Potato Disease Detection Using MultiNet: A Deep Neural Network with Multi-Scale Feature Fusion

This project is submitted to the Department of Computer Science and Engineering, Dhaka International University, in partial fulfillment to the requirements of Bachelor of Science (B. Sc.) in Computer Science and Engineering (CSE).

Submitted by

NAME	REG. NO	ROLL NO
Tariqul Islam	CS-D-58-19-111689	07
Md. Raihanur Rahman	CS-D-58-19-111724	08
Rumon Islam	CS-D-58-19-111750	12
Sumsun Nahar	CS-D-58-19-112001	45
Dola Saha	CS-D-58-19-112004	46

Project Work, CSE-425
Batch: 58 (1st Shift), Session (2019-2020)

Supervised by

Md. Rezaul Islam
Lecturer
Dept. of Computer Science & Engineering
Dhaka International University

Knowledge is Power
DIU
Estd. 7th April, 1995

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF SCIENCE AND ENGINEERING
DHAKA INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH

FEBRUARY-2024

SUPERVISOR'S STATEMENT

This is to certify that the project paper entitled as **"Potato Disease Detection Using MultiNet: A Deep Neural Network with Multi-Scale Feature Fusion"** submitted by Tariqul Islam, Roll No:- 07; Md. Raihanur Rahman, Roll No:- 08; Rumon Islam, Roll No:- 12; Sumsun Nahar, Roll No:- 45; Dola Saha, Roll No:- 46; has been carried out under my supervision. This project has been prepared in partial fulfillment of the requirement for the Degree of B.Sc. in Computer Science & Engineering, Department of Computer Science & Engineering, Dhaka International University, Dhaka, Bangladesh.

Supervisor's Signature

Date: …………………

..........…...…………………………………………

Md. Rezaul Islam
Lecturer
Dept. of Computer Science & Engineering
Dhaka International University

APPROVAL

The project report as **"Potato Disease Detection Using MultiNet: A Deep Neural Network with Multi-Scale Feature Fusion"** submitted by Tariqul Islam, Md Raihanur Rahman, Rumon Islam, Sumsun Nahar, and Dola Saha to the Department of Computer Science & Engineering, Dhaka International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents.

Board of Honorable Examiners

1. Chairman ....

Prof. Dr. A. T. M. Mahbubur Rahman
Dean,
Faculty of Science and Engineering,
and Chairman,
Dept. of Computer Science and Engineering,
Dhaka International University

2. Internal Member ....

Prof. Md. Abdul Based
Additional Director,
Research and publication Cell
Dhaka International University

3. Internal Member ....

Mst. Jahanara Akhtar
Professor
Dept. of Computer Science and Engineering,
Dhaka International University

4. Supervisor and Member ....

Md. Rezaul Islam
Lecturer,
Dept. of Computer Science and Engineering,
Dhaka International University

5. External Member ....

Dr. Md. Manowarul Islam
Associate Professor,
Dept. of Computer Science and Engineering,
Jagannath University

DECLARATION

We hereby declare that; this project has been carried out by us and it has been submitted for the award of the B.Sc. degree. We also certify that this project was prepared by us for the purpose of fulfillment of the requirements for the Bachelor of Science (B.Sc.) in Computer Science & Engineering.

Authors Signature

....
Tariqul Islam
B.Sc. In CSE, Roll No: 07
Reg. No: CS-D-58-19-111689
Batch:58(1st shift), Session: 2019-2020
Dhaka International University

....
Md. Raihanur Rahman
B.Sc. In CSE, Roll No: 08
Reg. No: CS-D-58-19-111724
Batch:58(1st shift), Session: 2019-2020
Dhaka International University

....
Rumon Islam
B.Sc. In CSE, Roll No: 12
Reg. No: CS-D-58-19-111750
Batch:58(1st shift), Session: 2019-2020
Dhaka International University

....
Sumsun Nahar
B.Sc. In CSE, Roll No: 45
Reg. No: CS-D-58-19-112001
Batch:58(1st shift), Session: 2019-2020
Dhaka International University

....
Dola Saha
B.Sc. In CSE, Roll No: 46
Reg. No: CS-D-58-19-112004
Batch:58(1st shift), Session: 2019-2020
Dhaka International University

Supervisor's Signature

Date: ……………

....
Md. Rezaul Islam
Lecturer
Dept. of Computer Science and Engineering
Dhaka International University

ABSTRACT

Potatoes hold significant agricultural importance in Bangladesh, ranking third after rice and wheat, and contributing substantially to the nation's economy. Despite Bangladesh being a major global potato producer, leaf diseases pose considerable challenges to growth and quality. Timely disease detection is crucial for boosting productivity and advancing agricultural digitization. This study aims to employ machine-learning algorithms on a limited dataset of leaf images for disease detection. Utilizing deep learning and machine learning techniques, the research focuses on categorizing potato leaves into distinct disease groups using "The Plant Village Dataset." A novel infrastructure called "MultiNet" is introduced, leveraging transfer learning to effectively categorize various potato leaf diseases. "MultiNet" enables rapid and accurate diagnostics through two multi-class classifications: (1) Early Blight, Late Blight, and Healthy; and (2) Early Blight Normal, Early Blight Serious, Late Blight, Healthy, Insect infected, LeafRoll Virus infected, and Virus infected. The system integrates three pre-trained models—ResNet50, DenseNet-201, and VGG16—to extract features from images, followed by concatenation to create a hybrid structure. Results demonstrate an impressive overall classification accuracy of 99.83% for three classes and 98% for seven classes. These outcomes highlight the potential impact and utility of the "MultiNet" framework in agriculture, suggesting promising applications within the sector.

ACKNOWLEDGEMENTS

We would like to pay our gratitude to the Almighty Allah who created us with not only the ability to design and program this system but also the power of practice.

We would also like to express our sincere thanks to our respected supervisor **Md. Rezaul Islam, Lecturer**, Department of CSE, Dhaka International University for his continuous encouragement, motivation and professional guidance during the work of this project. We would like to thank all the faculty members for their valuable time spent on requirements analysis and evaluation of the project work.

We would like to express our sincere and cordial gratitude to the people those who have supported us directly, provided mental encouragement. We are also thankful to our family and friends who have contributed directly or indirectly the development word and its associated activities.

We warmly thank **Prof. Dr. A. T. M. Mahbubur Rahman,** Dean, Faculty of Science and Engineering and Chairman, Department of Computer Science and Engineering and **Prof. Md. Abdul Based,** Additional Director, Research and Publication cell, Dhaka International University for their valuable advice and moral support. Their extensive discussion around work and interesting exploration in operations has been very helpful for this study.

Finally, we would like to dedicate this project to our teachers for their love, encouragement and professional guidance throughout the project.

DEDICATION

Dedicated to:

Our Parents

&

Teachers

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

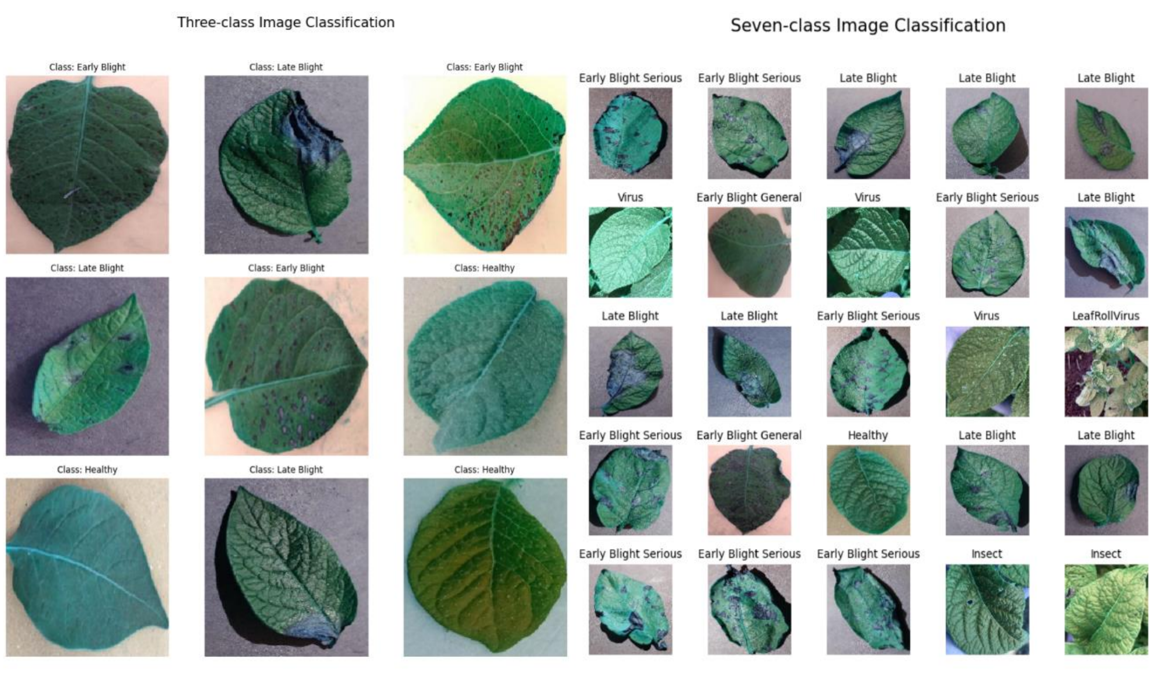
Chapter 1 INTRODUCTION

1.1 OVERVIEW

The potato (scientifically known as Solanum tuberosum L.) stands as the world's most widely cultivated crop, playing a pivotal role in addressing nutritional requirements as the third-largest contributor to the global food supply [1, 2]. While Bangladesh is predominantly associated with rice consumption, it also showcases a substantial cultivation and consumption of potatoes annually, with their popularity steadily on the rise. However, plant diseases pose frequent threats to crop yield, exerting significant economic repercussions by reducing both producer and distributor income and driving up consumer prices. Prior to harvest, potatoes undergo disease assessment at the outset of their growth cycle.

1.2 COMMON POTATO DISEASES IN BANGLADESH

Various ailments can afflict potato crops, manifesting in different parts of the plant's leaves. Noteworthy among these are foliage issues, early blight, and late blight. Early blight, instigated by the fungal agent Alternaria Solani, and late blight, provoked by the bacterium Phytophthora infestans (2002), pose significant fungal hazards that can disrupt potato cultivation and affect national finances. Additionally, Potato Leaf Roll (PLR), activated by Polerovirus, leads to the curling of potato foliage, with further leaf impairment caused by various viruses and insects. **Figure 1.1** shows the sample images from three-class dataset and seven-class dataset.



.

Figure 1.1: Sample images from three-class and seven-class dataset

1.3 TRADITIONAL METHOD

Traditionally, farmers and local experts have relied on visual inspections of crops to optimize pesticide use and minimize production losses. However, this approach encounters challenges like time consumption, lack of expertise, and potential errors. To address these obstacles, an automated system that accurately identifies and categorizes damaged plant leaves is crucial. Plant diseases have typically been diagnosed through subjective visual estimation, leading to unpredictability and inaccuracies. Unlike these traditional techniques, scientists have developed a tool to identify healthy or diseased plant leaves by measuring the wavelengths of their spectrum of light [3]. Another approach is to use the Polymerase Chain Reaction method to extract the DNA from the leaves [4] or perform Polymerase Chain Reaction in real time [5]. However, implementing these strategies into practice is expensive and time-consuming since they require expert operation, certain experimental settings, and a large-scale application of agricultural safeguards for crops. Fortunately, advancements in machine learning (ML), artificial intelligence (AI), and computer vision (CV) have facilitated the development of automated diagnosis systems for plant diseases. These technologies provide efficient and precise identification of leaf diseases without human intervention. Notably, Deep Learning (DL) has particularly emerged as a potent tool in agriculture, transforming disease control and crop enhancement endeavors [6].

1.4 OUR PROPOSED MULTINET

This research introduces an automated system based on the 'MultiNet' framework for classifying potato diseases. The framework aims to classify leaf images into three primary classes (Early Blight, Late Blight, and Healthy) and seven sub-classes (Early Blight General, Early Blight Serious, Late Blight, Virus Infected, Insect Infected, LeafRollVirus Infected, and Healthy), offering a comprehensive solution for disease management in potato crops. We employ three pre-trained CNN models, namely ResNet-50, DenseNet-201, and VGG16, to extract features from the images. These extracted features are then combined using a concatenation layer. The fused features undergo fine-tuning, involving adjustments to dense layers, batch normalization layers, and dropout layers in the model. For the classification task, we utilize the ultimate dense layer with a softmax activation function.

1.5 ADVANTAGES OF OUR PROPOSED METHOD

This approach offers several significant advantages, which are outlined below:

- The 'MultiNet' framework employs feature extraction methods that bypass segmentation, thus obviating the necessity for manually engineered feature extraction techniques typically used in traditional machine learning approaches.
- The proposed infrastructure showcases impressive performance across both three-class and seven-class classifications of potato leaf images.
- Evaluation of the 'MultiNet' framework's performance extends to both large and small datasets, affirming its competence in achieving high performance across varying data scales.

1.6 STRUCTURE OF REMAINING CHAPTERS

This paper is structured as follows:

Section 2 presents the literature review. Following this, Section 3 provides a detailed explanation of the proposed 'MultiNet' framework, including discussions on data preprocessing and augmentation strategies. Section 4 extensively outlines the experimental setup and results. In Section 5, a comparative analysis of the results is conducted, along with exploration of potential future directions. Finally, Section 6 serves as the conclusion of the paper.

Chapter
2

LITERATURE REVIEW

2.1 OVERVIEW

Recent research has explored various methodologies for identifying and categorizing potato diseases, each showcasing notable achievements.

2.2 RELATED WORKS

In a previous study by Islam et al. [7], they utilized the support vector machine (SVM) algorithm for potato leaf segmentation in the potato village dataset. This method achieved a 95% accuracy in categorizing diseases across more than 300 photos. Despite concerns about potential output data loss with RGB imaging, this study focused on SVM and segmentation techniques. In another study by Hu YH et al. [8], Hyperspectral imaging features were applied to analyze potato leaves under late blight stress, using a dataset 60 leaves. Least Squares-Support Vector Machine (LS-SVM) models were developed using both raw and preprocessed data, exhibited optimal performance with a discrimination accuracy of 94.87%. Trishita A. et al. [9] achieved a 98% detection success using "Deep Learning" models, including Support Vector Machine (SVM) and Convolution Neural Network (CNN), on a trained sample dataset of "healthy and unhealthy" plant leaves. Tiwari et al. [10] used a previously trained VGG19 architecture for extracting features and used a variety of classification algorithms for classification, such as neural network models, k-nearest neighbor, and support vector machines. To distinguish between early and late blight illnesses in potato leaves, the model was trained using the Plant Village dataset. Significantly, no testing was done on untested data; instead, the model's performance, which yielded a 97.8% accuracy rate, was assessed only on the training set. Rabbia, M. et al. [11] introduced a system utilizing the efficient pre-trained DenseNet-201 architecture, which tackled class imbalance in the data by adjusting the reweighted cross-entropy loss function. Their model was evaluated on five classes: Early Blight, Late Blight, Healthy, Potato Leaf Roll, and Potato Verticillium_wilt, attaining an impressive accuracy of 97.2%. Meanwhile, Rashid, J. et al. [12] utilized the YOLOv3 image segmentation technique to extract features and developed a CNN model. They achieved a high accuracy of 99.75% in detecting early blight and late blight potato diseases. Chugh, G et al. [13] developed a CNN model utilizing the Inception V3 architecture and Adam Optimizer for the diagnosis and classification of diseases in potato plants, including early and late blight. They attained a classification accuracy of 90% over the test dataset, training the model on 2152 images sourced from the Plant Village Dataset. Iqbal A. et al. [14] conducted segmentation on 450 images depicting both healthy and diseased potato leaves sourced from the Plant Village database. They employed seven classifier algorithms to recognize and classify the leaves based on their health. Notably, the Random Forest classifier achieved an accuracy of 97%. In [15], Tarik M. et al. constructed a CNN model that underwent training on a dataset comprising more than 2034 images depicting unhealthy potatoes and leaves. The training encompassed seven disease types, including Early Blight, Potato Leaf Roll Virus, Hollow heart of potato, Scab of Potato, Soft rot of potato, Potato Tuber Worm, and Virus-related disease. Their findings underscored the effectiveness of CNN for object detection in this context, with the model achieving an impressive accuracy of

99.23%. Geetharamani, G. et al. [16] introduced a deep Convolutional Neural Network (CNN) model designed to distinguish between healthy and diseased leaves across various crops. Their training utilized the Plant Village dataset, encompassing 38 different crop types featuring images of both diseased and healthy leaves, as well as background images. However, the model's emphasis was not specifically on diseases of a single potato crop. Additionally, training on specific regional datasets for the USA and Switzerland proved ineffective in detecting potato leaf diseases in the Pakistani region. In [17], Kamal et al. presented Modified MobileNet and Reduced MobileNet models designed for identifying plant leaf diseases. These models incorporated Depth-wise Separable Convolutional Neural Network instead of the conventional Convolutional Neural Network by modifying the architecture of MobileNet. The models demonstrated an accuracy of 98.34%. In [18], Lee et al. created a Convolutional Neural Network model for identifying Early Blight, Late Blight potato leaves, and healthy potato leaves. The researchers utilized the Plant Village dataset specific to a particular region and did not conduct testing on data not previously encountered, achieving an accuracy of 99%. Barman et al. [19] introduced a Self-Build CNN (SBCNN) model designed for the detection of early blight, late blight, and healthy classes in potato leaf diseases. The model was trained using the Plant Village dataset tailored for a specific region. However, it is important to note that the model was not validated on unseen test data. Despite this, it achieved an accuracy of 96.75%. Lastly, Zhang et al. [20] developed a Faster RCNN model with 97.1% accuracy in detecting tomato diseases, with each technique utilizing the Plant Village dataset and aiming for increased precision in disease identification.

2.3 PROBLEM WITH PREVIOUS WORKS

To accurately identify potato leaf diseases, researchers create numerous study articles; however, they accomplish this task by employing broad approaches, limited information, and no fine-tuning. We attempt to create better projects by utilizing big data and fine-tuning to make our model more efficient.

Chapter 3 METHODOLOY

3.1 OVERVIEW

In this segment, the procedure for categorizing potato leaf images is delineated, employing a variety of multi-scale transfer learning models. **Figure 3.1** illustrates the 'MultiNet' framework devised specifically for this classification task. The framework initiates by loading images and retrieving corresponding labels from the dataset. Prior to dataset partitioning, various preprocessing techniques are applied, followed by extensive data augmentation to increase dataset size. Consequently, the model undergoes separate training on three classes of potato leaf sourced from Plant Village and seven classes from both Plant Village and diverse internet datasets. The assessment of the 'MultiNet' framework involves testing images. Further details regarding the components of this framework are provided in subsequent subsections

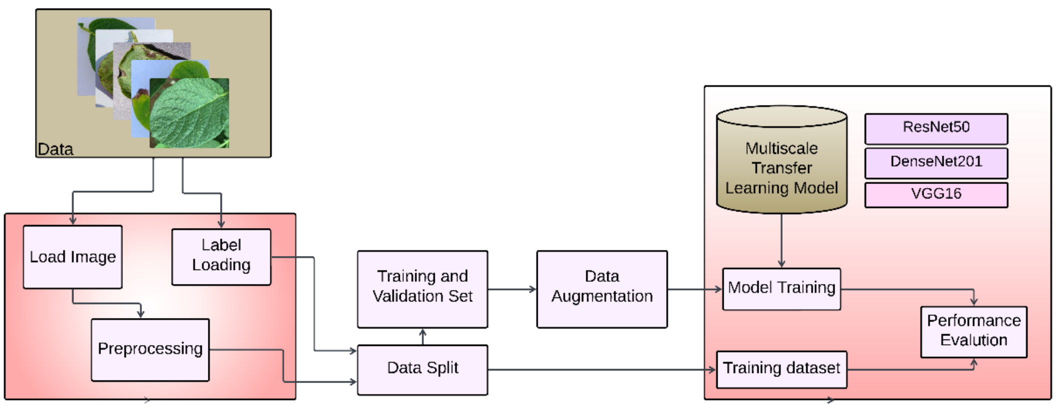


Figure 3.1: Suggested the "MultiNet" framework for classifying images of potato leaf diseases.

3.2 DATA ACQUISITION

In the initial phase, we gathered images of potato leaves from the publicly accessible Plant Village Dataset (Updated), which encompasses over 4000 images related to Potato Leaves diseases. From this dataset, we selected 1000 images each for three classes: Late Blight, Early Blight, and Healthy leaves. Expanding our dataset to seven classes, we incorporated four additional disease types. **Table 3.1** provide supplementary information about the three-class and seven-class datasets

Table 3.1: Further details regarding the two datasets prior to applying data augmentation.

Class	Name of Diseases	No of total Images	No. of train images	No. of test images
Three-class classification	Early Blight	1006	800	206
	Late Blight	1000	800	200
	Healthy	1005	800	205
Seven-class classification	Early Blight General	1006	800	206
	Early Blight Serious	1078	800	278
	Late Blight	1000	800	200
	Healthy	1005	800	205
	Insect	1016	810	206
	Potato Leaf Roll Virus	853	653	200
	Potato Leaf Virus	1000	800	200

Notably, a more severe form of Early Blight, with 1078 images, was distinguished from the regular Early Blight in the Plant Village Dataset (Updated). The remaining three disease types, including 1016 potato leaves affected by insects, 853 potato leaves infected by LeafRollVirus, and 1000 images of potato leaf virus, were manually collected from Google sources.

3.3 PREPROCESSING

Preprocessing data involves modifying raw data prior to feeding it into machine learning or deep learning algorithms. Directly training a convolutional neural network with raw images could lead to subpar classification performance; hence, preprocessing is vital for improving efficiency. This process accelerates training by implementing techniques like centering and scaling. Before inputting the images into the finely-tuned multiscale transfer learning model, several preprocessing steps have been implemented. The pre-processing methods employed in this context include the following:

3.3.1 Resizing

Before feeding the images into the fine-tuned multi-scale transfer learning model (refer to **Figure 3.1**), various preprocessing steps were executed. The images, initially in RGB format with three channels, each with an 8-bit depth, underwent adjustments. In the case of the three-class dataset, where the original images had dimensions of 700 x 460 pixels, we resized them to 224 x 224 pixels, in line with transfer learning principles. Similarly, for the seven-class dataset, the images were resized to 224 x 224 pixels, maintaining consistent dimensionality as per transfer learning principles. Additionally, to expedite training and minimize memory usage, each image was converted into a NumPy array.

3.3.2 Shuffling

Next, we undertook the process of image shuffling, a deliberate step intended to enhance the model's training efficiency by presenting data in a randomized order. This randomization serves to deter the model from discerning patterns solely based on the sequential organization of the data. Employing techniques tailored for mathematical contexts, we carefully manipulate sensitive data to safeguard operational metrics and computer-installed Key Performance Indicators (KPIs) throughout the database. As shuffling ensures the preservation of the integrity of every mathematical distribution, it facilitates the secure utilization of production data for essential tasks like testing and training.

3.3.3 Data Splitting

The datasets undergo a tripartite division. Initially, 70% of the data is dedicated to training, succeeding which emerges a testing phase. Ultimately, the remaining 30% is split evenly between testing and validation, completing the dataset allocation process.

3.3.4 Data Augmentation

Various data augmentation techniques serve multiple objectives, such as increasing dataset size, addressing overfitting concerns, and enhancing the model's overall robustness [21]. Following the application of data augmentation strategies (**Table 3.2**), the dataset size for the three-class dataset increased from 3011 to 21077, and for the seven-class dataset, it expanded from 6958 to 48706.

Table 3.2: Strategies for Augmenting Data with specified parameter values.

Techniques of Data Augmentation	Value of Parameter
Rotation range	90
Zoom range	2
Width shift range	.4
Height shift range	.4
Shearing range	.5
Vertical flip	True
Horizontal flip	True

3.3.5 Data Augmentation in Keras Library

The deep learning library, Keras, provides us the ability to augment the images automatically when training our model. In the Keras library, the ImageDataGenerator class provides a simple and easy way to augment our image data. The main advantage of using this ImageDataGenerator class is that it provides real-time data augmentation, which means it generates augmented images while the model is in the training stage and it requires lower memory usage as well. It is the responsibility of ImageDataGeneator class to ensure that the model receives new transformed images after each epoch and directly returns those transformed images to the model, instead of adding them to the original dataset. This is to make sure that the model does not overfit by seeing the original images multiple times. Different techniques from ImageDataGenerator that have been used in our research are given bellow:

3.3.5.1 Rotation Augmentation

The ImageDataGenerator class enables us to enhance an image by randomly rotating it clockwise by a specified number of degrees between 0 and 360 degrees. Here, we define the range of degrees for the rotation of our image datasets by setting an integer value of 90 in the rotation_range argument.

3.3.5.2 Zoom Augmentation

With this zoom augmentation approach, the original image is zoomed in, and the pixel values are either interpolated or additional pixels are added around the image. Here, we set the zoom_range argument's value to 2.

3.3.5.3 Shift Augmentation

The shift augmentation technique is the process of moving all the pixels of the image in one direction, such as vertically or horizontally while keeping the size and image dimensions the same. The ImageDataGenerator class has arguments such as height_shift_range for vertical shift and width_shift_range for the horizontal shift. In our research purpose we have used both shift range. We set the shift_range settings to 0.4, meaning that each image's height and width are shifted by 40%.

3.3.5.4 Shearing Augmentation

Shear Transformation involves distorting an image, distinguishing itself from rotation where the entire image rotates. Unlike rotation, shear focuses on fixing one axis while stretching the image at a specific angle, termed the shearing angle. This stretching effect is distinct from rotation. The shear_range parameter is a floating-point value that signifies the shearing angle in degrees, measured in a counter-clockwise direction. For our research purpose, we set the value for shear_range to 0.5

3.3.5.5 Flip Augmentation

Image flipping is an image augmentation technique that generates new images by reversing the rows and columns of image pixels in vertical and horizontal flips, respectively. We can flip an image along its vertical or horizontal axis using Keras' ImageDataGenerator class, which contains arguments like vertical_flip and horizontal_flip. In simple terms, a horizontal flip flips the rows and columns in a horizontal direction, whereas a vertical flip flips them in a vertical one. Therefore, in the ImageDataGenerator constructor, we have set the True argument for both vertical and horizontal flips. By default, the value for both arguments are false. **Table 3.3** provides supplementary details on the datasets after the implementation of the data augmentation techniques.

Table 3.3: Further details regarding two datasets after the application of data augmentation

Class	Name of Diseases	No of total Images	No. of train images	No. of test images
Three-class classification	Early Blight	7042	5600	1442
	Late Blight	7000	5600	1400
	Healthy	7035	5600	1435
Seven-class classification	Early Blight General	7042	5600	1442
	Early Blight Serious	7546	5600	1946
	Late Blight	7000	5600	1400
	Healthy	7035	5600	1435

Insect	7112	5670	1442
Potato Leaf Roll Virus	5971	4571	1400
Potato Leaf Virus	7000	5600	1400

3.4 MULTI-SCALE TRANSFER LEARNING MODEL

The creation of a transfer learning model that integrates multiple scales involves blending three renowned pre-trained CNN models: ResNet-50, DenseNet-201, and VGG16. This fusion represents a holistic approach to transfer learning, leveraging the unique strengths of each model to enhance overall performance. These models independently extract intricate details from images of potato leaves. Following this, the extracted features are merged via a fully connected layer to execute the classification task, as illustrated in **Figure 3.2**. Upon amalgamating all features, the proposed "MultiNet" framework encompasses 57,217,287 parameters, tripling the parameter count of individual architectures ResNet50, DenseNet-201, and VGG16. The subsequent sections delve into comprehensive explanations of both the foundational formation of each selected CNN model, which has undergone prior training, and the fine-tuning process.

3.4.1 ResNets - Residual Networks

The ResNet50 model, a variant of the ResNet (Residual Network) architecture, assumes a crucial role as the primary feature extractor in our proposed model. Originating from the work of Kaiming He and his team [22], ResNet50 stands distinguished for its depth, boasting 50 layers. Over the years, deep convolutional neural networks have spearheaded a series of breakthroughs in the realm of image recognition and classification. A prevailing trend involves delving deeper to tackle more intricate tasks and enhance classification or recognition accuracy. Nonetheless, training deeper neural networks has posed challenges, notably the vanishing gradient problem [23] and the degradation problem [24]. Residual learning is fundamentally about tackling challenges in deep neural network training. By introducing residual connections, data seamlessly transfers between layers, easing the complexities of training deep networks. This innovative architecture empowers the ResNet50 model to effectively capture complex patterns and hierarchical features in diverse datasets. ResNet50 stands out for its outstanding performance in image categorization tasks, making it a preferred choice in various computer vision applications. In our proposed model, ResNet50 serves as a reliable feature extractor, excelling in isolating and encoding essential features from input data. Its ability to discern intricate representations is attributed to its depth and streamlined connections, enhancing overall efficiency and resilience. Our dedication to utilizing cutting-edge architectures for superior feature extraction capabilities is highlighted by the adoption of ResNet50 in our proposed model.

3.4.2 DenseNet - Dense Convolutional Network

Gao Huang, Zhuang Liu, and Laurens van der Maaten are acknowledged for pioneering the DenseNet model, which was previously trained [25]. In 2017, this model achieved the highest classification accuracy across the ImageNet, CIFAR-10, and CIFAR-100 datasets. Unlike ResNet, DenseNet introduces a novel approach where each layer is directly connected to every other layer. This interconnected design facilitates seamless information sharing within the network, resulting in improved overall performance and training efficiency [26]. The cornerstone of this research's feature extraction lies in the pre-trained DenseNet-201 model. With its 201 deep Convolutional Neural Network (CNN) layers, DenseNet-201 is strategically arranged to alleviate overfitting concerns in smaller datasets. Additionally, by addressing challenges related to gradient descent, DenseNet-201 significantly enhances the ImageNet database [27]. Leveraging its multiple advanced CNN layers, DenseNet-201 is positioned to extract higher-level and crucial features compared to architectures such as GoogLeNet, AlexNet, and ResNet.

3.4.3 VGGNet - Visual Geometry Group Network

The VGG model, conceptualized by Andrew Zisserman and Karen Simonyan in 2013, emerged as a pioneering solution initially for the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) prototype in 2014. Affiliated with Oxford's Visual Geometry Group (VGG), this model garnered attention for its groundbreaking performance in image localization, clinching the top rank. On individual test scales, VGG showcased a top-1 error rate of 25.5% and a top-5 error rate of 8.0%. Across diverse test scales, VGG maintained a top-1 error rate of 24.8% and a top-5 error rate of 7.5%. Moreover, in the 2014 ImageNet competition, VGG secured the second position, impressively achieving a top-5 error rate of 7.3%, which further diminished to 6.8% post-submission [28]. In our current study, we adopt the VGG16 variation as our primary feature extractor. This variant comprises thirteen convolution layers and three fully interconnected layers, defining its architectural backbone. Employing a window dimension of (3×3) across all convolution layers, the model incorporates approximately 64 filters, expandable to 128 and eventually 256 filters based on availability. In later stages, up to 512 filtrations become feasible. Five max-pooling layers contribute to spatial pooling, strategically positioned after successive convolutional layers. The max-pooling operation employs a (2×2) pixel window with a stride size of 2 [29]. Within the VGG network, the ReLU activation function is uniformly applied to every hidden layer, underlining its foundational role in enhancing model performance.

3.5 OTHER STATE-OF-THE-ART MODELS

In order to demonstrate the effectiveness of our suggested "MultiNet" framework, we also train our datasets into a number of separate models, such as Xception, ResNet-50, DenseNet-201, MobileNetV2, NASNetMobile, VGG16, and VGG19. A short description remaining four models are given below:

3.5.1 VGG19

The deep convolutional neural network (CNN) model VGG-19 is part of the Visual Geometry Group (VGG) family of neural network models. It was created by professors at Oxford University, particularly those from the Visual Geometry Group, and is renowned for its understated yet exquisite architecture. The VGG-19 architecture, which has 19 layers total—16 convolutional layers and 3 fully linked layers—is an extension of the original VGG-16 architecture [30]. The input images that the network receives are fixed in size, usually 224 by 224 pixels. This network is made up of several convolutional blocks with several convolutional layers with tiny 3x3 filters inside of each block. A max-pooling layer for spatial downsampling and a Rectified Linear Unit (ReLU) activation function come after each convolutional block. The deeper the network, the more filters there are. There are completely connected layers for higher-level reasoning near the conclusion of the network. The last layer uses the softmax activation function to provide output probabilities for various classifications. The primary feature of VGG-19 is its uniform and simple architecture, which is achieved through the use of small 3x3 convolutional filters throughout. The depth of the network allows it to learn hierarchical features from low-level edges and textures to high-level object parts and concepts.

3.5.2 MobileNetV2

Google researchers just unveiled MobileNetV2 which has 53 deep layers. This is essentially an improvement on MobileNetV1, making it even more potent and efficient [31]. The pointwise convolution in V1 either doubled or maintained the number of channels. On the other hand, it reduces the number of channels in V2. Another important thing in the bottleneck residual block is a residual connection. It just works the same as in ResNet [22]. The main network, width multiplier 1, 224 x 224, requires 3.4 million parameters and 300 million multiply-add computations. The researchers investigated the performance trade-offs for input resolutions ranging from 96 to 224 and width multipliers of 0.35 to 1.4. The computational cost of the network varies from 7 billion additions to 585 million adds, while the model size spans from 1.7 million to 6.9 million parameters. The bottleneck residual block has three convolution layers. A 1 x 1 expansion layer makes up the initial layer. It increases the amount of data (or channels) that pass through it. The expansion factor determines how much of the data is extended. This hyperparameter can be obtained by weighing various trade-offs in architecture. Six is the expansion factor that is used by default. The depth-wise convolution layer is the second layer. The activation function is ReLU6. All that exists is min(max(x, 0), 6). With the exception of the project layer, every layer has a batch normalization layer and an activation function (ReLU6). The projection layer only uses batch normalization because the authors discovered that using ReLU6 to introduce nonlinearity will result in lower performance because the projection layer's output is low dimensional.

3.5.3 NASNetMobile

Artificial Neural Networks (ANN) use Neural Architecture Search (NAS) as the most recent DL approach. Three components make up this proposal from the Google Brain team from 2016: search space, search strategy, and performance estimation [32]. In search space, complex performances, fully-connectedness, max-pooling, and other features are sought after, and connections between the layers that create fully functional network architectures are then verified. In order to sample the population of network design candidates, the search strategy uses reinforcement learning and random search, with child model performance rewards (highest accuracy, time management). As this is going on, the primary goal of performance estimation is to minimize the amount of computational power or time that network design requires, allowing performance to be estimated at the search strategy position and resulting in child model performance rewards. [33, 34].

3.5.4 Xception

Xception is a deep convolutional neural network architecture based on Depthwise Separable Convolutions [35]. Researchers at Google developed it. In depthwise convolution, each filter processes a single channel of the input image, while in pointwise convolution, a 1x1 dimensional filter iterates over all input points. The three structures that make up the Xception architecture are the entry flow, middle flow, and exit flow. These three structures are made up of 14 modules total—4 modules, 8 modules, and 2 modules, respectively—that together include 36 convolution layers. All modules have residual connections, with the exception of the entrance flow's initial and last modules. The four modules that make up the entrance flow of the Xception architecture each have two convolution layers. The Xception architecture can be summarized as a linear stack of convolution layers that are separable depth-wise and have residual connections. This facilitates the definition and modification of the architecture.

3.6 FINE-TUNING

Figure 3.2 shows the integration of various transfer learning models utilizing distinct fully connected layers to categorize potato disease groups. After individually extracting features, each pre-trained model employs GlobalAveragePooling2D simultaneously to transform all layers into vector forms by computing the mean value of each input channel. Subsequently, the individual vectors are merged into a unified vector using the concatenate layer. Following this step, the concatenate layer is utilized to combine each of the distinct vectors into a single vector. Six layers are then employed in the subsequent stage to refine the integrated characteristics for our classification problem, ultimately leading to a softmax activation function. As emphasized in [36], overfitting presents a significant challenge for neural networks with deep layers, occurring when the model excessively fits to the training set, thus adversely affecting the performance of the test set. To address this issue, our approach incorporates two dropout layers. During the model development phase, 40% of the data is discarded in the first dropout layer, while 20% of the data is discarded in the second dropout layer. As indicated in [36], this approach not only helps mitigate overfitting but also significantly accelerates the training process.

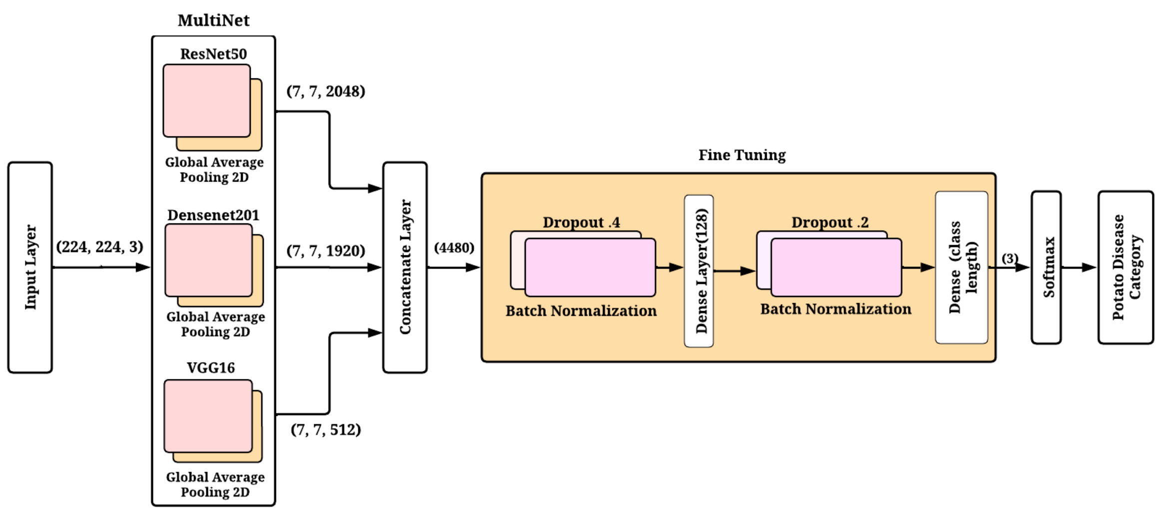


Figure 3.2: Adapted the "MultiNet" framework through fine-tuning.

Batch normalization and softmax are commonly used in tandem within deep learning architectures. BatchNorm is typically applied to the input of every layer except the output layer, thereby enhancing training dynamics. Softmax activation, on the other hand, is frequently employed in the output layer for multi-class classification tasks, facilitating the derivation of class probabilities. In our classification approach, the integration of two batch_normalization layers hold significant importance. These layers play a pivotal role in rescaling all data and ensuring consistent normalization throughout the network. By rescaling the data, the training process experiences a notable acceleration, reducing sensitivity to network initialization and significantly enhancing efficiency. In our methodology, a dense layer, functioning as a fully connected layer, establishes dense connections between neurons from both preceding and current layers. This layer meticulously processes input data, generating an output corresponding to the input. We integrate two dense layers in our approach, reserving the last layer specifically for classification purposes. To streamline the categorization process, we employ a softmax activation function, which calculates the likelihood of each possible outcome depending on the anticipated class's characteristics. Spanning from 0 to 1, the softmax activation function recognizes attributes typically linked with the anticipated class, thus facilitating neuron activation correspondingly. The mathematical representation of the softmax activation function is articulated as follows:

$$\text{softmax}\ (z_i) = \frac{\exp\ (z_i)}{\sum_{j=1}^{n} \exp\ (x_j)} \qquad (1)$$

Table 3.4 illustrates the results obtained through the incorporation of different transfer learning models with fully connected layers, a process conducted during the development phase of the "MultiNet" framework. Designed specifically for classifying three distinct categories, the framework's fully connected architecture culminates in a terminal layer consisting of three neurons. In cases of multiclass classification involving seven classes, the corresponding dense layer is equipped with seven neurons. It is noteworthy to emphasize the consistent and cohesive structure upheld by the envisioned "MultiNet" framework throughout its design. This consistency serves as a prominent characteristic of the proposed framework, irrespective of its application in either a multiclass scenario involving three classes or one encompassing seven classes.

Table 3.4: Summary of the 'MultiNet' Framework.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	[ ]
resnet50 (Functional)	(None, 7, 7, 2048)	23587712	['input_1[0][0]']
densenet201 (Functional)	(None, 7, 7, 1920)	18321984	['input_1[0][0]']
vgg16 (Functional)	(None, 7, 7, 512)	14714688	['input_1[0][0]']
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	['resnet50[0][0]']
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1920)	0	['densenet201[0][0]']
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0	['vgg16[0][0]']
concatenate (Concatenate)	(None, 4480)	0	['global_average_pooling2d[0][0]', 'global_average_pooling2d_1[0][0]', 'global_average_pooling2d_2[0][0]']
dropout (Dropout)	(None, 4480)	0	['concatenate[0][0]']
batch_normalization (BatchNormalization)	(None, 4480)	17920	['dropout[0][0]']
dense (Dense)	(None, 128)	573568	['batch_normalization[0][0]']
dropout_1 (Dropout)	(None, 128)	0	['dense[0][0]']
batch_normalization_1 (BatchNormalization)	(None, 128)	512	['dropout_1[0][0]']
dense_1 (Dense)	(None, 3)	387	['batch_normalization_1[0][0]']
Total params: 57216771 (218.26 MB) Trainable params: 56925379 (217.15 MB) Non-trainable params: 291392 (1.11 MB)			

Chapter 4 | Experimental Configuration, Evaluation Metrics and Results Analysis

4.1 OVERVIEW

This section details the experimental setup, including the configuration and hyperparameters used, and highlights the results obtained by leveraging the "MultiNet" framework obtained from "The Plant Village," in conjunction with manually collected datasets. Additionally, a comprehensive comparative analysis is undertaken to assess the proposed methodology against the distinct pre-trained architectures of individual Convolutional Neural Networks (CNNs).

Algorithm 4.1: Identification and Categorization of Potato Leaf Diseases

Input:

1. Potato Disease Training Set T1, Validation set V1, and Testing set T2
2. Learning rate (a)
3. Number of Epochs (b)
4. Batch size (c)
5. The total number of images that each batch covers (d)

   Output: w -> CNN pre-trained models' weight.

Start:

6. Transform individual image in the training set into dimensions 224 x 224.
7. Employ a data augmentation technique to increase the size of the dataset.
8. Retrieve features from the images by utilizing pre-existing CNN models such as ResNet50, DenseNet-201, and VGG16.
9. Merge the extracted characteristics by employing concatenate layers.
10. Configure the fine-tuning layers such as dense, batchnormalization, dropout, and softmax.
11. Initialize the parameters of the pre-trained CNN model as a, b, c, d.
12. Assess the proposed framework to determine and confirm the initial weights.
13. for each iteration from 1 to b:
    - Choose a batch of size d from T1.
    - Compute the loss function and carry out forward propagation.
    - Apply backpropagation to optimize and modify the weights indicated by 'w.'.
14. end for

4.1 EXPERIMENTAL CONFIGURATION

Following the proposal, the 'MultiNet' infrastructure was established leveraging Keras, an open-source software package facilitating the integration of neural networks within Python. Both model development and validation were conducted utilizing Google Colab, a cloud-based service. Users can harness a Tesla T4 GPU with 12 GB GDDR5 storage capacity through this complimentary platform.

4.2 EVALUATION METRICS

The proposed framework's efficacy is assessed through a range of statistical metrics, encompassing Accuracy, Precision, Recall, False-positive rate (FPR), True negative rate (TNR), F1-Score, Mean

Squared Error (MSE), and Mean Absolute Error (MAE). These metrics draw upon values extracted from the confusion matrix, which comprises True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Below are the individual formulations for these measurement indicators:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (2)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (4)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \quad (5)$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN+FP}} \quad (6)$$

$$\text{F1-Score} = 2 \times \frac{\text{Pr}\times\text{Re}}{\text{Pr+Re}} \quad (7)$$

True Positive (TP) indicates a situation where the 'MultiNet' infrastructure accurately identifies the positive type of Potato disease.

True Negative (TN) indicates a situation where the proposed infrastructure accurately identifies the absence of the negative Potato disease type.

False Positive (FP) indicates a situation where the infrastructure inaccurately identifies the positive type of Potato disease.

False Negative (FN) indicates a situation where the infrastructure mistakenly identifies the absence of the negative Potato disease type.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{i}(Y_i - X_i)^2 \quad (8)$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{i}|Y_i - X_i| \quad (9)$$

Where in equation 8 and 9,

n is the number of data points

Y_i are the observed values

X_i are the predicted values

4.3 TRAINING AND PARAMETER OPTIMIZATION

The training process of the proposed 'MultiNet' infrastructure for three potato diseases from the Plant Village dataset is depicted in **Figure 4.1**, with simulation results showcased. **Table 4.1** details the hyperparameter values utilized during training. Notably, the efficacy of model training heavily relies on the selection of optimizer and loss functions for gradient descent. In this regard, we select Adam as the optimizer function, capitalizing on the advantages of both AdaGrad and RMSProp optimizers. This choice is made to effectively address sparse gradients commonly encountered in large datasets. Categorical cross-entropy is chosen as the loss function, given the binary classification nature of our work on the Plant Village dataset. To optimize the learning process, a learning rate value is crucial, and we address this challenge by selecting a rate of 0.0001 to mitigate undesirable behavior associated with excessively large or small learning rates.

Additionally, a small batch size of 32 is utilized, promoting effective model generalization. The suggested framework undergoes 50 training epochs, reaching a training accuracy exceeding 98% and a validation accuracy of 100% by the 21st epoch.

Table 4.1: The parameter setting employed in the training phase of the 'MultiNet' infrastructure.

Dataset	Parameter	Value
Three-class and seven-class classification	Optimizer	adam
	Learning Rate	.0001
	Loss Function	categorical_crossentropy
	Metrics	accuracy
	Batch Size	32
	Epochs	50

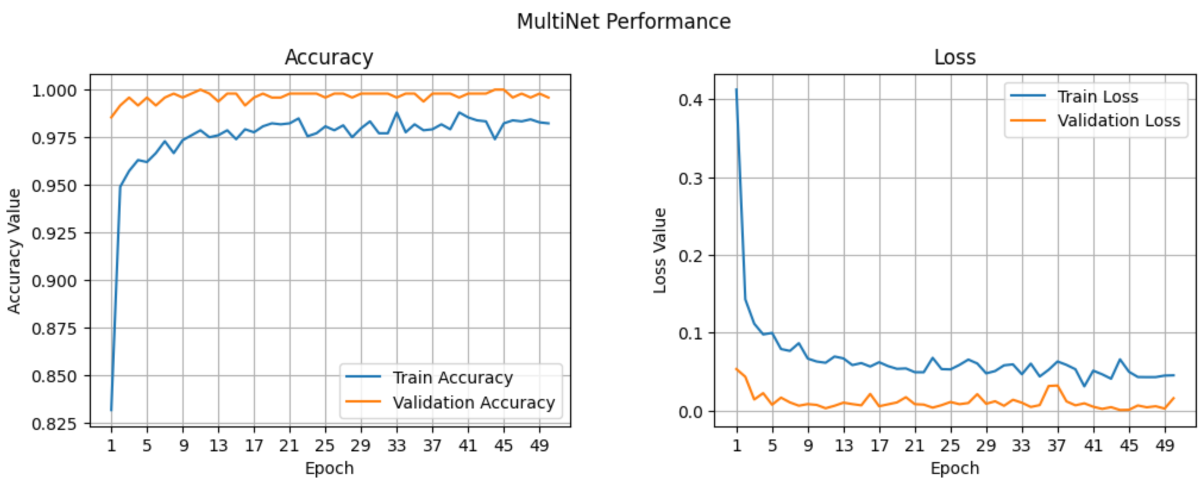


Figure 4.1: Training advancement for the three-class dataset: (a) accuracy in training and validation (increased values indicate improved performance), and (b) loss in training and validation (decreased values indicate improved performance

The absence of overfitting during training is evident in **Figure 4.1(a)**, and the graph for the loss function in **Figure 4.1(b)** illustrates a significant decrease in the loss value. Despite minor fluctuations attributed to the constrained group of 32 images, the training process, on the whole, produces favorable outcomes. Similarly, for the second dataset, encompassing seven diseases, the progression of accuracy and loss during the training of the proposed 'MultiNet' framework is depicted in **Figure 4.1**. The hyperparameter values utilized for training align with those employed for the Plant Village dataset, as detailed in **Table 4.1**. The 'MultiNet' framework undergoes 50 epochs of training, attaining a training accuracy of 94 percent right after the 24th epoch, with validation accuracy surpassing 97 percent. **Figure 4.2(b)** showcases a loss function curve with minimal fluctuation, indicating that the loss value is nearly zero throughout the training and validation phases. Moreover, **Figure 4.2(a)** provides visual confirmation that the model does not exhibit overfitting during the training process.

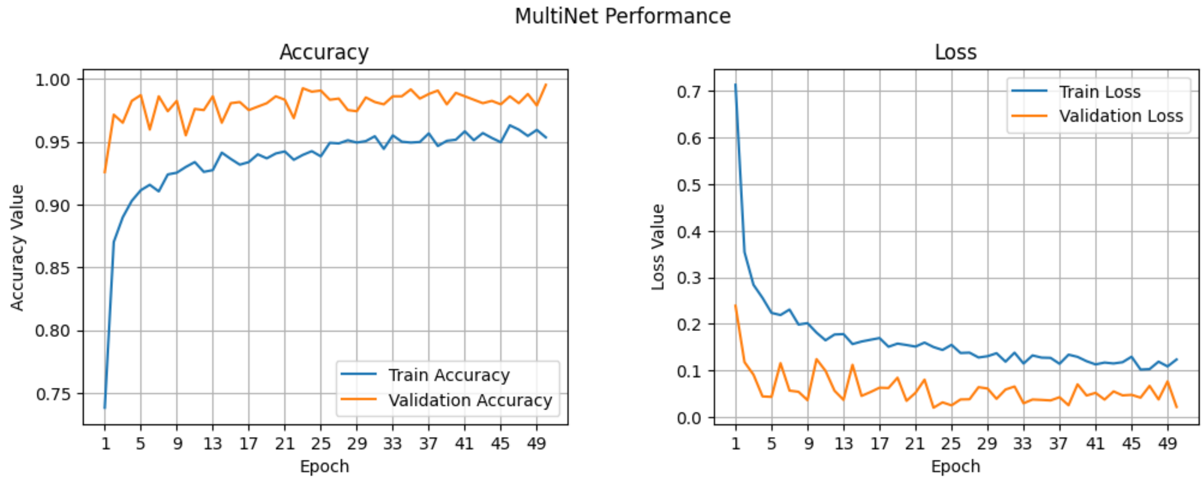


Figure 4.2: Training advancement for the seven-class dataset: (a) accuracy in training and validation (increased values indicate improved performance), and (b) loss in training and validation (decreased values indicate improved performance).

4.4 RESULT ANALYSIS

The ROC (Receiver Operating Characteristics) curve and confusion matrix for the "MultiNet" framework applied to the Plant Village dataset are shown in **Figure 4.3**, which focuses on three-class disease identification. This framework amalgamates ResNet50, DenseNet-201, and VGG16, leveraging their combined features to categorize images of potato leaf into Healthy,

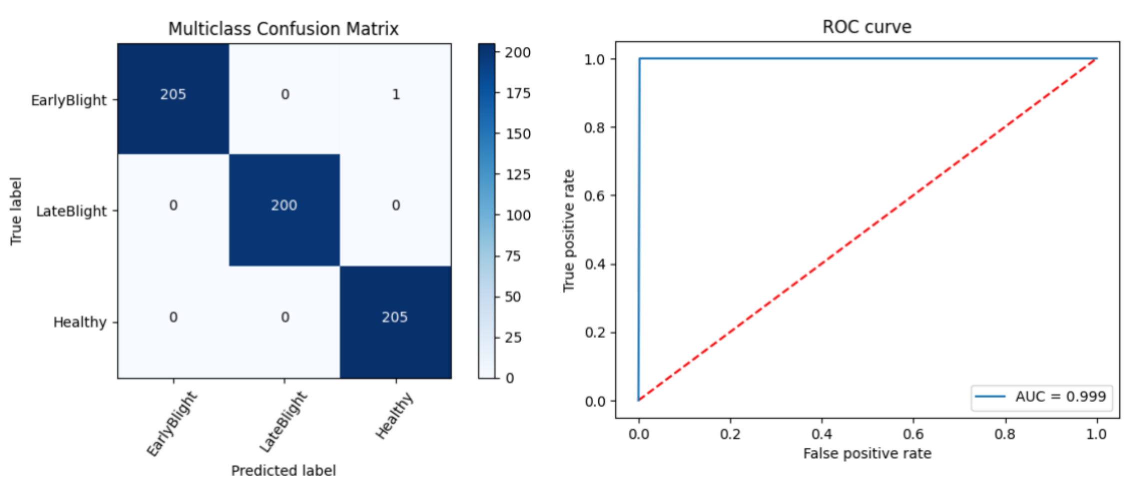


Figure 4.3: The evaluation of the 'MultiNet' infrastructure on a three-class dataset, including a) Three-class Confusion Matrix and b) ROC curve analyses

Early Blight, or Late Blight classes. As illustrated in **Figure 4.3(a)**, the 'MultiNet' framework accurately classifies 205, 200, and 205 leaf images corresponding to Early Blight, Late Blight, and Healthy Potato leaves, respectively. Impressively, only one misclassification occurs for Early

Blight leaves, with no errors in classifying Late Blight and Healthy images. The ROC curve in **Figure 4.3(b)** attests to the model's high consistency, achieving an area value of 0.999. Furthermore, individual evaluations of the transfer learning models on the Plant Village datasets are conducted to provide a comprehensive understanding of the 'MultiNet' framework's efficiency. **Table 4.2** shows the information from the classification report obtained by using the "MultiNet" infrastructure for three-class categorization. Notably, for three-class, each class's average precision, recall, and f1-score are 1.00, demonstrating the effectiveness of our suggested approach.

Table 4.2: Classification report based on use of the proposed "MultiNet" infrastructure for three-class potato disease

Class	Precision	Recall	F1-score	Support
Early Blight	1.00	1.00	1.00	206
Late Blight	1.00	1.00	1.00	200
Healthy	1.00	1.00	1.00	205
Accuracy			1.00	611
Macro avg	1.00	1.00	1.00	611
Weighted avg	1.00	1.00	1.00	611

Table 4.3 provides a comparison between the 'MultiNet' infrastructure and seven additional models based on transfer learning, offering an insightful analysis of their respective performances for three -class dataset. Remarkably, the 'MultiNet' framework surpasses all other cutting-edge models, showcasing an exceptional average precision, recall, and F1-score of 0.998. Additionally, it exhibits impressive model efficiency, reflected in MSE and MAE values of 0.007 and 0.003, respectively. Notably, Xception also delivers commendable performance with an F1-score of 0.997, closely trailing the MultiNet Framework.

Table 4.3: Outcomes derived from the utilization of the Proposed 'MultiNet' infrastructure with seven distinct transfer-learning models on datasets pertaining to three-class Potato Diseases.

Model	**Class 3**					
	Accuracy	F1-Score	Precision	Recall	MSE	MAE
VGG16	.995	.995	.995	.995	.015	.008
VGG19	.995	.995	.995	.995	.020	.010
ResNet-50	.961	.961	.961	.961	.152	.077
DenseNet-201	.993	.993	.993	.993	.026	.013
MobileNetV2	.989	.989	.989	.989	.040	.021
NASNetMobile	.980	.980	.980	.980	.069	.036
Xception	.997	.997	.997	.997	.013	.007
Proposed Method	.998	.998	.998	.998	.007	.003

Conversely, VGG16 and VGG19 exhibit comparable F1-scores of 0.995, indicating strong performance. DenseNet201 performs well, achieving an F1-score of 0.993. On the other hand, ResNet50, NASNetMobile, and MobileNetV2 demonstrate F1-scores of 0.961, 0.980, and 0.989, respectively, in the three-class disease detection category. Using the 'MultiNet' framework, **Figure 4.4** illustrates the ROC curve and confusion matrix for the seven-class dataset. This dataset involves identifying various potato disease categories, namely Early Blight General (EarlyBlightG), Early Blight Serious (EarlyBlightS), Late Blight, Virus infected leaf, Insect infected leaf, LeafRollVirus infected leaf, and Healthy leaf.

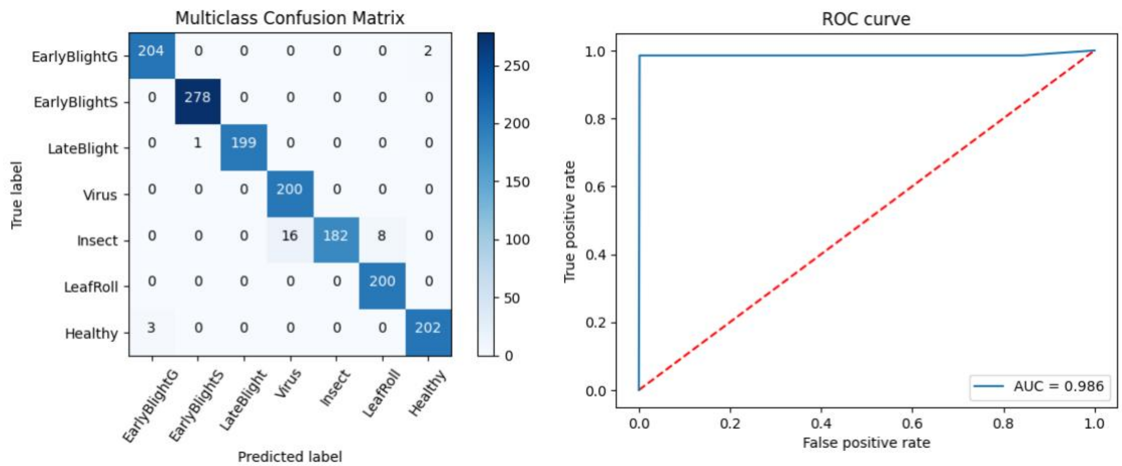


Figure 4.4: The evaluation of the 'MultiNet' infrastructure on a seven-class dataset, including a) Seven-class Confusion Matrix and b) ROC curve analyses.

Examining **Figure. 4.4(a)**, the framework correctly classifies 204, 278, 199, 200, 182, 200, and 202 images for Early Blight General, Early Blight Serious, Late Blight, Virus infected Leaf, Insect infected Leaf, Leafroll Virus infected Leaf, and Healthy Leaf, respectively. However, misclassifications occur, notably with two Early Blight General images, one Late Blight image, twenty-four Insect infected leaf images, and three Healthy images. Importantly, the 'MultiNet' framework exhibits no misclassifications for Early Blight Serious, Virus infected, and LeafRoll Virus infected images. The model showcases robust consistency and applicability, as evidenced by an area value of 0.986 as depicted in **Figure 4.4(b)**. This high area value underlines the model's strong performance and versatility across various scenarios. **Table 4.4** shows the information from the classification report obtained by using the "MultiNet" infrastructure for seven-class categorization. This table displays the f1-scores for Early Blight General and Healthy at 0.99, and Early Blight Serious and Late Blight at 1.00. However, the f1-scores for the remaining three classes—infested with viruses, infected with insects, and infected with LeafRoll viruses—are, respectively, 0.94, 0.98, and 0.96.

Table 4.4: Classification report based on use of the proposed "MultiNet" infrastructure for three-class potato disease

	Precision	Recall	F1-Score	Support
Early Blight General	0.99	0.99	0.99	206
Early Blight Serious	1.00	1.00	1.00	278
Late Blight	1.00	0.99	1.00	200
Healthy	0.99	0.99	0.99	205
Insect	1.00	0.88	0.94	206
Potato Leaf Roll Virus	0.96	1.00	0.98	200
Potato Leaf Virus	0.93	1.00	0.96	200
Accuracy			0.98	1495
Macro avg	0.98	0.98	0.98	1495
Weighted avg	0.98	0.98	0.98	1495

In **Table 4.5**, a comparative analysis reveals the 'MultiNet' framework's high performance, boasting an average precision, recall, and f1-score of 0.98, along with MSE of 0.137 and MAE of 0.037. Surprisingly, the system outperforms seven modern models in identifying the presence of potato disease categories. Within this set, DenseNet-201, ResNet50, VGG16, MobileNetV2, NASNetMobile, and Xception achieve classification accuracies surpassing 95%. However, VGG19 exhibits subpar performance specifically in detecting the multiclass potato disease category. This highlights the efficacy of the framework in surpassing its counterparts, especially in comparison to these widely used pre-trained models.

Table 4.5: Outcomes derived from the utilization of the Proposed 'MultiNet' infrastructure with seven distinct transfer learning models on datasets pertaining to seven-class Potato Diseases

Model	**Class 7**					
	Accuracy	F1-Score	Precision	Recall	MSE	MAE
VGG16	.960	.960	.960	.960	.170	.076
VGG19	.794	.80	.84	.794	1.60	.430
ResNet-50	.978	.978	.978	.978	.075	.036
DenseNet-201	.977	.977	.977	.977	.072	.037
MobileNetV2	.953	.954	.961	.952	.205	.072
NASNetMobile	.978	.978	.979	.978	.119	.039
Xception	.970	.970	.970	.970	.099	.052
Proposed Method	.98	.98	.98	.98	.137	.037

.

Chapter 5 DISCUSSION

5.1 OVERVIEW

This paper introduces a novel approach to multiclass classification, specifically targeting three-class and seven-class categorizations, by employing a 'MultiNet' infrastructure on potato leaf images. The evaluation of the 'MultiNet' infrastructure is conducted using two distinct datasets.

5.2 ASSESSMENT OF THE "MULTINET" APPROACH IN COMPARISON WITH EXISTING APPROACHES

Table 5.1 presents a thorough comparison of the performance of the 'MultiNet' infrastructure with prior studies that employed the same dataset. These studies, however, utilized diverse architectures, parameters, and depth sizes, making the comparison comprehensive and encompassing various structural aspects. The aim is to assess how the 'MultiNet' framework fares against existing literature with differences in these critical elements. Kamal et al. [17] introduced Modified MobileNet and Reduced MobileNet models for plant leaf disease identification. These models utilized depthwise separable convolution in lieu of the traditional convolution layer, altering the MobileNet architecture. Training sessions involved diverse crops from the Plant Village dataset, encompassing photographs of plant leaves from a specified geographical region.

In a related context, Liang et al. [37] introduced a network for plant disease identification and severity approximation, incorporating a residual structure and shuffle units inspired by the architecture of ResNet-50. Bonik, C. et al. [38] presented a CNN model which is trained on potato leaf images from the Plant Village dataset. A CNN model for identifying early blight, late blight, and healthy potato leaves was presented by Khalifa et al. [39]. The model was trained using the region-specific Plant Village dataset. A Convolutional Neural Network (CNN) model developed by Rozaqi and Sunyoto [40] aims to differentiate between early and late blight on the leaves of potatoes as well as healthy leaves. Their model was trained using the region-specific Plant Village dataset. In order to distinguish between healthy potato leaves and two frequent diseases—early and late blight—Sanjeev et al. [41] used a feedforward neural network (FNN). They used the Plant Village dataset to train and test their model. In order to distinguish between potato leaves that have been damaged (by early or late blight) and those that have not, Barman et al. provided a self-constructed CNN framework in [19]. The model was trained using the region-specific Plant Village dataset. Using a previously trained VGG19 framework for the extraction of features, Tiwari et al. [10] classified data from the Plant Village dataset using different classifiers, including SVM, neural, and KNN. Lee et al. [18] built a CNN-based method to differentiate between healthy and diseased (early blight, late blight) potato leaves using the Plant Village dataset. Nevertheless, no data that the model had not seen before was used for testing. Using the data collected by Plant Village, Islam et al. [7] constructed a multi-support vector machine (SVM) and segment-oriented model for identifying the existence of potato illnesses. Their findings emphasized the potential for accuracy improvement in the model. Rabbia, M. et al. [11] proposed a system utilizing the Efficient DenseNet-201 architecture that has been pre-trained, addressing class imbalance in data through reweighted cross-entropy loss function adjustments. The results in **Table 5.1** highlight that the proposed framework achieves superior prediction accuracy in identifying multiclass potato diseases compared to previous studies. Remarkably, we have attained an outstanding mean classification accuracy of 99.8% for the three-class dataset through the effective integration of all pre-trained models. Additionally, our approach yielded a commendable mean classification accuracy of 98% for the seven-class dataset, demonstrating the successful integration of pre-trained models.

Table 5.1: Assessing the 'MultiNet' framework through comparisons with alternative methods in both binary and multi-class classification scenarios.

Reference	Methodology	Disease	Dataset	Accuracy
1. [42] Chen, J. et al. 2023	MobileNet -V2	3 Classes (Healthy, Late Blight, Early Blight)	PlantVillage	97.73%
2. [17] Kamal, K. et al. 2019	Modified MobileNet	3 Classes (Healthy, Late Blight, Early Blight)	PlantVillage	98.34%
3. [37] Liang, Q. et al. 2019	ResNet50	3 Classes (Healthy, Late Blight, Early Blight)	PlantVillage	98%
4. [38] Bonik, C. et al. 2023	CNN	3 Classes (Healthy, Late Blight, Early Blight)	PlantVillage	94.2%
5. [39] Khalifa, N.E.M. et al. 2021	CNN	Binary (Late Blight and Early Blight)	PlantVillage	98%
6. [41] Sanjeev, K. et al. 2021	FFNN	Binary (Late Blight and Early Blight)	PlantVillage	96.5%
7. [11] Rabbia, M. et al. 2022	DenseNet201	5 classes (Healthy, Late Blight, Early Blight, Leaf Roll, Potato Verticillum_wilt)	PlantVillage, Manual	97.2%
8. [40] Rozaqi, A.J. et al. 2020	CNN	Binary (Late Blight and Early Blight)	PlantVillage	92%
9. [19] Barman, U. et al. 2020	SBCNN	Binary (Late Blight and Early Blight)	PlantVillage	96.75%
10. [10] Tiwari, D. et al. 2020	SVM, KNN and Neural Net	Binary (Late Blight and Early Blight)	PlantVillage	97.8%
11. [18] Lee, T.Y. et al. 2020	CNN	Binary (Late Blight and Early Blight)	PlantVillage	99%

12. [7] Islam, M. et al. 2017	Segment and Multi SVM	Binary (Late Blight and Early Blight)	PlantVillage	95%
13. Proposed Method	ResNet-50+DenseNet-201+VGG16	MultiClass (Early Blight, Late Blight, and Healthy)	PlantVillage	99.84%
14. Proposed Method	ResNet-50+ DenseNet-201+VGG16	MultiClass (Early Blight General, Early Blight Serious, Late Blight, Healthy, Insect infected, LeafRollVirus, and Virus infected)	PlantVillage, Manual	98%

.

5.3 SAMPLE OUTPUT

Figure 5.1 and **5.2** shows the prediction of three-class and seven-class dataset images by our framework. Our approach exhibits advantages over existing methods in the literature, such as the avoidance of time-consuming and potentially ineffective processes associated with utilizing different feature extractor methods, especially when dealing with a large number of images. The suggested framework offers a segmentation-free methodology that eliminates the need for any manually crafted features.

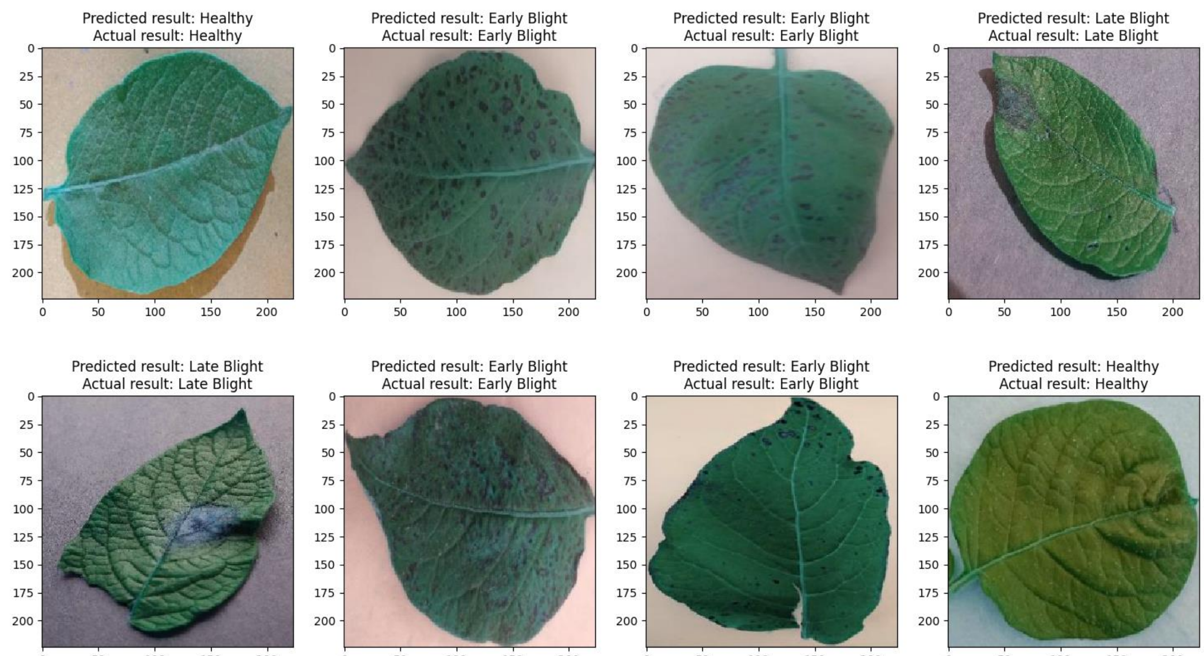


Figure 5.1: ‘‘MultiNet” predicted three-class dataset image

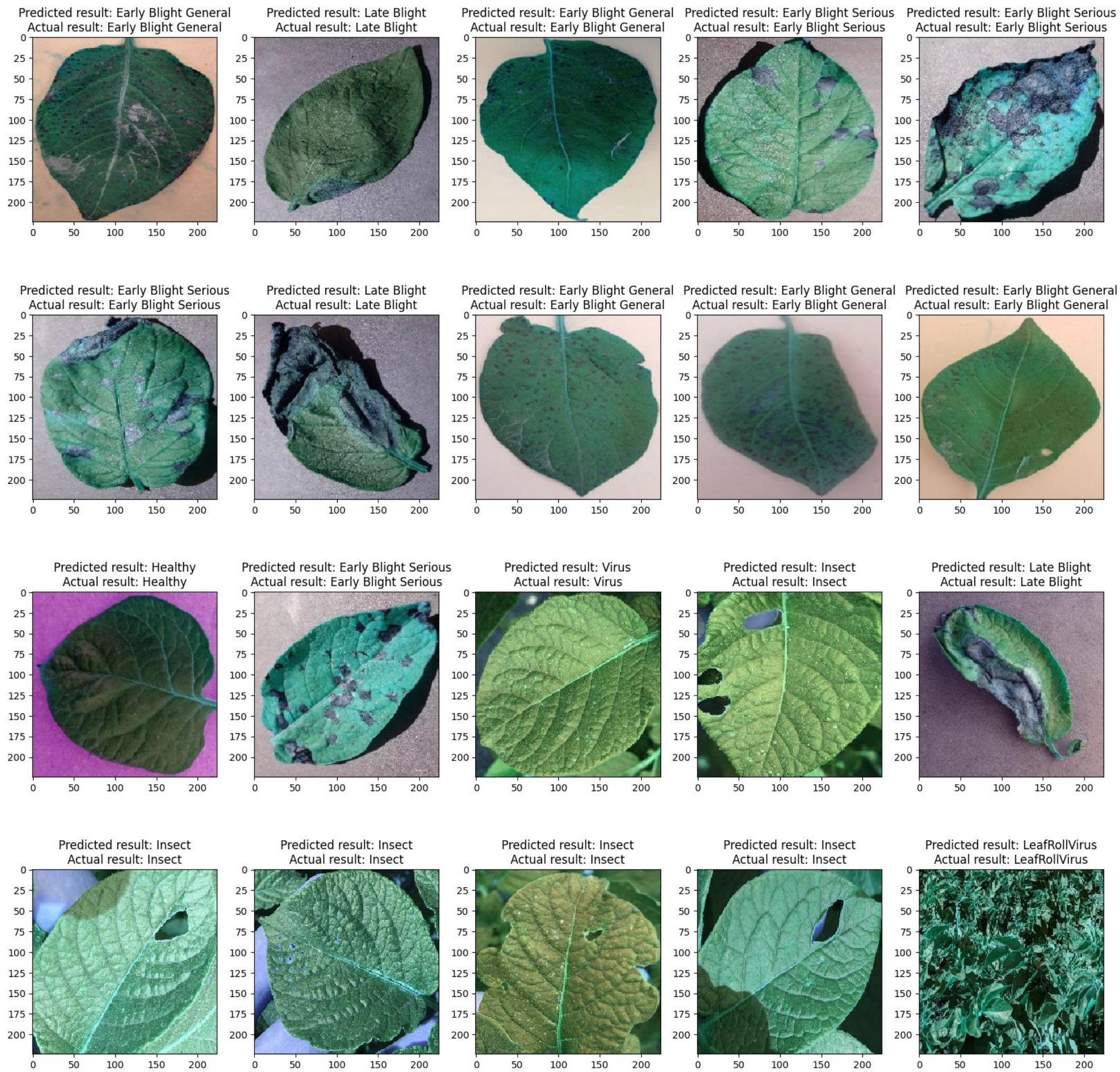


Figure 5.2: ‘‘MultiNet” predicted seven-class dataset images.

Chapter 6 CONCLUSION & FUTURE WORK

6.1 CONCLUSION

In this study, we present a novel deep learning framework, termed 'MultiNet,' designed to enhance the precision and reliability of the detection of potato leaf disease from images. This 'MultiNet' infrastructure makes use of transfer learning methods. Additionally, it incorporates feature extraction techniques. This approach allows multiple CNN models that have been pretrained to retrieve characteristics simultaneously, which are then merged for the classification process. The model is subjected to training and testing on image datasets of varying sizes, showcasing its versatility across diverse datasets. Experimental results showcase the 'MultiNet' framework's exceptional performance, achieving classification accuracy of 99.8% and 98% for three-class and seven-class datasets, respectively. This outperformance extends to individual CNN models that have been pretrained and all other most modern models reported in the literature. The strong experimental outcomes confirm the effectiveness of our suggested strategy.

6.2 FUTURE WORK

The framework will be used in future Developing real-time monitoring systems and early warning mechanisms can help farmers detect potato diseases at their earliest stages. By leveraging IoT (Internet of Things) devices, wireless sensor networks, and cloud computing, farmers can receive timely alerts about potential disease outbreaks, enabling proactive management strategies. Integration of potato disease detection systems with precision agriculture techniques can optimize resource allocation, minimize chemical usage, and improve crop health management. This may involve combining disease detection data with other agronomic factors such as soil moisture, nutrient levels, and weather conditions to provide holistic recommendations for farmers. Moreover, we are confident that our 'MultiNet' approach could potentially be applicable in other fields, including real-time malfunction evaluation, virtualized rice damage evaluation, and localization for identifying distinct imagery. Overall, our ‘MultiNet’ approach holds promise for advancing agriculture towards more sustainable, efficient, and resilient practices, ultimately benefiting farmers, consumers, and the environment.

References

[1] Salan, M. S. A., Hossain, M. M., Sumon, I. H., Rahman, M. M., Kabir, M. A., & Majumder, A. K. ((2022). Measuring the impact of climate change on potato production in Bangladesh using Bayesian Hierarchical Spatial-temporal modeling. *Plos one*, *17*(11), e0277933.

[2] Kadakoğlu, B., & Karli, B. (2022). Economic analysis of potato production in Afyonkarahisar province. *KSU J Agric Nat*, *25*, 581-588.

[3] Sasaki, Y., Okamoto, T., Imou, K., & Torii, T. (1998). Automatic diagnosis of plant disease-Spectral reflectance of healthy and diseased leaves. *IFAC Proceedings Volumes*, *31*(5), 145-150.

[4] Henson, J. M., & French, R. (1993). The polymerase chain reaction and plant disease diagnosis. *Annual review of phytopathology*, *31*(1), 81-109.

[5] Koo, C., Malapi-Wight, M., Kim, H. S., Cifci, O. S., Vaughn-Diaz, V. L., Ma, B., ... & Han, A. (2013). Development of a real-time microchip PCR system for portable plant disease diagnosis. *PloS one*, *8*(12), e82704.

[6] Singh, U. P., Chouhan, S. S., Jain, S., & Jain, S. (2019). Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. *IEEE access*, *7*, 43721-43729.

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

[8] Hu, Y. H., Ping, X. W., Xu, M. Z., Shan, W. X., & He, Y. (2016). Detection of late blight disease on potato leaves using hyperspectral imaging technique. *Guang pu xue yu guang pu fen xi= Guang pu*, *36*(2), 515-519.

[9] Acharjee, T., Das, S., & Majumder, S. (2023). Potato Leaf Diseases Detection Using Deep Learning. *International Journal of Digital Technologies*, *2*(1).

[10] Tiwari, D., Ashish, M., Gangwar, N., Sharma, A., Patel, S., & Bhardwaj, S. (2020, May). Potato leaf diseases detection using deep learning. In *2020 4th international conference on intelligent computing and control systems (ICICCS)* (pp. 461-466). IEEE.

[11] Mahum, R., Munir, H., Mughal, Z. U. N., Awais, M., Sher Khan, F., Saqlain, M., ... & Tlili, I. (2023). A novel framework for potato leaf disease detection using an efficient deep

learning model. *Human and Ecological Risk Assessment: An International Journal*, *29*(2), 303-326.

[12] Rashid, J., Khan, I., Ali, G., Almotiri, S. H., AlGhamdi, M. A., & Masood, K. (2021). Multi-level deep learning model for potato leaf disease recognition. *Electronics*, *10*(17), 2064.

[13] Chugh, G., Sharma, A., Choudhary, P., & Khanna, R. (2020). Potato leaf disease detection using inception V3. *Int. Res. J. Eng. Technol*, *7*(11), 1363-1366.

[14] Iqbal, M. A., & Talukder, K. H. (2020, August). Detection of potato disease using image segmentation and machine learning. In *2020 international conference on wireless communications signal processing and networking (WiSPNET)* (pp. 43-47). IEEE.

[15] Tarik, M. I., Akter, S., Al Mamun, A., & Sattar, A. (2021, February). Potato disease detection using machine learning. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (pp. 800-803). IEEE.

[16] Geetharamani, G., & Pandian, A. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers & Electrical Engineering*, *76*, 323-338.

[17] Kamal, K. C., Yin, Z., Wu, M., & Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. *Computers and electronics in agriculture*, *165*, 104948.

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

[19] Barman, U., Sahu, D., Barman, G. G., & Das, J. (2020, July). 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)* (pp. 682-687). IEEE.

[20] Zhang, Y., Song, C., & Zhang, D. (2020). Deep learning-based object detection improvement for tomato disease. *IEEE access*, *8*, 56607-56614.

[21] Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016, November). Understanding data augmentation for classification: when to warp?. In *2016 international conference on digital image computing: techniques and applications (DICTA)* (pp. 1-6). IEEE.

[22] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[23] Habibzadeh, M., Jannesari, M., Rezaei, Z., Baharvand, H., & Totonchi, M. (2018, April). Automatic white blood cell classification using pre-trained deep learning models: Resnet and inception. In *Tenth international conference on machine vision (ICMV 2017)* (Vol. 10696, pp. 274-281). SPIE.

[24] Wichrowska, O., Maheswaranathan, N., Hoffman, M. W., Colmenarejo, S. G., Denil, M., Freitas, N., & Sohl-Dickstein, J. (2017, July). Learned optimizers that scale and generalize. In International conference on machine learning (pp. 3751-3760). PMLR.

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

[26] Hao, W., & Zhang, Z. (2019). Spatiotemporal distilled dense-connectivity network for video action recognition. Pattern Recognition, 92, 13-24

.

[27] Lumini, A., & Nanni, L. (2019). Deep learning and transfer learning features for plankton classification. Ecological informatics, 51, 33-43

[28] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[29] Scherer, D., Müller, A., & Behnke, S. (2010, September). Evaluation of pooling operations in convolutional architectures for object recognition. In International conference on artificial neural networks (pp. 92-101). Berlin, Heidelberg: Springer Berlin Heidelberg.

[30] Shah, S.R., et al., Comparing Inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A Case Study on Early Detection of a Rice Disease. Agronomy, 2023. 13(6): p. 1633.

[31] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[32] Addagarla, S. K., Chakravarthi, G. K., & Anitha, P. (2020). Real time multi-scale facial mask detection and classification using deep transfer learning techniques. *International Journal*, *9*(4), 4402-4408.

[33] He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, *212*, 106622.

[34] Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8697-8710).

[35] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).

[36] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

[37] Liang, Q., Xiang, S., Hu, Y., Coppola, G., Zhang, D., & Sun, W. (2019). PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. *Computers and electronics in agriculture*, *157*, 518-529.

[38] Bonik, C. C., Akter, F., Rashid, M. H., & Sattar, A. (2023, January). A Convolutional Neural Network Based Potato Leaf Diseases Detection Using Sequential Model. In *2023 International Conference for Advancement in Technology (ICONAT)* (pp. 1-6). IEEE.

[39] Khalifa, N. E. M., Taha, M. H. N., Abou El-Maged, L. M., & Hassanien, A. E. (2021). Artificial intelligence in potato leaf disease classification: a deep learning approach. *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*, 63-79.

[40] Rozaqi, A. J., & Sunyoto, A. (2020, November). Identification of disease in potato leaves using Convolutional Neural Network (CNN) algorithm. In *2020 3rd International Conference on Information and Communications Technology (ICOIACT)* (pp. 72-76). IEEE.

[41] Sanjeev, K., Gupta, N. K., Jeberson, W., & Paswan, S. (2021). Early prediction of potato leaf diseases using ANN classifier. *Oriental Journal of Computer Science and Technology*, *13*(2, 3), 129-134.

[42] Chen, J., Deng, X., Wen, Y., Chen, W., Zeb, A., & Zhang, D. (2023). Weakly-supervised learning method for the recognition of potato leaf diseases. *Artificial Intelligence Review*, *56*(8), 7985-8002.