Deep-CNN for Plant Disease Diagnosis Using

Low Resolution Leaf Images

|  |  |
| --- | --- |
| By |  |
| Ashiqur Rahman, | ID:16172103385 |
| Md Hafiz Al Foisal, | ID:16172103116 |
| Md Hafijur Rahman, | ID:16172103388 |
| Md Ranju Miah, | ID:16172103376 |
|  |  |

Submitted in partial ful llment of the requirements of the degree of

Bachelor of Science in

Computer Science and Engineering



Department of Computer Science and Engineering

Bangladesh University of Business and Technology

July 2021

Declaration

We do hereby declare that the research works presented in this thesis entitled “Deep-CNN for Plant Disease Diagnosis Using Low Resolution Leaf Image" are the results of our own works. We further declare that the thesis has been compiled and written by us. No part of this thesis has been submitted elsewhere for the requirements of any degree, award or diploma, or any other purposes except for publications. The materials that are obtained from other sources are duly acknowledged in this thesis.

Approval

We do hereby acknowledge that the research works presented in this thesis entitled "Deep-CNN for Plant Disease Diagnosis Using Low Resolution Leaf Images" result from the original works carried out by Dr. Muhammad Firoz Mridha, Chairman and Associate Professor, Department of Computer Science and Engineering, Bangladesh University of Business and Technology. We further declare that no part of this thesis has been submitted elsewhere for the requirements of any degree, award or diploma, or any other purposes except for publications. We further certify that the dissertation meets the requirements and standard for the degree of Doctor of Philosophy in Computer Science and Engineering.

Dr. Muhammad Firoz Mridha

Chairman and Associate Professor

Department of CSE

Acknowledgement

We would like to express our heartfelt gratitude to the almighty Allah who o ered upon our family and us kind care throughout this journey until the ful lment of this research.

Also, we express our sincere respect and gratitude to our supervisor Dr. Muhammad Firoz Mridha, Chairman and Associate Professor, Department of Computer Science and Engineering, Bangladesh University of Business and Technology(BUBT). Without his guidance, this research work would not exist. We are grateful to him for his excellent supervision and for putting his utmost e ort into developing this project. We owe him a lot for his assistance, encouragement, and guidance, which has shaped our mentality as a researcher.

Finally, we are grateful to all our faculty members of the CSE department, BUBT, to make us compatible to complete this research work with the proper guidance and supports throughout the last four years.

Abstract

Plant diseases present a threat to food security, but early recognition is problematic in many areas due to a lack of necessary facilities. Plant disease diagnosis is critical in agriculture since diseases frequently restrict plant production capacity. Manual strategies to identify plant diseases, on the other hand, are often temporal, challenging, and lengthy. As a consequence, agricultural automation with automated identification of plant diseases is widely preferred. In the modern time, for the advancement of computer vision, detecting diseases utilizing leaf representation of a specific plant has already been implemented. Despite these challenges, most of the implemented models could only identify diseases of a particular plant using high-resolution images, which is quite expensive from a farmer's position. Because of the variation in leaf colours, aspect ratios, and congested backgrounds, detecting plant disease by low-quality images is difficult. With the advancement of deep convolutional neural networks (DCNNs), the field of object recognition from low-resolution images has seen significant progress. This paper explores an efficient plant disease identification model that combines multiple plant diagnosis for lowresolution images. The model inherits a multilabel classification system to classify both the plant and the specific disease simultaneously. We gathered data for the study and analysis from online articles, including leaf images of tomatoes, corn, and apples. For our research, we have

used various stand-ard convolutional neural network (CNN) architectures such as Xception, ResNet, DenseNet, and MobileNet to get better performance in this task. The result comparison looks like this: Xception, ResNet, DenseNet performs better only on High-resolution images, and MobileNet performs well in low-resolution images.

List of Tables

4.1 The table summarizes the plants and varieties of disease that the accumulated dataset includes. . . . . . . . . . . . . . . . . . . . . 37

4.2 This table represents the accuracy, precision, and recall of different architectures. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

List of Figures

|  |  |  |
| --- | --- | --- |
| 3.1 | The figure illustrates the workflow of the proposed system. . | 29 |
| 3.2 | The neural network architecture of plant diagnosis from leaf images | 31 |
| 3.3 | The Figure shows the disorder images of Apple (top left), Corn (bottom left), and Tomato (right). | 38 |
| 5.1.1 | semester Gantt Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 45 |
| 5.1.2 | semester Gantt Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 46 |
| 5.1.3 | semester Gantt Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 46 |
|  |  |  |

List of Abbreviations

**ANN** Artificial Neural Network.

**DL** Deep Learning

**CNN** Convolutional Neural Network.

**DCNN** Deep Convolutional Neural Network.

**ReLU** Rectified Linear Activation Function.

Contents

|  |  |  |  |
| --- | --- | --- | --- |
| Declaration | | | 2 |
| Approval | | | 3 |
| Acknowledgement | | | 4 |
| Abstract | | | 5 |
| List of Tables | | | 7 |
| List of Figures | | | 8 |
| List of Abbreviations | | | 9 |
| 1 | Introduction | | 12 |
|  | 1.1 | Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 12 |
|  | 1.2 | Problem Statement . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 13 |
|  | 1.3 | Problem Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 13 |
|  | 1.4 | Research Objectives . . . . . . . . . . . . . . . . . . . . . . . . . . . | 14 |
|  | 1.5 | Motivations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 15 |
|  | 1.6 | Flow of the Research . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 15 |
|  | 1.7 | Significance of the Research . . . . . . . . . . . . . . . . . . . . . . | 16 |
|  | 1.8 | Research Contribution . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
|  | 1.9 | Thesis Organization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
|  | 1.10 | Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 18 |
| 2 | Background | | 19 |
|  | 2.1 | Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
|  | 2.2 | Literature Review . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
|  | 2.3 | Problem Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 2.4 | Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 27 |
| 3 | Proposed Model | | | 28 |
|  | 3.1 | Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 28 |
|  | 3.2 | Feasibility Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 28 |
|  | 3.3 | Requirement Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 29 |
|  | 3.4 | Research Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 29 |
|  |  | 3.4.1 | Data Pre-processing . . . . . . . . . . . . . . . . . . . . . . . | 30 |
|  |  | 3.4.2 Image Preprocessing. . . . . . . . . . . . . . . . . . . . . . . . . | | 30 |
|  |  | 3.4.3 | Baseline Architecture. . . . . . . . . . . . . . . . . . . . . . . . | 30 |
|  |  | 3.4.4 | Transfer Learning . . . . . . . . . . . . . . . . . . . . . . . . . | 33 |
|  | 3.5 | Design, Implementation, and Simulation . . . . . . . . . . . . . . | | 35 |
|  | 3.6 | Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 35 |
| 4 Implementation, Testing, and Result Analysis | | | | 36 |
|  | 4.1 | Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 36 |
|  | 4.2 | Dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 36 |
|  | 4.3 | System Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 38 |
|  | 4.4 | Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 38 |
|  | 4.5 | Experiments Result and Comparisons . . . . . . . . . . . . . . . | | 40 |
|  | 4.6 | Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 41 |
| 5 | Standards, Constraints, Milestones | | | 42 |
|  | 5.1 | Standards (Sustainability) . . . . . . . . . . . . . . . . . . . . . . . . . | | 42 |
|  | 5.2 | Impacts (on Society) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 42 |
|  | 5.3 | Ethics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 43 |
|  | 5.4 | Challenges . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 43 |
|  | 5.5 | Constraints . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 44 |
|  | 5.6 | Timeline and Gantt Chart . . . . . . . . . . . . . . . . . . . . . . . . . | | 44 |
|  | 5.7 | Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 47 |
| 6 | Conclusion | |  | 48 |
|  | 6.1 | Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 48 |
|  | 6.2 | Future Works and Limitations . . . . . . . . . . . . . . . . . . . . . . | | 49 |

1 Introduction

1.1 Introduction

A disease occurring in food crops is a significant threat that has a significant economic effect, including its productivity. Plant diseases are responsible for 10–16 percentage of crop damage worldwide every year. Plant diseases have an impact on the growth of individual plants, and early detection is crucial. So if disease outbreaks are not identified in advance, food disruption will grow. Plant disease study is associated with 2 the examination of visually noticeable patterns on plants and especially on leaves. It takes quite a lot of fieldwork, such as looking at any leaf or plant to find symptoms to detect diseases at an early stage. This method could be sped up using advanced computational methods to analyze real-time images of plants or leaves and recognize the disease. Numerous Machine Learning (ML) frameworks were used to detect and identify plant diseases, such as SVM classifiers, KNN classifiers, ANN classifiers, etc. Several works have been done on high-resolution image datasets when analyzing some literature on detecting or diagnosing plant disease. Most farmers have no access to high-resolution cameras and cannot afford to detect diseases of plants. Consequently, a system for diagnosing and recognizing plant diseases on low-resolution images that would be both

effective and highly accurate must be developed. And so, we are determined

to develop a framework that can effectively detect the patterns of low-resolution images. We have chosen an already experimented dataset for this work, and we will work on three different plants as Apples, tomatoes, and corn.

1.2 Problem Statement

The main goal of agriculture is to detect autoimmune diseases through the leaves of trees.Therefore, timely detection of autoimmune diseases of these plants increases the quality and quantity of agriculture. Due to the large number of crops grown, even an agronomist and pathologist can often fail to identify Diseases in trees imagining disease-affected leaves. Plant-borne diseases have become a major problem, leading to a decline in the quality of agricultural products Automatic detection of plant diseases is a must, it can prove large fields in crops and leaves appear so that the disease can be easily detected. So this is a software solution to solve the problem of autoimmune diseases of plants.

1.3 Problem Background

Work has been done on this for many years. There are many farmers who cannot take pictures with high resolution. Diagnosis is made by taking pictures with high resolution but there are many farmers who cannot take pictures with high resolution. So in the case of those who will not be able to take pictures with high resolution, we can easily diagnose the disease by taking pictures with low-resolution. So we can easily diagnose the

disease from all kinds of images, it will be a great benefit to farmers.

1.4 Research Objectives

The analysis objectives were to implement a system that selects plant diseases in low-resolution images. Scope and responsibilities of plant pathology is unmeasured. Its ultimate goal is to prevent and control plant diseases of economic importance. The presented system is tested on a historical data-set and presents targeted results.

The objectives of our research work are as follows:

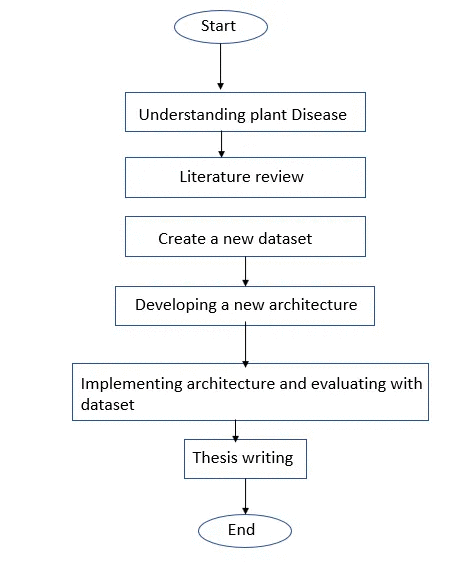
* the living entities that cause diseases in plants;
* the non-living entities and the environmental conditions that cause disorders in plants;
* the mechanisms by which the disease-causing agents produce diseases;
* the interactions between the disease-causing agents and host plant in relation to overall environment; and
* the method of preventing or management the diseases and reducing the losses/damages caused by diseases.

1.5 Motivations

During the past decade since it many critical applications, such as try for Deep-CNN for Plant Disease Diagnosis Using Low Resolution Leaf Images. Modern researchers are trying to develop the best approaches to address these problems. Enormous research work has already done that gives us reasonable solutions for few FER problems. However, no specific architecture has developed yet that identify Deep-CNN for Plant Disease Diagnosis Using Low Resolution Leaf Images dataset. Finally, a CNN approach was applied to dataset and achieve a promising result to identify Deep-CNN for Plant Disease Diagnosis Using Low Resolution Leaf Images.

1.6 Flow of the Research

We did the research work in a few steps, first we analyzed how to diagnose plant diseases. Then we reviewed them, then we created a new dataset by doing different kinds of research on these diseases. We have built and implemented new architectures accordingly. We applied the overall method and proposed a model. Below is an example of how we completed the research work step by step



1.7 Significance of the Research

We focused on food security in particular plant diseases that are a threat to food. Using Convolutional Neural Network (CNN) we apprehend the 14 diseases of Tomato, Corn, and Apple. In this research, we use standard CNN architecture like Xception, DensNet, ResNet, and MobileNet. Consequent upon we see that MobileNet gives the best results upon the other architectures on the low-resolution images.

1.8 Research Contribution

The overall contribution of the research work includes,

* We have investigated previously done researchers to determine the general knowledge behind the architectures and their inabilities in deep learning.
* We gathered images of 3 plants (Tomato, Apple, and Corn) 14 diseases to construct our study and evaluate the deep-CNN architecture.
* We tested six image recognition baseline strategies, including Densenet, Inception, Mobilenet, ResNet, VGG, and Xception, and explored MobileNet compelling low-resolution plant leaf images.

1.9 Thesis Organization

The thesis work is organised as follows. Chapter 2 highlights the background and literature review on the eld of the Plant Disease Recognition(PDR) system. Chapter 3 consists of the analysis as well as the combination of our proposed architecture in detail Requirements, feasibility, research methods.

Chapter 4 includes the details of the tests and evaluations performed to evaluate our proposed architecture. Chapter 5 explains the Standards, Impacts, Ethics, Challenges, the Constraints, Timeline, and Gantt Chart. Finally, Chapter 6 contains the overall conclusion of our thesis work.

1.10 Summary

This chapter comprises a broad overview of the problem such as what are we specifically targeting, what are the purposes of our thesis work along with the motivation of the output of the thesis work. This section also represents the overall steps on which we carried out our thesis work.

2 Background

2.1 Introduction

The advancement of computer vision, detecting diseases utilizing leaf representation is introduced with modern technology. There are various approaches that focus on high-resolution image datasets, where most farmers have no access to high-resolution cameras and cannot afford to detect diseases of plants. However, before this study, no work has introduced on low-resolution image detection. This paper explores an efficient plant disease identification model that combines multiple plant diagnoses for low-resolution images.

2.2 Literature Review

Several works have been done on high-resolution image datasets when analyzing some literature on detecting or diagnosing plant disease. Most farmers have no access to high-resolution cameras and cannot afford to detect diseases of plants.

Rahul et al. [1] present disease detection of cassava brown leaf spot responsible for Cercospora heningsii is proved to hold, using equations 1 & 2, three main areas namely Mani Road, Boca, and Anuve employed its span and wideness in Vanua Levu.. Using the Simple Random Sampling (SRS) approach the Z transect (Zigzag) method was applied to obtain the disease on cassava leaves and hold its prevalence. Anuve(42.9%) has the highest incidence, Mani Road(38.2%) following, and the lowest Boca(36.4%) using equation 1. Using equation 2 100% prevalence is hold.

Manisha et al. [2] present a pomegranate fruit disease automated tool that will perform on the web. Firstly, the image is resized, according to the desired resolution, and color, morphology, and CCV are recognized as being good or important as parameters, and using the k-means algorithm the clustering is done. SVM is used to get infected or non-infected notation that is accomplished by a huge dataset. The result is 82% accurate of their employed approach.

Ilaria et al. [3] present a multilingual web-based model to identify plant disease. Strawberry fruit is selected for the case study. The farmer on their farm will observe indications and these indications will measure with images provided in the system. The outcome will be the identification of fruit disease. The web-based system consists of users and superusers. Superuser has authority

Shiv et al. [4] present an image processing way to detect and identify fruit disease. The fruit is selected for apple and getting diseases on apple rot, apple blotch for the experiments. K-means clustering helps to segment

images. To get features from input images Color coherence vector, Histogram, Local Binary patterns, complete local binary patterns are considered. To get proper results vector machine is used.

Monica et al. [5] present image processing fruit disease detection. Apples, Grapes, and Mangoes are considered for experiments. To get characteristics from input images Morphology, color, and texture characteristics vectors are considered. Morphology provides the highest (90%) accurate results comparing with others vectors. To get the disease and approximate the weight of fruits, image processing techniques are used. weight adjustment from images is performed by Backpropagation which is stored in the learning database.

Xiaoou et al. [6] a search engine called Web-Scale Image Search Engine has been proposed that will perform its work on a text-based basis. J. Cui et al has employed an alternative way to rank up text base search results to use of adaptive visual similarity concept. An input image is first separated into one of the many predetermined portions, and the images are ranked up based on generic similarity characteristics.

Feng et al. [7] has implemented, IGroup, a useful algorithm that is able to do a bunch of web images based on keywords turned out and then employ all the resulting images to the corresponding bunch based on optical or textual characteristics.

Zhiguo et al. [8] employed keyword propagation using WordNet. WordNet employs references between words named Synonym, Hyponym, and Hypernymon based on three dimensions respectively. In essence TSN (Term

Semantic Network), a popular confederation mining algorithm i.e. Apriori algorithm is used here, also the keywords are worthen using sound frequency and opposite document frequency.

J. Krapac et al. [9] present the role of generic classifiers, which is working on query-relative characteristics. They combine textual characteristics as visual characteristics based on the presence of query words in web pages, image metadata, and visual histogram representation of images.

David et al. [10] present a method for removing distinctive invariant characteristics from images that work to find well similarities between different types of objects and scenes. The characteristics are shown to be consistently maintained strongly in terms of scale and rotation of the image and show a fairly strong similarity to the Eiffel distortion, the change in 3D view sight, the additional information of different sound, and the change in illumination.

D S Guru et al. [11] proposed an algorithm to classify flower images using a KNN classifier. The characteristics which are extracted are Textural and Gabor characteristics. The flower image is first segmented into different parts from which features are extracted. The feature values are then obeyed by the KNN-classifier for alignment.

Wenjiang et al. [12] have created new spectrum indicators to identify winter wheat diseases. Three pets of yellow rust, aphids, and powdery mildew winter wheat are taken for study. Topical wavelengths and many different types of disease were plucked using the RELIEF-F algorithm. The given accuracies of these new categories for healthy and unhealthy leaves with aphids, powdery mildew, and yellow rust were 91.6%,86.5%, 85.2%, and 93.5% separately.

Monica et al. [13] use image processing to find out disease and the fruit grading. They have used artificial neural networks for finding diseases. Two databases were created one has stored trained disease images and the other is query images for implementation. Training databases used Back propagation to adjust the weight. Three feature vectors are taken, namely, textures, color, and morphology. Better accuracies are given by morphological features than the other two features.

Zulkifli et al. [14] represent the healthy status of chilli plant the leaf images were considered. Their mission is to show the applied chemicals are on the affected chilli plant only. MATLAB is used for feature plucked and image identification. Pre-processing is brought about by using morphological operations, edge detection, and Fourier filtering. To implement computerized images digital cameras were used and GUI builds using the LABVIEW software tool.

Mrunalini et al. [15] while plucking out the property from an image the leaf image is significant at the time of the segmentation. Infected leaf analyses are performed comparing between Otsu threshold and the k-means clustering. K-means algorithm provides better accuracy instead of using the Otsu threshold.

H. Al-Hayri et al. [16] RGB images are used to detect disease on a page. The green pixels were discovered after the application of the K-mean clustering method, and various marginal values were obtained using the Otsu method. For feature plaque, the method with color is used. HSI Translator is used to converting RGB images. SGDM matrix texture statistics are taken and properties are considered using the GLCM function.

Chunxia et al. [17] FPGA and DSP-based measures are used to monitor

plant diseases. The FPGA controls the field plant image or video data for monitoring and part of the processing is performed by DSP TMS320DM642 and the flow of data transmission is nRF24L01 single-chip 2.4 GHz. There are two ways to take data from users and multi-channel is used to reduce the cost of the system.

Ferentinos et al. [18] Open-source datasets have been presented based on a special convoluted neural network for plant disease detection, and a useful model is the VGG Convulsive Neural Network.

Barbedo et al. [19] found different diseases which have an impact on different parts of the same leaf. To find the proper disease, the leaf is segmented by researchers into isolated symptoms that are taken individually and faults that were portion of bunches taken as a group. To get a proper solution limited datasets were impact and techniques were used to categorized plant disease.

Kawasaki et al. [20] present a method for achieving high categorization ability using the convoluted neural network and K-fold cross-validation fetch and acquire an average accuracy.

Durmus et al. [21] present plant disease detection using deep learning that is able to find diseases in real-time data gathered by the robot. They are able to find physical changes in leaves that were audited by RGB cameras and AlexNet and SqueezeNet were used to examine the network.

Shantanu et al. [22] present rice disease discovery using pattern recognition and software is proposed to find out disease using the affected

images of rice plants. HIS model is used to divide images after getting the desired portion of the affected leaf and then the boundary and spot identification is performed.

Savita et al. [23] represent a survey on different types of plant leaf disease classification. An example k-nearest-neighbor method is likely suitable also simplified of all algorithms for order prediction. If training data is failed to separate to determine optimal parameters in SVM then the drawback occurred.

Sanjay et al. [24] proposed four types of implementation ways, the first is, a color form is created on RGB image input because RGB is able to create color and turned into an image of RGB, that is, the color descriptor is made by HSI. Detecting and deleting green pixels using threshold values happens in the second step. Third, using pre-mathematical marginal layers, the green pixels are removed and masking is done for the useful steps that appear first in this step when an image is placed and lastly segmentation is done.

Mrinalini et al. [25] Introduce techniques for classifying and identifying plants affected by various diseases. A machine learning-based recognition system will prove to be highly effective in the Indian economy as it saves effort, money, and time. The method used to draw the feature set is the color co-appearance method. Neural networks are chosen to automatically detect disease on the page. The proposed method seems to significantly support the correct recognition of the leaves and makes less effort to calculate the vapor and root disease method.

Arivazhagan et al. [26] present the diagnostic process consists of several steps, including four main steps: first, for the input RGB image, a color

conversion structure is taken and then, Detecting and deleting green pixels using threshold values happens, which is further followed by the partitioning process, and the texture figures are calculated to obtain useful sections. Finally, the classifier is used to find the actual diseases. The proposed algorithm is proved on the five hundred leaves database.

Ananda et al. [27] presented a method for detecting and instantly identifying plant diseases using artificial neural networks (ANNs) and various image processing techniques. Depending on the given method ANN identifier for identification and Gabor filter for properties extraction, it gives better results with an identification rate of up to 91%. An ANN-based identifier identifies several plant diseases and uses the summation of textures, colors, and properties to identify these diseases.

Ramesh et al. [28] proposed to identify anomalies that arise on plants in their natural environment. To remove occlusion, they shot photographs with a plain backdrop. In terms of precision, the method was applied to other machine learning models. The identifier was tested using 160 images of papaya leaves and the Random Forest identifier.

Mohanty et al. [29] represent a deep convolutional neural network that is used a pre-trained open-source dataset containing Figure 54306 to identify 14 types of crops and 26 diseases. The given method has acquired 99.35 percent ability in the test set.

Sharma et al. [30] proposed a work investigating a possible solution for automatic detection in plants by training convolutional neural network (CNN) models with segmented image data. They balanced the F-CNN model and S-

CNN model and experimented with this work is double performance.

2.3 Problem Analysis

The advancement of computer vision, detecting diseases utilizing leaf representation is introduced with modern technology. There are various approaches that focus on high-resolution image datasets, where most farmers have no access to high-resolution cameras and cannot afford to detect diseases of plants. In this paper we are determined to develop a framework that can effectively detect the patterns of low-resolution images.

2.4 Summary

This chapter investigated and reviewed the latest techniques of detecting diseases utilizing leaf representation, including the drawbacks. The thesis’s target is to eliminate the imperfections as much as possible and introduced a new combined approach to investigate plants disease.

3 Proposed Model

3.1 Introduction

Here, we discuss the feasibility analysis of plant disease identification using its leaves and basic needs demand in this architecture. This chapter broadly describes the model’s exact architecture and all the information collected from the web and previous work.

3.2 Feasibility Analysis

This thesis project wanted one supervisor, four researchers, and it took eleven months to be produced. The research needed technical, software, and hardware support. The Thesis also wanted its dataset generation, which the researchers make. Evaluation makes it accurate, which is also required for this research. The work is executed by a countless collection of data, weighing the genuine feasibility of the information. Besides, this thesis project didn’t need financial support.

3.3 Requirement Analysis

To conduct the proposed architecture of the overall requirements, include,

• High configured device.  
• Image input device.  
• Open-source software libraries for scientific computations.  
• Open-source software libraries to implement the deep learning model.

3.4 Research Methodology

In this part, we discussed the methodology of our proposed architecture. This segment has four subsections where this subsection shows the model gradually with a broad discussion. Moreover in figure 3.1 shows the workflow of the proposed model.

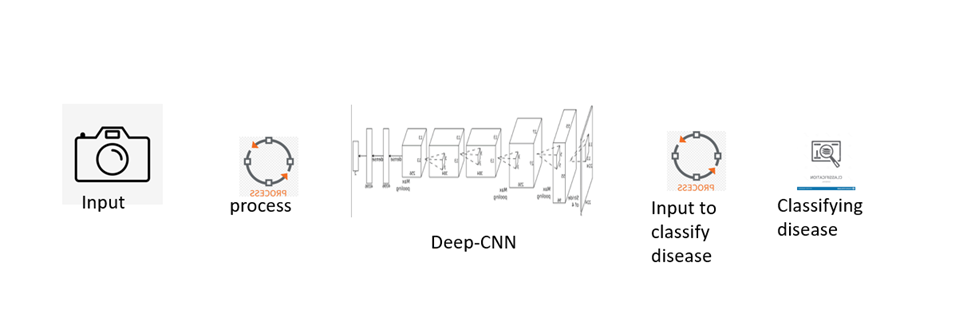


Figure 3.1. The figure illustrates the workflow of the proposed system (from  
left to right)

3.4.1 Data Pre-processing

Various CNN architectures are introduced and compared to diagnose plant diseases. The basic concept and layers of different CNN architectures are instructed in the following sections.

3.4.2 Image Preprocessing

Normalization is a process in image analysis that alters the range of pixel intensities [16]. Data normalization assures that each input data has an equal distribution. Since the CNN architectures need the input images to be of the same form, each image data is restructured into 128 by 128 pixels, resulting in faster CNN convergence. As a consequence, each channel of the reconfigured leaf images is normalized as follows [17]:



Here D is the single-channel leaf image matrix, n refers to the number of rows, and m means the number of columns of the leaf image matrix.

3.4.3 Baseline Architecture

Convolutional Neural Networks are renowned for their ability to identify patterns in images. This field allows and enables machines to understand the world in the same way that individuals do and then apply that knowledge to several things and processes such as Image Recognition, Image Analysis, classification, etc. The first step is to feed the image's pixels in the form of arrays to the neural network's input

layer (used to classify images). The hidden layers extract features by

conducting various measurements. Convolutional neural networks are feedforward neural networks commonly used to interpret visual images by analyzing information and are referred to as a ConvNet [31].

To recognize and identify objects in an image, a convolutional neural network is used. A convolution neural network has many hidden layers that enable the extraction of information from images. The essential layers are given below with a brief discussion:

We emphasize analyzing reviewed CNN architecture for detecting plant disease on low-resolution images in this article. The CNN architecture is depicted in Figure 2, with the input layer (images of the leaves from the dataset), convolutional layers, a dense layer, and an output layer.

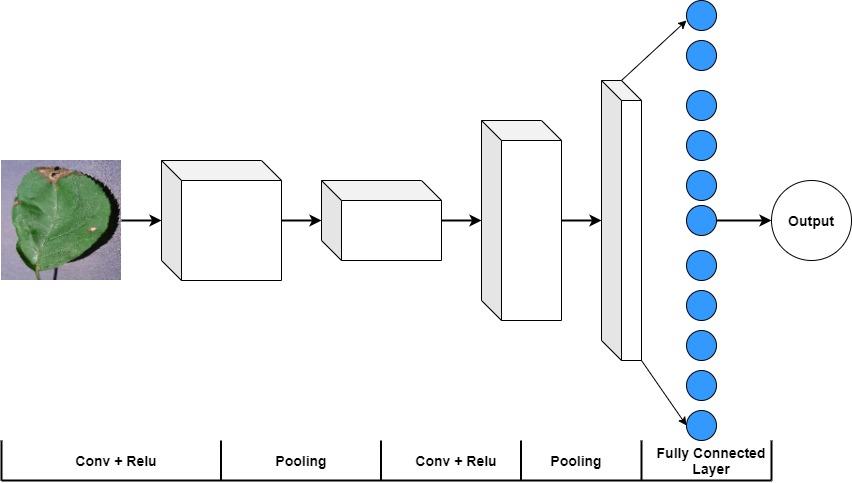


Figure3.2 The neural network architecture of plant diagnosis from leaf images.

**Input layer:** In the input layer, the original images of different plants from the dataset are input. But the authentic images are reconstructed with different widths, heights in the shape of 120 \* 120 \* 3 before being given to the architecture as inputs.

**Convolutional layers:** This is the first step in extracting useful

information from images. The convolution process is performed by plenty of filters (matrix of values known as weights that have been trained to detect unique features) in a convolution layer. CNN utilizes filters, also identified as kernels, to determine what features, such as edges, are present in images. The filter iteratively checks the images to see if the function it is designed to detect is available. The mathematical procedure can be found in[19]. The Batch Normalization process converts the images into a standard shape based on the mean and variance of a particular batch. It also improves the reliability of the network and leads to faster convergence.

**ReLU Layer:** The rectified linear unit (ReLU) is an abbreviation. After the function maps have been removed, they must be moved to a ReLU layer. This is a non-linear activation function, and its sole aim is to incorporate non-linearity into the network. The ReLU function is straightforward; values less than or equal to zero become zero, while all positive values remain unchanged.

**Pooling layer:** Pooling is a type of downsampling that reduces the dimensions of a feature map. Reduce the redundancy in the input function to boost up the training process and decrease the amount of storage used by the network. To generate a pooled feature map, the rectified feature map is now passed through a max-pooling pooling layer. Max pooling dramatically decreases the size of the image, lowering the amount of system memory and the number of operations conducted further in the network.

**Fully Connected Layer:** The following step in the process is known as flattening (convert all the resultant 2-Dimensional arrays into a single

long continuous linear vector). With the fully connected operation of a neural network, the input image is flattened into a feature vector and transferred through a network that predicts the output possibilities. This fully connected layer is followed by several dense layers of neurons, which eventually produce accurate observations.

For this analysis, the MobileNet [19] architecture provides the highest performance. Apart from the first layer, the MobileNet architecture is based on depth-wise separable convolutions. The first layer is a dense or full convolutional layer. After each sheet, batch normalization and ReLU non-linearity are applied. On the other hand, the output layer is a fully connected layer with no non-linearity that flows into the softmax for classification. Stridden convolution is being used for both depthwise convolutions and the first fully convolutional layer in downsampling. When depthwise and pointwise convolution are considered as different layers, the overall number of layers for MobileNet is 28.

3.4.4 Transfer Learning

In general, transfer learning refers to a method in which a model trained solely on a single problem is used in any manner on a second related issue. It is a common method in computer vision since it helps us to create accurate models while saving time. Instead of beginning the learning process from scratch, transfer learning begins with patterns learned while solving a different set of problems.

The first step is to obtain the pre-trained model that will be used to solve the specified task. To go for one of the frameworks, such as ResNet or Xception, to initialize the base model. The base model would generally

have much more units in the final output layer than desired and would need to remove the final output layer. Then, add a final output layer that is task-compatible. Freezing the layers from the pre-trained model is crucial because all previous learning would be lost if not. The following move introduces additional trainable layers, which will convert existing features into predictions on the new dataset. After training the new layers on the dataset, we will have a model that can make predictions on the dataset. And we can improve the performance by fine-tuning. Fine-tuning is accomplished by unfreezing the base model or a part of it and training the whole model on the data set at a shallow learning rate. The low learning rate would improve the model's efficiency on the new dataset while avoiding overfitting.

3.5 Design, Implementation, and Simulation

In figure 3.1, the workflow describes the overall idea of the proposed architecture. By using python, all the alluded moves of the prototype are implemented[31]. The Convolutional architectures using Keras are implemented. There are some different implementations, calculations, and Numpy[32] is used. The dataset utilized to test the design is specifically embedded, and no varieties or choices are created whereas trying the architecture.

3.6 Summary

Here we illustrated the proposed architecture plant disease diagnosis identification method. This model structure conducts the deep CNN.

4 Implementation, Testing, and Result Analysis

4.1 Introduction

In this segment, we illustrated the proposed architecture plant disease diagnosis identification method. This model structure conducts the deep convolutional neural network approach.

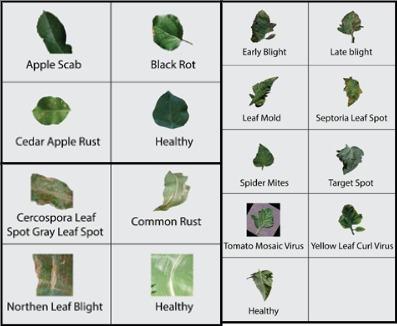
4.2 Dataset

Three openly prepared plant disease datasets are applied throughout the investigations. The images were gathered from Kaggle. It carries pictures of 3 separate plants where every plant has a different number of related diseases. Table 1 reveals how the database is categorized in terms of plant varieties and illnesses. Besides, the number of images employed in every class is presented in Table 1. Figure 1 illustrates the three affected apple leaves, namely, Apple scab, Black rot, and Cedar apple rust, with a healthful leaf image. Also, in Figure 1, dysfunctional images of corn are displayed: Cercospora leaf spot, Common rust, and Northern leaf blight. Nine disorders of tomatoes, including Early blight, Late blight, Leaf mold, Septoria leaf spot, Spider mites, Target spot, Tomato mosaic virus, and

yellow leaf curl virus, are exhibited in Figure 1.

**Table 4.1.** The table summarizes the plants and varieties of disease that the accumulated dataset includes.





**Fig. 1.** The Figure shows the disorder images of Apple (top left), Corn (bottom left), and Tomato (right).

4.3 System Setup

We used Python for data collection, pre-processing, tests, and model evaluation [33]. And Keras [34] is used to implement the neural network architecture also for deep learning models. We used Numpy [35], which is a Python library for performing simple mathematical operations. Then TensorFlow [36] is used to produce neural network GPU execution.

4.4 Evaluation

This section is designed with some subsections, including evaluation

metrics, experimental setup, experiments, and comparison that we have done for this specific task. The datasets on which the tests were performed are given in Table 1.

An algorithm or method needs to estimate how superior it is by measuring the performance relevant to that algorithm. The main difficulty in assessing each process is selecting instructing and testing sets, introducing disparity in model performance. Almost every performance metrics is founded on a confusion matrix that constructs true-positive (TP), true-negative (TN), false-positive(FP), and false-negative (FN) [33] values.

According to the following equations, we evaluated various baseline architectures by using the confusion matrix's accuracy, precision, and recall evaluation metrics.

**Accuracy:** By following the equation, we can find the accuracies of the model.

**Precision:** This equation will give the precision of the model, and it can be stated as

**Recall:** The equation which helps to gain an efficient recall can be stated as

4.5 Experiments Result and Comparisons

We use the Kaggle dataset to implement our Deep-CNN architecture. Experimented the proposed system using various pre-trained Keras-implemented transfer learning models like Mobilenet, MobileNetV2, Xception, ResNet50, ResNet50V2, ResNet101, DenseNet121, DenseNet169, DenseNet201, InceptionV3, VGG16, VGG19, and so on in this section. As the pre-trained learned features were applied in the new diagnosis task, so it certainly identifies the diseases from the infected plants. Thus, we present the effectiveness of each model to determine which had the best accuracy, recall, and precision. Table 4.2 shows the dimensions of the developed system. We could see that MobileNet performs admirably, with 99.9 % accuracy, 99.08% precision, and 99.87% recall. This is the present state of the performance, as far as we can learn. Here, we applied different architectures such as MobileNet, ResNet, DenseNet, Inception, VGG, Xception, and their latest version, producing better performance. However, MobileNet provides the best performance for disease diagnosis in various plants. The concept of transfer learning was greatly used to run the experiment using these CNN architectures.

**Table 4.2.** This table represents the accuracy, precision, and recall of different architectures.



We can recognize that Xception, ResNet, and DesnseNet don’t give us a better result than Mobilenet. The table shows that DenseNet and ResNet are good but MobileNet accuracy is higher than other deep CNN architecture. MobileNet performs better in lower resolution.

This paper broadly investigated the significance of the Deep CNN(MobileNet) to classify plant disease and founds satisfactory performance.

4.6 Summary

Finally, the given output confirmed that this model executes and provides better accuracy on plant disease diagnosis classification by analyzing this section.

5 Standards, Constraints, Milestones

This part explains the standards, Impacts, Ethics, and challenges of this research work. After that, the Constraints and alternatives of this work are described in detail. At last, the time limit, tasks, and Milestones of the proposed system are showed.

5.1 Standards (Sustainability)

We make sure this research work will be sustainable for an extended time. In recent years, Plant disease diagnosis is liked by researchers. Plant disease diagnosis is individual choices hot topics in the research field. Plant disease diagnosis is an image identification problem using low-quality pictures of a leaf that can be useful for Farmers and the country's economies. Furthermore, we used CNN for implementing our model in recent deep learning close in which is part of a neural network. The resources that we used in this thesis work will be convenient for a longer time. So we can say that it will be sustainable.

5.2 Impacts (on Society)

Plant diseases are a severe problem and have a significant impact on our

society. It also has an economic effect because Bangladesh is a cultivated country and 60-70% of people live on agriculture. Plant disease has a significant impact on export. Our research will variously help farmers as we know that farmers can't afford expensive devices. They have only mobile phone devices. We made a compatible system that gives a better result using a low-resolution picture of the mobile. This thesis work has had a vital influence on rural people as they can use it more effectively than the previous system. They can take pictures of plant leaves, and the system will help detect diseases of the plant.

5.3 Ethics

The plant diseases diagnosis structure has a significant number of available application field, where we used the Kaggle dataset and models is pre-trained. The plant disease diagnosis does not have much security threat, but it will damage economically if this system uses badly. The system’s installation should assist personal or country’s economic worries and should not be applied for any motive that increases food security threat. All the information, collection of the dataset needs to be under moral standards.

5.4 Challenges

Though plant disease diagnosis architectures are intensifying mainly, the organization is making these technologies with recent information security challenges. This system is used primarily for identifying diseases of the plant by detecting specific plant leaves. Sometimes the architecture can’t see unusually, and the farmer may use a better

camera, which can create a problem to detect the disease. Misuse also creates a threat for manipulating rural people. Awful people can make trouble for those people who can’t use this system.

5.5 Constraints

Here we discuss various constraints like components constraints, design constraints, and hardware constraints.

 This entire research work is done for an image dataset. This system required a high configuration device because many images require at least an octa-core processor to work our proposed model. Nonetheless, No Graphical external power is needed.

This component is used to train our model:

*•*Minimum processor: Intel Core i3 (8th gen)

*•*Minimum memory: 4GB (DDR4, 2400bus)

*•*Video Input: HD Video Input Device

 Remember, The cost depends on the current market price and region because of hardware products availability.

5.6 Timeline and Gantt Chart

We divided our work timeline into three parts because we have three

semesters for all the work. We followed our supervisor's instructions

and strategy to execute our thesis work. At first, we submitted our project through a project proposal, searched previous work, and reviewed it. After that, For planning and analyzing the entire system, we made a proposed prototype.

In the semester, we implemented our proposed model and built the dataset. Finally, our entire structure has been created by us. We have tested with our dataset in the last semester and reported the whole process of work. Meanwhile, our conference paper has been accepted that we wrote.

We draw a Gantt chart (figure 5.1,5.2,5.3) to represent our timeline, the entire thesis work process, and the execution to all. We have a three-semester time limit, and each semester has four months. So we have a total of 12 months to complete the research.

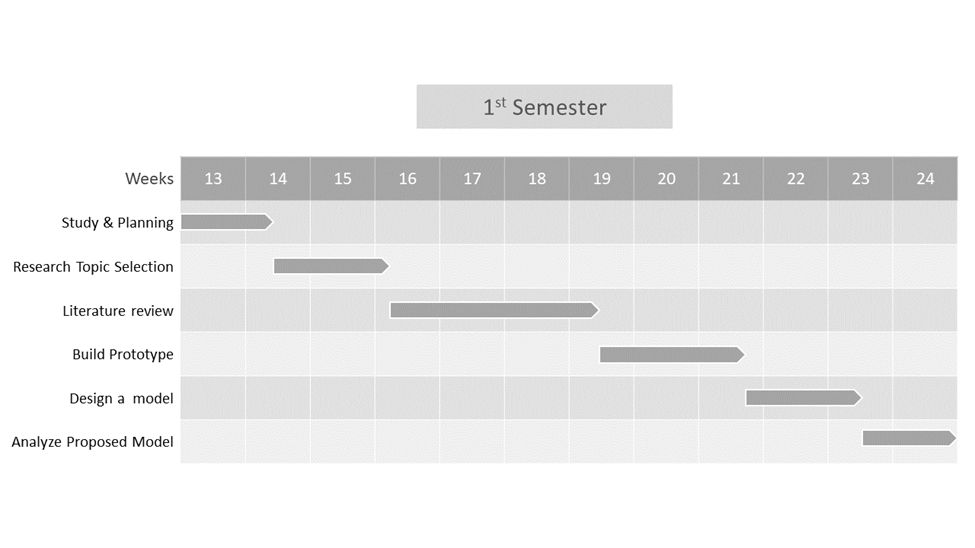


Figure 5.1 1st semester Gantt Chart

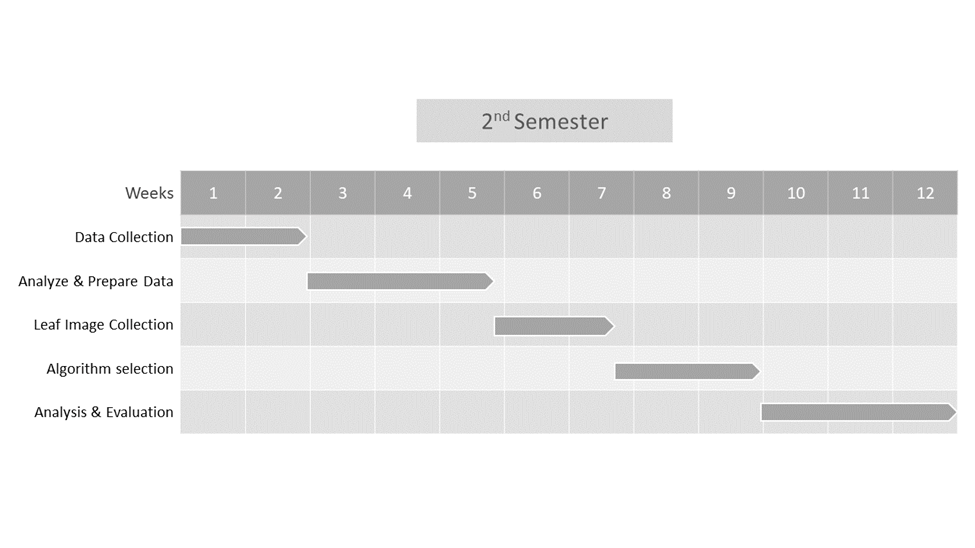


Figure 5.2 2nd semester Gantt Chart.

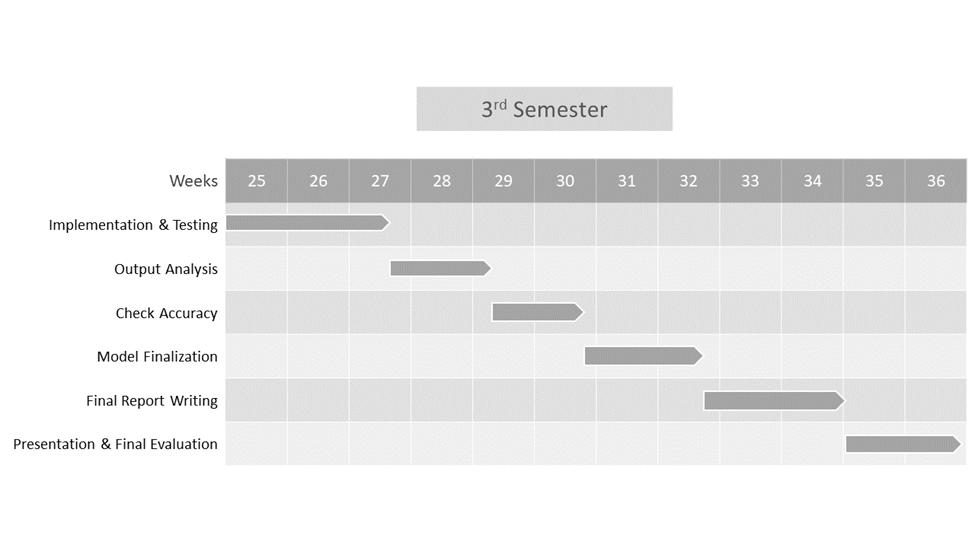
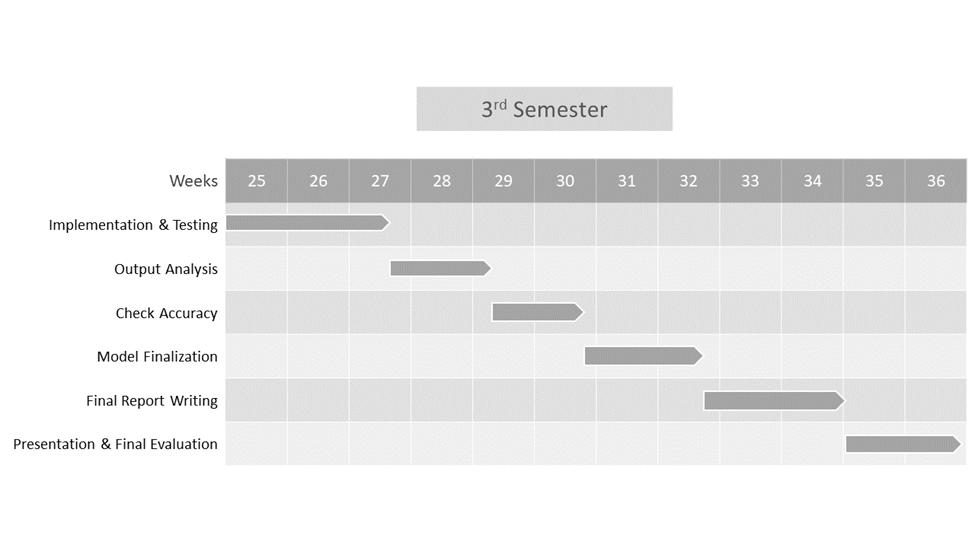
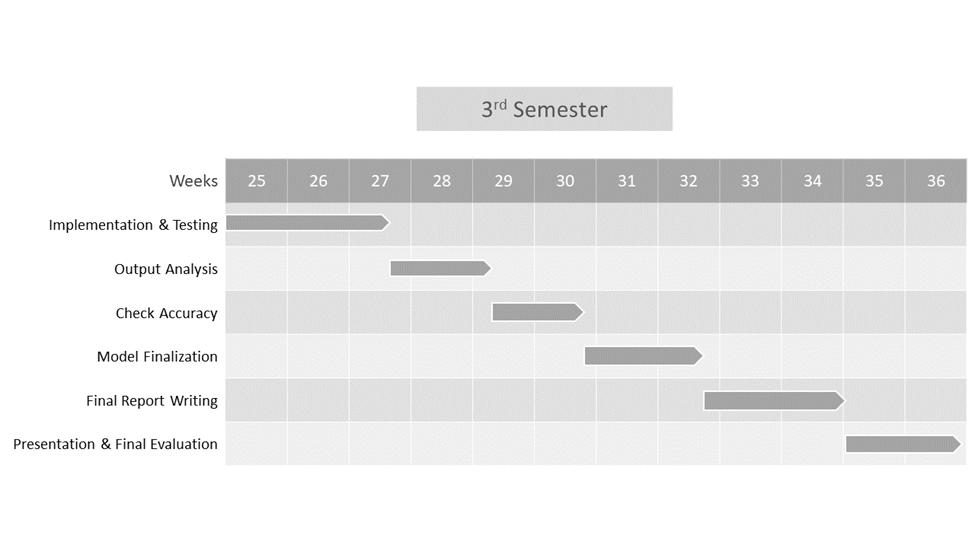


Figure 5.3 3rd semester Gantt Chart



5.7 Summary

Though, the episode shortly describes the patterns, impacts, ethics, challenges of the research task. Furthermore, alternatives, schedules, assignments, constraints, and milestones of the suggested work are shown.

6 Conclusion

6.1 Introduction

This section incorporates and justifies a multi-plant diagnosis approach based on various image classifier baselines for low-resolution images. We used a transfer learning strategy to train and evaluate our method systematically. We collected a publicly introduced dataset that consisted of different images, and we evaluated three different plants, apples, corns, and tomatoes, with several different diseased leaf images. We find that separate and distinct convolution and skipped connections greatly enhance plant disease detection efficiency. The paper focused on image processing using CNN for its excellent feature extraction capability and superiority over conventional systems in terms of real-time, precision, and resilience. We evaluate and observe different architectures such as Xception, ResNet, DenseNet, and MobileNet, and in comparison, MobileNet performs better for low-resolution plant leaf images.

The paper experiments and appraises a plant diseases diagnosis using a low-quality image process using the ImageNet and Kaggle dataset built by us. We prepared a deep neural network method of CNN to practice and test our process correctly. We observe that the Deep-

CNN, precisely the MobileNet approach, gives better performance for plant diseases diagnosis. We give our all expertise, the architecture offered in this paper is the primary architecture that identifies plant diseases applying novel Deep-CNN models.

6.2 Future Works and Limitations

This research has a vital influence on the farmer to predict the plant disease diagnosis. This research work will reduce financial damage by identifying diseases of plants. We plan to add some new features like plant disease from video using deep learning models that will be an outstanding achievement. By testing six architecture of CNN we built, but new one like GRU, LSTM needs to be tested. This work has lots of potential in the future. To reduce losses of feature extraction CNN need to change its architecture. We are confident that perhaps the objectives of this research work will lead the road for intensive studies on image processing methods in the future and improve the knowledge and usefulness of object classification based on CNN.

References

[1] Rahul Ravneel Prasad, Bradley Alungo. Prevalence and Incidence of Cassava (Manihot esculenta) Brown Leaf Spot Disease Caused By Cercospora heningsii in Macuata Province, Vanua Levu, Fiji.

[2] Manisha Bhange and H.A.Hingoliwala. Smart Farming: Pomegranate Disease Detection Using Image Processing.

[3] Ilaria Pertot, Tsvi Kuflik, Igor Gordon, Stanley Freeman, Yigal Elad, Identificator: A web-based tool for visual plant disease identification, a proof of concept with a case study on strawberry.

[4] Shiv Ram Dubey, Anand Singh Jalal, Detection and Classification of Apple Fruit Diseases using Complete Local Binary Patterns IEEE.

[5] Monika Jhuria, Ashwani Kumar, Rushikesh Borse, Image Processing For Smart Farming: Detection of Disease and Fruit Grading.

[6] Xiaoou Tang, Fellow, IEEE, Ke Liu, Jingyu Cui, Student Member, IEEE, Fang Wen, Member, IEEE and Xiaogang Wang, Member, IEEE, “IntentSearch:Capturing User Intention for One-Click Internet

Image Search.

[7] Feng Jing, Changhu Wang, Yuhuan Yao, Kefeng Deng, Lei Zhang, Wei-Ying Ma, “IGroup: Web Image Search Results Clustering.

[8] Zhiguo Gong, Chan Wa Cheang, and Leong Hou U, “Web Query Expansion by WordNet”.

[9] J. Krapac, M. Allan, J. Verbeek, and F. Jurie, “Improving web image search results using query-relative classifiers.

[10] David G. Lowe, “Distinctive Image Features from Scale-Invariant Key points.

[11] D S Guru, Y. H. Sharath, S. Manjunath, “Texture Features and KNN in Classification of Flower Images”.

[12] Wenjiang Huang, Qingsong Guan, Juhua Luo, Jingcheng Zhang, Jinling Zhao, Dong Liang, Linsheng Huang, and Dongyan Zhang, “New Optimized Spectral Indices for Identifying and Monitoring Winter Wheat Diseases”.

[13] Monica Jhuria, Ashwani Kumar, and Rushikesh Borse, “Image Processing For Smart Farming: Detection Of Disease And Fruit Grading”.

[14] Zulkifli Bin Husin, Abdul Hallis Bin Abdul Aziz, Ali Yeon Bin Md Shakaff Rohani Binti S Mohamed Farook, “Feasibility Study on Plant Chili Disease Detection Using Image Processing Techniques”.

[15] Mrunalini R. Badnakhe, Prashant R. Deshmukh, “Infected Leaf Analysis and Comparison by Otsu Threshold and k-Means Clustering”.

[16] H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik and Z. ALRahamneh, “Fast and Accurate Detection and Classification of Plant Diseases”.

[17] Chunxia Zhang, Xiuqing Wang, Xudong Li, “Design of Monitoring and Control Plant Disease System Based on DSP&FPGA”.

[18] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis.

[19] Barbedo, J. G. A. (2019). Plant disease identification from individual lesions and spots using deep learning.

[20] Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2015, December). Basic study of automated diagnosis of viral plant

diseases using convolutional neural networks.

[21] Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning.

[22] Santanu Phadikar and Jaya Sil, “Rice Disease Identification using Pattern Recognition”.

[23] Savita N. Ghaiwat, Parul Arora. Detection and Classification of Plant Leaf Diseases Using Image processing Techniques: A Review. International Journal of Recent Advances in Engineering & Technology.

[24] Prof. Sanjay B. Dhaygude, Mr.Nitin P.Kumbhar. Agricultural plant Leaf Disease Detection Using Image Processing. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.

[25] Mrunalini R. Badnakhe and Prashant R. Deshmukh. An Application of K-Means Clustering and Artificial Intelligence in Pattern Recognition for Crop Diseases.

[26] S. Arivazhagan, R. Newlin Shebiah, S. Ananthi, S. Vishnu Varthini. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features.

[27] Anand.H.Kulkarni, Ashwin Patil R. K. Applying image processing technique to detect plant diseases.

[28] Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N., & Vinod, P. V. (2018, April). Plant disease detection using machine learning.

[29] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection.

[30] Sharma, P., Berwal, Y. P. S., & Ghai, W. (2020). Performance analysis of deep learning CNN models for disease detection in plants using image segmentation.

[31] Kabir, M. M., Ohi, A. Q., Rahman, M. S., & Mridha, M. F. (2020, December). An Evolution of CNN Object Classifiers on Low-Resolution Images. In 2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET) (pp. 209-213). IEEE.

[32] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[33] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual CreateSpace. Scotts Valley, CA.

[34] Chollet, F. (2015). Keras: Deep learning library for theano and tensorflow. URL: https://keras. io/k, 7(8), T1.

[35] Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. Computing in science & engineering, 13(2), 22-30.

[36] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (pp. 265-283).