Predicting cotton leaf disease based on image classification using deep learning

Submitted By

Fahima Sultana Prity

ID: 191-35-2783

28th Batch

Supervised By

Afsana Begum

Assistant Professor & Coordinator

Department of Software Engineering

Daffodil International University

This Report Submitted in Partial Fulfillment of the Academic Requirements for the Degree of Bachelor of Science in Software Engineering

APPROVAL

This thesis titled on "**Predicting cotton leaf disease based on image classification using deep learning**", submitted by **Fahima Sultana Prity (ID: 191-35-2783)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

-- **Chairman**

Dr. Imran Mahmud
Head and Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

-- **Internal Examiner 1**

Afsana Begum
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Fazle Elahe

-- **Internal Examiner 2**

Dr. Md. Fazle Elahe
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

-- **External Examiner**

Mohammad Abu Yousuf, PhD
Professor
Institute of Information Technology
Jahangirnogor University

DECLARATION

I hereby declare that this thesis paper has been done by me under the supervision of **Afsana Begum**, Assistant Professor & Coordinator, Department of Software Engineering (SWE), Daffodil International University. I also declare that neither this report nor any part of this report has been submitted elsewhere for award of any degree by me.

<u>Submitted By</u>

Fahima

Fahima Sultana Prity

ID: 191-35-2783

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

<u>Certified By</u>

Afsana Begum

Assistant Professor & Coordinator

Department of Software Engineering

Daffodil International University

ACKNOWLEDGEMENT

First of all, I want to express my gratitude to Almighty Allah for enabling me to finish the thesis.

I would like to thank my supervisor, Mrs. Afsana Begum, for her persistent assistance, inspiration, energy, and knowledge sharing with my thesis and research work. Her guidance aided me in finding research solutions and reaching my finalized theory.

I would like to express my sincere appreciation to all of the teachers in our software engineering department for their thoughtful aid, faith, advice, and consistent support during my studies period.

I would like to thanks all the good-wishers surrounding who helped and inspired me to make the task complete.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

TABLE OF CONTENTS

CONTENTS	PAGE
APPROVAL	i
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	vi

CHAPTER	PAGE
CHAPTER 1:	
INTRODUCTION	1-5
1.1 Introduction	1
1.2 Background	2
1.3 Motivation	2-3
1.4 Problem Statement	3
1.5 Research Questions	3-4
1.6 Research Objective	4
1.7 Research Scope	4-5
1.8 Thesis Organization	5
CHAPTER 2:	
Literature Review	6-9
2.1 Introduction	6
2.2 Related Works	6-9
CHAPTER 3:	
Research Methodology	10-20
3.1 Research Subject and Instrumentation	10
3.2 Data Collection Procedure	10
3.2.1 Data Collection	11-12

CONTENTS **PAGE**

3.3 Statistical Analysis	12
3.4 Proposed Methodology/Applied Mechanism	13
3.4.1.1 Image Acquisition	13
3.4.1.2 Data Preprocessing	14
3.4.2.3 Data Splitting	14
3.5 Introduction To Convolutional Neural Network (CNN)	15-16
3.5.1.1 Convolution Layer	16-17
3.5.1.2 Activation Function(ReLu)	17
3.5.1.3 Max Pooling Layer	18
3.5.1.4 Fully Connected Layer	18
3.6 Implementation Requirements	19-20
3.6.1 Model Construction	19
3.6.2 Model Compilation	19
3.6.3 Training the Model	20
3.7 Experiment	20
CHAPTER 4:	
Experimental Results and Discussion	21- 27
4.1 Introduction	21
4.2 Experimental Setup	21
4.3 Experimental Result	22
4.3.1 Training Process	22
4.3.1.1 Training and Validation Loss	22
4.3.1.2. Training and Validation Accuracy	23

CONTENTS **PAGE**

4.4 Evaluation Matrix	23-25
4.5 Confusion Matrix	26
4.6 Prediction	27
CHAPTER 5:	
CONCLUSION AND FUTURE SCOPE	28
5.1 Conclusion	28
5.2 Future Scope	28
REFERENCES	29-30

LIST OF FIGURES **PAGE**

Figure3.2.1.1: Sample of dataset classes	11
Figure3.3.1: Image volume for different classes	12
Figure3.4.1: Proposed Methodology	13
Figure3.5.1: Proposed CNN Architecture	16
Figure3.5.1.1.1: Convolution Layer	17
Figure3.5.1.3.1: Max pooling	18
Figure 4.3.1.1.1: Training and Validation loss	22
Figure 4.3.1.2.1: Model Training and Validation accuracy	23
Figure 4.4.1: The performance of CNN model	25
Figure 4.5.1: Confusion Matrix	26
Figure 4.6.1: Predicted Result	27

ABSTRACT

Cotton is one of Bangladesh's most important agricultural products, but it faces a number of challenges or constraints in the leaf. Most of the time, these constraints are identified as diseases and pests that are difficult to detect with the naked eye. The goal of this study was to create a model to improve cotton leaf disease detection and prediction using CNN, based a deep learning technique. This study used a raw dataset containing bacterial blight, curl virus, fusarium wilt, verticillium wilt, and healthy leaves to accomplish this. The dataset split into 80:20 which boosted the generalization of the CNN model. For this research, the dataset has nearly 2535 images where 80% were accessed for training purposes. This developed model is implemented using python version 3.9.13, and the model is equipped with the deep learning package called Keras, TensorFlow, and Jupyter which are used as the developmental environment. This model achieved an accuracy of 96.88% for identifying classes of leaf disease in cotton plants. This paper aided the agricultural sector in transitioning away from traditional or manual disease and pest detection methods in order to achieve breakthrough results. Large farms will greatly benefit from this automated process for reducing monitoring work.

Keywords: Deep learning, Convolutional Neural Network, Cotton leaf diseases.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Among Bangladesh's cash crops, cotton is one of the most profitable after tea because it is used in the country's garment export industry. Cotton is the most widely used fiber in the world for the production of clothing and other textile products. It benefits the national economy by creating direct and indirect jobs in the agricultural and industrial sectors. On the other hand, cotton is susceptible to many diseases, which damage the crop and significantly reduce yield. Furthermore, crop yield is constantly threatened by diseases and insect pests. Plant diseases and insect attacks are estimated to cost the global economy between 20% and 40% of annual crop production. Approximately $220 billion and $70 billion will be lost to the global economy.

Over the last few decades, researchers have become increasingly interested in sustainable agriculture, which addresses issues such as plantation cycles, damaged plants, and disease prevention. Along with growth, a crop's life cycle also involves prevention and alignment diagnosis. Globally, researchers are now using machine learning algorithms or deep learning algorithms to predict and diagnose plant diseases to eliminate long-term losses. The main objective of this project is to replace the conventional diagnosis procedure for cotton leaf diseases with a deep learning-based Convolutional Neural Network (CNN) algorithm.

1.2 Background

Plant leaf disease is difficult to detect and diagnose with the naked eye, which takes time and causes a sustained economic loss for any nation. To address this issue, a reliable model has been created using a machine learning algorithm, which will aid in the accurate diagnosis of leaf diseases. The proposed model can be integrated with the collected data picture to detect cotton disease leaves where my dataset has 4 diseased classes and one healthy cotton leaf class. The image dataset was preprocessed using resizing and rescaling methods before the model was trained using a deep-learning based CNN technique. I used the sequential model to create the most accurate and best-fitting model possible.

1.3 Motivation

Although our country's garment industry is growing by the day, the proportion of cotton cultivation is not very significant, even though cotton is the main component of the garment or textile industry. One of the reasons for the low production of cotton in Bangladesh is that cotton leaves are quickly infected by various fungal or bacterial strains, as a result of which farmers are losing interest in its production day by day. As they are not very educated, they cannot figure out why this is happening or even understand the situation. In most cases, it causes a lack of understanding of the disease and the issues. Therefore, they can't take the proper steps or make the appropriate decisions at the right time.

It worries me a lot and hurts my feelings at the same time. For this reason, I want to do something for the farmers in my country. By the way, the modern world is advancing rapidly, so why should our farmers and agriculture stay behind? And that's why I reached the idea which is "Predicting cotton leaf disease based on image classification using deep learning".

1.4 Problem Statement

The agricultural sector's contribution to our country's GDP is decreasing year by year. One of the primary reasons for this is the significant crop losses caused by pests and diseases in the field. Every year, our country suffers significant economic losses as a result of farmers' inability to detect the right disease at the right time due to a lack of information. So, with farmers and the agriculture industry in mind, I present a deep learning-based model that can predict whether a leaf is diseased or healthy based on an image of a real plant leaf.

1.5 Research Question

I'm working on cotton leaf disease prediction using a deep learning approach to get the best results, and where I must carefully examine the algorithms and choose the appropriate algorithm. The research question was as follows:

RQ1: What are the main environmental and symbiotic factors that can cause some cotton leaf diseases in Bangladesh and how can agriculture and farmers be benefited by identifying and predicting these diseases?

RQ2: How can I choose the most effective model for predicting cotton foliar disease based on my dataset?

1.6 Research Objective

The main objective of this thesis is to develop an automatic leaf disease predictor and determine which classified disease it is based on my raw dataset. This thesis has the following objectives:

- Sort the images into five Categories.
- Expand the training dataset.
- Train with CNN with our dataset.
- Save the model's progress.
- Save the best weight for feature prediction.
- Predict leaf disease to see leaf structure from the unknown test dataset.

1.7 Research Scope

The scope of this thesis is exiting helpful for:-

- It can help in the early detection of damaged plants.
- It will help agriculturists and farmers to take future steps
- Will prevent loss of the agriculture sector.
- Will boost our economy.
- Can improve the efficiency of automated plant disease detection.

1.7 Thesis Organization

The first chapter presents the initiative's objective, motivation, research questions, and anticipated results. The second chapter discusses what has already been done in this state. The theoretical discussion of this research work is related in Chapter 3. This section expanded on the statistical methods used in this work to discuss the theoretical portion of the research. This chapter also demonstrates the procedural approaches of CNN and machine or deep learning classifiers. The experimental results, performance evaluation, and discussion of the results are presented in Chapter 4. In this chapter, some experimental images are shown to help realize the project. Chapter 5 summarizes the study, its limitations, and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In its simplest form, a literature review is an examination of previously published academic books, papers, dissertations, conference proceedings, and other resources. In this area review provides an overview, a description, and a critical assessment of a subject, problem, or field of study. Contrast it with a work review, a less formal method for summarizing a piece of work.

2.2 Related Work

(Afzaal, et. al., 2021) [1] Stated that a deep learning-based model was used to autonomously detect and segment seven types of strawberry diseases, with approximately 2500 datasets being addressed. The researchers proposed a model based on the Mask R-CNN architecture for instance segmentation in these seven diseases. A ResNet backbone was also used, as well as a systematic approach to data augmentation that allowed for target disease segmentation and a final mean accuracy of 82.43%.

To predict tomato disease, (Chen, et. al., 2022) proposed an Android based platform using Alexnet transform architecture. The dataset contains 18,345 training data and 4,585 test data to build the predictive model. Firstly the data is divided into ten labels for tomato leaf diseases, each with 64 × 64 RGB pixels. Here the model used the Adam optimizer, and finding high model accuracy with a number of epochs of 75, batch size of 128. [3]

(Jenifa, et. al., 2019) used a deep convolutional neural network-based model for the automatic detection of cotton leaf diseases. Here a disease image is taken as an input image and converted into a gray image using the MATLAB tool. The model was trained with 500 leaf images and 100 cotton leaves for testing. However the classification result wasn't as good as expected. [4]

(Ma, et. Al., 2018) in [5], was proposed a deep convolutional neural network model to handle symptom-based recognition of four cucumber diseases. To reduce the chance of overfitting, data augmentation methods were used to enlarge the datasets. The authors compare the results of DCNN with conventional classifiers Random Forest and Support Vector Machine, as well as AlexNet. (Krishnamoorthy, et. al., 2021) applied an InceptionResNetV2 a type of CNN model utilized with a transfer learning approach for recognizing diseases in rice leaf images. They optimized the parameters of the proposed model for the classification task and obtained an accuracy of 95.67%. [6]

(Arivazhagan, et. al., 2018) Reported that deep learning techniques were applied to detect foliar diseases in various mango trees. The researchers used five different leaf diseases from different samples of mango leaves, where they addressed about 1200 datasets. The CNN framework was trained with more than 600 images, with 80% used for training and 20% for testing. The classification accuracy can be increased further if more images are provided in the dataset by tuning the parameters of the CNN model. [7]

(Bhimte, et. al., 2018) employed an SVM classifier to categorize cotton leaves using the images' color and texture as acceptable features. The photos were taken using a digital camera among cotton fields. There are several preprocessing methods used, including filtering, background removal, and augmentation. To extract the sick segmented portion from the cotton leaf, color-based segmentation is used. Feature extraction is carried out on a segmented image. 130 photos make up the dataset for the proposed work. Out of these, the classifier is trained using 50 images of bacterial blight, 50 images of magnesium shortage, and 30 images of health. A random image dataset is provided for testing. [8]

(Shruthi, el. at., 2019) described the stages of a general plant disease detection system and a comparison of strategies for detecting plant diseases using machine learning. In this paper, a comparison of five different machine learning classification techniques is made in order to identify plant diseases. The outcome demonstrates that the CNN classifier accurately and more frequently diagnoses more diseases. [9]

(Panigrahi, et. al., 2020) focuses on supervised machine learning techniques for maize plant disease detection using plant images. The proposed methodology was used to train the classification model using labeled image data. The RF classifier has the highest accuracy (79.23%) among the rest of the classification models for disease detection in testing image data.

In [15] (Warne, et. al., 2015) identified cotton leaf diseases through image processing. Here, the input image is preprocessed using histogram equalization, and the K-means algorithm is used to segment the datasets.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Research Subject and Instrumentation

Recent years have seen a huge response in the use of deep learning algorithms and machine learning algorithms for any kind of detection and prediction. This study is about predicting cotton leaf disease. So, I employed convolutional neural networks based on deep learning to predict cotton diseases. As part of my research, I used a high-configuration laptop equipped with a 500 GB hard drive and 240 GB solid-state drive. For coding, I used the Python programming language with a variety of library packages including TensorFlow, Keras, matplotlib, NumPy, pandas, etc. As a data training and testing tool, I used Jupiter Notebook where Anaconda Prompt was used to open the tool.

3.2 Data Collection Procedure

The main purpose of this paper is to classify and predict cotton leaf diseases. For the research, I collected different samples of cotton leaves taken from the field.

3.2.1 Data collection

Image data is the main source of this paper. Acquiring a large number of leaf samples in a short period had not been so smooth. However, while collecting the images I had to keep in mind some things like image size, resolution, image quality and also adequate knowledge about cotton diseased leaf syndrome. For this study, samples were placed into a total of five classes, four diseased and one healthy. During data collection, 2535 images of data are captured by an Android phone camera which is distributed in five categories such as bacterial blight, curl virus, healthy leaves, verticillium wilt, and fusarium wilt, the model will be trained with this dataset, as shown in Figure 3.2.1.1.

a) Bacterial blight

b) Curl virus

c) Healthy leaf

d) Verticillium Wilt

e) Fusarium Wilt

Figure3.2.1.1: Sample of dataset classes: (a) bacterial blight, (b) Curl virus, (c) Healthy leaf, (d) Verticillium Wilt, and (e) Fusarium Wilt

3.3 Statistical Analysis

In total, my dataset contains 2535 images of cotton leaves. Here the mobile phone camera is used to take pictures from different angles of that cotton leaf. Figure 3.2.2 shows how many images the dataset contains of bacterial blight, curl virus, healthy leaves, Verticillium wilt, and Fusarium wilt.

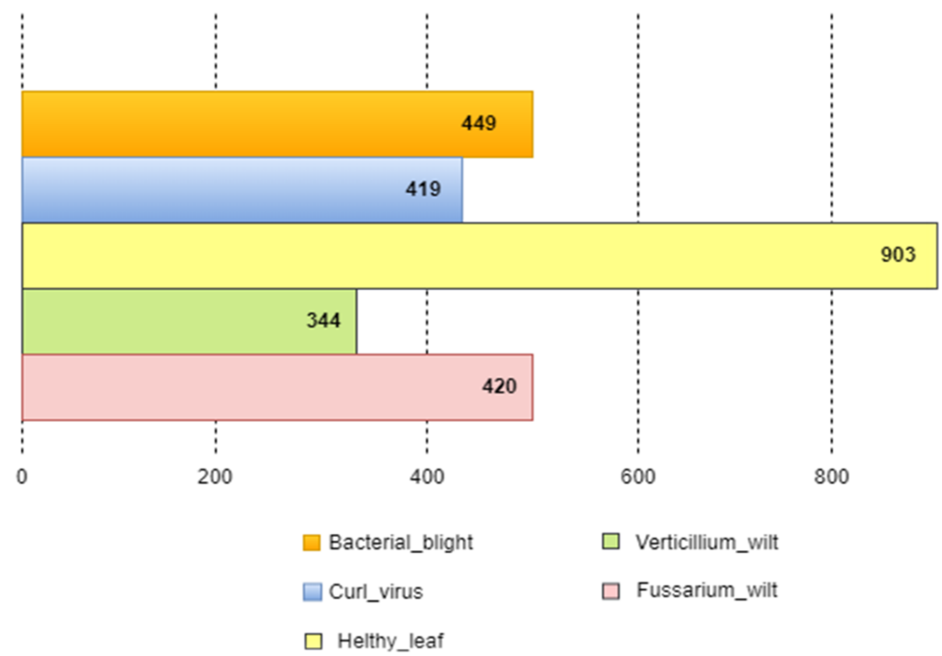

Figure3.3.1: Image volume for different classes.

3.4 Proposed Methodology/Applied Mechanism

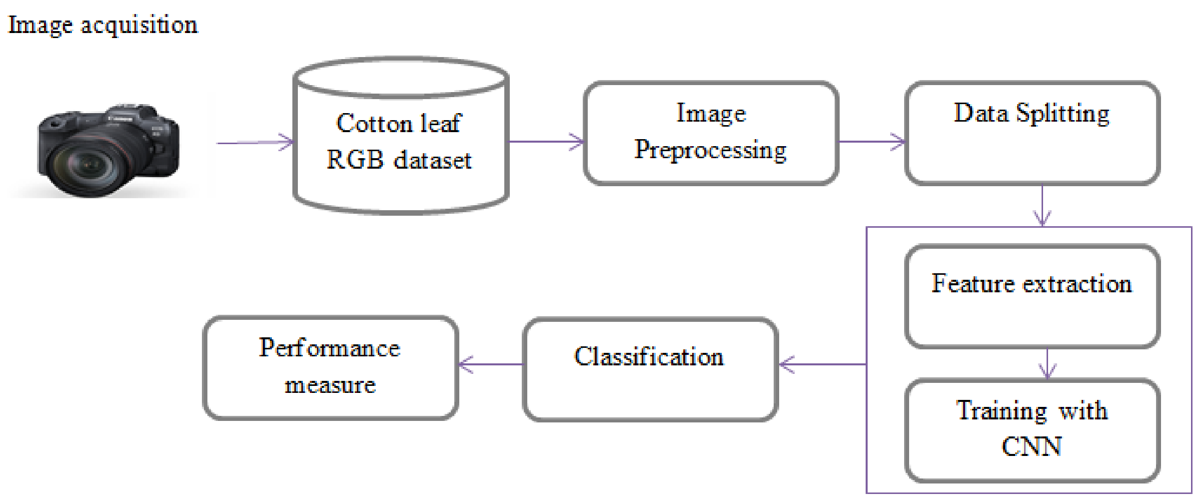

Figure3.4.1: Proposed Methodology

3.4.1.1 Image Acquisition

Image acquisition is the act of retrieving an image from a source, usually through a camera hardware system. This is the first step in image capture. A digital camera is used here for the digital acquisition of this research project. Cotton leaves were photographed using a digital camera attached to a magnifying lens. These images are attached to the PC for the next preprocessing.

3.4.1.2 Data Preprocessing

The first thing I do in pre-processing is scaling, here the NumPy array of images is 0 to 255 which is an RGB scale. I divided it by 255 so I can get a number between 0 and 1 with tf. Keras. sequential layers. I rescaled the images to 1.0/255. Additionally, I resized the images into 256 by 256 pixels as inputs. Because when I train the model and predict the results but the dimensions of the images are not the same then it creates a problem. At that time the resize and rescale layer takes care of that problem. A second step I take during preprocessing is data augmentation to make the model more robust. For that, I am using TensorFlow API RandomFlip and RandomRotation.

3.4.1.3 Data Splitting

After preprocessing the dataset, I split the dataset into training and testing. Training samples are used to train the model and test samples are used to evaluate its performance. And to see how well the model performs on unseen data and achieves target results. The dataset has been split into 80% and 20%. In this case, 80% is allocated towards training, 10% is used for validation and the remaining 10% is used for testing.

3.5 Introduction To Convolutional Neural Network (CNN):

Over the past few decades, deep learning has proven to be a very valuable tool due to its ability to process large amounts of data. However, one of the most popular deep neural networks is the convolutional neural network. CNN is very useful because it reduces human effort by automatically detecting features.

CNN architecture has two main parts. The first is feature extraction, where a convolution layer separates and identifies different features of the image to produce feature maps. Here the feature extraction network consists of many pairs of convolutional or pooling layers. The feature map is then normalized using an activation function which is ReLu. This technique can be repeated multiple times. The second part is classification, a fully connected layer that uses the output from the convolution process and predicts the image class based on the features extracted in the previous stage. As a general rule, I feed cotton images into an input layer which includes a convolutional layer and a maximum pooling layer. The convolutional layer helps break down the images into features, while the maximum pooling layer helps reduce the features by analyzing them. The results of this process are given as input to a fully connected neural network framework that drives the final classification decision. Softmax functions are used to evaluate the reliability of a model and ensure the prediction is between 0 and 1. As shown in Figure 3.5.1.

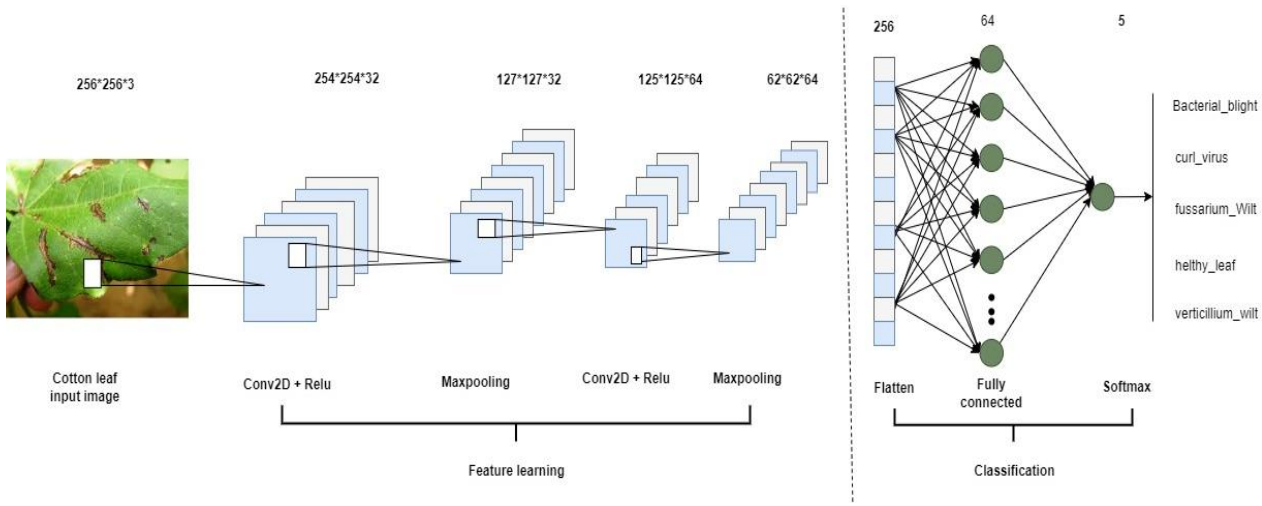

Figure3.5.1: Proposed CNN Architecture.

3.5.1.1 Convolution Layer

The convolution layer extracts features from an input image. This is a two-input mathematical function. One is a filter or kernel, and the other is an image matrix. Here filters are smaller than input data or matrix, so a dot product is applied between a patch of input and filter. Basically, a dot product is the product of the element-wise multiplication of the input patch and the filter patch, which is summed to produce a single value. A smaller filter (set of weights) is intentionally used so it can be multiplied multiple times at various points in the input array using the same filter (set of weights) by running from left to right, top to bottom.

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$

Input matrix Filter Convoluted Map

Figure3.5.1.1.1: Convolution Layer

3.5.1.2 Activation Function (ReLu)

The Rectified Linear Unit is abbreviated as ReLU which is a non-linear function or piecewise linear function that will directly output the input if it is positive, otherwise, it will be zero. It is employed in nonlinear operations f(x) = max(0, x). The goal of ReLU is to identify nonlinearities in convolutions. I will increase the non-linearity of an image using this function. Here, ReLu prevents the problem of Vanishing Gradient.

3.5.1.3 Max Pooling Layer

When the photos are huge, pooling layers are used to reduce the dimension of the feature map. Maximum pooling is a pooling layer that selects large elements from the region of the feature map covered by the filter. The 2X2 filter size as shown in Figure 3.10.1, is used to minimize the size of the matrix which helps us to reduce the computation and processing time.

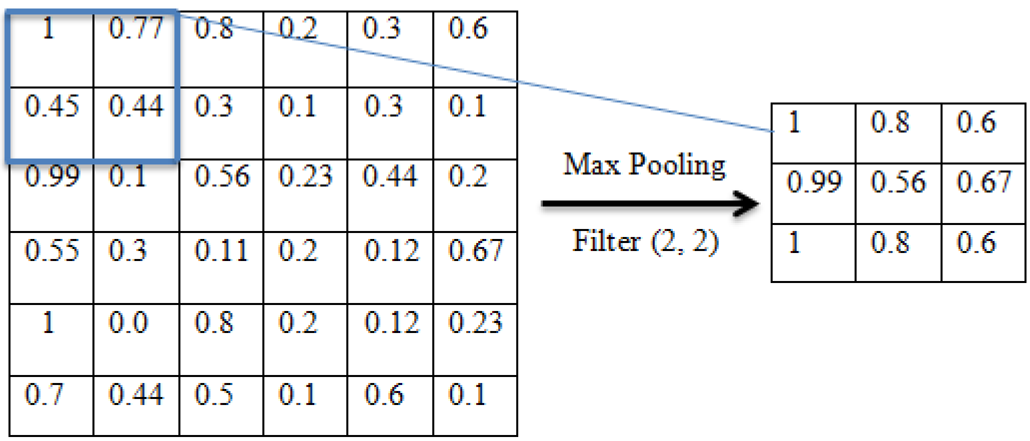

Figure3.5.1.3.1: Max pooling

3.5.1.4 Fully Connected Layer

A fully connected layer is the last layer in CNN which gives the final probabilities for each class. This layer is a combination of the Affine function y = Wx + b and the Non-Linear function (ReLu), which is called 1 FC or 1 Hidden Layer. The fully Connected layer takes input from Flatten Layer and passed it to the Affine function and then to the Non-Linear function. Add multiple layers based on our model.

3.6 Implementation Requirements

3.6.1 Model Construction

The CNN algorithm is used to construct a sequential model. Keras aids in the construction of this model layer by layer. Convolution 2D layers are used to deal with input images, which can be viewed as a 2-dimensional matrix, and dense layers are used to deal with output images. The kernel size as a filter matrix was 3x3. This model's activation function was ReLU. Images must be entered into this model as an input size of 256 by 256. This means the image's height and weight should be 256. Flatten layer served as a link between the convolution 2D layer and the dense layer. Softmax was added to the model as an activation function for prediction based on the highest probability.

3.6.2 Model Compilation

Three parameters optimizer, loss, and metrics are used to compile the model. The learning rate was managed by the optimizer. The optimizer utilized was called "Adam". A good optimizer will change the learning rate as the train progresses. I will utilize 'categorical cross entropy' as the loss function. I'll use the "accuracy" matrix to examine the accuracy score on the validation set and train set when the model was trained.

3.6.3 Training the model

I train the model by using the 'fit ()' function on the model with the train data, validation data, batch size, and epoch parameters. I specified the number of epochs to cycle the model across the data in the fit function. I need to increase the number of epochs up to a particular point to improve the suggested model.

3.7 Experiment

A CNN model is tested with 50 epochs in this study to reduce validation and training loss. This paper presents a CNN model based on deep learning that is divided into two parts: training and prediction. The dataset contains 2535 images in total. 80% of these are used for training, while the remaining 20% are used for validation and testing. The dataset is divided into five categories: bacterial blight, curl virus, healthy leaves, verticillium wilt, and fusarium wilt, with images labeled with 0, 1, 2, 3, and 4. In this case, the model.predict function accurately predicts the output images from the original classes..

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The prediction of cotton disease is a state which describes if a cotton leaf is healthy or unhealthy. This chapter presents and discusses the results of a study that aims to regulate cotton plant disease. In this study, I used the sequential approach to develop the CNN (Convolutional Neural Network) architecture model. However, the model examines the test and train accuracy at the end of the training, and I discover the anticipated result. In this chapter, I'll display the outcome and quickly analyze it. In future, I'll try to create an android-based mobile application to forecast cotton illness.

4.2 Experimental Setup

In this study, a large image data set was used for cotton leaf disease diagnosis. Working with a large data set necessitates an effective analysis technique as well as an experimental setup. For coding, a graded version of "Jupiter Notebook" with Python version 3.9.13 was used. TensorFlow framework was installed, and the necessary libraries, such as TensorFlow, Keras, matplotlib, NumPy, os, and sequential, were imported into the notebook. The Python programming language is used to create and train machine learning models in this case. Because it takes a long time to complete an epoch when training a deep learning model, I used a high-configuration PC with at least 4 GB RAM and a 500 GB hard disk.

4.3 Experimental Result

4.3.1 Training process

The dataset has been separated into 80 and 20 percent for training and validation, respectively, with 20 percent of the validation data being split into 10 percent for test data. The training progress was measured over a period of 50 epochs.

4.3.1.1 Training and Validation loss

In this graph, training loss was 0.13 after one epoch, however it dropped to 0.08 after 50 epochs. In the case of loss of validity, it starts at 0.98 and drops to 0.04 after 50 epochs. As shown as figure 4.3.1.1.1

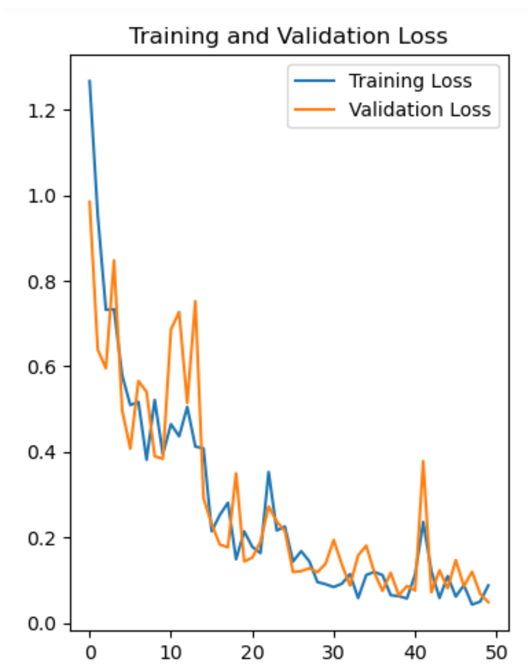

Figure 4.3.1.1.1: Training and Validation loss

4.3.1.2 Training and Validation accuracy

In this area, 1.0 and 0 are considered the best and worst values. The training and validation accuracy in the first epoch were 0.48 and 0.65, respectively, as shown in figure 4.3.1.1.1. After 50 epochs, model training accuracy was around 0.97 percent and validation accuracy was around 0.98 percent. It's a good percentage, in my op

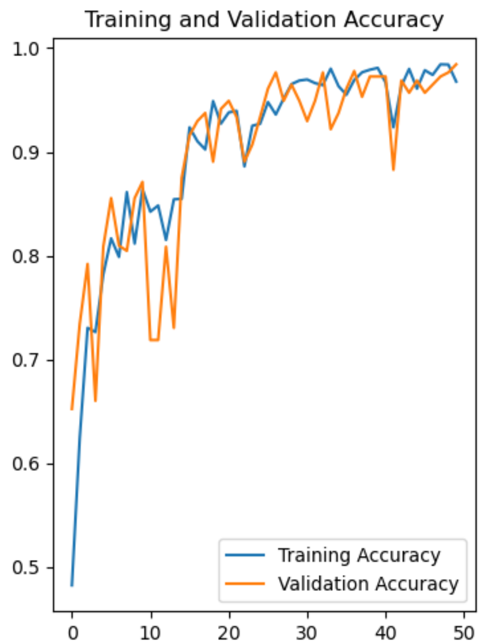

Figure 4.3.1.2.1: Model Training and Validation accuracy

4.4 Evaluation Matrix

The obtained results are discussed in this section. It is used to assess the classifier model's performance in classifying the cotton leaf test image dataset. Key metrics such as precision, recall, and F1-score were used to design and test image classifiers.

The key metrics are shown in the equations below: where accuracy measures a classifier's overall efficiency. The degree to which data labels match the classifier's positive labels is measured by accuracy. The recall of a classifier measures its ability to detect positive labels. The F1 score is the weighted average of precision and recall.

1. True Positive: The classification or identification was correct.

2. False Positive: Misclassification or identification. It represents the type of error we make.

3. False Negative: Incorrectly rejected.

4. True negative: A proper denial.

Accuracy: Accuracy describes the performance of the model across all classes. It evaluates how effectively an algorithm classifies a dataset.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

- Precision: The proximity of two or more measurement values is referred to as precision. The value of accuracy changes due to observational errors. used to ascertain whether exact measurements are repeatable or reliable. A collection of reliable and important statistical data is what is referred to as precision. The finding of recurrent or constant text values is the result of high-precision measurement. Lower precision denotes a variable value for the measurement. Precision indicates how well the positive labels assigned by the classifier match the data labels. when recovering data.

Precision = TP / (TP + FP)

- Recall: The recall fraction is the proportion of the total number of relevant episodes of real recovery that can be recalled. Our model conveys the degree of genuine positive recall by classifying it as genuine (positive). We know that if a False Negative has a high cost, we'll choose our best model based on the Recall model metric. This matrix answers the question, "Was the ratio of actual positives correctly identified?"

Recall = TP / (TP+FN)

- F1-score: The F1- score is calculated as the weighted average of Precision and Recall.

F1-score = 2 × (Recall × Precision) / Recall + Precision

The performance results of using CNN model to build deep learning classifiers was reported in the Figure 4.4.1. The method was implemented in Jupiter Notebook. These results used a train-test split of 80-20 ratio to evaluate the classifiers, from 20% of test dataset 10% used for validation.

Method	Accuracy	Precision	Recall	F1-score
CNN	96.88%	95.56%	96.91%	96.23%

Figure 4.4.1: The performance of CNN model.

4.5 Confusion Matrix

The confusion matrix is essential for assessing deep and machine learning classification performance. Calculating the confusion matrix can help us gain a better understanding of the kinds of mistakes our system makes as well as the appropriate actions. Figure 4.5.1 illustrates how the model mistook 4 photos of bacterial blight for another class, Fusarium wilt, it may happened due to the images' low pixel.

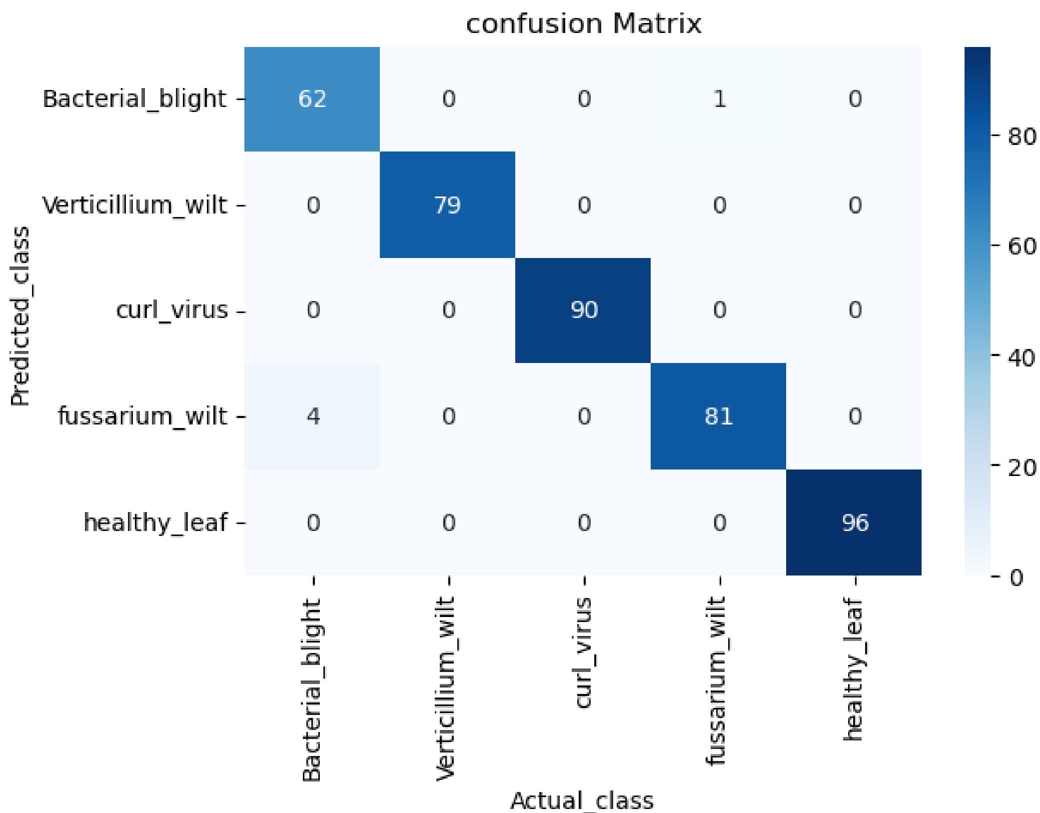

Figure 4.5.1: Confusion Matrix

4.6 Prediction

In figure 4.6.1, this displays the highly confident prediction outcome of my model.

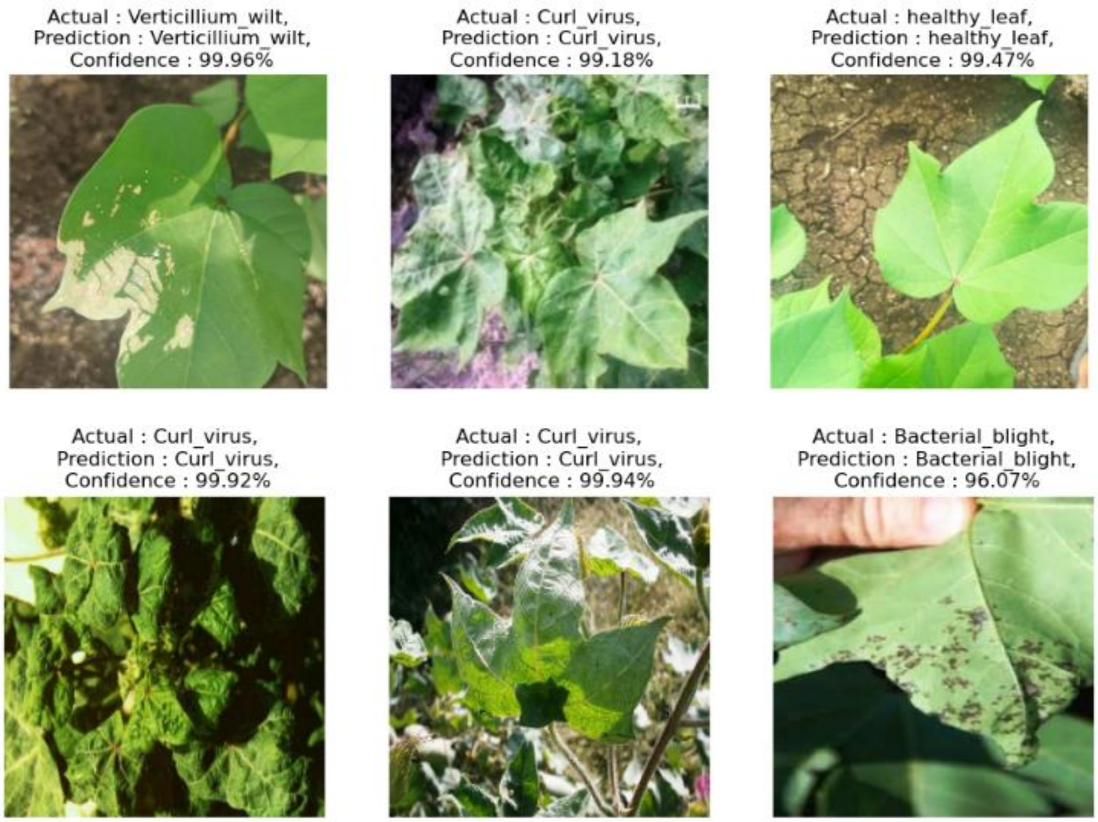

Figure 4.6.1: Predicted Result

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The deep learning-based model was built with Python and the TensorFlow package, with Jupyter serving as the development environment. In this research study, various experiments were carried out in order to obtain an efficient model by customizing various parameters such as the number of epochs, augmentation, and regularization methods. The epoch numbers are very significant in improving model performance by 6.3%, where I trained the model with 20 epochs and then increased the epochs to 50. For the test dataset, the proposed prototype had the best accuracy, coming in at 96.88%. This study also makes highly accurate predictions regarding cotton leaf diseases. The creation of such automated systems helps farmers and professionals recognize cotton illnesses and pests by looking at the visible symptoms on the leaves. The obtained results demonstrate that the designed system for farmers is extremely beneficial in terms of reducing the complexity, time, and cost of diagnosing diseased leaves.

5.2 FUTURE WORK

In the future, I'll strive to create a mobile app for Android devices that can diagnose leaf diseases and be used by all kinds of users. This allows you to take a picture of a cotton leaf immediately to check on its health or not.

REFERENCES

[1] Afzaal, U., Bhattarai, B., Pandeya, Y. R., & Lee, J. (2021). An instance segmentation model for strawberry diseases based on mask R-CNN. *Sensors*, *21*(19), 6565.

[3] Chen, H. C., Widodo, A. M., Wisnujati, A., Rahaman, M., Lin, J. C. W., Chen, L., & Weng, C. E. (2022). AlexNet convolutional neural network for disease detection and classification of tomato leaf. *Electronics*, *11*(6), 951.

[4] Jenifa, A., Ramalakshmi, R., & Ramachandran, V. (2019, December). Cotton leaf disease classification using deep convolution neural network for sustainable cotton production. In *2019 IEEE international conference on clean energy and energy efficient electronics circuit for sustainable development (INCCES)* (pp. 1-3). IEEE.

[5] Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., & Sun, Z. (2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and electronics in agriculture*, *154*, 18-24.

[6] Krishnamoorthy, N., Prasad, L. N., Kumar, C. P., Subedi, B., Abraha, H. B., & Sathishkumar, V. E. (2021). Rice leaf diseases prediction using deep neural networks with transfer learning. *Environmental Research*, *198*, 111275.

[7] Arivazhagan, S., & Ligi, S. V. (2018). Mango leaf diseases identification using convolutional neural network. *International Journal of Pure and Applied Mathematics*, *120*(6), 11067-11079.

[8] Bhimte, N. R., & Thool, V. R. (2018, June). Diseases detection of cotton leaf spot using image processing and SVM classifier. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 340-344). IEEE.

[9] Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019, March). A review on machine learning classification techniques for plant disease detection. In *2019 5th International conference on advanced computing & communication systems (ICACCS)* (pp. 281-284). IEEE.

[10] Panigrahi, K. P., Das, H., Sahoo, A. K., & Moharana, S. C. (2020). Maize leaf disease detection and classification using machine learning algorithms. In *Progress in Computing, Analytics and Networking* (pp. 659-669). Springer, Singapore.

[11] Kaur, P., Harnal, S., Tiwari, R., Upadhyay, S., Bhatia, S., Mashat, A., & Alabdali, A. M. (2022). Recognition of leaf disease using hybrid convolutional neural network by applying feature reduction. *Sensors*, *22*(2), 575

[12] Saleem, M. H., Potgieter, J., & Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants*, *8*(11), 468.

[13] Sabrol, H., & Satish, K. (2016, April). Tomato plant disease classification in digital images using classification tree. In *2016 International Conference on Communication and Signal Processing (ICCSP)* (pp. 1242-1246). IEEE.

[15] Warne, P. P., & Ganorkar, S. R. (2015). Detection of diseases on cotton leaves using K-mean clustering method. *International Research Journal of Engineering and Technology (IRJET)*, *2*(4), 425-431.