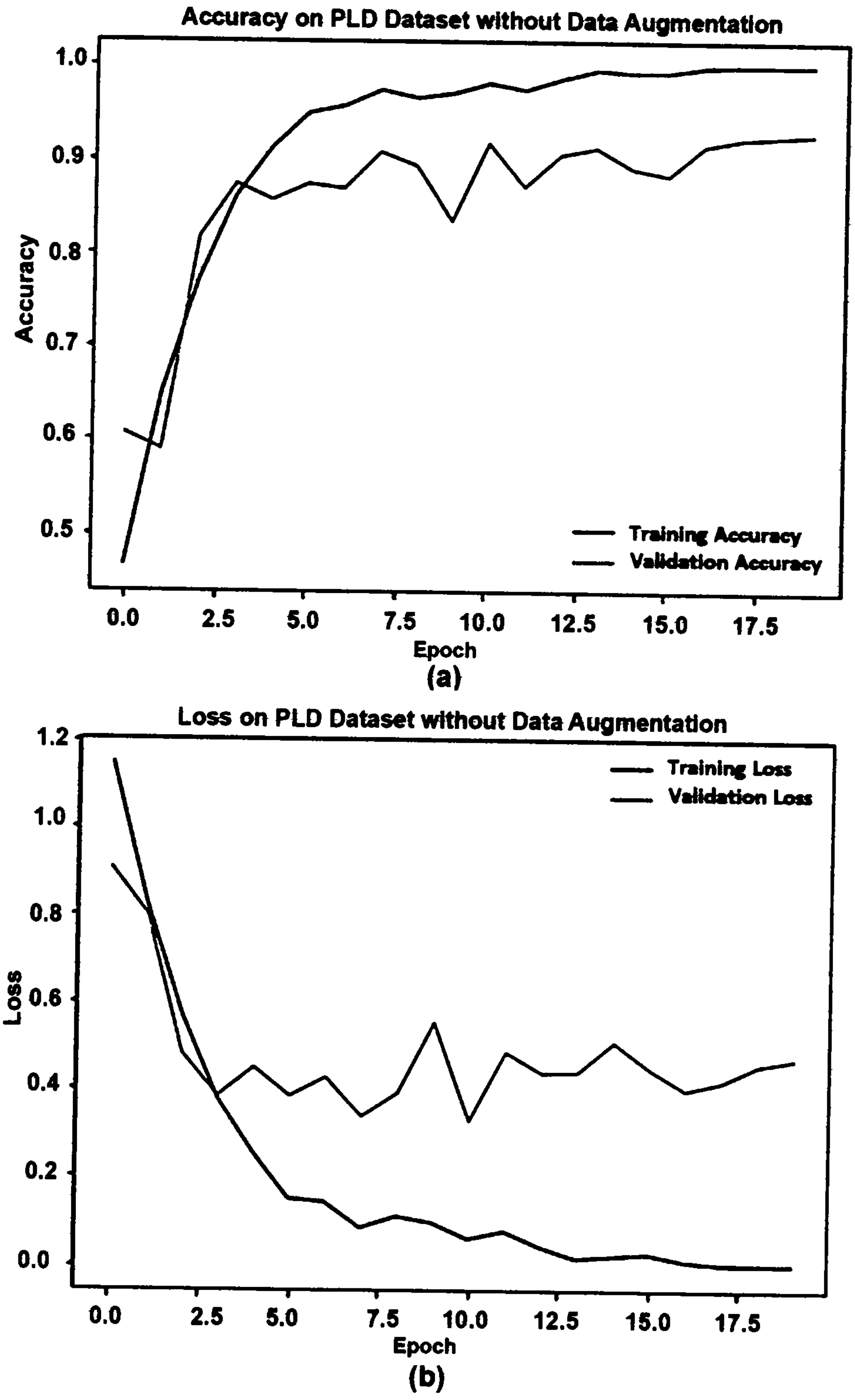


Figure 5.12: (**a**) Accuracy Graph of *PDDCNN* on *PLD* without Data Augmentation. (**b**) Loss Graph of *PDDCNN* on *PLD* without Data Augmentation.

The assessment metrics for the proposed PDDCNN model on the PLD dataset are shown in Table 5.7 without using data augmentation methods. Table 5.7 shows that for early blight, healthy, and late blight, the model's accuracy is 93.87%, 84.47%, and 92.25%, respectively. For early blight, healthy, and late blight, the model's accuracy is 87%, 94%, and 90%, respectively. For early, healthy, and late blight, the model's recall is 92%, 85%, and 88%, respectively. According to the model, early blight, healthy, and late blight obtained f1 scores of 96%, 92%, and 94%, respectively. According to the results, the suggested PDDCNN model performed well on the PLD dataset without data augmentation techniques. The healthy class's accuracy and other evaluation metrics are lower than those for the other two classes, as illustrated in Figure 5.13.

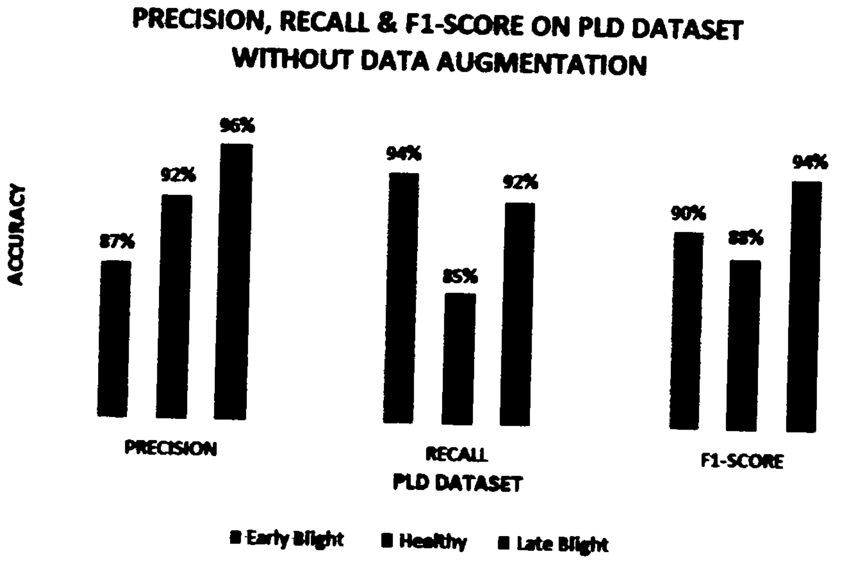


Figure 5.13: *PDDCNN* Precision, Recall and F1-Score on the *PLD* Dataset without Data Augmentation.

The proposed *PDDCNN* model's confusion matrix on the *PLD* dataset without any data augmentation approaches is shown in Figure 5.14. The algorithm accurately identified 153 samples of early blight as early blight, while it incorrectly identified five samples as healthy and five as late blight (see Figure 5.14). The model correctly classified 87 healthy samples as healthy and misclassified 15 as early blight. The proposed method accurately identified 131 samples of late blight as late blight, misclassified three as healthy, and incorrectly identified eight samples of early blight. The model's overall accuracy is 91.15%, meaning that the model correctly classified 91.15% of all samples in the dataset. The model was a misclassification rate of 8.85%, indicating that the model incorrectly classified 8.85% of the samples. As can be seen in Figure 5.14, the model did quite well with the early and late blight classes but had trouble classifying with the healthy class, incorrectly labeling 15 samples as early blight.

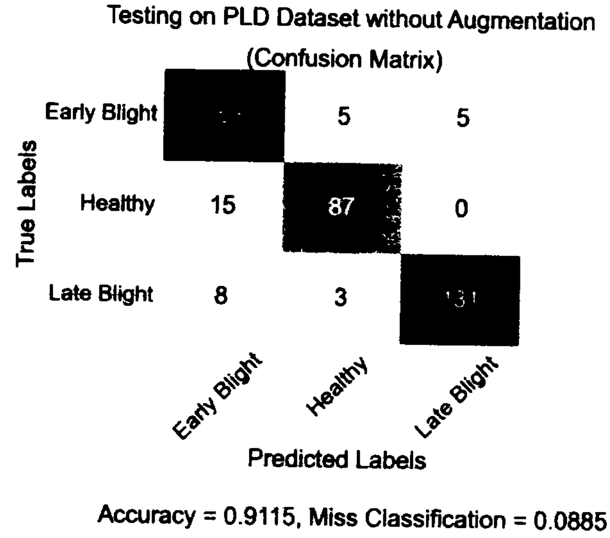


Figure 5.14: The Confusion Matrix of *PDDCNN* on *PLD* without Using Data Augmentation.

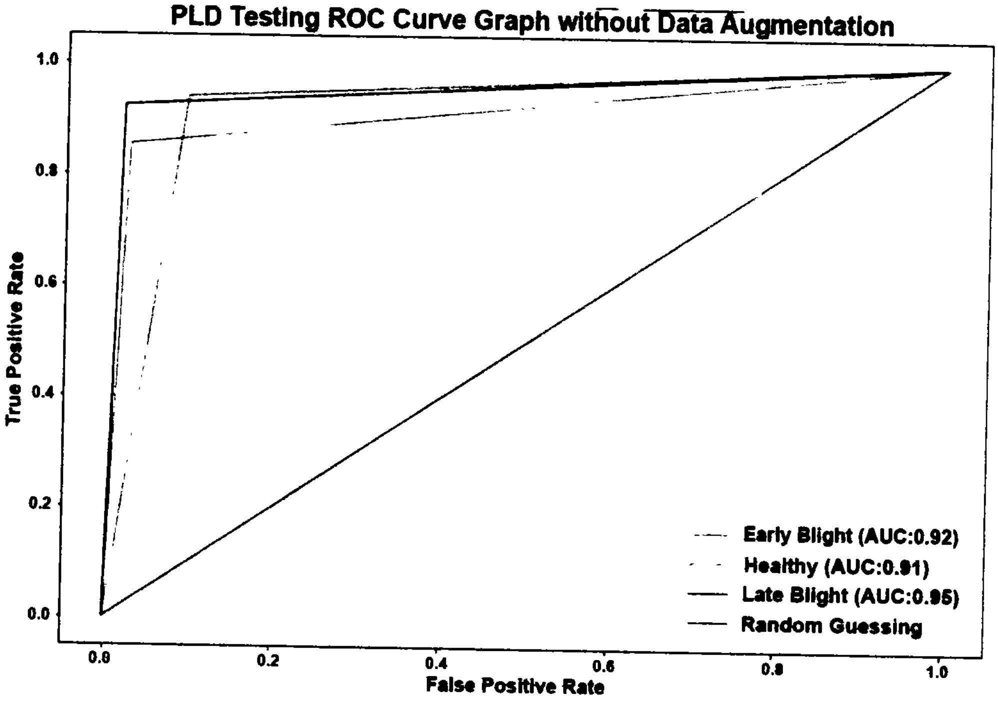


Figure 5.15: *ROC* Curve on the *PLD* Dataset Without Applying Augmentation Techniques.

The proposed *PDDCNN* model's ROC curve on the *PLD* dataset without any data augmentation approaches is shown in Figure 5.15. The AUC can take on values between 0 and 1, with 0.5 being a randomly selected classifier and 1 representing a perfect classifier. Figure 5.15 shows the model has good discriminatory power for the early blight class, with an AUC of 92%. The model's ability to discriminate healthy from unhealthy samples is slightly lower than for early blight, as indicated by the AUC for the healthy class being 91%. The model performs well in classifying instances of late blight, as evidenced by the 95% AUC. The model's performance, especially its discriminatory power, can be gauged through the ROC curve and AUC values. Figure 5.15 shows that the model performs well at classifying the *PLD* dataset, as indicated by the high AUC scores.

The proposed strategy performed poorly across the evaluation metrics without using data augmentation techniques on the PLD dataset. It implied that a massive amount of data would be needed to train the proposed approach. When there was less information to work with, overfitting became an issue. The overfitting could be eliminated by enhancing the dataset with the help of different data augmentation methods or increase the dataset to millions of images.

5.3.2 The Proposed *PDDCNN* Model Performance on the PlantVillage Dataset

The suggested model was trained using data from sources outside the *PLD* dataset to increase its generalizability. The potato harvest was the primary focus of the investigation. We used the potato leaves from the open-source dataset PlantVillage [161] to evaluate the *PDDCNN* approach.

The proposed *PDDCNN* strategy was tested twice on the PlantVillage dataset. We applied the data augmentation approaches to the training set in the first experiment. The second experiment did not involve the data augmentation techniques to the training set of the PlantVillage dataset. The same model and parameters were utilized in the first experiment used in the preceding section utilizing the data augmentation approaches applied to the training set of the PlantVillage dataset. The proposed method's generality was checked utilizing training, validation, and testing. The experiment findings revealed that the proposed technique obtained 99%, 92.31%, and 95% accuracies for early blight, healthy, and late blight classes, respectively. The suggested technique achieved 96.71% average accuracy on the PlantVillage dataset in Figure 5.16. Figure 5.17 displays the training and validation accuracy and loss for each epoch. The results demonstrated that the suggested method achieved outstanding identification rates when the data augmentation strategies were used in the training set of the PlantVillage dataset, which indicated the proposed *PDDCNN* model's generalization.

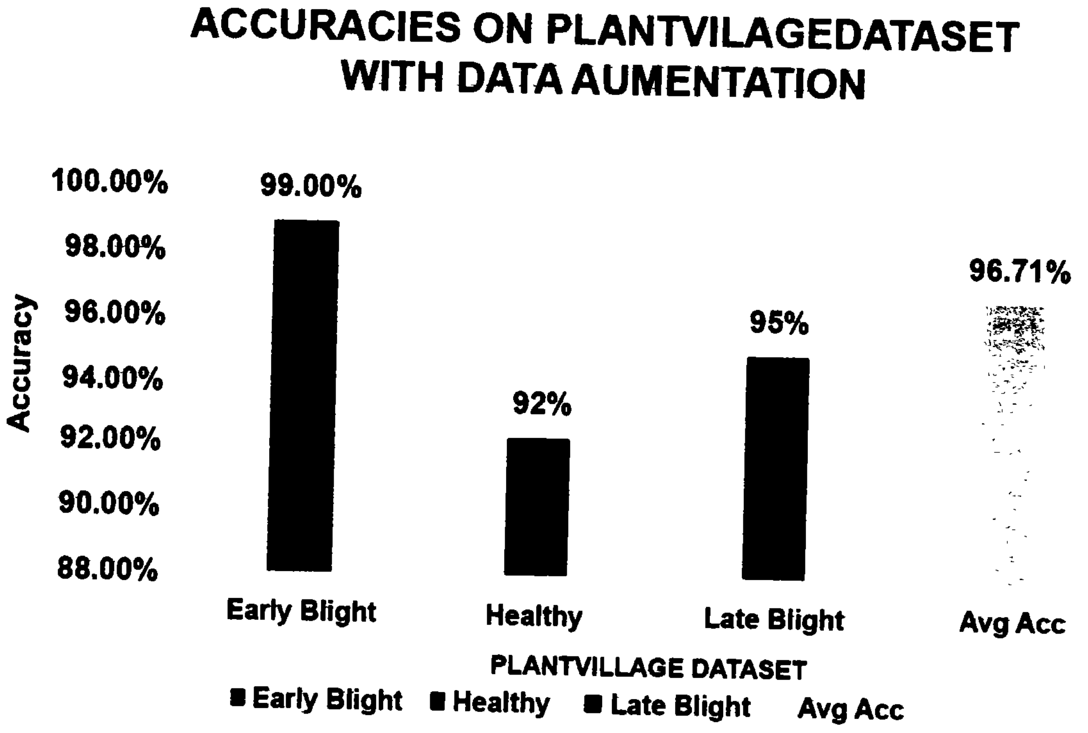


Figure 5.16: Accuracies Graph of PDDCNN on PlantVillage with Data Augmentation.

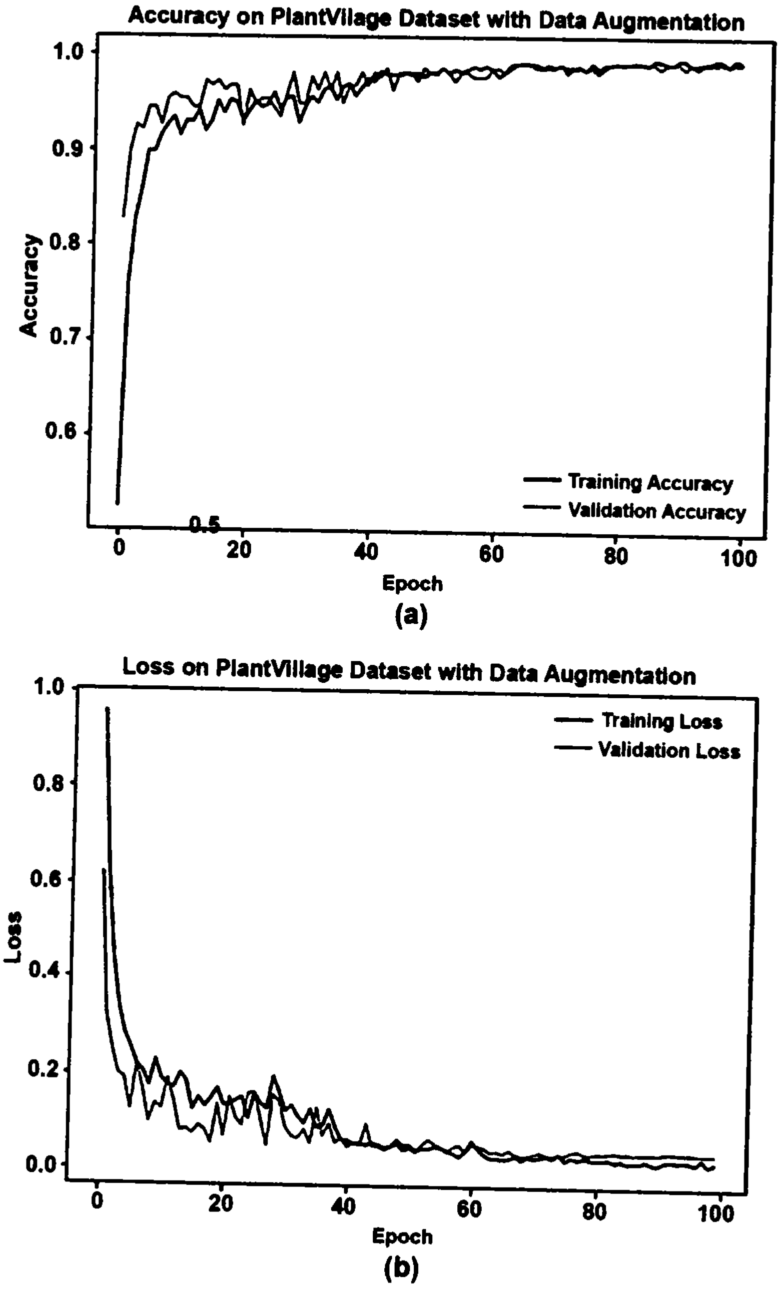


Figure 5.17: (a) Accuracy Graph of PDDCNN on PlantVillage using Data Augmentation. (b) Loss Graph of PDDCNN on PlantVillage using Data Augmentation.

The test set's accuracy, recall, and f1 score are further used to evaluate the suggested approach. The outcomes showed excellent performance across all classes in the PlantVillage dataset using data augmentation approaches. Table 5.8 and Figure 5.18 show that the performance of the proposed

PDDCNN model was outstanding after applying data augmentation strategies to the training set of the PlantVillage dataset. Using the PlantVillage dataset, the *PDDCNN* achieved 95% precision for early stages, 100% for healthy stages, and 98% precision for late stages of blight, and it attained 99%, 92%, and 95% recall for early blight, healthy, and late blight classes. The system achieved f1 scores of 97%, 96%, and 96% for early, healthy, and late blight classes, respectively. All three plant diseases were detected and correctly classified by the suggested *PDDCNN* model, with the results showing high accuracy, precision, recall, and f1 score. The model performed best for early blight, then late blight, then healthy classes, in that order, as depicted in Figure 5.18.

Table 5.8: Classification Accuracies, Precision, Recall & F1-Score of the Proposed PDDCNN Model on the PlantVillage Dataset Using Data Augmentation Techniques.

	Performance Measures	Early Blight	Healthy	Late Blight	Average
With Data Augmentation	Accuracy	99%	92.31%	95%	96.71%
	Precision	95%	100%	98%	-
	Recall	99%	92%	95%	-
	F1-Score	97%	96%	96%	-

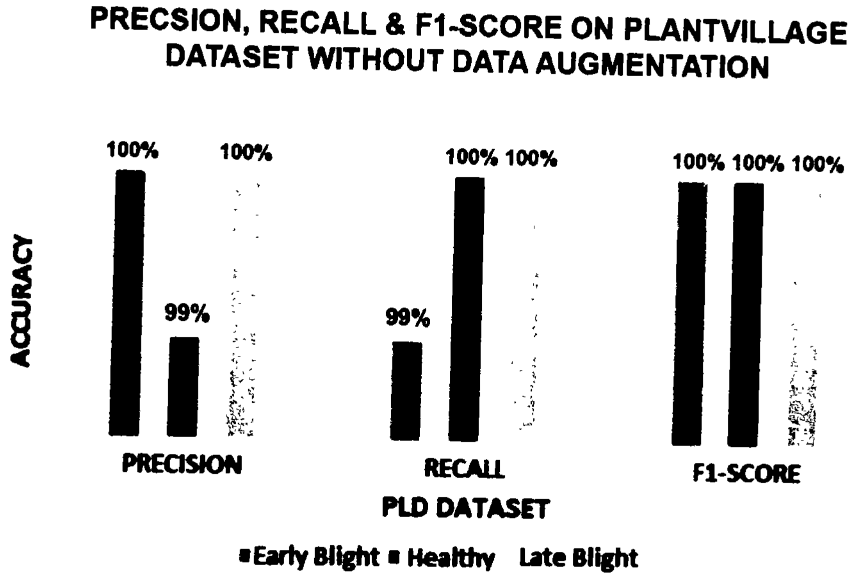


Figure 5.18: PDDCNN Precision, Recall and F1 Score on PlantVillage with Data Augmentation Techniques.

Figure 5.19 also depicts the confusion matrix for the PlantVillage dataset after data augmentation techniques have been applied to the training set. The suggested *PDDCNN* model successfully classified 99 out of 100 images as having early blight disease, 12 out of 13 images as healthy leaves, and 95 out of 100 images of late blight leaves. On the PlantVillage dataset, with data augmentation techniques applied to the training set, the proposed *PDDCNN* model achieved a 96.71% overall classification accuracy. After using data augmentation approaches to the PlantVillage dataset's training set, the suggested model achieved a 3.29% misclassification ratio across all classes. The results showed that by applying data augmentation techniques to a training set comprised of data from both the PLD and PlantVillage datasets, the suggested *PDDCNN* method could obtain high-quality prediction results.

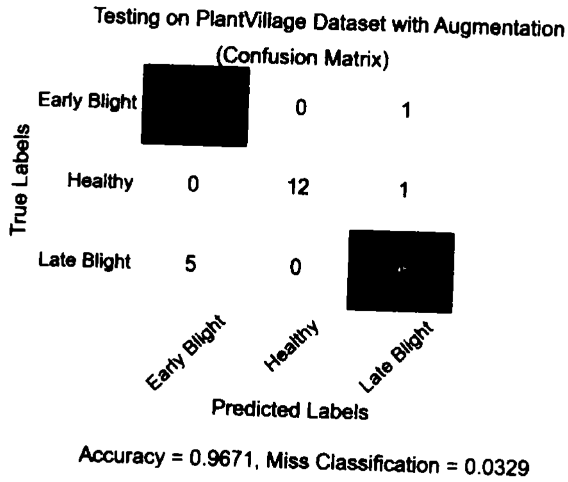


Figure 5.19: PDDCNN Confusion Matrix on PlantVillage with Data Augmentation Techniques.

Figure 5.20 shows that we used ROC measurements to verify the effectiveness of our data-augmentation strategies on the PlantVillage dataset. The blue hue represents the disease in its first stages. The colour orange represents a fit and active group. Late blight class is depicted in green, while those who take a more haphazard approach is shown in blue. A good area under the *ROC* curve was obtained for early blight (97%), healthy (96%), and late blight (97%), indicating successful classification on the PlantVillage dataset when data augmentation techniques were used to the training set of the proposed *PDDCNN* model. The latter was superior to the PlantVillage dataset since it provided more training data for the proposed model. The proposed technique performed worse on the PlantVillage dataset across the board than the *PLD* dataset.

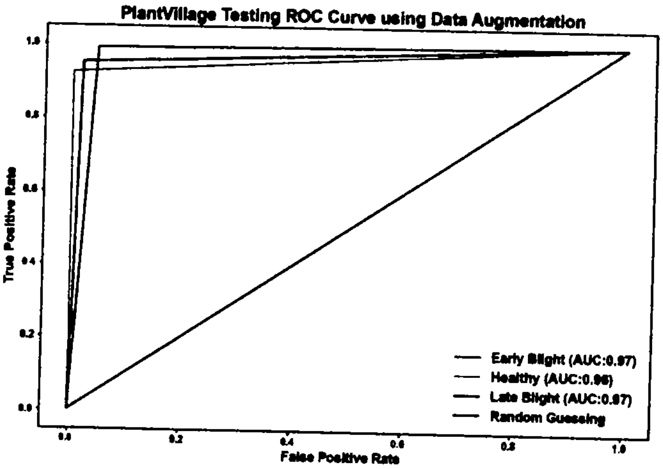


Figure 5.20: PDDCNN ROC Curve on PlantVillage with Applying Data Augmentation.

The effectiveness of the proposed *PDDCNN* model was measured in a second experiment that used the PlantVillage dataset as training data without any data augmentation approaches. The same experimental setup was used, but only 20 epochs were used this time. The experimental results showed that the proposed approach had a 91% success rate for early blight, a 100% success rate for healthy, and a 96% accuracy for late blight. As seen in Figure 5.21, the suggested strategy achieved an average accuracy of 93.90% on the PlantVillage dataset without applying data augmentation techniques to the training set. Figure 5.21 displays the overall performance of the proposed *PDDCNN* approach on the PlantVillage dataset without data augmentation techniques used for the training set, including the training and validation accuracies and losses.

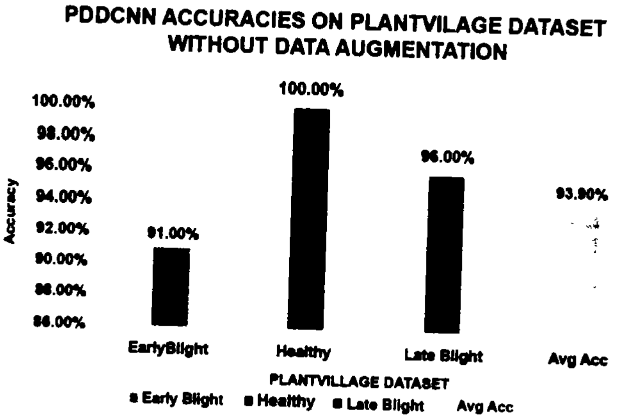
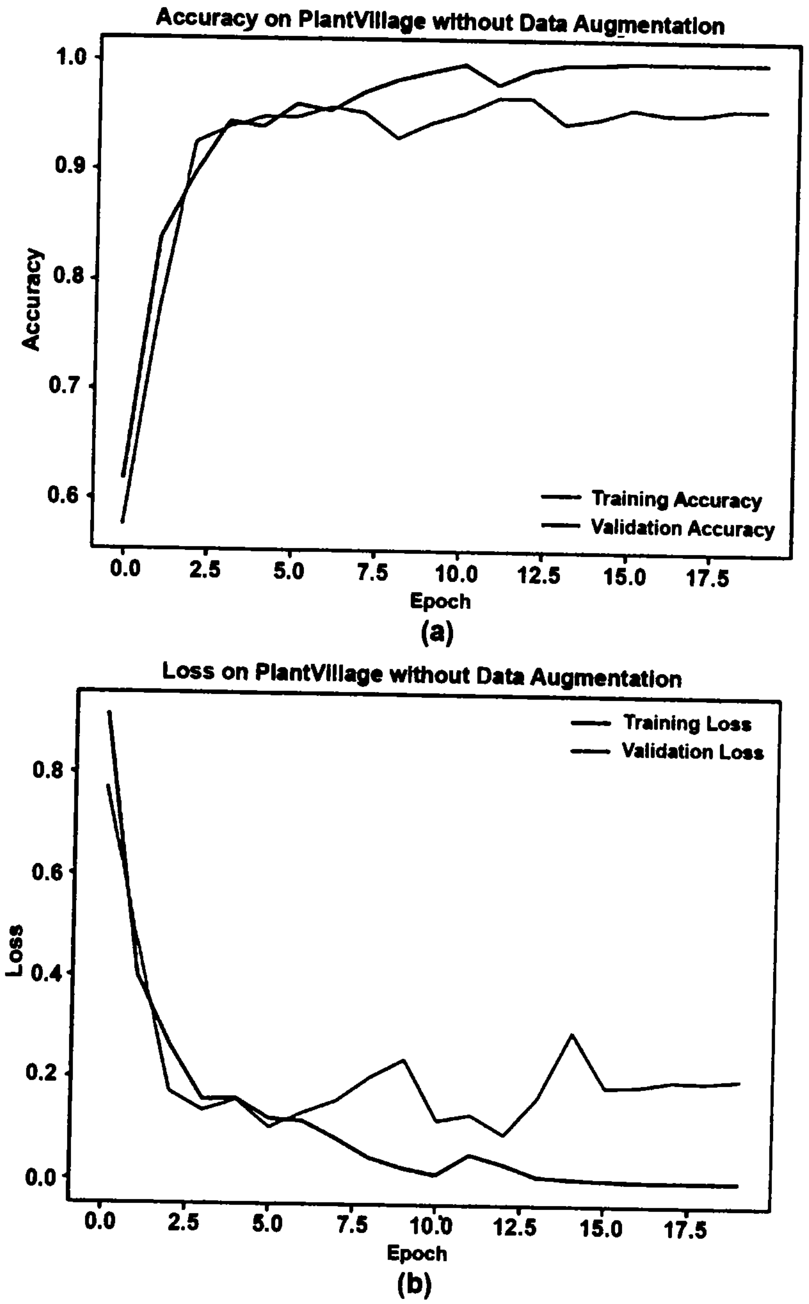


Figure 5.21: Accuracies Graph of PDDCNN on PlantVillage without Applying Data Augmentation.



(a)

(b)

Figure 5.22: (a) Accuracy Graph of PDDCNN on PlantVillage without Data Augmentation. (b) Loss Graph of PDDCNN on PlantVillage without Data Augmentation.

Table 5.9 and Figure 5.23 display the results of precision, recall, and f1 score calculations performed to validate the suggested method's performance further. The suggested technique achieved 87%, 94%, and 90% precision on the early, healthy, and late blight classes, respectively, on the

PlantVillage dataset. It had 92% recall for early blight, an 85% recall for healthy, an 88% recall for late blight, and an F1 score of 96%, 92%, and 94%, respectively. When data augmentation was used on the PlantVillage dataset's training set, the findings indicated poorer performance across the board.

Table 5.9: Classification Accuracies, Precision, Recall & F1-Score of the Proposed PDDCNN Model on the PlantVillage Dataset.

	Performance Measures	Early Blight	Healthy	Late Blight	Average
Without Data Augmentation	Accuracy	91%	100%	96%	93.90%
	Precision	97%	93%	91%	-
	Recall	91%	100%	96%	-
	F1-Score	94%	96%	94%	-

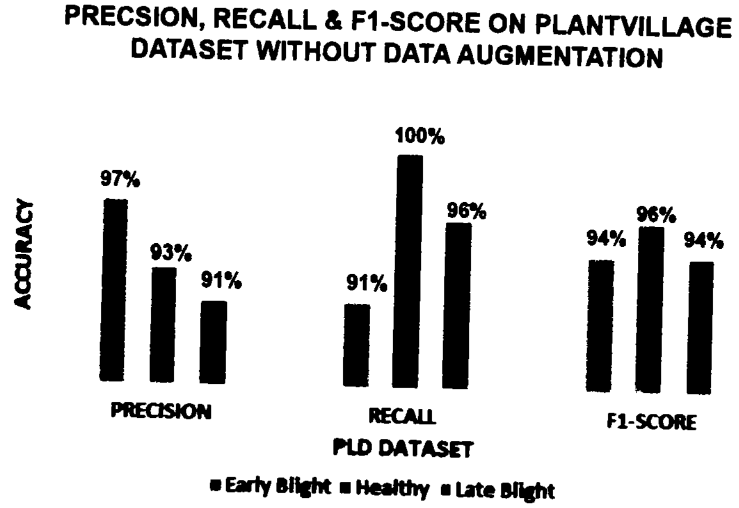


Figure 5.23: PDDCNN Precision, Recall and F1 Score on PlantVillage without Data Augmentation Techniques.

Figure 5.24 depicts the confusion matrix of the proposed *PDDCNN* approach without any data augmentation techniques applied to the test set. The suggested model correctly identified 91 out of 100 leaves with early blight. The analysis was spot-on for all healthy leaves (13) and 96 of the 100 photos of leaves with late blight. Without data augmentation strategies, the suggested model on the PlantVillage dataset showed lesser classification performance, with 93.90% accuracy and a 6.10% misclassification rate, as shown in Figure 5.24.

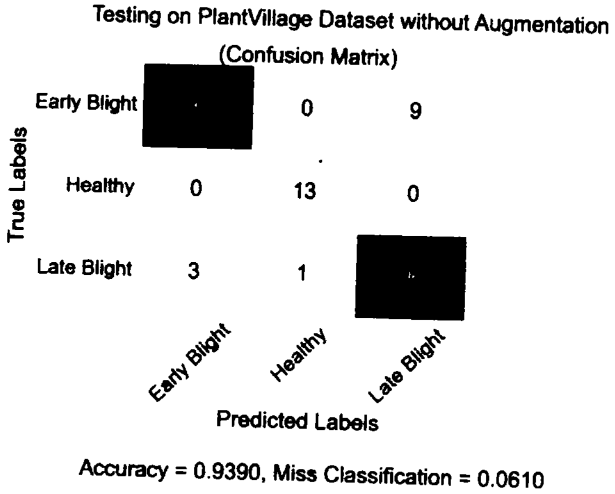


Figure 5.24: PDDCNN Confusion Matrix on PlantVillage without Data Augmentation Techniques.

Without resorting to data augmentation approaches, the ROC curve (Figure 5.25) was utilized to evaluate the efficacy of the suggested strategy on the PlantVillage dataset. The light blue colour indicates the early blight, and the orange colour represents the healthy class. The green colour denotes the late blight class, and the blue colour denotes random guessing. The ROC curve graph showed that the early blight had 94%, healthy had 100% and late blight had 94% area under the curve.

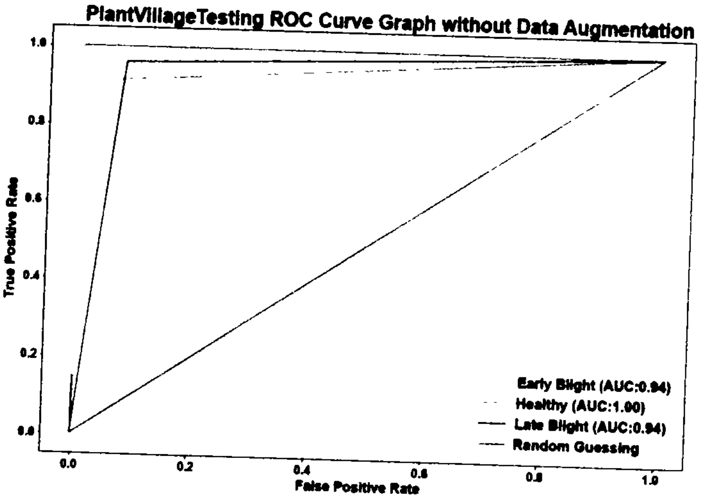


Figure 5.25: PDDCNN ROC Curve on PlantVillage without Applying Data Augmentation.

When data augmentation techniques were not applied to the PlanVillage dataset's training set, the results showed that the proposed PDDCNN algorithm performed less accurately. The proposed strategy could have performed better because it was trained with a smaller sample of data. To train, deep learning requires an enormous amount of information. Therefore, thê dataset needs to be huge to improve the performance of the suggested strategy.

5.3.3 Cross Dataset Performance

The detection of crop diseases in Pakistan was an area that needed more study. Two experiments were run on a combined dataset to evaluate how well the proposed strategy worked. We trained on the PlantVillage dataset and tested on the PLD dataset, both with the same experimental setup as the first experiment. We used the PLD dataset for training and the PlantVillage dataset for testing in the second set of experiments. As shown in Table 5.10 and Figure 5.26, the proposed method had an accuracy of 48.89% in the first experiment and 86.38% in the second experiment. This finding lends credence to the idea that regional differences in plant diseases and species are attributable to climate change, climatic variation, and other environmental factors.

Table 5.10: Classification Accuracies of the Proposed PDDCNN Model Training on PlantVillage and Testing on PLD and Training on PLD and Testing on the PlantVillage Dataset.

Training Dataset	Testing Dataset	Early Blight Accuracy	Healthy Accuracy	Late Blight Accuracy	Total Testing Images	Overall Accuracy
PlantVillage	PLD	95.71%	08.82%	23.94%	807	48.89%
PLD	PlantVillage	92.00%	100%	79.00%	213	86.38%

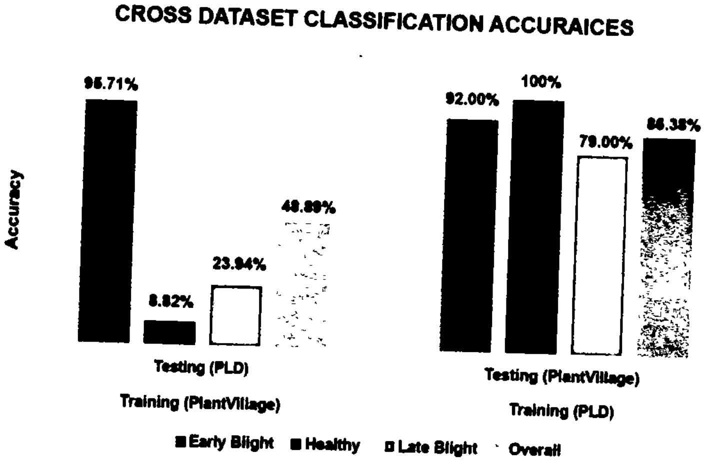


Figure 5.26: PDDCNN Cross Dataset Classification Accuracies.

5.3.4 Accuracies Comparison of Proposed Method with State-of-the-Art Methods

We compared the proposed PDDCNN model to the VGG16 [224], [263], DenseNet_121 [28], DenseNet169 [28] and Xception [264] models on the PLD dataset through a process called transfer learning. For this reason, all tests utilized identical conditions and data enhancement strategies. The precision of contemporary deep learning methods is displayed in Table 5.11. Table 5.11 displays the results of various models on the PLD dataset, including the accuracy attained by each: VGG16, 40.05%; InceptionResNetV2, 99.26%; DenseNet121, 99.26%; DenseNet169, 99.53%; Xception, 99.26%; and the suggested PDDCNN model, 99.75%. According to the findings, the proposed PDDCNN model was the most accurate (99.75%), whereas VGG16 was the least accurate (40.5%). Compared to the VGG16, InceptionResNetV2, DenseNet121, DenseNet169, and Xception, the suggested PDDCNN approach required fewer training parameters. Table 5.11 shows that the state-of-the-art models have more trainable parameters than the proposed PDDCNN model. The VGG16, Inception ResNetV2, DenseNet121, DenseNet169, and Xception models, for example, each had 14,716,227, 20,867,627, 7,040,579, and 12,647,875 trainable parameters. Table 5.11 demonstrates that compared to the state-of-the-art models, the suggested PDDCNN model saved a lot of computational expenses and required less time to train the model than all of them except the DenseNet121 model, which had the lowest number of parameters at 7,040,579. The suggested PDDCNN model required fewer convolutional layers, which required fewer parameters and a lower overall computing cost.

Table 5.11: Comparison with State-of-the-Art Techniques.

Model	Total Parameters	Accuracy
VGG16 [224]	14,716,227	40.05%
InceptionResNetV2 [263]	20,867,627	99.26%
DenseNet_121 [28]	7,040,579	99.26%
DenseNet169 [28]	12,647,875	99.53%
Xception [264]	20,867,627	99.26%
PDDCNN	8,578,611	99.75%

5.3.5 Accuracies Comparison of Proposed Method with Existing Studies

Table 5.11 displays the results of a comparison between the suggested technique and state-of-the-art methodologies and represents the generalizability of the proposed approach. According to the results, the proposed deep learning model outperformed the current gold standard by a wide margin. The proposed method could have been more precise than the state-of-the-art methods. The proposed method's effectiveness was evaluated compared to other methods currently used for detecting diseases in potato leaves. According to Table 5.12, the proposed method has the best accuracy (99.75%) compared to previous research. Tiwari et al. [166] obtained 97.80% accuracy; however, they used pre-trained models with many trainable parameters (i.e., 143,667,240). Therefore the proposed PDDCNN model fared better. Potato disease detection was performed using the PlantVillage dataset, which featured fewer photos and an uneven distribution of classes. The 14-layer architecture used by Khalifa et al. [162] was reported to have achieved 98.00% accuracy at a high computational cost. PlantVillage was used, a dataset with fewer photos and unbalanced classes. With only 10,089,219 trainable parameters with the PlanVillage dataset's imbalanced classes and fewer parameters, Lee et al. [167] achieved 99.00% accuracy. With only 8,578,611 parameters, the suggested PDDCNN model outperformed state-of-the-art methods to achieve an accuracy of 99.75% at a reduced computational cost.

Table 5.12: Comparison with Existing Studies.

Existing Study	Total Parameters	Accuracy
Rozaqi and Sunyoto [163]	6,812,995	92.00%
Islam et al. [168]	-	95.00%
Sanjeev et al. [164]	-	96.50%
Barman et al. [165]	16,407,395	96.98%
Tiwari et al. [166]	143,667,240	97.80%
Khalifa et al. [162]	-	98.00%
Lee et al. [167]	10,089,219	99.00%
PDDCNN	8,578,611	99.75%

5.4 Chapter Summary

The use of deep learning algorithms for plant leaf disease detection has the potential to improve crop yields and quality. In this chapter, we suggested a multi-tiered deep learning model for identifying potato leaf illnesses, which can be used for easy and rapid classification of potato leaf diseases. The potato leaves in the plant image were initially extracted using the YOLOv5 image segmentation algorithm. In order to distinguish between early and late blight potato illnesses in potato leaf images, a unique potato leaf disease detection convolutional neural network (PDDCNN) was created at the second level. Furthermore, environmental factors were also considered.

Images of potato leaves captured in Central Punjab, Pakistan, responded favourably to the proposed PDDCNN method. Experiments used both the non-augmented and augmented versions of the PlantVillage and PLD datasets. The proposed PDDCNN algorithms were shown to be the most effective across all datasets. The effectiveness of the proposed method was evaluated against other approaches for disease detection in potato leaves. Both the PlantVillage and PLD datasets were more sensitive to variations in environmental factors and disease symptoms because of the PLD dataset's high false-positive rate using state-of-the-art and current methodologies. Its 99.75% accuracy, high precision on the PLD dataset, few parameters, low computational cost, and fast speed are its defining features.

Chapter 6

Conclusions and Future Directions

To support agricultural processes, plant disease detection is one major area of interest for research community. For plant disease detection one approach is through classification and segmentation of plant leaves. To address the problem of classification and segmentation, AI based approaches may be employed. One such approach is using deep learning algorithms. This research set its scope for identifying the infected regions of leaves for various plant species including that of potato and guava. In particular, due to the striking resemblance in morphology among many plant species, it is challenging to accurately segment and classify them. Additionally, early leaf disease identification is difficult due to differences in crop species, disease symptoms, and environmental conditions. As a result of these variances, early diagnosis of leaf diseases is often difficult. Researchers have created several machine learning methods to identify the affected area, identify plant species, and spot diseases on leaves. Important aspect is to detect the disease in real time mode. This research focuses on addressing the problem considering the four main areas including Plant Species Recognition, Identification of leaf's Infected Patches, Leaf Segmentation, and Leaf multiple diseases. For the inaccuracies among achieving these tasks, the overall complexity and accuracy of the system gets compromised. To address these problems, this research focused on the following key points:

1. To identify the specie of the plant from its leaf, a deep ensemble technique was developed. It recognizes the potato and guava species using leaf images. It enables the model to identify the potato and guava species of different varieties in the Central Punjab region of Pakistan.

2. A Modified MobileNetV2 and U-Net (GIP-MU-NET) Segmentation Technique was developed to detect the infected patches from plant leaves. It will help the farmers with recommending pesticides specifically for the infected region of the plants. Thus, the approach is

expected to save costs, improve crop yield, and enhance the farmer's income.

3. Three methods have been developed to detect infected patches from the crop leaves. The first novel method is the Guava Leaf Segmentation Model (GLSM). GLSM segments the guava leaves, which was a challenging task due to limited scale and scope of the data and its variances due to geographic specific information. This technique helps to segment the guava leaves having complex backgrounds. The second model is developed for extracting potato leaves from the potato plant images with a complex background based on YOLOv5. Another segmentation technique based on MobileNetV3-UNet has been created. It segments the diseased and healthy leaves of guava and potato.

4. Diseases in potato and guava leaves can be identified using two distinct deep-learning approaches. The first model, the PDDCNN model, was created to detect potato single leaf disease from a leaf from different varieties of potatoes in the Central Punjab region of Pakistan. This model's ability to identify late and early blight disorders in potatoes improved crop yields and early diagnosis of diseases for Pakistani farmers. The other deep learning model, Guava Multiple Leaf Diseases Detection (GMLDD), has been developed to detect multiple leaf diseases on a single leaf. The GMLDD model classifies anthracnose, nutrient deficiency, wilt diseases, insect attack, and healthy leaves of different guava diseases on a single leaf. It helps them to detect the multiple diseases of guava in real-time. Both methods will benefit Pakistan's agricultural sector by allowing farmers to identify diseases early, reducing losses, and increasing crop yields and profits for farmers.

5. The existing methods did not correctly identify the Pakistani region potato leaf diseases because all the current practices were only trained on the specific dataset that did not include the images of the crops from Pakistani region. Shape, symptoms, leaf color, variety, and environmental conditions are only a few of the many ways in which diseases of potatoes and guavas differ around the world. As a result, the current techniques are limited in identifying guava and potato infections in Pakistan. Therefore, there is a dire need to develop a new dataset to detect the Pakistani region's potato and guava leaves' diseases so that farmers in Pakistan can determine the diseases in their early stages, enhance crop yield, and boost the country's economy. For this purpose, we developed the following datasets from the Central Punjab region of Pakistan:

   (a) **Development of Plant Species Dataset (PLSD):** The first-ever Plant Species Dataset (PLSD) is developed to classify the guava, java plum, and potato plant leaf species from the central Punjab region of Pakistan.

(b) **Development of Potato Leaf Dataset (PLD):** This dataset may help the researchers with the opportunity to research Pakistani potato varieties and diseases. For this purpose, Potato Leaf Dataset (PLD) has been developed to identify late, early blight diseases, and healthy potato leaves. There is only one instance of a disease affecting a potato leaf in this data collection.

(c) **Development of Guava Leaf Disease Dataset (GLDD):** The Guava Leaf Disease Dataset (GLDD) was created to identify many guava leaf diseases from a single leaf. It presents multiple disorders, including anthracnose, nutrient deficiency, wilt diseases, and insect attack on a single leaf. The guava leaf images were collected from the Central Punjab region of Pakistan. We believe this dataset enables the computer-based disease detection techniques to analyze more than one guava disease on a single leaf.

6. The Guava Multiple Leaf Diseases Detection (GMLDD) approach was designed to identify plant diseases in real-time for optimal performance. In light of this progress, the suggested methods have become more effective in terms of their accuracy. However, the new PDD-CNN approach improves efficiency by reducing the ideal number of parameters compared to existing approaches. The developed PDDCNN model works better than before because it can now achieve maximum accuracy.

6.1 Suggestions for Future Work

The field of plant leaf disease segmentation and recognition has vast potential for future development. Future research will focus on developing deep learning techniques, plant leaf disease datasets of various plant leaf diseases, and plant leaf disease segmentation techniques. To accomplish this, we can emphasize the following ideas in the future:

1. Chapters 3, 4, and 5 mentioned that a particular crop's image sequences were used to train and evaluate deep-learning models. The study will be expanded to incorporate a single identification model for leaf disease recognition across multiple crop species. Moreover, it could also be extended to real-time plant leaf disease recognition, which requires developing online learning techniques.

2. Currently, we are working on the plant leaf disease detection of three cops. Still, future research should be extended to classify the other important crops cultivated in Punjab, such

as corn, wheat, rice, tomato, chili, onion, and various vegetables.

3. Regardless of the findings of this research, more research work is required on plant leaf disease recognition, not only for the crops cultivated in Punjab, this research will also be extended to recognize the plant leaf diseases of other crops being produced in different regions of Pakistan.

4. Thus, the focus of future research should be the recognition of multiple diseases on a single leaf recognition of the physiological disease and pests diseases.

Appendix A

Dataset Authentication, Acceptance Letter, Proof of Submission, & Research Statistics

A.1 Dataset Authentication Letters

Here we have dataset authentication letters. To ensure the validation of our self-created datasets, we authenticated our datasets from Department of Plant Pathology, Pir Mehar Ali Shah ARID Agriculture University, Rawalpindi. Some of the datasets are publicly available on Kaggle and remaining datasets will be public after publication of articles. These letters are presented in sequence below.

1. The Potato Leaf Dataset (PLD) authentication letter.

2. The Guava Patches Dataset (GPD) authentication letter.

3. The Plant Species Dataset (PLSD) authentication letter.

4. The Guava Leaf Diseases Dataset (GLDD) authentication letter.

PIR MEHR ALI SHAH

ARID AGRICULTURE UNIVERSITY RAWALPINDI

Department of Plant Pathology

TO WHOM IT MAY CONCERN

The Potato Leaf Dataset (PLD) made for the digital recognition of Potato Blight diseases (detail under given) from field sown Coroda, Mozika and Sante varieties of Potato in the District Okara (Central Punjab) region of Pakistan was created and utilized by Mr. Javed Rashid (PhD Scholar in Computer Science), Registration No. 155-FBAS/PHDCS/F16 at the Department of Computer Science & Software Engineering, International Islamic University, Islamabad. Following labeled classes of infected/non-infected were determined by using the deep learning approach.

Class Labels	Samples
Early Blight	1628
Late Blight	1414
Healthy	1020
Total Samples	**4062**

The Potato Leaf Dataset (PLD) images were submitted for authentication of the dataset. The Potato Leaf Dataset (PLD) is hereby validated as aforementioned classes.

Prof. Dr. Tariq Mukhtar
Chairman

Chairman
Department of Plant Pathology
PMAS-Arid Agriculture University Rawalpindi

PIR MEHR ALI SHAH
ARID AGRICULTURE UNIVERSITY RAWALPINDI
Department of Plant Pathology

No. PMAS-AAUR/PP/_____ Dated: ____________

TO WHOM IT MAY CONCERN

The Guava Patches Dataset (GPD) made for the digital recognition of Infected patches (detail under given) from field sown Choti Surahi, Bari Surahi, Gola, Golden and Sadabahar varieties of Guava in the District Okara (Central Punjab) region of Pakistan was created and utilized by Mr. Javed Rashid (PhD Scholar in Computer Science). Registration No. 155-FBAS/PHDCS/F16 at the Department of Computer Science & Software Engineering, International Islamic University, Islamabad. Following labeled classes of infected/non-infected were determined by using the deep learning approach.

Class Labels	Samples
Infected Patches	1196
Total Samples	**1196**

The Guava Patches Dataset (GPD) images were submitted for authentication of the dataset. The Guava Patches Dataset (GPD) is hereby validated as aforementioned classes.

(Dr. Traiq Mukhtar)
Chairman
CHAIRMAN
Department of Plant Pathology
PMAS-Arid Agriculture University Rawalpindi

PIR MEHR ALI SHAH
ARID AGRICULTURE UNIVERSITY RAWALPINDI
Department of Plant Pathology

No. PMAS-AAUR/PP/_____

Dated: ____________

TO WHOM IT MAY CONCERN

The Plant Species Dataset (PLSD) made for the digital recognition of Guava, Java Plum, and Potato leaf species (detail under given) from field in the District Okara (Central Punjab) region of Pakistan was created and utilized by Mr. Javed Rashid (PhD Scholar in Computer Science), Registration No. 155-FBAS/PHDCS/F16 at the Department of Computer Science & Software Engineering, International Islamic University, Islamabad. Following labeled classes of infected/non-infected were determined by using the deep learning approach.

Class Labels	Samples
Guava Leaf Species	1900
Potato Leaf Specie.	1900
Java Plum Leaf Species	1880
Total Samples	**5680**

The Plant Species Dataset (PLSD) images were submitted for authentication of the dataset. The Plant Species Dataset (PLSD) is hereby validated as aforementioned classes.

(Dr. Traiq Mukhtar)
Chairman

PIR MEHR ALI SHAH
ARID AGRICULTURE UNIVERSITY RAWALPINDI
Department of Plant Pathology

No. PMAS-AAUR/PP/____ Dated: __________

TO WHOM IT MAY CONCERN

The Guava Leaf Diseases Dataset (GLDD) made for the digital recognition of nutrient deficiency, anthracnose, wilt diseases, insect attack, and healthy (detail under given) from field sown Choti Surahi, Bari Surahi, Gola, Golden and Sadabahar varieties of Guava ir the District Okara (Central Punjab) region of Pakistan was created and utilized by Mr. Javed Rashid (PhD Scholar in Computer Science) Registration No. 155-FBAS/PHDCS/F16 at the Department of Computer Science & Software Engineering, International Islamic University Islamabad. Following labeled classes of infected/non-infected were determined by using the deep learning approach

Class Labels	Samples
Anthracnose	2286
Nutrient deficiency	1419
Wilt	509
Insert attack	1532
Healthy	495
Total Samples	6241

The Guava Leaf Diseases Dataset (GLDD) images were submitted for authentication of the dataset. The Guava Leaf Diseases Dataset (GLDD) is hereby validated as aforementioned classes.

(Dr. Traiq Mukhtar)
Chairman

CHAIRMAN
Department of Plant Pathology
PMAS-Arid Agriculture University, Rawalpindi

Bibliography

[1] J. Hasell and M. Roser, "Famines," *Our World in Data*, 2020, https://ourworldindata.org/famines.

[2] M. E. Chowdhury, A. Khandakar, S. Ahmed, F. Al-Khuzaei, J. Hamdalla, F. Haque, M. B. I. Reaz, A. Al Shafei, and N. Al-Emadi, "Design, construction and testing of iot based automated indoor vertical hydroponics farming test-bed in qatar," *Sensors*, vol. 20, no. 19, p. 5637, 2020.

[3] E.-C. Oerke, "Crop losses to pests," *The Journal of Agricultural Science*, vol. 144, no. 1, pp. 31–43, 2006.

[4] R. Yadav, Y. Kumar Rana, and S. Nagpal, "Plant leaf disease detection and classification using particle swarm optimization," in *International Conference on Machine Learning for Networking*. Springer, 2018, pp. 294–306.

[5] 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, 2018.

[6] Å. Bergman, J. Heindel, S. Jobling, K. Kidd, and R. Zoeller, "Who (world health organization)/unep (united nations environment programme): State of the science of endocrine disrupting chemicals-2012," *Geneva, UNEP/WHO*, p. 296, 2013.

[7] Y. Guo, X. Li, Z. Zhao, H. Wei, B. Gao, and W. Gu, "Prediction of the potential geographic distribution of the ectomycorrhizal mushroom tricholoma matsutake under multiple climate change scenarios," *Scientific reports*, vol. 7, no. 1, pp. 1–11, 2017.

[8] K. Zaheer and M. H. Akhtar, "Potato production, usage, and nutritiona review," *Critical reviews in food science and nutrition*, vol. 56, no. 5, pp. 711–721, 2016.

[9] A. Devaux, P. Kromann, and O. Ortiz, "Potatoes for sustainable global food security," *Potato Research*, vol. 57, no. 3, pp. 185–199, 2014.

[10] GoP, "Economic survey of pakistan 2013-14," *Economic Advisors Wing. Ministry of Finance, Government of Pakistan, Islamabad*, 2014.

[11] M. Ali, "Dynamics of vegetable production, distribution and consumption in asia," AVRDC, Tech. Rep., 2000.

[12] K. Farooq, A. Mubarik, and Y. Aqsa, "Potato cluster feasibility and transformation study," *2020). Cluster Development Based Agriculture Transformation Plan Vision-2025. Project*, no. 131, p. 434, 2020.

[13] Anonymous, "Crops area and production, districts wise, 2020-21," *MINISTRY OF NATIONAL FOOD SECURITY & RESEARCH*, 2020.

[14] M. Usman, W. A. Samad, B. Fatima, and M. H. Shah, "Pollen parent enhances fruit size and quality in intervarietal crosses in guava (psidium guajava)," *International Journal of Agriculture and Biology*, vol. 15, no. 1, 2013.

[15] A. Kareem, M. J. Jaskani, B. Fatima, and B. Sadia, "Clonal multiplication of guava through softwood cuttings under mist conditions," *Pak. J. Agri. Sci*, vol. 50, no. 1, pp. 23–27, 2013.

[16] K. Sanda, H. Grema, Y. Geidam, and Y. Bukar-Kolo, "Pharmacological aspects of psidium guajava: An update," *International Journal of Pharmacology*, vol. 7, no. 3, pp. 316–324, 2011.

[17] P. Officer, "Food and agriculture organization of the united nations," *FAO, Italy*, 2016.

[18] A. Mehmood, M. J. Jaskani, I. A. Khan, S. Ahmad, R. Ahmad, S. Luo, and N. M. Ahmad, "Genetic diversity of pakistani guava (psidium guajava l.) germplasm and its implications for conservation and breeding," *Scientia Horticulturae*, vol. 172, pp. 221–232, 2014.

[19] A. Mehmood, M. J. Jaskani, S. Ahmad, and R. Ahmad, "Evaluation of genetic diversity in open pollinated guava by ipbs primers." *Pakistan Journal of Agricultural Sciences*, vol. 50, no. 4, 2013.

[20] Anonymous, "Agriculture statistics year book, 2011," *Ministry of National Food Security and Research, Islamabad, Pakistan*, 2011.

[21] A. Kelman, M. C. Shurtleff, M. J. Pelczar, and R. M. Pelczar, *plant disease*

— *Importance, Types, Transmission, & Control*, 2019. [Online]. Available: https://www.britannica.com/science/plant-disease

[22] Y. Sasaki, T. Okamoto, K. Imou, and T. Torii, "Automatic diagnosis of plant disease-spectral reflectance of healthy and diseased leaves," *IFAC Proceedings Volumes*, vol. 31, no. 5, pp. 145–150, 1998.

[23] J. M. Henson and R. French, "The polymerase chain reaction and plant disease diagnosis," *Annual review of phytopathology*, vol. 31, no. 1, pp. 81–109, 1993.

[24] C. Koo, M. Malapi-Wight, H. S. Kim, O. S. Cifci, V. L. Vaughn-Diaz, B. Ma, S. Kim, H. Abdel-Raziq, K. Ong, Y.-K. Jo *et al.*, "Development of a real-time microchip pcr system for portable plant disease diagnosis," *PloS one*, vol. 8, no. 12, 2013.

[25] A. Kaplan and M. Haenlein, "Siri, siri, in my hand: Whos the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence," *Business Horizons*, vol. 62, no. 1, pp. 15–25, 2019.

[26] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[29] D. Sinwar, V. S. Dhaka, M. K. Sharma, and G. Rani, "Ai-based yield prediction and smart irrigation," in *Internet of Things and Analytics for Agriculture, Volume 2*. Springer, 2020, pp. 155–180.

[30] D. N. Patel, S. L. Joshi, and V. Ravikumar, "Agriculture monitoring system using iot: A survey," in *Soft Computing: Theories and Applications*. Springer, 2020, pp. 631–648.

[31] A. dos Santos Ferreira, D. M. Freitas, G. G. da Silva, H. Pistori, and M. T. Folhes, "Weed detection in soybean crops using convnets," *Computers and Electronics in Agriculture*, vol. 143, pp. 314–324, 2017.

[32] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, "An in-field automatic wheat disease diagnosis system," *Computers and electronics in agriculture*, vol. 142, pp. 369–379, 2017.

[33] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.

[34] S. Desai, R. Nayak, and R. Patel, "Identifying plant diseases using deep convolutional neural networks," in *Recent Advances in Communication Infrastructure*. Springer, 2020, pp. 95–104.

[35] Y. Liu, C. Cen, Y. Che, R. Ke, Y. Ma, and Y. Ma, "Detection of maize tassels from uav rgb imagery with faster r-cnn," *Remote Sensing*, vol. 12, no. 2, p. 338, 2020.

[36] R. N. Strange and P. R. Scott, "Plant disease: a threat to global food security," *Annu. Rev. Phytopathol.*, vol. 43, pp. 83–116, 2005.

[37] D. Zhai, R. Shi, J. Jiang, and X. Liu, "Rectified meta-learning from noisy labels for robust image-based plant disease classification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 18, no. 1s, pp. 1–17, 2022.

[38] B. Labatut, "Fao regional office for latin america and the caribbean," *Food and Agriculture Organization of the United Nations*, 2017.

[39] GoP, "Population projections for the year 2007-2030," *Ministry of Planning, Development and Reforms, Government of Pakistan, Islamabad*, 2016.

[40] Plantix. [Online]. Available: https://www.plantix.net/en/

[41] P. Kumar, V. K. Gupta, A. K. Tiwari, and M. Kamle, *Current Trends in Plant Disease Diagnostics and Management Practices*. Springer, 2016.

[42] T. T. N. Nguyen, V. Le, T. Le, V. Hai, N. Pantuwong, and Y. Yagi, "Flower species identification using deep convolutional neural networks," in *AUN/SEED-Net Regional Conference for Computer and Information Engineering*, 2016.

[43] I. Gogul and V. S. Kumar, "Flower species recognition system using convolution neural networks and transfer learning," in *2017 fourth international conference on signal processing, communication and networking (ICSCN)*. IEEE, 2017, pp. 1–6.

[44] Q. Xiao, G. Li, L. Xie, and Q. Chen, "Real-world plant species identification based on deep convolutional neural networks and visual attention," *Ecological Informatics*, vol. 48, pp. 117–124, 2018.

[45] C. S. Pereira, R. Morais, and M. J. Reis, "Deep learning techniques for grape plant species identification in natural images," *Sensors*, vol. 19, no. 22, p. 4850, 2019.

[46] M. Lasseck, "Image-based plant species identification with deep convolutional neural networks." in *CLEF (Working Notes)*, 2017.

[47] N. Van Hieu and N. L. H. Hien, "Recognition of plant species using deep convolutional feature extraction," *Int J Emerg Technol*, 2020.

[48] S. Kaur and P. Kaur, "Plant species identification based on plant leaf using computer vision and machine learning techniques," *Journal of Multimedia Information System*, vol. 6, no. 2, pp. 49–60, 2019.

[49] D. Bisen, "Deep convolutional neural network based plant species recognition through features of leaf," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 6443–6456, 2021.

[50] J. Wei Tan, S.-W. Chang, S. Abdul-Kareem, H. J. Yap, and K.-T. Yong, "Deep learning for plant species classification using leaf vein morphometric," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 17, no. 1, pp. 82–90, 2018.

[51] N. Van Hieu and N. L. H. Hien, "Automatic plant image identification of vietnamese species using deep learning models," *arXiv preprint arXiv:2005.02832*, 2020.

[52] D. Bambil, H. Pistori, F. Bao, V. Weber, F. M. Alves, E. G. Gonçalves, L. F. de Alencar Figueiredo, U. G. Abreu, R. Arruda, and I. M. Bortolotto, "Plant species identification using color learning resources, shape, texture, through machine learning and artificial neural networks," *Environment Systems and Decisions*, vol. 40, no. 4, pp. 480–484, 2020.

[53] A. J. Hati and R. R. Singh, "Artificial intelligence in smart farms: plant phenotyping for species recognition and health condition identification using deep learning," *AI*, vol. 2, no. 2, pp. 274–289, 2021.

[54] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, "Leafnet: A computer vision system for automatic plant species identification," *Ecological Informatics*, vol. 40, pp. 50–56, 2017.

[55] V. M. Araújo, A. S. Britto Jr, L. S. Oliveira, and A. L. Koerich, "Two-view fine-grained classification of plant species," *Neurocomputing*, vol. 467, pp. 427–441, 2022.

[56] S. Mahmudul Hassan and A. Kumar Maji, "Identification of plant species using deep learning," in *Proceedings of International Conference on Frontiers in Computing and Systems*. Springer, 2021, pp. 115–125.

[57] U. Erkan, A. Toktas, and D. Ustun, "Hyperparameter optimization of deep cnn classifier for plant species identification using artificial bee colony algorithm," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2022.

[58] S. A. Pearline and V. S. Kumar, "Performance analysis of real-time plant species recognition using bilateral network combined with machine learning classifier," *Ecological Informatics*, vol. 67, p. 101492, 2022.

[59] K. Nagasubramanian, S. Jones, A. K. Singh, S. Sarkar, A. Singh, and B. Ganapathysubramanian, "Plant disease identification using explainable 3d deep learning on hyperspectral images," *Plant Methods*, vol. 15, no. 1, p. 98, 2019.

[60] Q. Wu, K. Zhang, and J. Meng, "Identification of soybean leaf diseases via deep learning," *Journal of The Institution of Engineers (India): Series A*, pp. 1–8, 2019.

[61] S. B. Jadhav, V. R. Udupi, and S. B. Patil, "Identification of plant diseases using convolutional neural networks," *International Journal of Information Technology*, pp. 1–10, 2020.

[62] A. Karlekar and A. Seal, "Soynet: Soybean leaf diseases classification," *Computers and Electronics in Agriculture*, vol. 172, p. 105342, 2020.

[63] J. G. A. Barbedo, L. V. Koenigkan, B. A. Halfeld-Vieira, R. V. Costa, K. L. Nechet, C. V. Godoy, M. L. Junior, F. R. A. Patricio, V. Talamini, L. G. Chitarra *et al.*, "Annotated plant pathology databases for image-based detection and recognition of diseases," *IEEE Latin America Transactions*, vol. 16, no. 6, pp. 1749–1757, 2018.

[64] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi, "Cucumber leaf disease identification with global pooling dilated convolutional neural network," *Computers and Electronics in Agriculture*, vol. 162, pp. 422–430, 2019.

[65] J. Kianat, M. A. Khan, M. Sharif, T. Akram, A. Rehman, and T. Saba, "A joint framework of feature reduction and robust feature selection for cucumber leaf diseases recognition," *Optik*, vol. 240, p. 166566, 2021.

[66] M. A. Khan, A. Alqahtani, A. Khan, S. Alsubai, A. Binbusayyis, M. M. I. Ch, H.-S. Yong, and J. Cha, "Cucumber leaf diseases recognition using multi level deep entropy-elm feature selection," *Applied Sciences*, vol. 12, no. 2, p. 593, 2022.

[67] K. Zhang, Z. Xu, S. Dong, C. Cen, and Q. Wu, "Identification of peach leaf disease infected

by xanthomonas campestris with deep learning," *Engineering in Agriculture, Environment and Food*, 2019.

[68] S. Yadav, N. Sengar, A. Singh, A. Singh, and M. K. Dutta, "Identification of disease using deep learning and evaluation of bacteriosis in peach leaf," *Ecological Informatics*, vol. 61, p. 101247, 2021.

[69] P. Bedi and P. Gole, "Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network," *Artificial Intelligence in Agriculture*, vol. 5, pp. 90–101, 2021.

[70] 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, 2017.

[71] W. Guoxiong Zhou, A. Zhang, and C. M. He, "Rapid detection of rice disease based on fcm-km and faster r-cnn fusion," *IEEE Access*, vol. 7, 2019.

[72] S. Ramesh and D. Vydeki, "Recognition and classification of paddy leaf diseases using optimized deep neural network with jaya algorithm," *Information Processing in Agriculture*, 2019.

[73] M. A. Azim, M. K. Islam, M. M. Rahman, F. Jahan *et al.*, "An effective feature extraction method for rice leaf disease classification," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 463–470, 2021.

[74] J. Chen, D. Zhang, A. Zeb, and Y. A. Nanehkaran, "Identification of rice plant diseases using lightweight attention networks," *Expert Systems with Applications*, vol. 169, p. 114514, 2021.

[75] N. Krishnamoorthy, L. N. Prasad, C. P. Kumar, B. Subedi, H. B. Abraha, and V. Sathishkumar, "Rice leaf diseases prediction using deep neural networks with transfer learning," *Environmental Research*, vol. 198, p. 111275, 2021.

[76] S. K. Upadhyay and A. Kumar, "A novel approach for rice plant diseases classification with deep convolutional neural network," *International Journal of Information Technology*, vol. 14, no. 1, pp. 185–199, 2022.

[77] K. Archana, S. Srinivasan, S. P. Bharathi, R. Balamurugan, T. Prabakar, and A. Britto, "A novel method to improve computational and classification performance of rice plant disease identification," *The Journal of Supercomputing*, pp. 1–21, 2022.

[78] M. Sharma, C. J. Kumar, and A. Deka, "Early diagnosis of rice plant disease using machine learning techniques," *Archives of Phytopathology and Plant Protection*, vol. 55, no. 3, pp. 259–283, 2022.

[79] R. R. Patil and S. Kumar, "Rice-fusion: A multimodality data fusion framework for rice disease diagnosis," *IEEE Access*, 2022.

[80] W. Bao, J. Zhao, G. Hu, D. Zhang, L. Huang, and D. Liang, "Identification of wheat leaf diseases and their severity based on elliptical-maximum margin criterion metric learning," *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100526, 2021.

[81] Z. Jiang, Z. Dong, W. Jiang, and Y. Yang, "Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning," *Computers and Electronics in Agriculture*, vol. 186, p. 106184, 2021.

[82] J. G. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosystems engineering*, vol. 172, pp. 84–91, 2018.

[83] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access*, vol. 6, pp. 30 370–30 377, 2018.

[84] R. A. Priyadharshini, S. Arivazhagan, M. Arun, and A. Mirnalini, "Maize leaf disease classification using deep convolutional neural networks," *Neural Computing and Applications*, pp. 1–9, 2019.

[85] W. Setiawan, A. Ghofur, F. H. Rachman, and R. Rulaningtyas, "Deep convolutional neural network alexnet and squeezenet for maize leaf diseases image classification," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 2021.

[86] S.-q. PAN, J.-f. QIAO, W. Rui, H.-l. YU, W. Cheng, K. TAYLOR, and H.-y. PAN, "Intelligent diagnosis of northern corn leaf blight with deep learning model," *Journal of Integrative Agriculture*, vol. 21, no. 4, pp. 1094–1105, 2022.

[87] Z. Li, G. Zhou, Y. Hu, A. Chen, C. Lu, M. He, Y. Hu, and Y. Wang, "Maize leaf disease identification based on wg-marnet," *Plos one*, vol. 17, no. 4, p. e0267650, 2022.

[88] 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. 59 069–59 080, 2019.

[89] L. Li, S. Zhang, and B. Wang, "Apple leaf disease identification with a small and imbalanced dataset based on lightweight convolutional networks," *Sensors*, vol. 22, no. 1, p. 173, 2021.

[90] H. Sun, H. Xu, B. Liu, D. He, J. He, H. Zhang, and N. Geng, "Mean-ssd: A novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 189, p. 106379, 2021.

[91] Z. u. Rehman, M. A. Khan, F. Ahmed, R. Damaševičius, S. R. Naqvi, W. Nisar, and K. Javed, "Recognizing apple leaf diseases using a novel parallel real-time processing framework based on mask rcnn and transfer learning: An application for smart agriculture," *IET Image Processing*, 2021.

[92] M. G. Selvaraj, A. Vergara, H. Ruiz, N. Safari, S. Elayabalan, W. Ocimati, and G. Blomme, "Ai-powered banana diseases and pest detection," *Plant Methods*, vol. 15, no. 1, p. 92, 2019.

[93] V. G. Krishnan, J. Deepa, P. V. Rao, V. Divya, and S. Kaviarasan, "An automated segmentation and classification model for banana leaf disease detection," *Journal of Applied Biology & Biotechnology Vol*, vol. 10, no. 01, pp. 213–220, 2022.

[94] M. A. Khan, T. Akram, M. Sharif, M. Awais, K. Javed, H. Ali, and T. Saba, "Ccdf: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep cnn features," *Computers and electronics in agriculture*, vol. 155, pp. 220–236, 2018.

[95] S. Singh, M. Bergerman, G. Hoheisel, K. Lewis, and T. Baugher, "Comprehensive automation for specialty crops (casc)–developing a more sustainable and profitable us specialty crop industry," 2011.

[96] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia computer science*, vol. 133, pp. 1040–1047, 2018.

[97] T. R. Gadekallu, D. S. Rajput, M. Reddy, K. Lakshmanna, S. Bhattacharya, S. Singh, A. Jolfaei, and M. Alazab, "A novel pca–whale optimization-based deep neural network model for classification of tomato plant diseases using gpu," *Journal of Real-Time Image Processing*, vol. 18, no. 4, pp. 1383--1396, 2021.

[98] S. Ashwinkumar, S. Rajagopal, V. Manimaran, and B. Jegajothi, "Automated plant leaf disease detection and classification using optimal mobilenet based convolutional neural networks," *Materials Today: Proceedings*, vol. 51, pp. 480–487, 2022.

[99] R. Thangaraj, S. Anandamurugan, and V. K. Kaliappan, "Automated tomato leaf disease classification using transfer learning-based deep convolution neural network," *Journal of Plant Diseases and Protection*, vol. 128, no. 1, pp. 73–86, 2021.

[100] M. E. Chowdhury, T. Rahman, A. Khandakar, M. A. Ayari, A. U. Khan, M. S. Khan, N. Al-Emadi, M. B. I. Reaz, M. T. Islam, and S. H. M. Ali, "Automatic and reliable leaf disease detection using deep learning techniques," *AgriEngineering*, vol. 3, no. 2, pp. 294–312, 2021.

[101] Y. Altuntaş and F. KOCAMAZ, "Deep feature extraction for detection of tomato plant diseases and pests based on leaf images," *Celal Bayar University Journal of Science*, vol. 17, no. 2, pp. 145–157, 2021.

[102] A. Sembiring, Y. Away, F. Arnia, and R. Muharar, "Development of concise convolutional neural network for tomato plant disease classification based on leaf images," in *Journal of Physics: Conference Series*, vol. 1845, no. 1. IOP Publishing, 2021, p. 012009.

[103] A. A. Gomaa and Y. El-Latif, "Early prediction of plant diseases using cnn and gans," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, 2021.

[104] A. W. Muhamad, Y. A. Jasim, M. I. S. Windi, M. G. Saeed, and S. D. AbdulAmeer, "High-performance deep learning to detection and tracking tomato plant leaf predict disease and expert systems," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 10, no. 2, pp. 97–122, 2021.

[105] S. Nandhini and K. Ashokkumar, "Improved crossover based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18 583–18 610, 2021.

[106] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with c-gan synthetic images," *Computers and Electronics in Agriculture*, vol. 187, p. 106279, 2021.

[107] A. Bhujel, N.-E. Kim, E. Arulmozhi, J. K. Basak, and H.-T. Kim, "A lightweight attention-

based convolutional neural networks for tomato leaf disease classification," *Agriculture*, vol. 12, no. 2, p. 228, 2022.

[108] H.-C. Chen, A. M. Widodo, A. Wisnujati, M. Rahaman, J. C.-W. Lin, L. Chen, and C.-E. Weng, "Alexnet convolutional neural network for disease detection and classification of tomato leaf," *Electronics*, vol. 11, no. 6, p. 951, 2022.

[109] T. Vadivel and R. Suguna, "Automatic recognition of tomato leaf disease using fast enhanced learning with image processing," *Acta Agriculturae Scandinavica, Section BSoil & Plant Science*, vol. 72, no. 1, pp. 312–324, 2022.

[110] E. Cengil and A. Çınar, "Hybrid convolutional neural network based classification of bacterial, viral, and fungal diseases on tomato leaf images," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, p. e6617, 2022.

[111] H. Tarek, H. Aly, S. Eisa, and M. Abul-Soud, "Optimized deep learning algorithms for tomato leaf disease detection with hardware deployment," *Electronics*, vol. 11, no. 1, p. 140, 2022.

[112] M. S. Al-gaashani, F. Shang, M. S. Muthanna, M. Khayyat, and A. A. Abd El-Latif, "Tomato leaf disease classification by exploiting transfer learning and feature concatenation," *IET Image Processing*, 2022.

[113] B. Gokulnath *et al.*, "Identifying and classifying plant disease using resilient lf-cnn," *Ecological Informatics*, vol. 63, p. 101283, 2021.

[114] S. Zhao, Y. Peng, J. Liu, and S. Wu, "Tomato leaf disease diagnosis based on improved convolution neural network by attention module," *Agriculture*, vol. 11, no. 7, p. 651, 2021.

[115] G. Hu, H. Wu, Y. Zhang, and M. Wan, "A low shot learning method for tea leaf's disease identification," *Computers and Electronics in Agriculture*, vol. 163, p. 104852, 2019.

[116] G. Hu, H. Wang, Y. Zhang, and M. Wan, "Detection and severity analysis of tea leaf blight based on deep learning," *Computers & Electrical Engineering*, vol. 90, p. 107023, 2021.

[117] L. X. B. Sorte, C. T. Ferraz, F. Fambrini, R. dos Reis Goulart, and J. H. Saito, "Coffee leaf disease recognition based on deep learning and texture attributes," *Procedia Computer Science*, vol. 159, pp. 135–144, 2019.

[118] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector

for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2017.

[119] T. Maenpaa, "The local binary pattern approach to texture analysis: Extensions and applications." 2004.

[120] J. G. Esgario, P. B. de Castro, L. M. Tassis, and R. A. Krohling, "An app to assist farmers in the identification of diseases and pests of coffee leaves using deep learning," *Information Processing in Agriculture*, vol. 9, no. 1, pp. 38–47, 2022.

[121] R. I. Hasan, S. M. Yusuf, M. S. M. Rahim, and L. Alzubaidi, "Automated masks generation for coffee and apple leaf infected with single or multiple diseases-based color analysis approaches," *Informatics in Medicine Unlocked*, vol. 28, p. 100837, 2022.

[122] S. K. Noon, M. Amjad, M. Ali Qureshi, and A. Mannan, "Computationally light deep learning framework to recognize cotton leaf diseases," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–16, 2021.

[123] A. R. Luaibi, T. M. Salman, and A. H. Miry, "Detection of citrus leaf diseases using a deep learning technique," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, p. 1719, 2021.

[124] R. Sujatha, J. M. Chatterjee, N. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocessors and Microsystems*, vol. 80, p. 103615, 2021.

[125] M. Ji, L. Zhang, and Q. Wu, "Automatic grape leaf diseases identification via unitedmodel based on multiple convolutional neural networks," *Information Processing in Agriculture*, 2019.

[126] X. Xie, Y. Ma, B. Liu, J. He, S. Li, and H. Wang, "A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks," *Frontiers in plant science*, vol. 11, p. 751, 2020.

[127] P. Kaur, S. Harnal, R. Tiwari, S. Upadhyay, S. Bhatia, A. Mashat, and A. M. Alabdali, "Recognition of leaf disease using hybrid convolutional neural network by applying feature reduction," *Sensors*, vol. 22, no. 2, p. 575, 2022.

[128] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image clas-

sification using convolutional neural networks vgg," *Artificial Intelligence in Agriculture*, 2022.

[129] J. Chen, J. Chen, D. Zhang, Y. A. Nanehkaran, and Y. Sun, "A cognitive vision method for the detection of plant disease images," *Machine Vision and Applications*, vol. 32, no. 1, pp. 1–18, 2021.

[130] H. Qi, Y. Liang, Q. Ding, and J. Zou, "Automatic identification of peanut-leaf diseases based on stack ensemble," *Applied Sciences*, vol. 11, no. 4, p. 1950, 2021.

[131] S. Uğuz and N. Uysal, "Classification of olive leaf diseases using deep convolutional neural networks," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4133–4149, 2021.

[132] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease," *IEEE Access*, vol. 7, pp. 43 721–43 729, 2019.

[133] J. Yao, Y. Wang, Y. Xiang, J. Yang, Y. Zhu, X. Li, S. Li, J. Zhang, and G. Gong, "Two-stage detection algorithm for kiwifruit leaf diseases based on deep learning," *Plants*, vol. 11, no. 6, p. 768, 2022.

[134] V. Ravi, V. Acharya, and T. D. Pham, "Attention deep learning-based large-scale learning classifier for cassava leaf disease classification," *Expert Systems*, vol. 39, no. 2, p. e12862, 2022.

[135] J. Anitha and N. Saranya, "Cassava leaf disease identification and detection using deep learning approach," *International Journal of Computers, Communications and Control*, vol. 17, no. 2, 2022.

[136] U. K. Lilhore, A. L. Imoize, C.-C. Lee, S. Simaiya, S. K. Pani, N. Goyal, A. Kumar, and C.-T. Li, "Enhanced convolutional neural network model for cassava leaf disease identification and classification," *Mathematics*, vol. 10, no. 4, p. 580, 2022.

[137] E. Elfatimi, R. Eryigit, and L. Elfatimi, "Beans leaf diseases classification using mobilenet models," *IEEE Access*, 2022.

[138] R. Ganesh Babu and C. Chellaswamy, "Different stages of disease detection in squash plant based on machine learning," *Journal of Biosciences*, vol. 47, no. 1, pp. 1–14, 2022.

[139] S. Ibrahim, N. Hasan, N. Sabri, K. A. F. Abu Samah, and M. Rahimi Rusland, "Palm leaf nu-

trient deficiency detection using convolutional neural network (cnn)," *International Journal of Nonlinear Analysis and Applications*, vol. 13, no. 1, pp. 1949–1956, 2022.

[140] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosystems Engineering*, vol. 180, pp. 96–107, 2019.

[141] G. Geetharamani and A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Computers & Electrical Engineering*, vol. 76, pp. 323–338, 2019.

[142] K. Kamal, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Computers and Electronics in Agriculture*, vol. 165, p. 104948, 2019.

[143] A. Khamparia, G. Saini, D. Gupta, A. Khanna, S. Tiwari, and V. H. C. de Albuquerque, "Seasonal crops disease prediction and classification using deep convolutional encoder network," *Circuits, Systems, and Signal Processing*, pp. 1–19, 2019.

[144] A. A. Ahmed and G. H. Reddy, "A mobile-based system for detecting plant leaf diseases using deep learning," *AgriEngineering*, vol. 3, no. 3, pp. 478–493, 2021.

[145] S. Thenmozhi, R. J. Lakshmi, I. Ibrahim, and R. Mohan, "A novel plant leaf ailment recognition method using image processing algorithms," 2021.

[146] H. Kushal, K. Swathi, K. Nithyusha, A. Lavanya *et al.*, "Agridoc: Classification and prediction of plant leaf diseases," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 14, pp. 3909–3932, 2021.

[147] G. G. AR, R. Kannadasan, M. H. Alsharif, A. Jahid, and M. A. Khan, "Categorizing diseases from leaf images using a hybrid learning model," *Symmetry*, vol. 13, no. 11, p. 2073, 2021.

[148] S. A. Wagle *et al.*, "Comparison of plant leaf classification using modified alexnet and support vector machine." *Traitement du Signal*, vol. 38, no. 1, 2021.

[149] V. Tiwari, R. C. Joshi, and M. K. Dutta, "Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images," *Ecological Informatics*, vol. 63, p. 101289, 2021.

[150] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of plant-leaf diseases using cnn and transfer-learning approach," *Electronics*, vol. 10, no. 12, p. 1388, 2021.

[151] N. Ahmad, H. M. S. Asif, G. Saleem, M. U. Younus, S. Anwar, and M. R. Anjum, "Leaf image-based plant disease identification using color and texture features," *Wireless Personal Communications*, vol. 121, no. 2, pp. 1139–1168, 2021.

[152] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using efficientnet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, 2021.

[153] N. Kaur *et al.*, "Plant leaf disease detection using ensemble classification and feature extraction," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 11, pp. 2339–2352, 2021.

[154] 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.

[155] Y. Li and X. Chao, "Semi-supervised few-shot learning approach for plant diseases recognition," *Plant Methods*, vol. 17, no. 1, pp. 1–10, 2021.

[156] Y. He, Q. Gao, and Z. Ma, "A crop leaf disease image recognition method based on bilinear residual networks," *Mathematical Problems in Engineering*, vol. 2022, 2022.

[157] W. Albattah, M. Nawaz, A. Javed, M. Masood, and S. Albahli, "A novel deep learning method for detection and classification of plant diseases," *Complex & Intelligent Systems*, pp. 1–18, 2021.

[158] S. S. Gaikwad, S. S. Rumma, and M. Hangarge, "Fungi affected fruit leaf disease classification using deep cnn architecture," *International Journal of Information Technology*, pp. 1–10, 2022.

[159] A. K. Singh, S. Sreenivasu, U. Mahalaxmi, H. Sharma, D. D. Patil, and E. Asenso, "Hybrid feature-based disease detection in plant leaf using convolutional neural network, bayesian optimized svm, and random forest classifier," *Journal of Food Quality*, vol. 2022, 2022.

[160] B. Wang, "Identification of crop diseases and insect pests based on deep learning," *Scientific Programming*, vol. 2022, 2022.

[161] D. Hughes, M. Salathé *et al.*, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[162] N. E. M. Khalifa, M. H. N. Taha, L. M. Abou El-Maged, and A. E. Hassanien, "Artificial intelligence in potato leaf disease classification: A deep learning approach," in *Ma-*

chine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges. Springer, 2021, pp. 63–79.

[163] A. J. Rozaqi and A. Sunyoto, "Identification of disease in potato leaves using convolutional neural network (cnn) algorithm," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT).* IEEE, 2020, pp. 72–76.

[164] K. Sanjeev, N. K. Gupta, W. Jeberson, and S. Paswan, "Early prediction of potato leaf diseases using ann classifier," *Oriental Journal of Computer Science and Technology*, vol. 13, no. 2, 3 and 4, 2020.

[165] U. Barman, D. Sahu, G. G. Barman, and J. Das, "Comparative assessment of deep learning to detect the leaf diseases of potato based on data augmentation," in *2020 International Conference on Computational Performance Evaluation (ComPE).* IEEE, 2020, pp. 682–687.

[166] D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel, and S. Bhardwaj, "Potato leaf diseases detection using deep learning," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS).* IEEE, 2020, pp. 461–466.

[167] T.-Y. Lee, J.-Y. Yu, Y.-C. Chang, and J.-M. Yang, "Health detection for potato leaf with convolutional neural network," in *2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN).* IEEE, 2020, pp. 289–293.

[168] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, "Detection of potato diseases using image segmentation and multiclass support vector machine," in *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE).* IEEE, 2017, pp. 1–4.

[169] S. Ali, M. Hassan, J. Y. Kim, M. I. Farid, M. Sanaullah, and H. Mufti, "Ff-pca-lda: Intelligent feature fusion based pca-lda classification system for plant leaf diseases," *Applied Sciences*, vol. 12, no. 7, p. 3514, 2022.

[170] Q. Liang, S. Xiang, Y. Hu, G. Coppola, D. Zhang, and W. Sun, "Pd2se-net: Computer-assisted plant disease diagnosis and severity estimation network," *Computers and electronics in agriculture*, vol. 157, pp. 518–529, 2019.

[171] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional networks," in *Proceedings of the Conference Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[172] M. R. Howlader, U. Habiba, R. H. Faisal, and M. M. Rahman, "Automatic recognition of guava leaf diseases using deep convolution neural network," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2019, pp. 1–5.

[173] A. F. Al Haque, R. Hafiz, M. A. Hakim, and G. R. Islam, "A computer vision system for guava disease detection and recommend curative solution using deep learning approach," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2019, pp. 1–6.

[174] A. Almadhor, H. T. Rauf, M. I. U. Lali, R. Damaševičius, B. Alouffi, and A. Alharbi, "Ai-driven framework for recognition of guava plant diseases through machine learning from dslr camera sensor based high resolution imagery," *Sensors*, vol. 21, no. 11, p. 3830, 2021.

[175] P. Perumal *et al.*, "Guava leaf disease classification using support vector machine," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 7, pp. 1177–1183, 2021.

[176] X.-F. Wang, D.-S. Huang, J.-X. Du, H. Xu, and L. Heutte, "Classification of plant leaf images with complicated background," *Applied mathematics and computation*, vol. 205, no. 2, pp. 916–926, 2008.

[177] N. Valliammal and S. Geethalakshmi, "Plant leaf segmentation using non linear k means clustering," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 3, p. 212, 2012.

[178] L.-H. Ma, Z.-Q. Zhao, and J. Wang, "Apleafis: an android-based plant leaf identification system," in *International Conference on Intelligent Computing*. Springer, 2013, pp. 106–111.

[179] Z.-Q. Zhao, L.-H. Ma, Y.-m. Cheung, X. Wu, Y. Tang, and C. L. P. Chen, "Apleaf: An efficient android-based plant leaf identification system," *Neurocomputing*, vol. 151, pp. 1112–1119, 2015.

[180] I. K. Raji and K. Thyagharajan, "An analysis of segmentation techniques to identify herbal leaves from complex background," *Procedia Computer Science*, vol. 48, pp. 589–599, 2015.

[181] K. Itakura and F. Hosoi, "Automatic leaf segmentation for estimating leaf area and leaf inclination angle in 3d plant images," *Sensors*, vol. 18, no. 10, p. 3576, 2018.

[182] D. Ward, P. Moghadam, and N. Hudson, "Deep leaf segmentation using synthetic data," *arXiv preprint arXiv:1807.10931*, 2018.

[183] D. Kuznichov, A. Zvirin, Y. Honen, and R. Kimmel, "Data augmentation for leaf segmentation and counting tasks in rosette plants," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019, pp. 0–0.

[184] J. P. Kumar and S. Domnic, "Image based leaf segmentation and counting in rosette plants," *Information Processing in Agriculture*, vol. 6, no. 2, pp. 233–246, 2019.

[185] J. Bell and H. M. Dee, "Leaf segmentation through the classification of edges," *arXiv preprint arXiv:1904.03124*, 2019.

[186] J. Gimenez-Gallego, J. D. Gonzalez-Teruel, M. Jimenez-Buendia, A. B. Toledo-Moreo, F. Soto-Valles, and R. Torres-Sanchez, "Segmentation of multiple tree leaves pictures with natural backgrounds using deep learning for image-based agriculture applications," *Applied Sciences*, vol. 10, no. 1, p. 202, 2019.

[187] D. Li, G. Shi, W. Kong, S. Wang, and Y. Chen, "A leaf segmentation and phenotypic feature extraction framework for multiview stereo plant point clouds," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2321–2336, 2020.

[188] J. Praveen Kumar and S. Domnic, "Rosette plant segmentation with leaf count using orthog~~onal transform and deep convolutional neural network," *Machine Vision and Applications*,~~ vol. 31, no. 1, pp. 1–14, 2020.

[189] K. Yang, W. Zhong, and F. Li, "Leaf segmentation and classification with a complicated background using deep learning," *Agronomy*, vol. 10, no. 11, p. 1721, 2020.

[190] N. Nikbakhsh, Y. Baleghi, and H. Agahi, "A novel approach for unsupervised image segmentation fusion of plant leaves based on g-mutual information," *Machine Vision and Applications*, vol. 32, no. 1, pp. 1–12, 2021.

[191] Z. M. Amean, T. Low, and N. Hancock, "Automatic leaf segmentation and overlapping leaf separation using stereo vision," *Array*, vol. 12, p. 100099, 2021.

[192] M. A. Putra. (2020) Potato leaf disease dataset. Kaggle. [Online]. Available: https://www.kaggle.com/datasets/muhammadardiputra/potato-leaf-disease-dataset

[193] N. TILAHUN, "Developing potatos leaves disease detection model using convolutional neu-

ral network," Ph.D. dissertation, ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY, 2020.

[194] H. T. Rauf and M. I. U. Lali, "A guava fruits and leaves dataset for detection and classification of guava diseases through machine learning," *Mendeley Data*, vol. 1, 2021.

[195] O. M. Dalvi. (2022) Guava disease dataset. Kaggle. [Online]. Available: https://www.kaggle.com/datasets/omkarmanohardalvi/guava-disease-dataset-4-types

[196] M. Šulc and J. Matas, "Fine-grained recognition of plants from images," *Plant Methods*, vol. 13, no. 1, pp. 1–14, 2017.

[197] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in *2015 IEEE international conference on image processing (ICIP)*. IEEE, 2015, pp. 452–456.

[198] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planqué, A. Rauber, S. Palazzo, B. Fisher *et al.*, "Lifeclef 2015: multimedia life species identification challenges," in *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 2015, pp. 462–483.

[199] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *European conference on computer vision*. Springer, 2012, pp. 502–516.

[200] O. Söderkvist, "Computer vision classification of leaves from swedish trees," 2001.

[201] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, "Performance improvement of leaf identification system using principal component analysis," *International Journal of Advanced Science and Technology*, vol. 44, no. 11, pp. 113–124, 2012.

[202] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *2007 IEEE international symposium on signal processing and information technology*. IEEE, 2007, pp. 11–16.

[203] A. Krizhevsky and I. Sutskever, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.

[204] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmen-

tation for sar target recognition," *IEEE Geoscience and remote sensing letters*, vol. 13, no. 3, pp. 364–368, 2016.

[205] G. Ali, T. Ali, M. Irfan, U. Draz, M. Sohail, A. Glowacz, M. Sulowicz, R. Mielnik, Z. B. Faheem, and C. Martis, "Iot based smart parking system using deep long short memory network," *Electronics*, vol. 9, no. 10, p. 1696, 2020.

[206] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[207] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[208] L. Holzbauer. (2019, December) Convolutional neural networks explained with american ninja warrior. Insight. [Online]. Available: https://blog.insightdatascience.com/convolutional-neural-networks-explained-with-american-ninja-warrior-c6649875861c

[209] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE international symposium on circuits and systems*. IEEE, 2010, pp. 253–256.

[210] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[211] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, pp. 611–629, 2018.

[212] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[213] B. Coppin, *Artificial intelligence illuminated.* Jones & Bartlett Learning, 2004.

[214] C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," in *Artificial intelligence and statistics*. PMLR, 2016, pp. 464–472.

[215] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[216] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz,

and D. Menotti, "A robust real-time automatic license plate recognition based on the yolo detector," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–10.

[217] D. Sun, L. Liu, A. Zheng, B. Jiang, and B. Luo, "Visual cognition inspired vehicle re-identification via correlative sparse ranking with multi-view deep features," in *International conference on brain inspired cognitive systems*. Springer, 2018, pp. 54–63.

[218] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5561–5569.

[219] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, "A real-time detection algorithm for kiwifruit defects based on yolov5," *Electronics*, vol. 10, no. 14, p. 1711, 2021.

[220] S. Sun, M. Jiang, N. Liang, D. He, Y. Long, H. Song, and Z. Zhou, "Combining an information-maximization-based attention mechanism and illumination invariance theory for the recognition of green apples in natural scenes," *Multimedia Tools and Applications*, vol. 79, no. 37, pp. 28 301–28 327, 2020.

[221] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Computers and electronics in agriculture*, vol. 158, pp. 20–29, 2019.

[222] K. Willis, "State of the worlds plants 2017 report. royal botanic gardens, kew, richmond, surrey, 100 pp," 2017.

[223] S. Y. Arafat, M. I. Saghir, M. Ishtiaq, and U. Bashir, "Comparison of techniques for leaf classification," in *2016 Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*. IEEE, 2016, pp. 136–141.

[224] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[225] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[226] H. Yang and P. A. Bath, "The use of data mining methods for the prediction of dementia:

evidence from the english longitudinal study of aging," *IEEE journal of biomedical and health informatics*, vol. 24, no. 2, pp. 345–353, 2019.

[227] M. Wang, H. Wang, J. Wang, H. Liu, R. Lu, T. Duan, X. Gong, S. Feng, Y. Liu, Z. Cui *et al.*, "A novel model for malaria prediction based on ensemble algorithms," *PloS one*, vol. 14, no. 12, p. e0226910, 2019.

[228] E. L. Ray and N. G. Reich, "Prediction of infectious disease epidemics via weighted density ensembles," *PLoS computational biology*, vol. 14, no. 2, p. e1005910, 2018.

[229] T. K. Yamana, S. Kandula, and J. Shaman, "Superensemble forecasts of dengue outbreaks," *Journal of The Royal Society Interface*, vol. 13, no. 123, p. 20160410, 2016.

[230] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[231] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[232] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[233] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[234] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[235] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and D. Erhan, "Going deeper with convolutions (googlelenet)," *Journal of Chemical Technology and Biotechnology*, 2016.

[236] S. Safari, A. Baratloo, M. Elfil, and A. Negida, "Evidence based emergency medicine; part 5 receiver operating curve and area under the curve," *Emergency*, vol. 4, no. 2, p. 111, 2016.

[237] M. Anas, S. S. Roy, K. S. Srivastava, and J. Chakraborty, "Plant diseases classification using neural network: Alexnet," in *Deep Learning Applications in Image Analysis*. Springer, 2023, pp. 133–147.

[238] M. T. S. Abirami, "Application of image processing in diagnosing guava leaf diseases," *International Journal of Scientific Research and Management*, vol. 5, no. 7, pp. 5927–5933, 2017.

[239] A. G. Awan and G. Yaseen, "Global climate change and its impact on agriculture sector in pakistan," *American Journal of Trade and Policy*, vol. 4, no. 1, pp. 41–48, 2017.

[240] K. Song, X. Sun, and J. Ji, "Corn leaf disease recognition based on support vector machine method," *Transactions of the CSAE*, vol. 23, no. 1, pp. 155–157, 2007.

[241] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar, "Counting apples and oranges with deep learning: A data-driven approach," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 781–788, 2017.

[242] P. A. Dias, A. Tabb, and H. Medeiros, "Multispecies fruit flower detection using a refined semantic segmentation network," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3003–3010, 2018.

[243] J. Ubbens, M. Cieslak, P. Prusinkiewicz, and I. Stavness, "The use of plant models in deep learning: an application to leaf counting in rosette plants," *Plant methods*, vol. 14, no. 1, p. 6, 2018.

[244] J. Rashid, M. Ishfaq, G. Ali, M. R. Saeed, M. Hussain, T. Alkhalifah, F. Alturise, and N. Samand, "Skin cancer disease detection using transfer learning technique," *Applied Sciences*, vol. 12, no. 11, p. 5714, 2022.

[245] M. Sohail, G. Ali, J. Rashid, I. Ahmad, S. H. Almotiri, M. A. AlGhamdi, A. A. Nagra, and K. Masood, "Racial identity-aware facial expression recognition using deep convolutional neural networks," *Applied Sciences*, vol. 12, no. 1, p. 88, 2022.

[246] J. Rashid, I. Khan, G. Ali, S. H. Almotiri, M. A. AlGhamdi, and K. Masood, "Multi-level deep learning model for potato leaf disease recognition," *Electronics*, vol. 10, no. 17, p. 2064, 2021.

[247] H. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 861–870, 2022.

[248] ——, "Employing lexicalized dependency paths for active learning of relation extraction,"

INTELLIGENT AUTOMATION AND SOFT COMPUTING, vol. 34, no. 3, pp. 1415–1423, 2022.

[249] A. M. Roy and J. Bhaduri, "A deep learning enabled multi-class plant disease detection model based on computer vision," *AI*, vol. 2, no. 3, pp. 413–428, 2021.

[250] K. Wada, "labelme: Image Polygonal Annotation with Python," https://github.com/wkentaro/labelme, 2016.

[251] J. Rashid, I. Khan, G. Ali, S. u. Rehman, A. Shahid, Fahad, and T. Alkahlifah. (2021) Guava patches dataset (gpd). [Online]. Available: https://drive.google.com/drive/folders/1XSjsAlk5BRHPEhrlVs9cLBDZzXgNDx8l?usp=sharing

[252] ——. (2021) Guava leaf diseases dataset (gldd). [Online]. Available: https://drive.google.com/drive/folders/1vpgYn54j034ELwwu_5NuvMd0fjwQhXOr?usp=sharing

[253] Roboflow. (2021) Robflow. 2021 Roboflow, Inc. [Online]. Available: https://app.roboflow.com/

[254] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[255] M. Ji, L. Zhang, and Q. Wu, "Automatic grape leaf diseases identification via unitedmodel based on multiple convolutional neural networks," *Information Processing in Agriculture*, vol. 7, no. 3, pp. 418–426, 2020.

[256] X. Wang, J. Wei, Y. Liu, J. Li, Z. Zhang, J. Chen, and B. Jiang, "Research on morphological detection of fr i and fr ii radio galaxies based on improved yolov5," *Universe*, vol. 7, no. 7, p. 211, 2021.

[257] M. Kasper-Eulaers, N. Hahn, S. Berger, T. Sebulonsen, Ø. Myrland, and P. E. Kummervold, "Detecting heavy goods vehicles in rest areas in winter conditions using yolov5," *Algorithms*, vol. 14, no. 4, p. 114, 2021.

[258] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[259] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[260] S. Zhang, Y. Wu, C. Men, and X. Li, "Tiny yolo optimization oriented bus passenger object detection," *Chinese Journal of Electronics*, vol. 29, no. 1, pp. 132–138, 2020.

[261] J. Rashid, I. Khan, G. Ali, S. H. Almotiri, M. A. AlGhamdi, and K. Masood, "Potato leaf disease (pld) dataset," https://www.kaggle.com/datasets/rizwan123456789/potato-disease-leaf-datasetpld, June 2021.

[262] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[263] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[264] W. W. Lo, X. Yang, and Y. Wang, "An xception convolutional neural network for malware classification with transfer learning," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2019, pp. 1–5.