A Comparison of CNN Architectures Using Transfer Learning to Detect Paddy Leaf Diseases

BY

Al Imran Alvy

201-15-3255

FINAL YEAR DESIGN PROJECT REPORT

A portion of the requirements for the Bachelor of Science in Computer Science and Engineering degree have been met by this report.

Supervised By

Dr. Arif Mahmud

Associate Professor & Program Director MIS

Department of CSE

Daffodil International University

Co-Supervised By

Ms. Tasnim Tabassum

Lecturer

Department of CSE

Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

June 2024

APPROVAL

My Project titled "**A Comparison of CNN Architectures Using Transfer Learning to Detect Paddy Leaf Diseases**" submission was made by Al Imran Alvy, with the student ID 201-15-3255, to the Department of Computer Science and Engineering. This work, submitted to and accepted by the faculty of Daffodil International University to fulfill the prerequisites for a Bachelor of Science in Computer Science and Engineering, meets all the criteria both formally and substantively. Currently, the event is scheduled to occur in June of 2024.

BOARD OF EXAMINERS

Chairman

Narayan Ranjan Chakraborty (NRC)
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Ohidujjaman Tuhin (DOT)
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Tanzina Afroz Rimi (TAR)
Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

Dr. Md. Manowarul Islam
Associate Professor
Department of Computer Science & Engineering
Jagannath University

DECLARATION

I confirm that I have conducted this study under the supervision of **Dr. Arif Mahmud, Associate Professor and Program Director** at the Department of Computer Science and Engineering, Daffodil International University. I hereby declare that this project, in its whole, has not been previously submitted anywhere with the intention of acquiring any degree or certificate.

Supervised By:

Dr. Arif Mahmud
Associate Professor & Program Director MIS
Department of CSE
Daffodil International University

Co-Supervised by:

Ms. Tasnim Tabassum
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Al Imran Alvy
201-15-3255
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

The first thing I want to do is thank the All-Mighty Allah from the bottom of my heart. It was his special kindness that helped us finish this study project for the last year of my studies.

I are really grateful and wish my profound indebtedness to **Dr. Arif Mahmud, Associate Professor & Program Director MIS, Department of CSE Daffodil International University, Dhaka, and Co Supervisor Ms. Tasnim Tabassum, Lecturer, Department of CSE, Daffodil International University, Dhaka**. I want to thank my boss for helping me with this project by giving me a lot of information and interest in Machine Learning and Deep Learning. It would not have been possible to finish the project without his never-ending kindness, expert advice, constant support, and eager supervision, helpful feedback, reading many bad drafts and fixing them at every step.

I would like to express my heartiest gratitude to **Dr. Sheak Rashed Haider Noori, Professor and Head, of the Department of CSE,** and **Dr. S. M. Aminul Haque, Associate Professor and Associate Head, of the Department of CSE** for their kind help to finish my project and also to all of the faculty members and the staff of the CSE department of Daffodil International University for kind support and help on the technical and the administrative aspect of the study.

I would like to express my appreciation to all of my students from Daffodil International University who actively engaged in this discussion while also completing their course tasks.

Lastly, I would want to convey my profound appreciation to my parents for their enduring affection and unwavering tolerance that they have consistently shown towards me throughout my existence. They have consistently provided support and encouragement to me throughout my whole existence.

ABSTRACT

I provide a method for real-time detection of Paddy Leaf Disease in my article. Blast, brown spot, and bacterial leaf blight are among the most prevalent diseases that may harm rice plants in my nation. Farmers may be losing money due to decreased agricultural yields caused by harmful pathogens. The phrase "paddy leaf disease" describes a group of illnesses that impact rice plants. These diseases may be caused by fungi, bacteria, or viruses and because the leaves to become yellow, wilt, and eventually reduce the crop's output. For this reason, I need to find that issue quickly so that I may harvest additional crops. To enhance crop health, boost yields, and decrease economic losses, farmers may use deep learning to detect paddy leaf disease. This approach is quicker, more accurate, and more consistent than traditional methods of disease identification and diagnosis in rice plants. Discovering trends and patterns in illness data may also be aided by this, which can result in fresh perspectives and advancements in the agricultural sector. I need a procedure, such as data collecting, in order to complete the process or create that Deep-learning. From the internet, I get a dataset of images. The collection contains five different kinds of images. I proceed to pre-process the dataset. The next step is to train my model and put it through its paces in testing. Lastly, I use a number of algorithms, including DenseNet121, ResNet50V2, MobileNetV2, and EfficientNetB2. Using EfficientNetB2, I was able to get an accuracy of about 91%. The techniques used in each person's comparison statements are also included in the article's implementation component. To build the most accurate model for the conditions, this study also makes use of model validation techniques.

TABLE OF CONTENTS

CONTENTS	PAGE

LIST OF FIGURES

FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

At the start of the 21st century, paddy was still the most common type of rice used in human diets. It was the main source of energy and a major source of protein for about three billion people [16]. The Asia-Pacific region grows more than 90% of the world's rice [17].Complete crop protection is the goal of my project. Diseases like blast, sheath blight, and brown spot, which affect paddy leaves, may reduce rice harvest yields by as much as 30% in Bangladesh, where I live [18]. So it's very important for detecting paddy leaf disease. Visual examination has long been the method of choice for farmers, but it is laborious and not always reliable. Nevertheless, there is an increasing need to enhance the precision and effectiveness of paddy leaf disease detection in Bangladesh via the use of machine learning and other digital technologies. In an effort to enhance food security, decrease pesticide usage, and boost crop yields, several research projects and efforts have been initiated to create systems based on machine learning. Quickly identify and control paddy leaf diseases to protect crops from damage. To identify these issues, I'm experimenting with deep learning. Potentially useful for both farmers and researchers, deep learning might identify illnesses in rice leaves. Improving crop health, yields, and farmer economic losses, this quicker, more accurate, and consistent way of diagnosing rice plant diseases is a game-changer [19]. In order to improve farming operations, it might potentially reveal patterns and trends in diseases. Deep learning necessitates the collection of data. About 212 photo image collection from online platform and organized into 5 categories. Up next is the preprocessing of the dataset. Photos should be resized, cropped, and normalized in order to standardize the dataset. Lastly, in order to construct feature vectors for every picture, I need to extract form, color, and texture attributes from the pre-processed photographs. The next step is to construct a Deep learning model that can distinguish between healthy and damaged leaves based on photo feature vectors. After that, I evaluated the trained model's accuracy in detecting paddy leaf diseases by running it on new pictures. I want to use the model to really identify diseases in rice leaves in the future. To identify paddy leaf diseases, you may use EfficientNetB2, MobileNetV2, ResNet50V2, or DenseNet121. At its peak, EfficientNetB2 achieved an accuracy of 91% for me. So, a technology that can accurately forecast the chance of getting paddy leaf disease detection is urgently needed right now. If this method can be identified early on, it may preserve a number of harvests. The developers of the many Paddy Leaf Disease detection systems available today have done extensive research and shown a wide variety of classification and modeling techniques; nonetheless, each system has its own limitations. My main goal is to find a way to overcome these restrictions and create a system that can accurately and consistently forecast when a cardiac arrest will occur with little overhead.

1.2 Motivation

I am motivated to use deep learning for paddy leaf disease detection so that farmers may experience better crop health, higher yields, and fewer financial losses. Envision a subsistence farmer in a third-world nation who grows rice for a living. Some of the rice plants exhibit symptoms like leaf yellowing and other issues, as discovered by the farmer. Not only can the farmer not afford to lose their crop, but they also have no idea what's wrong or how to remedy it. This is where paddy leaf disease diagnosis based on deep learning comes in handy; it might provide a fast and accurate solution. The farmer may use their phone's camera to snap a photo of the diseased leaves, which could then be sent to an app that uses deep learning to diagnose the problem. In real-time, a trained Deep-learning model analyses the image and provides the farmer with instructions on how to fix the problem. As the illness inflicts less harm on the farmer's crops, the faster it can be identified and treated. Their economy will be more secure and their income will rise in direct proportion to the quantity and quality of their output. Researchers and policymakers may use Deep Learning-based technology to sift through mountains of data on paddy leaf diseases in an effort to combat the global food crisis by identifying trends and patterns in disease outbreaks and developing strategies to halt their spread.

1.3 Research Questions

I had a plethora of questions before I even began the inquiry. To identify paddy leaf diseases, I used deep-learning algorithms in this research. The following were among the most important research questions:

- How can we determine which deep learning algorithms are best for identifying various paddy leaf diseases?
- How can image pre-processing methods be used to enhance the accuracy of paddy leaf disease diagnosis using deep learning?

- How big of a dataset is best for training deep learning models to identify paddy leaf diseases?
- How might optimizing the hyper parameters of Deep learning models enhance the effectiveness of paddy leaf disease detection using Deep learning?
- Is there a way to make deep learning models more adaptable to diverse climates and locales so they can identify diseases in paddy leaves more accurately?

1.4 Main Objective

The goal of this research is to develop a system that uses deep learning and image processing to identify paddy leaf diseases.

- To train a deep learning network to reliably identify paddy leaf diseases using just leaf photos as input.
- To assess the effectiveness of the deep learning model, many measures such as accuracy, precision, recall, and F1 score are used.
- To provide a deep learning model interface that farmers and other interested parties may use easily.
- To do field experiments for assessing the effectiveness of the deep learning model in real-life conditions and get input from farmers and other users.
- To evaluate the deep learning model's efficacy in detecting paddy leaf diseases in comparison to other methods already in use, such as human inspection and other machine learning techniques.

1.5 Report Layout

My final report for the Thesis is following that structures:

- Titled "A Comparison of CNN Architectures Using Transfer Learning to Detect Paddy Leaf Diseases," the first chapter presents the key points. Aside from the study's rationale, intended results, goals, and introduction to my research, there are a number of other aspects to consider.
- In Chapter 2, you will find an overview of the research topic, relevant literature, challenges faced while conducting the study, and the results of related studies.

- The study methodology, datasets, suggested systems, implementation approach, data pretreatment, and improved model are all included in Chapter 3.
- The experimental design, confusion matrix, performance, and comparative analysis are all part of the Experimental Results and Discussion, which is included in Chapter 4.
- In chapter five, I detail the societal implications of my study on environmental impacts, etc.
- The project's conclusion and future scope are covered in chapter six.

1.6 Project Management and Finance

The management of a project entails adhering to the rules that have previously been established and making use of the appropriate strategies, abilities, information, and expertise in order to accomplish the objectives of the project. Because I usually need a camera to take pictures of the grass leaves here, I would appreciate it if you could send me some money. In order to ensure that a project is completed with the objectives and outcomes that were promised, the primary responsibility of project management is to plan and monitor the project. It is essential for all of the teams and partners to be able to communicate with one another, deal with any dangers, and make effective use of the resources and finances available to them. The organization of a project includes a number of critical components, including the facilitation of research and the incorporation of data into systems. As is the process of organizing and scheduling meetings. In order to keep track of how the study was progressing, the researchers found it helpful to write down the tasks that had been completed as well as those that were still outstanding. Throughout the course of time, these tools were added to an archive. It is now possible for you to access the archive by using the Madcaps messaging system. Studying, evaluating sources, analyzing, thinking critically, organizing, and writing are all necessary steps in the process of creating a research paper in the field of finance.

CHAPTER 2
BACKGROUND

2.1 Introduction

During my research, I made use of Transfer Learning, EfficientNetB2, MobileNetV2, ResNet50V2, and DenseNet121. All of the algorithms have a similar technical trait: they process photo sets using neural networks. The fields of picture detection and classification often use all of these methods. When put to use for processing photos, each of these approaches delivers respectable results. DenseNet121, EfficientNetB2, MobileNetV2, and ResNet50V2 are some of the most established instances of convolutional neural network design. My approach is compatible with object recognition models of up to six layers in complexity, among others.

2.2 Related Works

The field of plant disease detection has made great strides with the advent of deep learning and computer vision approaches. Research into these techniques for disease detection in leaves has recently exploded in popularity as a potential game-changer in the field of crop health monitoring and management. Numerous reports, articles, and research papers have been written by various authors on the topic. In this section, I will showcase a portion of their project:

Srdjan et al. [1] developed a model that uses deep convolutional networks to classify leaf pictures and detect plant illnesses. The model has the ability to distinguish between a healthy leaf and one afflicted with any of thirteen distinct illnesses. Additionally, it can accurately identify a healthy leaf within its natural environment. They have extensively trained Convolutional Neural Networks in the past. According to the outcomes of independent class testing, the model that was developed had an average accuracy of 96.3%. Akhtar et al. [2] examined the advantages and disadvantages of several machine-learning techniques for automating the analysis of plant diseases. Research has shown that the optimal classification results may be achieved by combining the discrete cosine transform (DCT), the discrete wavelet transform (DWT), and texture feature extraction. The suggested technique combines DCT and DWT characteristics for classification using Support Vector Machine (SVM) and obtains a maximum accuracy of 94.45%. Dheeb et al. [3] proposed a methodology for working with photographs that consists of the following key steps: Initially, the K-means methods were used to partition the pictures into clusters. During the second phase, the image fragments were inputted into a neural network that had undergone training. The photographs were captured at Al-Ghor, a district situated in Amman, the capital city of Jordan. They focused their investigation on five severe leaf diseases. These illnesses are referred to as early scorch, cottony mould, late scorch, and small whiteness. The experimental findings demonstrate that the neural network classifier, using statistical classification, has the potential to automatically identify leaf illnesses with an approximate accuracy of 91%. Kulkarni et al. [4] effectively classified many plant diseases using their method. The data was separated using a Gabor filter, and the disorders were classified using an artificial neural network. A plant disease classifier was developed using an artificial neural network to accurately categories and classify various plant diseases. This classifier used both color and texture data to categories objects. The test results indicate that the Artificial Neural Network (ANN), when using a specific set of features, exhibits superior performance in classification, achieving an accuracy rate of 91%. Al-Hiary et al. [5] propose doing research to develop an algorithm that can rapidly and reliably detect and categories plant illnesses. The first stage of the algorithm involves determining the method for obtaining RGB images. The subsequent action entails modifying the color scheme of the pre-existing RGB images. Next, the K-means clustering algorithms are employed to partition the images into segments. The Grey Level Co-occurrence Matrix (GLCM) is utilized to extract texture information from the selected regions. Neural networks are used for the purpose of categorizing items into distinct categories. Typically, this approach has a success rate of approximately 94%. Camargo et al. [6] developed an algorithmic approach to tackle the task of identifying plant diseases, which relied on image processing techniques. Prior to proceeding to the subsequent stage, it is necessary to apply a color transformation to the obtained image. Following this procedure, the photographs undergo enhancement using a Gaussian filter. The modified image undergoes segmentation to isolate the affected regions by accurately determining the optimal threshold. This is done to ensure more accurate diagnoses. Subsequently, a medical prognosis is allocated to each individual segment. Typically, this approach has a success rate of approximately 93.1%. Karmokar et al. [7] proposed different methods for detecting plant diseases that combine feature extraction and Neural Network Ensemble (NNE) techniques. The NNE approach offers an enhanced ability to generalize learning by training multiple neural networks and then combining their outputs. The sole objective of this specific procedure was to identify diseases that impact tea leaves, achieving a final testing accuracy of 91%. Rumpf et al. [8] proposed a technique for identifying and categorizing plant diseases that could be implemented using Support Vector Machine algorithms. This approach was utilized to address the issue of sugar beetroot diseases, and the findings are documented in this report. The classification accuracy varied between 65% and 90%, contingent upon the type of disease and its stage of progression. Patil et al. [9] proposed a model for detecting plant diseases by analyzing textural parameters such as inertia, homogeneity, and correlation. The gray-level co-occurrence matrix was used to extract texture properties from the picture, which are useful for identifying plant illnesses. Patil's model was utilized to identify numerous plant diseases. The researchers performed experiments on maize leaf samples, extracting color in order to ascertain the presence of diseases. Rakesh et al. [10] developed a method using support vector machines to predict plant diseases by creating weather-based prediction models. The researchers conducted a comparative analysis of the effectiveness of traditional multiple regression, two types of artificial neural networks (back propagation neural network and extended regression neural network), and support vector machines (SVM). The SVM-based regression technique revealed a more precise understanding of the relationship between environmental factors and the severity of sickness. This finding has potential implications for disease management. The discovery was highly significant. Kamljot et al. [11] describes a method for identifying the characteristics of plant diseases based on color, edge detection, and histogram matching. The study can be divided into two main components, as indicated above. MATLAB is provided with both healthy and diseased leaves during the initial stage of the process. Subsequently, the RGB color components are isolated and transformed into a grayscale image, followed by the application of the CANNY edge-detection algorithm. Next, we need to create a histogram for each individual component of both the healthy and sick leaf photographs. During the second step, the leaf is subjected to a reiteration of the testing process, which involves comparing all previously documented observations and arriving at a final conclusion. Panagiotis et al. [12] demonstrated the development and implementation of an artificial intelligence system for analyzing plant leaves and extracting precise geometric and morphological characteristics. The leaves can be classified according to these characteristics. The classification of plants as well as the initial diagnosis of certain plant diseases are dependent on the morphological characteristics of leaves. An integral component of this proposed system is an artificial vision system, also known as a camera. The system is equipped with a collection of image-processing techniques and a feedforward neural network classifier. A fuzzy surface selection strategy was employed to refine the selection of relevant features. The photographs with the smallest distances were selected and ordered based on their level of similarity to the query after performing the requested image calculation. Islam et al. [13] successfully determined the number of pixels in the paddy leaves that were impacted by the disease using innovative image processing techniques they developed. By employing the K-means clustering algorithm, the original image was partitioned into three distinct clusters based on color. This allowed for a more thorough exploration of the information. In these shots, both affected and unaffected areas of leaves were included to determine the percentage of photos with affected pixels. A pixel count was conducted for both the affected and unaffected regions. The final stage involved calculating the proportion of pixels that were impacted. The severity of the issue was evaluated and recommendations were provided for managing Leaf blast disease. Deshmukh et al. [14] proposed utilizing an image processing approach. They use a repository of images of paddy leaf as their source of information. The findings of this study could potentially facilitate the early identification of paddy diseases. This eradicates the subjective nature of conventional approaches and the inaccuracies that arise from human involvement. A more efficient pesticide dosage calculation system will enable farmers to reduce costs and minimize environmental damage. Pavithra et al. [15] used picture samples that exhibited visual indications of a disease to diagnose patients. The k-means segmentation technique was employed to identify and partition the affected regions into their respective segments. The color texture characteristics extracted from each segmented area were used as inputs for support vector machine and artificial neural network classifiers. The SVM classifier outperformed the ANN classifier in the conducted study. The study employs pattern recognition and image processing techniques to classify real-world agricultural diseases, thereby demonstrating practical application. On average, they attained a precision level of approximately 95.5%.

2.3 Comparative Analysis and Summary

Table 2.3.1. Shows a comparative analysis of previous research works.

Table1: Comparative analysis of previous research works

Reference	**Segmentation**	**Features**	**Classifier**	**Accuracy**

Kulkarni et.al 2016 [1]	K-Means Clustering	Texture Features	CNN	96.30%
Kulkarni et.al 2013 [2]	Threshold Segmentation	DCT Texture+ Feature DWT	SVM	94.45%
Kulkarni et.al 2011 [4]	images Segmentation	Texture Features	Neural Networks	91%
Al-Hiary et.al 2011 [5]	K-Means Clustering	Texture Features	Neural Networks	94%
Camargo et.al, 2009 [6]	Threshold Segmentation	Shape+ Texture+ Dispersio+Grayleve+Histogram of Frequencies	SVM	93.10%
Karmokar et.al, 2015 [7]	neural Segmentation	Texture Features	Neural Network Ensemble	91%

Islam et.al, 2015 [13]	Thresholding, Color-based Segmentation. K-Means Clustering	Texture Features	SM	48.65%
Pavithra et.al, 2015 [15]	K-Means Segmentation	Color Texture Features	SVM	95.50%

Previous research has shown that several authors have delved into this field of study. The research established a criterion of 96% accuracy as satisfactory. The majority of writers used the Support Vector Machine (SVM) and Neural Network techniques.

2.4 Scope of the Problem

The majority of farmers are unable to read or write, hence lacking the knowledge to operate this system. Consequently, my first task is to acquire the knowledge on how to use them. That will greatly impede the establishment of this kind of system. Another significant issue is the limited advancement of agricultural technology, particularly in the realm of artificial intelligence (AI). Consequently, due to its affordability, several farmers are unable to use it. In order to address this issue, some Intelligent Transportation Systems (ITS) have been developed. However, the successful implementation of these systems in my nation necessitates the installation of a significant number of surveillance cameras across the whole area. These novel programmers integrate electrical components with other technologies, including computer networks, sensors, and communication networks. It integrates data about vehicles, people, and roadways and transmits it instantly to enhance the safety, efficiency, and comfort of both public and private transportation. Efficiently managing my system necessitates a complex procedure and the expertise of skilled personnel, therefore requiring a top-notch instrument that performs admirably. Searching for information proved to be a challenging endeavor since it required capturing many photographs from various perspectives.

2.5 Challenges

Several challenging scenarios that may arise throughout the development of this study, considering the whole system, include:

- Data transfer: Preserving a substantial volume of data from a mobile device, such as the images I have taken with my phone, is a challenging task. Additionally, I intended to use a USB cable to facilitate the transmission of the photographs from my mobile device to the PC. I had a multitude of diverse images on my mobile devices, which resulted in a lengthy process of transferring them all.
- Data collection: Acquiring a substantial and varied dataset of photographs depicting both healthy and sick rice leaves might pose difficulties, especially when there are restrictions on accessing fields and crops.
- Time Complexity: Due to the purported nature of the device as a real-time detection system, I want to promptly identify it in order to minimize the processing time required for detection.
- Data pretreatment involves removing noise, correcting for lighting and color fluctuations, and normalizing the data for training. This procedure may be time-consuming and needs skill in image processing.
- Model selection and tuning: The process of selecting the most suitable deep learning model architecture and optimizing the hyper parameters for best performance may be complex and may need knowledge in machine learning and deep learning.
- Model training: The process of training a deep learning model may be computationally demanding and may need the use of specialized hardware such as GPUs or TPUs.
- Model assessment: Assessing the effectiveness of the deep learning model using suitable metrics and procedures like cross-validation and testing on new data may be difficult and necessitates proficiency in machine learning evaluation.
- Deployment of the trained deep learning model in a real-world environment may need extra efforts, such as integrating it with pre-existing systems or developing a user-friendly interface.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

With the aim of aiding farmers, I want to develop a technique for ascertaining the characteristics of the paddy leaf disease. I experimented with various combinations of EfficientNetB2, MobileNetV4, ResNet50V2, and DenseNet121 until I discovered a successful setup. Using the Tensorflow framework, I have developed my own pre-trained models based on EfficientNetB2, MobileNetV4, ResNet50V2, and DenseNet121 architectures. I created my own exclusive algorithms that were pre-trained and used an unprocessed collection of pictures as my training dataset.

3.2 Proposed System

Initially, I chose my dataset and then uploaded it to my Google Drive. I conducted my technique using Google Colab. I uploaded my dataset to Google Drive and then chose the Drive inside Colab. Next, I provide the five classes in order to thoroughly analyze them in my implementation. I included the three directories from my dataset in my implementation. Load all the photos as 128 by 128 pixel RGB color images. First, I convert the photos into an array. Then, I do a shift operation on the array and divide it. After that, I allocate around 20% of the data for testing purposes and the remaining 80% for training. Subsequently, I performed label encoding for all the dataset names. Next, I import four models. To achieve this, I have used a blend of transfer learning and pre-trained models such as EfficientNetB2, MobileNetV2, ResNet50V2, and DenseNet121. In order to choose such pre-trained models, I need to release something in my implementation. The last stage was assessing the four models I had created. I used pre-trained models, including EfficientNetB2, MobileNetV2, ResNet50V2, and DenseNet121, to train a model by extracting features from the photographs. Subsequently, I conducted an analysis of the obtained data. As a concluding measure, I constructed a trainable model using my own pre-trained models: EfficientNetB2, MobileNetV2, ResNet50V2, and DenseNet121. My pre-trained models consist of six layers: base-model, Flatten, Dropout, Dense512, Dense256, and dense5. In addition, I use the Adam optimizer and accuracy matrix for the purpose of compilation.

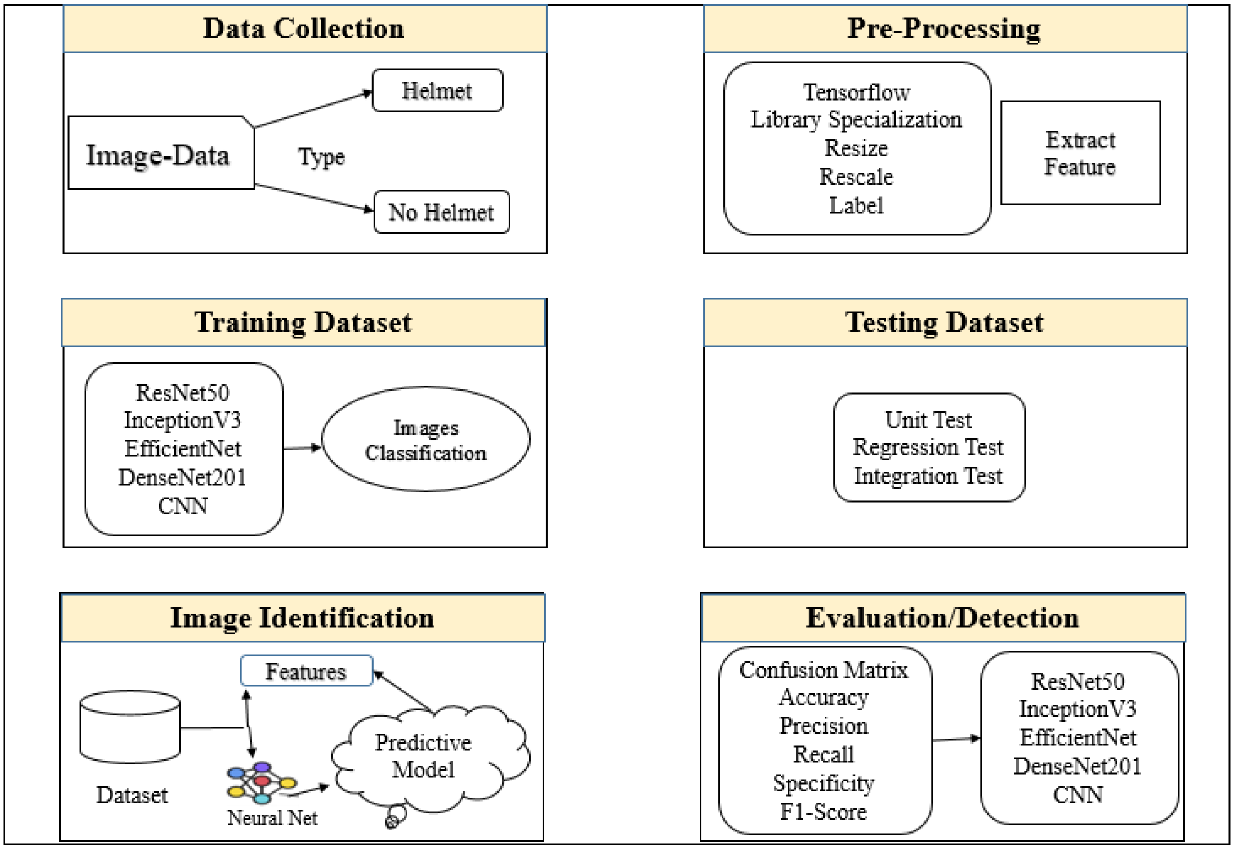

Figure 3.1: Working method of the proposed methodology.

3.3 Dataset

Getting the facts was the first thing I did to put my mental theory into action. A picture library was used as the base for building my model. That is, I was using mobile devices like smartphones and computers to collect data, takas, and pictures from web sources. The data in this set came from the field and showed all of the effects of paddy leaf. This happens after the picture data has been taken in its raw form. So that my model could load new data faster, I made the dots smaller. Out of the 212 pictures in the collection, each one had at least 128 pixels on the longest side. The information was split into three groups, and each group had five types. These were Brown Spot, Healthy Leaf, Leaf Blast, Leaf Blight, and Leaf Smut.

3.4 Implementation Procedure

3.4.1 Data Preprocessing

The importance of meticulous pre-processing of data cannot be overstated. The reason for this is that my collection consists of images of different dimensions. I use three distinct groups for testing, training, and validation purposes. Each group consists of a certain number of categories. I assess each of the three groups and five categories. If the visuals have varying dimensions, it is not possible to teach them accurately. In order to prevent this issue, I have adjusted the size and arrangement of the photos such that their longest side measures precisely 128 pixels. The dataset was trained using many models, such as EfficientNetB2, MobileNetV4, ResNet50V2, and DenseNet121, all of which need a large number of photos. I had sufficient information to adequately train my dataset. I have used RGB color-encoded photos and photographs containing distinct color channels. By using much superior photos throughout the dataset training process, I achieved improved outcomes.

3.4.2 Pre-Trained Model (MobileNetV2)

MobileNetV2 is an innovative approach to mobile vision and image classification that is based on convolutional neural networks. MobileNetV2 distinguishes itself from other models due to its minimal resource requirements for performing transfer learning. Given the frequent absence of a graphics processing unit (GPU) or inadequate computational capacity in mobile devices, embedded systems, and PCs, this technique is very suitable. Web browsers effectively use their advanced computational, visual processing, and storage capabilities to perform optimally.

I implement it on code-

1. Take dataset
2. MobileNet V2 Transfer Learning
3. Import Tensor flow and libraries
4. Prepare and train data
5. Creating training and validation sets
6. Preprocessing data
7. Initializing the base model
8. Add Extra layers for Pre-trained Model
9. Compile all the model

10. Train model
11. Visualizing the training and Validation performanc.

Here I used 6 layers. They are-

1. flatten_
2. Dropout_8
3. dense_512
4. Dense_256
5. dense_5
6. base_model

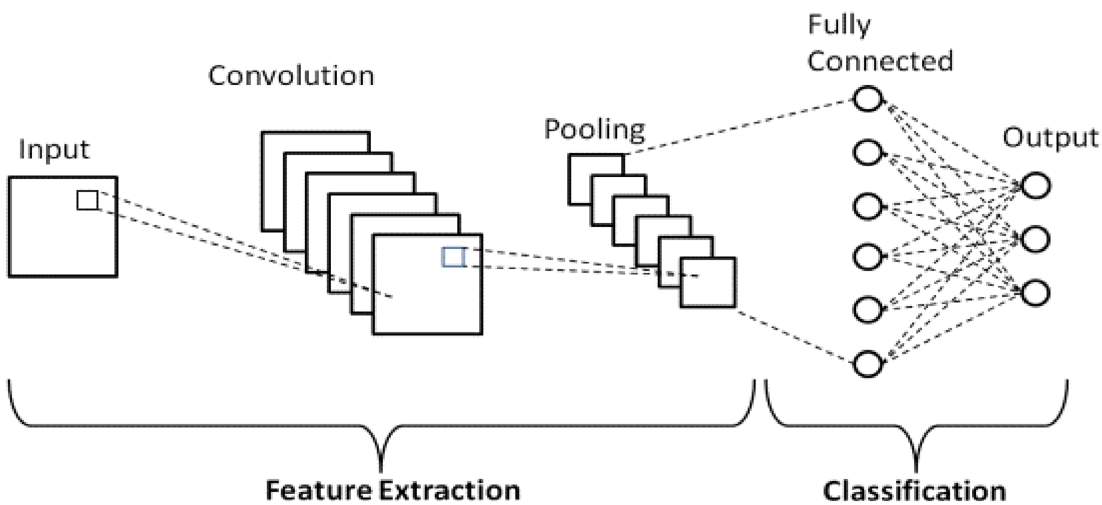

Figure 3.4.1: Working method of MobileNet V2

3.4.3 Pre-Trained Model (EfficientNetB2)

EfficientNetB2 used as a sophisticated variable to uniformly adjust the network's depth, breadth, and resolution. This enables the creation and effective scaling of convolutional neural networks. The EfficientNetB2 scaling approach does not arbitrarily scale network width, depth, and resolution. Instead, it expands these components evenly based on a predetermined set of scaling parameters. An efficient method to significantly boost computer processing power is by expanding the network's depth, breadth, and image size. The values for and may be determined using a simple grid search on the initial model. EfficientNetB2 use a sophisticated parameter to enhance the scale, complexity, and accuracy of a network in a proportionate manner. The compound scaling approach is based on the concept that in order to detect tiny components in an image, it is necessary to use a neural network with increased levels and channels, as well as a larger initial image.

I implement it on my code-

1. Classification of Custom Task.
2. Importing the EfficientNetB2 Dependencies.
3. Import EfficientNetB2 for Choosing EfficientNetB2 Model.
4. Create a Custom EfficientNetB2 for Training the Dataset.
5. Create the Custom EfficientNetB2 for the Training Job.

Here I used 6 layers. They are-

1. Flatten
2. Dropout_8
3. Dense_512
4. Dense_256
5. Dense_5
6. base_model

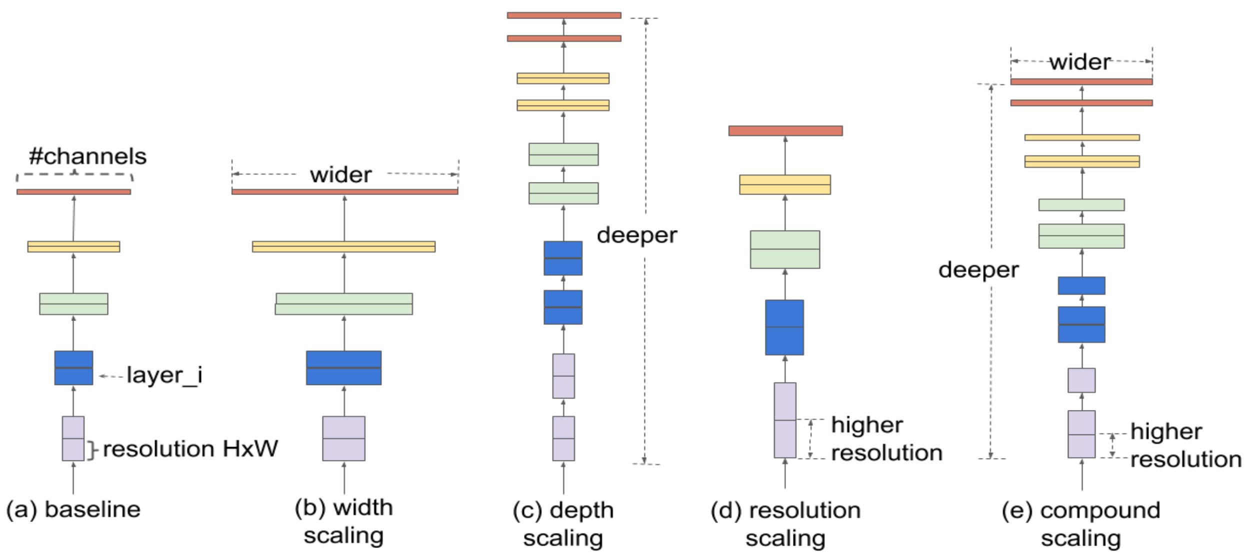

Figure 3.4.2: Working method of EfficientNetB2

3.4.4 Pre-Trained Model (DenseNet121)

DenseNet121 is a deep learning model architecture that employs dense connections across layers to facilitate feature reuse and enhance parameter economy. The architecture comprises of convolutional layers, dense blocks, and transition layers. During the training process, the model acquires the ability to minimize a loss function by using backpropagation and stochastic gradient descent optimization. DenseNet121 is very efficient for image classification jobs that have a scarcity of training data and include intricate visual characteristics.

I implement it on my code-

1. Install Monk.
2. Take Dataset.
3. Train the models.
4. Results Training.
5. Deploy the models through API.
6. Running the API.
7. Conclusion.

Here I used 6 layers. They are-

1. Flatten
2. Dropout_8
3. Dense_512
4. Dense_256
5. dense_5
6. base_model

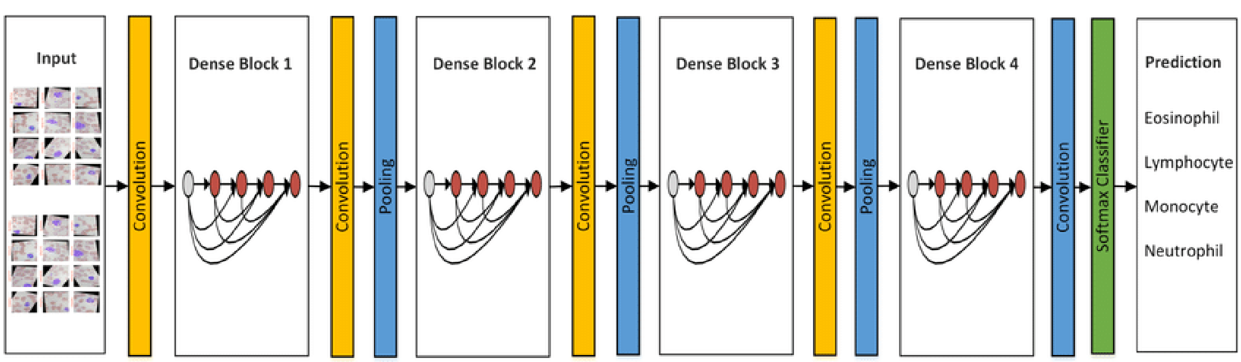

Figure 3.4.3: DenseNet121 model's architecture.

3.4.5 Pre-Trained Model (Resnet50V2)

The area of computer vision employs a deep learning model known as Residual Network (ResNet50V2). The term "convolutional neural network" (CNN) is used to describe this kind of neural network due to its architecture, which consists of several convolutional layers. Prior iterations of Convolutional Neural Networks (CNN) proved to be ineffective due to their inability to accommodate a large number of layers. Nevertheless, when the researchers included more layers, they confronted the predicament known as the "vanishing gradient" issue.

I implement it on my code-

1. Import required libraries and dataset
2. Partition and Visualization the Dataset.
3. Import the pre-trained machine learning model
4. Train and Evaluate Model
5. The model and classify images

Here I used 6 layers. They are-

1. Flatten
2. Dropout_8
3. Dense_512
4. Dense_256
5. dense_5
6. Base_model

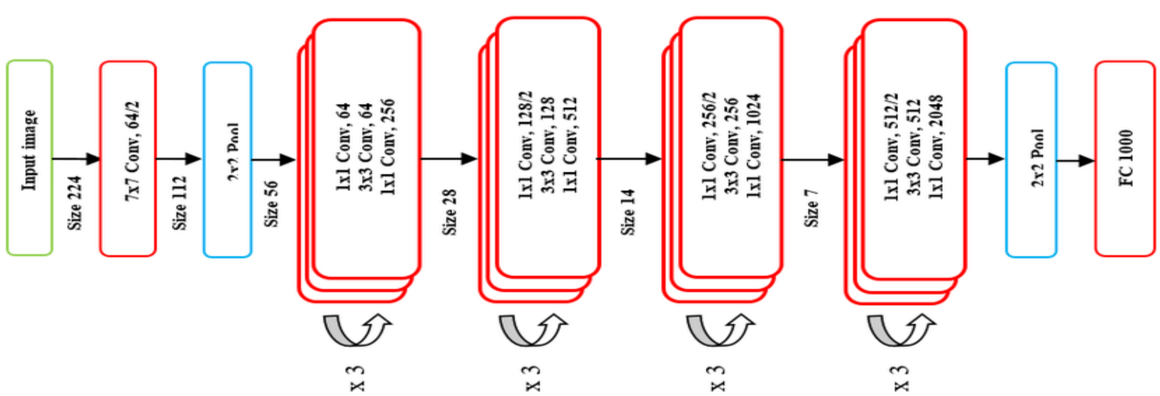

Figure 3.4.4: Resnet50V2 model Workflow.

3.4.6 Model Tuning

During the tuning phase, it is necessary to experiment with various options for a set of hyper parameters. I have increased the quantity of models in each batch to a total of 32 for my project. I refrained from modifying any of the classes in order to avoid any potential ambiguity in my collection. By changing the date to 20, the caliber of my models significantly improves.

3.4.7 Model Training

My study indicates that training is very necessary for the successful use of models. While preventing any additional alterations from being made to the networks that were trained in the past, namely EfficientNetB2, MobileNetV4, ResNet50V2, and DenseNet121. In order to complete the necessary training, I went through twenty training rounds with a total of thirty-two participants in each batch. During the time that I was eating, I had three different choices open to me. It is necessary to make adjustments to a number of hyperparameters included inside the setup file in order to train the model. The learning rate, batch size, picture filtering, and epoch count are all examples of hyperparameters that fall into this category. In order to facilitate future reference and the possibility of change, the structure of the model and its training parts have been kept. During the time that I am in the training loop, I load both the training dataset and the validation dataset concurrently. The Adam Optimizer with Cross-Entropy Loss is the method that I use to train the model so that I can guarantee its effectiveness. Following each time step in the validation set, the models are evaluated, and the most correct model is saved for use at a later time. The training and confirmation error and loss plots, in addition to the overtraining error and loss plot, are kept until the training is finished without being altered.

3.4.8 Experimental Environment

That Project was done on a machine that had an Intel(R) Core(TM) i9-7400U processor running at 3.40GHz and 4.10GHz, a 4GB NVIDIA Graphics card 730 graphics processing unit (GPU), 16 gp of random access memory (RAM), and Windows 11 installed. My model is run with the help of the open-source and free Google Colab Notebook. With the help of Tensorflow and Keras, the model was built.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Confusion Matrix

A confusion matrix is a table used in machine learning to figure out how well a classification model works. There are predictions that are right, wrong, mostly right, and completely wrong based on the model. When the model correctly predicted the positive class, these are called "true positives." When it correctly predicted the negative class, these are called "true negatives." When the model makes a fake positive, it says that the class is positive when it really is negative. When it makes a false negative, it says that the class is negative when really it is positive. You can use confusion matrices to figure out how well a classification method works generally and find its weak spots.

Figure 4.1.1 shows the confusion matrix of the pre-trained EfficientNetB2 model that used the photos I supplied for training. The number of photographs that were correctly recognised is shown along the diagonal.

	Brown Spot	Healthy Leaf	Leaf Blast	Leaf Blight	Leaf Smut
Brown Spot	37	3	0	2	1
Healthy Leaf	0	43	4	0	1
Leaf Blast	2	7	31	0	0
Leaf Blight	0	0	0	44	0
Leaf Smut	0	0	0	0	48

Figure 4.1.1: Confusion matrix of EfficientNetB2

Figure 4.1.2 shows the confusion matrix of the pre-trained EfficientNetB2 model that used the photos I supplied for training. The number of photographs that were correctly recognised is shown along the diagonal.

	Brown Spot	Healthy Leaf	Leaf Blast	Leaf Blight	Leaf Smut
Brown Spot	30	2	2	6	3
Healthy Leaf	0	42	6	0	0
Leaf Blast	1	7	29	1	2
Leaf Blight	1	1	0	42	0
Leaf Smut	5	1	2	3	37

Figure 4.1.2: Confusion matrix of MobileNetV2

Figure 4.1.3 shows the confusion matrix of the pre-trained EfficientNetB2 model that used the photos I supplied for training. The number of photographs that were correctly recognised is shown along the diagonal.

	Brown Spot	Healthy Leaf	Leaf Blast	Leaf Blight	Leaf Smut
Brown Spot	7	5	7	17	7
Healthy Leaf	0	34	14	0	0
Leaf Blast	0	14	20	6	0
Leaf Blight	1	4	1	36	2
Leaf Smut	0	5	1	8	34

Figure 4.1.3: Confusion matrix of ResNet50V2

Figure 4.1.4 shows the confusion matrix of the pre-trained EfficientNetB2 model that used the photos I supplied for training. The number of photographs that were correctly recognized is shown along the diagonal.

	Brown Spot	Healthy Leaf	Leaf Blast	Leaf Blight	Leaf Smut
Brown Spot	31	1	3	6	2
Healthy Leaf	0	42	6	0	0
Leaf Blast	2	10	26	1	1
Leaf Blight	2	0	0	41	1
Leaf Smut	3	1	1	10	33

Figure 4.1.4: Confusion matrix of DenseNet121

4.2 Result

Table 4.2.1: Performance of model's precision, recall, f1 score

Categories	Classes	Model			
		EfficientNetB2	MobileNetV2	ResNet50V2	DenseNet121
Brown Spot	Precision	95	81	88	82
	Recall	86	70	16	72
	F1 score	90	75	27	77
Healthy Leaf	Precision	81	79	55	78

	Recall	90	88	71	88
	F1 score	85	83	62	82
Leaf Blast	Precision	89	74	47	72
	Recall	78	72	50	65
	F1 score	83	73	48	68
Leaf Blight	Precision	96	81	54	71
	Recall	100	95	82	93
	F1 score	98	88	65	80
Leaf Smut	Precision	96	88	79	89
	Recall	100	77	71	69
	F1 score	98	82	75	78

Table 4.2.2: The performance of accuracy

Model	Accuracy
EfficientNetB2	91%
MobileNetV2	81%
ResNet50V2	59%
DenseNet121	78%

The table above shows all of the algorithms along with their f1-scores and the accuracies, precisions, recalls, and specificities that go with them. DenseNet121 has an f1-score of 78%, MobileNetV2 has f1-score of 81%, and ResNet50V2 has a f1-score of 59%. However, EfficientNetB2 has an f1-score of 91% and the best accuracy of all of my networks, which is clearly shown here.

4.3 Learning Curves

A learning curve, in its most basic definition, is a graph that displays the change in a model's success rate as a function of time or experience. One common use of learning curves in machine learning is troubleshooting algorithms that exhibit incremental improvement upon exposure to a training dataset. You can see the accuracy and loss from both the train and validation runs in the learning curve image.

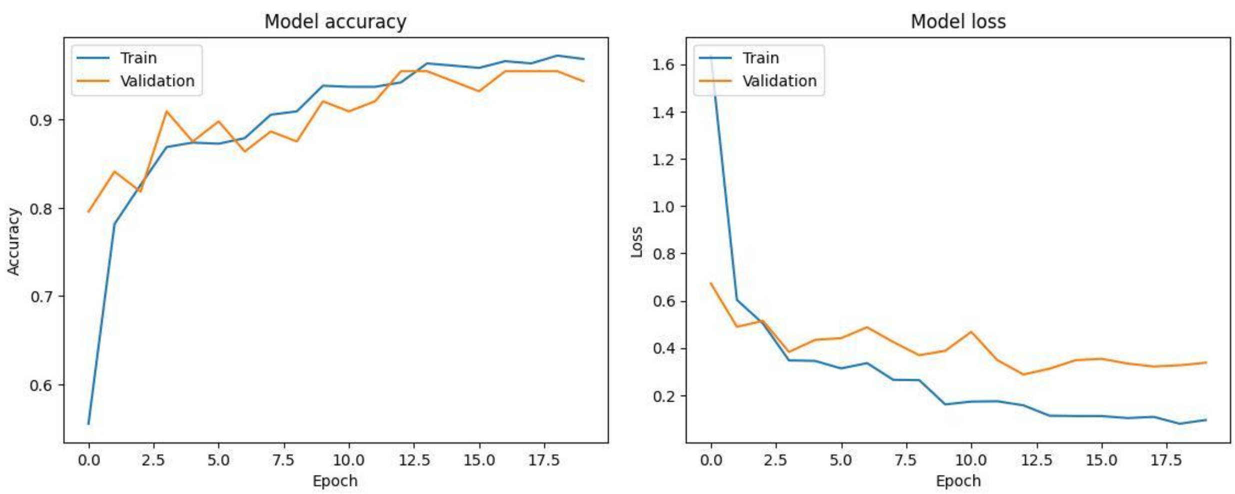

Figure 4.3.1: learning curve of custom EfficientNetB2.

Figure 4.3.1 exhibits the training and validation loss, as well as the accuracy, of EfficientNetB2 during the training and validation procedure. Both images are shown in the color blue to signify that they reflect the training accuracy and training loss. On the other hand, orange is used to indicate the metrics of validation loss and validation accuracy. At the beginning, the training loss was around 0.2761. Nevertheless, as the number of epochs rises, the loss diminishes. After many rounds, the loss function approaches a peak and remains stable at around 0.9103. According to my observation, the training had an efficacy of around 0.0725. As the number of epochs increases, the accuracy also increases. After many iterations, the precision stays consistent and converges to a value of 0.9832. Nevertheless, the validation loss stays constant during all epochs. The range of losses spans from 0.2761 to 0.9103, given an initial validation accuracy of 0.0725. Extending the length of epochs might potentially enhance the accuracy of confirmation. As the number of epochs rises, the validation accuracy stabilizes and approaches a final conclusion of 0.9832.

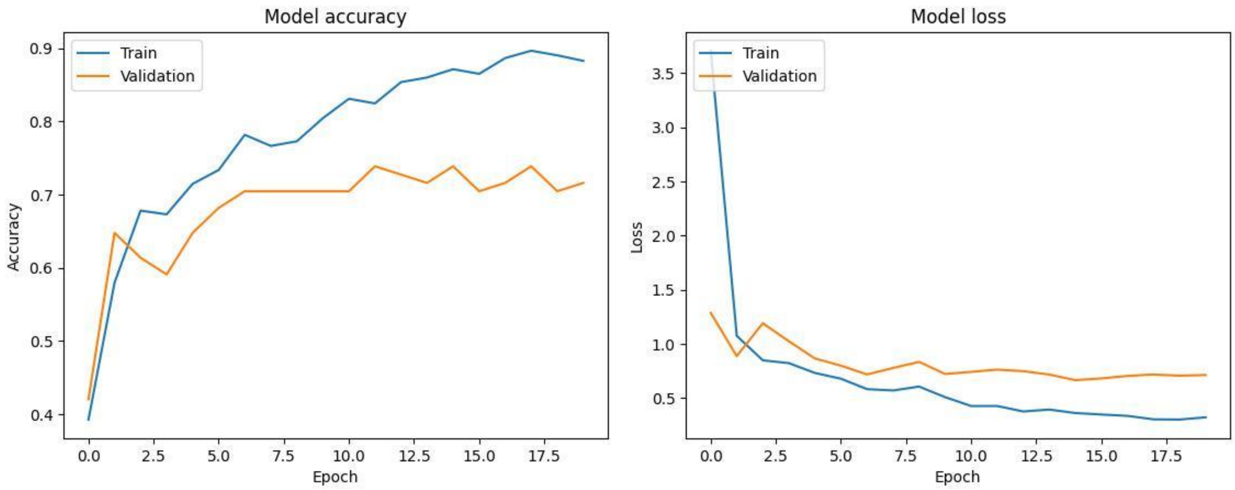

Figure 4.3.2: learning curve of MobileNetV2

Figure 4.3.2 illustrates the training and validation loss, as well as the accuracy, of EfficientNetB2 during the training and validation stages. Both representations are shown in blue color to signify that the training accuracy and training loss are denoted by the color blue. On the other hand, orange is used to indicate the metrics of validation loss and validation accuracy. At the beginning, the training loss was around 0.5683. Nevertheless, as the number of epochs rises, the loss diminishes. After many repetitions, the loss function achieves a peak and remains stable at about 0.8072. According to my observation, the instruction had an efficacy of around 0.1883. As the number of epoch's increases, the accuracy also increases. After many iterations, the precision stays consistent and converges towards a value of 0.9383. Nevertheless, the validation loss stays constant during all epochs. The range of losses spans from 0.5683 to 0.8072, whereas the initial validation accuracy was 0.1883. Extending the length of epochs might potentially enhance the accuracy of confirmation. As the number of epochs rises, the validation accuracy stabilizes and approaches a final result of 0.9383.

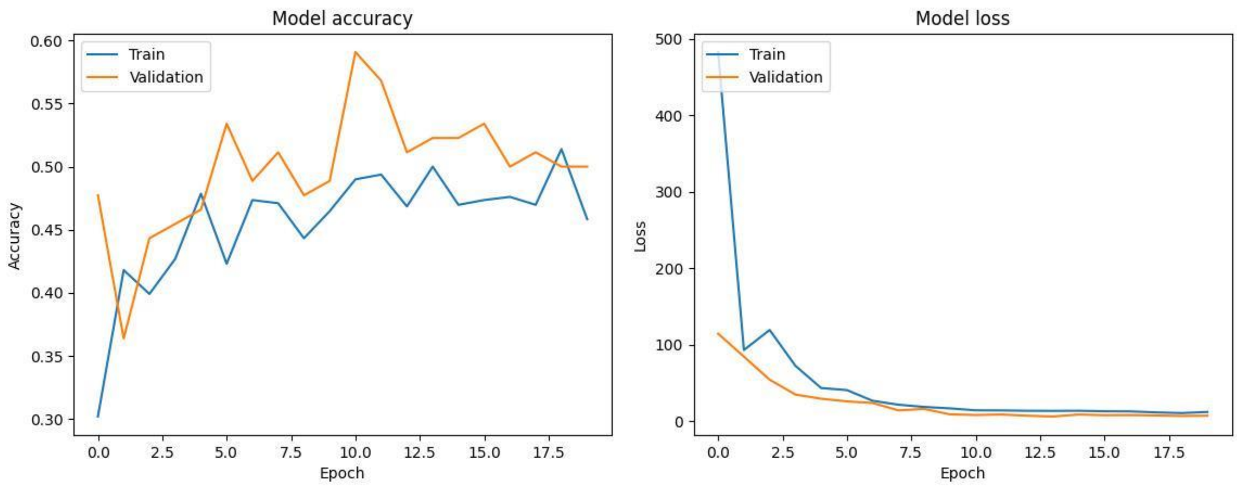

Figure 4.3.3: learning curve of ResNet50V2

Figure 4.3.3 illustrates the training and validation loss, as well as the accuracy, of EfficientNetB2 during the training and validation stages. Both representations are shown in the color blue to signify that the training accuracy and training loss are being represented by that particular color. On the other hand, orange is used to indicate the metrics of validation loss and validation accuracy. In the beginning, the training loss was around 4.1902. Nevertheless, as the number of epochs rises, the loss diminishes. After many cycles, the loss function approaches a peak and remains stable at around 0.5874. According to my observation, the instruction had an efficacy of around 3.2669. As the number of epoch's increases, the accuracy also increases. After many iterations, the precision stays consistent and converges to a value of 0.6031. Nevertheless, the validation loss stays constant during all epochs. The range of losses is from 4.1902 to 0.5874, whereas the first validation accuracy was 3.2669. Extending the length of epochs might potentially enhance the accuracy of confirmation. As the number of epochs rises, the validation accuracy plateaus and converges to a final result of 0.6031.

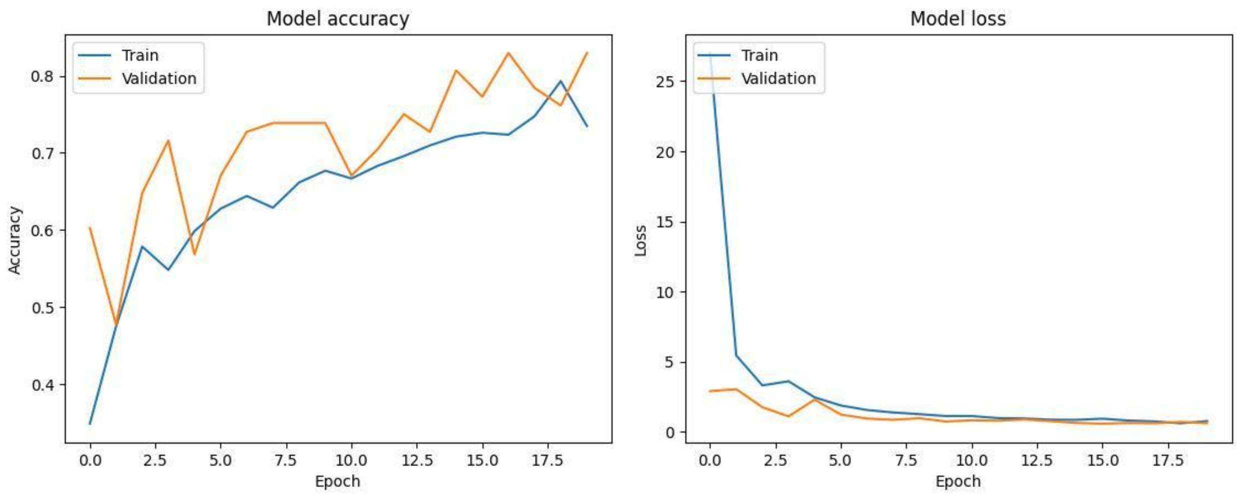

Figure 4.3.4: learning curve of DenseNet121

Figure 4.3.4 exhibits the training and validation loss, as well as the accuracy, of EfficientNetB2 during the training and validation procedure. Both illustrations are colored blue to signify that the training accuracy and training loss are shown using the color blue. On the other hand, orange is used to indicate the metrics of validation loss and validation accuracy. At the beginning, the training loss was around 0.5440. Nevertheless, as the number of epochs rises, the loss diminishes. After many repetitions, the loss function achieves a plateau and remains stable at about 0.7758. According to my observation, the instruction had an efficacy of around 0.3977. As the number of epochs increases, the accuracy also increases. After many iterations, the precision stays consistent and converges to a value of 0.8587. Nevertheless, the validation loss stays constant during all epochs. The range of losses spans from 0.5440 to 0.7758, given an initial validation accuracy of 0.3977. Extending the length of epochs might potentially enhance the accuracy of confirmation. As the number of epochs rises, the validation accuracy plateaus and the ultimate result approaches 0.8587.

CHAPTER 5

Impact on Society, Environment and Sustainability

5.1 Impact on Society

Deep learning has the capacity to significantly impact the globe, particularly in the field of agriculture, by effectively identifying the causes of paddy leaf disease. To minimize crop output losses, farmers should promptly detect and manage paddy leaf diseases. This not only ensures sufficient food availability for all individuals, but it also enhances the financial prosperity of farmers. Early detection and treatment of paddy leaf diseases might potentially reduce the reliance on chemical pesticides for farmers. Implementing this approach would enhance the sustainability of agricultural practices and contribute to improved environmental outcomes. Early detection of paddy leaf diseases may effectively prevent their spread. Deep learning has the capacity to expedite and enhance the identification of paddy leaf diseases compared to conventional approaches, such as visual inspection by a human observer. This would enable the testing of a greater variety of crops within a given timeframe. Providing farmers with training on the use of deep learning-based systems for diagnosing rice leaf diseases might enhance their decision-making abilities and enable them to adopt more effective preventive measures to safeguard their crops. Reducing crop losses due to disease will result in an increased supply of crops available for sale. This would result in a decrease in pricing for consumers and enhance accessibility of food for those in need. Ultimately, the use of deep learning in identifying paddy leaf disease has the promise of benefiting society. This may be achieved by increasing crop productivity, advocating for sustainable agricultural methods, enhancing farmer education, and expanding market access for products.

5.2 Impact on the Environment

Utilizing deep learning for the diagnosis of paddy leaf disease has the potential to contribute positively to the environment. Paddy leaf disease detection has been extensively discussed in several studies due to its notable social and environmental consequences. Deep learning has the potential to enhance the precision of pesticide use in agriculture, hence minimizing the overall need for pesticides. The prompt eradication of paddy leaf disease is essential to accomplish this task. This might have positive implications for the environment since pesticides have the potential to harm non-target species and contaminate both the soil and water. Deep learning algorithms have the potential to enhance the sustainability of agricultural practices and minimize environmental damage by decreasing the reliance on pesticides and promoting the use of precise and efficient treatments. Utilizing deep learning algorithms to detect illnesses in paddy leaves has the potential to prevent the transmission of diseases to non-target crops, hence preserving agricultural ecosystem diversity. Deep learning algorithms have the potential to assist agriculture in reducing its carbon footprint by eliminating the need for physical inspections and pesticides. In summary, using deep learning to detect paddy leaf disease might have a positive impact on the environment. This may be achieved by promoting the adoption of sustainable farming practices, reducing pesticide use, conserving biodiversity, and minimizing the carbon emissions associated with agriculture.

5.3 Sustainability Plan

The sustained effectiveness and influence of the system rely on a deep-learning-based approach for identifying paddy leaf illnesses that may be used over an extended period. I need high-quality datasets for the purpose of training and evaluating deep learning models, and I also need to effectively arrange and manage these datasets. Data collection, organization, and preservation should be integrated into a sustainability plan. In order for users to effectively use and get value from the system, it is imperative that farmers, local communities, and other stakeholders be actively engaged and included in its implementation. Incorporating participatory design into a sustainability plan is essential for engaging and gathering feedback from the community. In order for the system to function well, it is necessary for me to develop dependable and precise deep learning algorithms. An effective approach to achieving sustainability should include the use of crowdsourcing and other methodologies to create, evaluate, and enhance algorithms. I will construct and operate the technological framework that enables the system to be expanded and user-friendly. An effective approach to achieving sustainability should include the use of cloud computing and other initiatives aimed at creating scalable technologies. In order to ensure the longevity of the system, I will enhance the competencies and expertise of both users and stakeholders. In order to enhance capability, a comprehensive approach to long-term viability should include instructional initiatives and user-friendly interfaces. Furthermore, achieving long-term success is contingent upon maintaining financial viability. An effective approach to achieving sustainability should include collaborations between public and private entities, securing financial support via grants, and implementing pay-for-service frameworks. Ultimately, a comprehensive approach to use deep learning for the detection of paddy leaf disease necessitates the integration of data management, community involvement, algorithm advancement, technical infrastructure, skill development, and financial viability. This study focuses on the identification of paddy leaf disease using various algorithmic methods and with the use of deep learning.

5.4 Ethical Aspects

For the purpose of protecting the privacy of people whose data may be included in our dataset, I have done all in my power to secure the required permits and to adhere to ethical norms. Any information that may be used to identify a specific individual has been deleted or anonymized. I have implemented steps to reduce the impact of biases in our dataset and models, and I conduct performance evaluations on a regular basis in order to detect and rectify inequalities that exist across different demographic groups. By giving comprehensive documentation of my methods, making my code and trained models accessible to the public, and committing to continuing monitoring and assessment, I make openness and accountability a priority in my work. I am aware of the potential contribution that our study may make to society, especially in terms of enhancing agricultural productivity and ensuring food security. In order to guarantee that our results are used in a responsible and ethical manner, I promote conversation and engagement with the many stakeholders.

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

My capacity to identify plant diseases has been enhanced as a result of the knowledge included in this research. One of the long-term goals of this area of study is to develop an automated technique for identifying the degree of severity of an illness that has just been discovered. Furthermore, I have devised cutting-edge algorithms for identifying and categorizing the sickness, which are not only accurate but also simple to understand. Researchers will get a more nuanced knowledge of farmers' views on risk and how they may categories it if this experiment can be duplicated in a variety of circumstances, and the geometry of the picture capture that was employed makes it feasible for this to happen. As a consequence of this, the appropriate authorities could decide to implement procedures that will allow the general public to identify the problem and choose a course of action. In light of the fact that there is currently no system in place that allows for the timely identification and reporting of leaf diseases, this effort was created. It is via this method that protection is provided for farmers. Images of a certain category of leaf diseases should be extracted since it is essential for reasons of security. I will begin by classifying the images of leaf diseases into the appropriate groups. The first thing I do is collect the condition-based dataset via the categorization of pictures. In the event that any frames do not comply, they will either be destroyed or ignored. I am able to get an astounding 91% accuracy by using a dataset that has 212 rows and working with six distinct strategies. Throughout the study, I identify a variety of limits associated with my raw dataset. These limitations include a maximum number of right answers and a minimum number of outliers that is not zero. Taking into consideration that comparison, it is tough to get a superior outcome. Accuracy will be improved by locating a number that is comparable.

6.2 Future Work

Dataset 212 indicates that the research has been completed. By using those statistics, I was able to determine what the issue was with the leaf. I would want to continue my research on this topic in the future. In the not-too-distant future, I believe that additional strategies will be used in order to reduce the selection of features while simultaneously enhancing accuracy. The future will bring about the appearance of new leaf diseases, and I will include them in my studies. More modifications may be made to the approach in order to determine whether or not they make it more predictable and capable of being used on a wide scale. Last but not least, I would want to experiment with other methods of decision-making, such as multi-class voting systems and various forms of cutting-edge CNN, such as those that gather data and look at incentives. a willingness to experiment with new rules, such as those governing groups and methods of data collection.

Reference

[1] Sladojevic, S., Arsenovic, M., Anderla, A.,, D. and Stefanovic, D., 2016. Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016.

[2] Akhtar, A., Khanum, A., Khan, S.A. and Shaukat, A., 2013, December. Automated plant disease analysis (APDA): performance comparison of machine learning techniques. In 2013 11th International Conference on Frontiers of Information Technology (pp. 60-65). IEEE.

[3] Tijare, N.S. and Badnerkar, S.S., 2014. Image recognition based crop disease identification System: a survey. International Journal of Computer Science and Mobile Computing (IJCSMC), 3, pp.868-873.

[4] Kulkarni, A.H. and Patil, A., 2012. Applying image processing technique to detect plant diseases. International Journal of Modern Engineering Research, 2(5), pp.3661-3664.

[5] Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M. and Alrahamneh, Z., 2011. Fast and accurate detection and classification of plant diseases. International Journal of Computer Applications, 17(1), pp.31-38.

[6] Camargo, A. and Smith, J.S., 2009. An image-processing based algorithm to automatically identify plant disease visual symptoms. Biosystems engineering, 102(1), pp.9-21.

[7] Karmokar, B.C., Ullah, M.S., Siddiquee, M.K. and Alam, K.M.R., 2015. Tea leaf diseases recognition using neural network ensemble. International Journal of Computer Applications, 114(17).

[8] Rumpf, T., Mahlein, A.K., Steiner, U., Oerke, E.C., Dehne, H.W. and Plümer, L., 2010. Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. Computers and electronics in agriculture, 74(1), pp.91-99.

[9] Patil, J.K. and Kumar, R., 2012. Feature extraction of diseased leaf images. Journal of signal and image processing, 3(1), pp.60-63.

[10] Kaundal, R., Kapoor, A.S. and Raghava, G.P., 2006. Machine learning techniques in disease forecasting: a case study on rice blast prediction. BMC bioinformatics, 7, pp.1-16.

[11] Kailey, K.S. and Sahdra, G.S., 2012. Content-Based Image Retrieval(CBIR) For Identifying Image Based Plant Disease. International Journal of Computer Technology and Applications, 3(3).

[12] Tzionas, P., Papadakis, S.E. and Manolakis, D., 2005, October. Plant leaves classification based on morphological features and a fuzzy surface selection technique. In Fifth international conference on technology and automation, Thessaloniki, Greece (pp. 365-370).

[13] Islam, R. and Islam, M.R., 2015. An image processing technique to calculate percentage of disease affected pixels of paddy leaf. International Journal of Computer Applications, 123(12).

[14] Deshmukh, R. and Deshmukh, M., 2015. Detection of paddy leaf diseases. International Journal of Computer Applications, 975, p.8887.

[15] Pavithra, S., Priyadharshini, A., Praveena, V. and Monika, T., 2015. Paddy leaf disease detection using SVM classifier. International Journal of communication and computer Technologies, 3(01), pp.16-20.

[16] Trébuil, G., 2011. Rice production systems in Asia: The constant presence of an essential cereal on a continent in mutation.

[17] Papademetriou, M.K., 2000. Rice production in the Asia-Pacific region: issues and perspectives. Bridging the rice yield gap in the Asia-Pacific region, 220, pp.4-25.

[18] Islam, M.A., Shuvo, M.N.R., Shamsojjaman, M., Hasan, S., Hossain, M.S. and Khatun, T., 2021. An automated convolutional neural network based approach for paddy leaf disease detection. International Journal of Advanced Computer Science and Applications, 12(1).

[19] Rohan, D., Rohan, P., Rohith, B., Rohith, A., Rohith, T., Sri, K.R. and Reddy, M.R., PADDY LEAF DISEASE DETECTION USING PYTHON.

[20] Karmokar, B.C., Ullah, M.S., Siddiquee, M.K. and Alam, K.M.R., 2015. Tea leaf diseases recognition using neural network ensemble. International Journal of Computer Applications, 114(17).

Leaf

ORIGINALITY REPORT

18%	14%	8%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	9%
2	Submitted to Daffodil International University Student Paper	2%
3	Kazi Shakibur Rahman, Md. Rakibul Islam Rakib, M. Mirazus Salehin, Md. Rostom Ali, Anisur Rahman. "Assessment of paddy leaves disease severity level using image processing technique", Smart Agricultural Technology, 2024 Publication	1%
4	Submitted to CSU Northridge Student Paper	1%
5	"Communication and Intelligent Systems", Springer Science and Business Media LLC, 2024 Publication	<1%
6	Submitted to Grambling State University Student Paper	<1%