VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka – 590018

A

Project Report

On

"PLANT DISEASE DETECTION USING DEEP LEARNING"

Submitted By

ASHRITHA P	**(1KT17IS001)**
NIRMALA V	**(1KT17IS008)**
SHRAVANI H A	**(1KT17IS018)**
SHUBHASHINI PAL	**(1KT17IS019)**

Under the guidance

of

Mrs . SHRUTI B. P.

Assistant Professor

Dept.of ISE

SRI KRISHNA INSTITUTE OF TECHNOLOGY

Department of Information Science and Engineering

No.29, Hesaraghatta Main Road, Chimney hills, Chikkabanavara P.O., Bengaluru – 560090

2020-2021

SRI KRISHNA INSTITUTE OF TECHNOLOGY

No.29, Hesaraghatta Main Road, Chimney hills, Chikkabanavara P.O., Bengaluru – 560090

Department of Information Science and Engineering

CERTIFICATE

Certified that the Project Work entitled **"PLANT DISEASE DETECTION USING DEEP LEARNING"** carried out by **Ashritha P (1KT17IS001), Nirmala V (1KT17IS008), Shravani H A (1KT17IS018)** and **Shubhashini Pal (1KT17IS019),** bonafide students of **Sri Krishna Institute of Technology**, Bengaluru in partial fulfillment for the award of **Bachelor of Engineering** in **Information Science and Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2020-21. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements with respect to Project Work prescribed for the said degree.

Signature of Guide	**Signature of HOD**	**Signature of Principal**
Mrs. Shruti B. P.	**Dr. Hemalatha K.L.**	**Dr. Manjunatha. A.**
Assistant Professor	Professor and HOD	Principal
Dept. of ISE, SKIT	Dept. of ISE, SKIT	SKIT

ABSTRACT

Plants play an essential role in climate change, agriculture industry and a country’s economy. Thereby taking care of plants is very crucial. Just like humans, plants are affected by several disease caused by bacteria, fungi and virus. Identification of these disease timely and curing them is essential to prevent whole plant from destruction. This project proposes a deep learning-based model named plant disease detector. The model is able to detect several diseases from plants using pictures of their leaves. Plant disease detection model is developed using neural network. First, all augmentation is applied on dataset to increase the sample size. Later Convolution Neural Network (CNN) is used with multiple convolution and pooling layers. Plant Village dataset is used to train the model. After training the model, it is tested properly to validate the results. A set of data from Plant Village data will be used for testing purpose that contains images of healthy as well as diseased plants. Proposed model is expected to achieve high level accuracy. This study is focused on deep learning model to detect disease in plant leave. But, in future this model can be integrated with drone or any other system to live detect diseases from plants and report the diseased plants location to people so that they can be cured accordingly.

ACKNOWLEDGEMENT

The completion of Project Work brings with a sense of satisfaction, but it is never complete without thanking the persons responsible for its successful completion.

At the outset, we express our most sincere grateful acknowledgment to the holy sanctum **“Sri Krishna Institute of Technology”,** the temple of learning, for giving us an opportunity to pursue the degree course in Information Science and Engineering and thus helping us in shaping the career.

We express our deep sense of sincere gratitude to **Dr. Manjunatha. A, Principal**, **Sri Krishna Institute of Technology,** Bengaluru, for providing us an opportunity to continue our higher studies.

We extend our heartfelt sincere gratitude to **Dr. Hemalatha K.L.**, **Professor and HOD, Department of Information Science and Engineering**, Bengaluru, Sri Krishna Institute of Technology, for her valuable suggestions and support.

We extend our gratitude to our guide **Mrs. Shruti B. P., Assistant Professor, Department of Information Science and Engineering**, Bengaluru, who inspired us in taking up this project and guided with required necessities for carrying out of the Project Work.

We extend our heartfelt gratitude to our Project Coordinator **Mrs. Sandhya B. R., Assistant Professor, Department of Information Science and Engineering**, Sri Krishna Institute of Technology, Bengaluru, for her valuable and timely updations who has rendered us completing this Project Work in time.

We would like to thank all the teaching and non-teaching staff members in our **Department of Information Science and Engineering**, Sri Krishna Institute of Technology**,** Bengaluru, for their support.

Finally, we would like to thank all our friends and family members for their constant support, guidance and encouragement.

ASHRITHA P	**(1KT17IS001)**
NIRMALA V	**(1KT17IS008)**
SHRAVANI H A	**(1KT17IS018)**
SHUBHASHINI PAL	**(1KT17IS019)**

TABLE OF CONTENTS

CHAPTER NO.	CHAPTER NAME	PAGE NO.
	6.3 Pseudo Code for Automated Leaf Disease Classification Module	24
CHAPTER 7	**TESTING**	25
	7.1 Unit Testing	25
	7.2 Integration Testing	26
	7.3 System Testing	27
CHAPTER 8	**RESULTS AND DISCUSSION**	28
CHAPTER 9	**CONCLUSION AND FUTURE ENHANCEMENT**	30
	9.1 Conclusion	30
	9.2 Future Enhancement	30
BIBLIOGRAPHY		
APPENDIX	**APPENDIX A: ACRONYMS**	
	APPENDIX B: SNAPSHOTS	

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Plants play a vital role in economy and climate change. Since climate change has become a global issue also addressed in UN general assembly 2019, various countries are on mission to plant more and more trees and plants to keep climate balance. Many studies have proved that extinction of plants due to industry use have caused damage to ozone layer and thus resulting global warming. The rate of climate change forecast for the future is 10-100 times faster than the rate of deglacial warming.

These plants also play a major role in food industry as well. Balance of global food production is also a major issue. Apart from this in health care also plants play a vital role. Overall plants are very essential for human survival therefore it's also a worldwide concern to take care of them. Just like human health plants health can also be affected by several diseases. In economic terms, annual losses in food, fiber and ornamental production systems caused by plant pests and diseases are estimated in the hundreds of billions of Revised Manuscript received on April 21, 2020. These diseases are caused by fungi or fungal like organisms. However, other serious diseases of food and feed crops are caused by viral and bacterial organisms.

Some of the diseases may be of spreadable nature, means they may spread from one plant to other hence needed to be identified and taken care timely. It's very challenging job to detect disease in plants in very early stages. Some of the common symptoms of disease in plants are Leaf rust (common leaf rust in corn), Stem rust (wheat stem rust), Sclerotinia (white mold), Powdery mildew, Birds-eye spot on berries (anthracnose), Damping off of seedlings (phytophthora), Leaf spot (Septoria brown spot), Chlorosis (yellowing of leaves). These diseases can be identified by physical condition of plants leaves. The experts can identify either the plant is defected or not by looking at leaves stems or fruit. This approach requires having lots of human resources for this particular job. In this era of technology and automation it is not a very efficient approach, it would be much better if we have an automated system which detects disease in plants automatically. There are many researches already done to fill this purpose most of them utilize traditional machine learning approaches.

The purpose of this study is to create such automated system for detecting diseases in plants by using deep learning technique. Deep learning is subset of machine learning, to worry about domain expertise as no feature engineering is required in this, unlike learning. The advantage of deep learning over machine learning is that one does not need traditional machine learning approaches. Our system just like other previous researches utilizes images of plants leaves to detect disease in plants. Plant disease detector is computer vision based automated plant disease diagnostic system which utilizes machine learning techniques to correctly identify disease and healthy plants also the type of disease. For achieving so deep learning network for images like Convolution Neural Network (CNN) can be utilized. CNN is used for feature extraction. The CNN based network can be trained for detecting disease in plants by providing huge amount of images of healthy and sick plants and trained model in future can be used to predict the disease in plants by images of plants leaves.

1.1 Problem Statement

India is the land of Agriculture. It is the main occupation in India. Two-third of population is dependent on agriculture directly or indirectly. Agriculture is one of the major sectors of the Indian economy. It is present in the country for thousands of years. Over the years it has developed and the use of new technologies and equipment replaced almost all the traditional methods of farming. Besides, in India, there are still some small farmers that use the old traditional methods of agriculture because they lack the resources to use modern methods. Furthermore, this is the only sector that contributed to the growth of not only itself but also of the other sector of the country.

As agriculture struggles to support the rapidly growing global population, plant disease reduces the production and quality of food, fibre and biofuel crops. Losses may be catastrophic or chronic. Plant diseases have caused severe losses to farmers in several ways. Many plant diseases also affect the human health when consumed.

Presently there is no right approach in identifying the diseases in early stages. Most of the time it's miles the case that farmer is blind to the ailment type. In such cases we would like to use the emerging technologies like machine learning, to build a system that can identify various plant diseases based on the images of plants using Deep Learning Technique. This can help the farmers, economy of the country and also the lives of people dependent on agriculture, which is the source of food.

1.2 Objectives

- To detect diseases in plant leaves with better accuracy rates.
- To classify the detected leaves accurately using machine learning techniques.
- To achieve higher classification rate with the given training data.
- To classify the type of diseases in plant mainly Cercospora, Miner, Phoma, Rust.
- To reduce computational time with respect to the existing system.

1.3 Scope of the Project

We are setting up to design a system, which uses the deep learning technique. The system includes a set of data taken into CNN model using LeNet architecture with a data processing to recognize the output whether it is diseased or not. If it is diseased, the system then uses the RELU to further classify the type of disease. We train the model using the available dataset for normal abnormal disease images. In our system, abnormal images of four types are trained, namely, Cercospora, Miner, Phoma and Rust. For training the model, various features are extracted and classified accordingly using algorithm. The output provides the classification of the image.

1.4 Organization of the Report

The report has been organized into nine Chapters. Chapter 1 gives a brief Introduction about the project. This section states the problem with the Existing Systems, Objectives of the undertaken project and its Scope. Background works have been included in Chapter 2 under the name Literature Survey. Chapter 3 includes the System Requirements Specification. System Analysis is included as part of Chapter 4. Various designs of the undertaken project have been included in Chapter 5. Chapter 6 comprises of Project pseudo code. Various types of Testing carried out in the project have been defined in Chapter 7. Results of the project and comparisons of the same with Existing Systems are included in Chapter 8. Chapter 9 provides overall Conclusion of the project and Enhancements of the project that can be undertaken in future. Bibliography followed by Appendix section which includes Acronyms and Snapshots from the last section of the Report.

CHAPTER 2

LITERATURE SURVEY

Some of the papers have been surveyed with respect to the proposed work. Few of them are listed below

[1] Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks

Alternaria leaf spot, Brown spot, Mosaic, Grey spot, and Rust are five common types of apple leaf diseases that severely affect apple yield. However, the existing research lacks an accurate and fast detector of apple diseases for ensuring the healthy development of the apple industry. This paper proposes a deep learning approach that is based on improved Convolutional Neural Networks (CNNs) for the real-time detection of apple leaf diseases. In this paper, the Apple Leaf Disease Dataset (ALDD), which is composed of laboratory images and complex images under real field conditions, is first constructed via data augmentation and image annotation technologies. Based on this, a new apple leaf disease detection model that uses deep-CNNs is proposed by introducing the GoogLeNet Inception structure and Rainbow concatenation. Finally, under the hold-out testing dataset, using a dataset of 26,377 images of diseased apple leaves, the proposed INAR-SSD (SSD with Inception module and Rainbow concatenation) model is trained to detect these common apple leaf diseases. The experimental results show that the INAR-SSD model realizes a detection performance of 78.80% MAP on ALDD, with a high detection speed of 23.13 FPS. The results demonstrate that the novel INAR-SSD model provides a high-performance solution for the early diagnosis of apple leaf diseases that can perform real-time detection of these diseases with higher accuracy and faster detection speed than previous methods.

Drawback : Lacks in accuracy and fast detection.

[2] Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection

Plant disease, especially crop plants, is a major threat to global food security since many diseases directly affect the quality of the fruits, grains, and so on, leading to a decrease in agricultural productivity. Farmers have to observe and determine whether a leaf was infected by naked eyes. This process is unreliable, inconsistent, and error prone. Several works on deep learning techniques for detecting leaf diseases had been proposed. Most of them built their models based on limited resolution images using Convolutional Neural Networks (CNNs). In this research, we aim at detecting early disease on plant leaves with small disease blobs, which can only be detected with higher resolution images, by an Artificial Neural Network (ANN) approach. After a pre-processing step using a contrast enhancement method, all the infested blobs are segmented for the whole dataset. A list of several measurement-based features that represents the blobs are chosen and then selected based on their influences on the model's performance using a wrapper based feature selection algorithm, which is built based on a hybrid metaheuristic. The chosen features are used as inputs for an ANN. We compare the results obtained using our methods with another approach using popular CNN models (AlexNet, VGG16, ResNet-50) enhanced with transfer learning. The ANN's results are better than those of CNNs using a simpler network structure (89.41% vs 78.64%, 79.92%, and 84.88%, respectively). This shows that our approach can be implemented on low-end devices such as smartphones, which will be of great assistance to farmers on the field.

Drawback : Complex and time consuming due to long scripts which are used in the algorithm.

[3] A Unified Matrix-Based Convolutional Neural Network for Fine-Grained Image Classification of Wheat Leaf Diseases

Fine-grained image classification methods often suffer from the challenge that the subordinate categories within an entry-level category can only be distinguished by subtle differences. Crop disease classification is affected by various visual interferences, including uneven illumination, dew, and equipment jitter. It demands an effective algorithm to accurately discriminate one category from the others. Thus, the representational ability of algorithm needs to be strengthened to learn a robust domain-specific discrimination through an effective way. To address this challenge, a unified Convolutional Neural Network (CNN) denoting the Matrix-based Convolutional Neural Network (M-bCNN) was proposed. Its hallmark is the convolutional kernel matrix, whose convolutional layers are arranged parallelly in the form of a matrix, and integrated with DropConnect, exponential linear unit, local response normalization, and so on to defeat over-fitting and vanishing gradient. With a tolerable addition of parameters, it can effectively increase the data streams, neurons, and link channels of the model compared with the commonly used plain networks. Therefore, it will create more non-linear mappings and will enhance the representational ability with a tolerable growth of parameters. The images of winter wheat leaf diseases were utilized as experimental samples for their strong similarities among sub-categories. A total of 16,652 images containing eight categories were collected from Shandong Province, China, and were augmented into 83,260 images. The M-bCNN delivered significant improvements and achieved an average validation accuracy of 96.5% and a testing accuracy of 90.1%; this outperformed AlexNet and VGG-16. The M-bCNN demonstrated accuracy gains with a convolutional kernel matrix in fine-grained image classification.

Drawback : Multi-conceptual process, complex structure.

[4] Soybean Seed Counting Based on Pod Image Using Two-Column Convolution Neural Network

China's soybean supply and demand are seriously imbalanced. It is crucial to improve the level of soybean breeding. Hundred-grain weight is one of the most essential phenotypic parameters for crop breeding. Accurate soybean seed counting is a key step for 100-grain weight. There are several seed counting methods, which have their own limitations one way or the other. Among these, manual counting is time-consuming, electronic automatic seed counter devices are expensive and their counting speed is very slow, and the traditional digital image processing techniques are not suitable for seed counting based on individual pod images. This paper attempted to develop a method that would combine the density estimation based methods and the Convolution Neural Network (CNN)-based methods to accurately estimate the seed count from an individual soybean pod image with a single perspective. In this paper, we first introduced a new large-scale seed counting dataset, named Soybean-pod. The dataset contains 500 annotated pod images with a total of 32 126 seeds and is the largest annotated dataset for soybean seed counting so far. Simultaneously, we used annotation information to generate a ground-truth density map by convolving a Gaussian kernel and, then, devised a simple but effective method that would elucidate pod images to a seed density map using a Two-Column CNN (TCNN) and thus accomplish seed counting ultimately. We conducted relevant experiments from three aspects on the new dataset to verify the effectiveness of our model and method, which provided 13.21 Mean Absolute Error (MAE) and 17.62 Mean Squared Error (MSE). In addition, our research results showed that deep learning techniques can be easily adapted to precision tasks for plant phenotyping and breeding purposes.

Drawback : It is specifically designed for only this learning model.

[5] Detection and Classification of Leaf Disease Using Artificial Neural Network

Describes a diagnosis process that is mostly visual and requires precise judgment and also scientific methods. Image of diseased leaf is captured. As the result of segmentation Color HSV features are extracted. Artificial Neural Network (ANN) is then trained to distinguish the healthy and diseased samples. ANN classification performance is 80% better in accuracy.

Drawback : Accuracy rates are less, less clarity of database.

[6] Plant Disease Detection Using Image Processing Techniques

In this it describes the uploaded pictures captured by the mobile phones are processed in the remote server and presented to an expert group for their opinion. Computer vision techniques are used for detection of affected spots from the image and their classification. A simple color difference-based approach is followed for segmentation of the disease affected lesions. The system allows the expert to evaluate the analysis results and provide feedbacks to the famers through a notification to their mobile phones. The goal of this research is to develop an image recognition system that can recognize crop diseases. Image processing starts with the digitized color image of diseased leaf. A method of mathematics morphology is used to segment these images. Then texture, shape and color features of color image of disease spot on leaf were extracted, and a classification method of membership function was used to discriminate between the three types of diseases.

Drawback : Classification rates are less, failed to validate standard database.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Hardware Requirements

A Hardware requirement list is often accompanied by hardware compatibility list, essentially in the case of operating system.

• Processor	:	Core i3, GPU recommended
• Storage	:	160GB
• Memory	:	4 GB

3.2 Software Requirements

Software requirements deals with defining software reduces requirements and prerequisites that need to be installed on computer to provide optimal functioning of an application.

• Operating System	:	Windows7
• Language	:	Python 3.9.2 and Anaconda individual edition

3.3 Functional Requirements

1. **Hardware Interfaces:** The outside equipment interface utilized for ordering and looking is PCs of the customers. The PC's might be portable PCs with remote LAN as the web association.
2. **Software Interfaces:** The working Frameworks can be any rendition of windows.
3. **Performance Prerequisites:** The PC's utilized must be at least Pentium 4 machine with the goal that they can give ideal execution of the item.

3.4 Non-Functional Requirements

Non-utilitarian necessities are the capacities offered by the framework. It incorporates time imperative and requirement on the advancement procedure and models. The non-useful prerequisites are as per the following:

1. **Speed:** The application is aimed to decrease the time of computational speeds.
2. **Ease of Utilization:** The application is developed using GUI in python, where its can be used by a common man without the thorough knowledge.
3. **Reliability:** The rate of disappointments ought to be less than compared to other systems providing the same functionality, in other words the application is aimedto be noise free.
4. **Portability:** The application is made to run or work on several computers provided, it has required software environment.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Existing System

The Figure 4.1 below demonstrates the complete process flow of the system. The basic pre-processing module is followed by an important segmentation module. The color and texture features for each segmented cluster are extracted to train the system. The training phase is completed in two stages. The first stage classifier learns distinguishing features between a healthy and an unhealthy coffee leaf image sample. Classifiers in the second stage learn from features of infected clusters to classify a leaf image sample into one of the three disease categories. While training, observations are made and some rules are designed to ease classifier selection during testing. The testing phase starts with the same two initial modules. Among all the segmented clusters, if the one with maximum grey level value contains a single connected component, then the process stops giving healthy leaf as an output. In case this condition holds false, then depending upon the output of the first stage classifier a matching rule is referred to select a single classifier from the second stage. The output of the selected classifier gives the disease infecting the test leaf image sample.

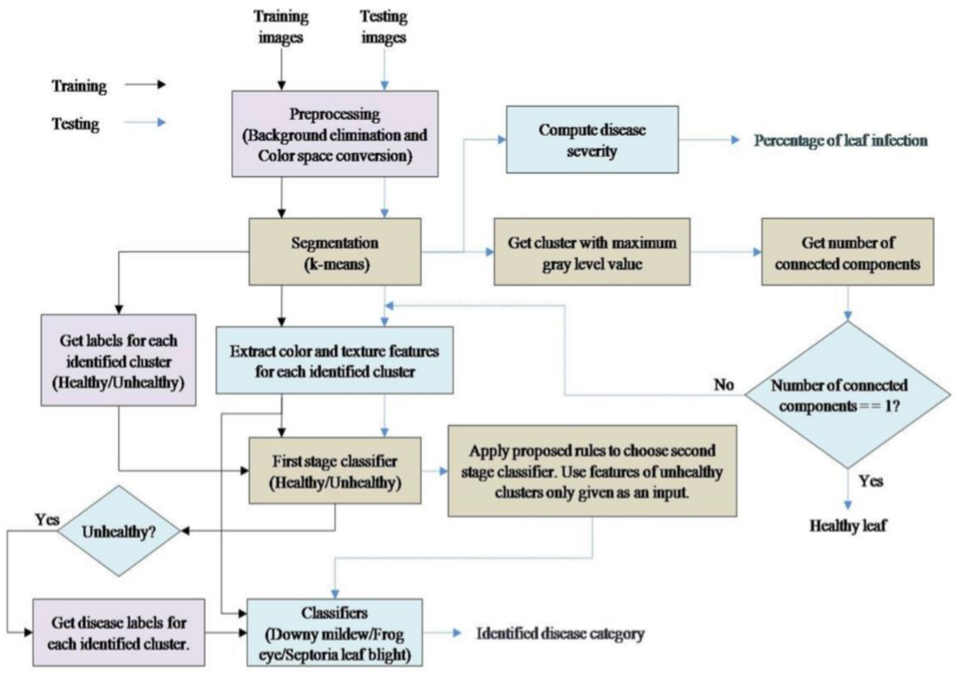

Training images
Testing images
Training
Testing
Preprocessing (Background elimination and Color space conversion)
Compute disease severity
Percentage of leaf infection
Segmentation (k-means)
Get cluster with maximum gray level value
Get number of connected components
Get labels for each identified cluster (Healthy/Unhealthy)
Extract color and texture features for each identified cluster
No
Number of connected components == 1?
Yes
Healthy leaf
First stage classifier (Healthy/Unhealthy)
Apply proposed rules to choose second stage classifier. Use features of unhealthy clusters only given as an input.
Yes
Unhealthy?
Get disease labels for each identified cluster.
Classifiers (Downy mildew/Frog eye/Septoria leaf blight)
Identified disease category

Figure 4.1: Complete Process Flow of the System

4.1.1 Preprocessing

The complex leaf image background is first removed to get a clear leaf sample before further processing. First, a Region of Interest (ROI) is selected manually followed by the computation of binary mask and its complements. The addition of the original image to the obtained complement removes the irrelevant background. Finally, cropping eliminates extra white spaces and completes background elimination process. The next pre-processing step is the color space conversion. The color space is converted from a device dependent RGB model into a device independent model. The proposed system uses L*a*b* (L* signifies the lightness, a* and b* are the chromaticity layers) color space. It closely resembles human perception and also splits information about chrominance better than other models The background eliminated resized RGB image is converted into L*a*b* first, then, the segmentation module is executed on a*b* channel. Using only two channels for color representation decreases the processing time as well. Moreover, L*a*b* has a strong aptness towards good segmentation results as compared to other color models.

4.1.2 Feature Extraction

The literature for plants disease detection conveys that color and texture play an important role in disease lesion classification. Thus, several color features, texture features, and their combinations are explored in this study to design a valuable system and to validate its performance

4.1.3 Classifiers

The dataset used is imbalanced thus for improved classification accuracy the proposed system uses three classifiers. One is to identify a healthy or infected cluster and two are utilized to resolve the clashing rules. CLFRHD learns healthy and infected clusters, CLFRFE_SLB differentiates frog eye from septoria leaf blight, and CLFRDM_SLB classifies between downy mildew and septoria leaf blight. CLFRHD uses features of all the clusters, CLFRFE_SLB and CLFRDM_SLB are trained from the features of infected clusters only. Any classifier can be chosen, but in the current implementation, SVM is used due to their numerous advantages in high-dimensional space. SVM deals beautifully with noisy data and also prevents over-fitting . A two-class SVM learns an optimal hyper-plane g(x) to properly split data points of the classes. Here, is a non-zero valued Lagrange multiplier, zn is its corresponding support vector, t is the size of input data, c is the threshold, and xn represents an input data point

$$g\ x = \Sigma\ 1 \leq n \leq t\ znan\ xn+x +c$$

4.2 Disadvantages of Existing System

1. Techniques which were used by researchers for Plant leaf classification resulted in accuracy drop. This needs to be addressed.
2. Many existing techniques couldn't extract enough features from the plant leaves. Hence, need for proper combination of feature extraction methods.
3. Images with overlapping leaves are difficult to classify.
4. Partial shading, multiple objects and non-centric images not covered.
5. Complex background, blur, low illumination.

4.3 Proposed System

The Figure 4.2 shows the steps taken for detection and classification of image. Inspired by AlexNet architecture, a Multilayer Convolutional Neural Network is proposed in this work for the classification of the leaves infected with the disease. The procedure of proposed method is revealed by the algorithm given below.

1. Acquire the real-time images of the Mango tree containing both diseased and non-diseased leaves and also, images from PlantVillage dataset.
2. Preprocess all the images for contrast enhancement using histogram equalization method and rescaling using central square crop method.
3. Assign the class labels to the images.
4. Categorize the images among training and testing dataset selecting from all the class labels.
5. Train the CNN with the help of training images.
6. Test the CNN with the help of testing images.
7. Validate the performance of the proposed model and compare the results with the other state-of-the-art approaches.

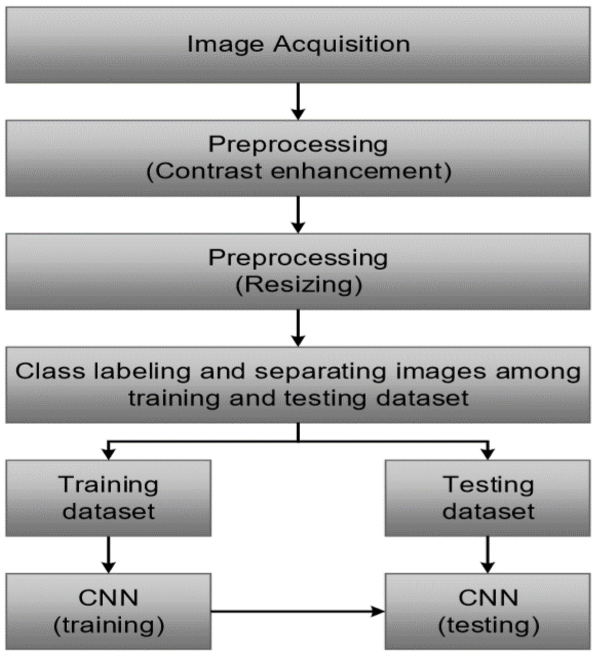


Image Acquisition
Preprocessing (Contrast enhancement)
Preprocessing (Resizing)
Class labeling and separating images among training and testing dataset
Training dataset
Testing dataset
CNN (training)
CNN (testing)

Figure 4.2 : Steps Taken for Detection and Classification of Image

4.4 Advantages of Proposed System

1. **Accuracy:** The classification of the detected leaves accurately using machine learning techniques.
2. **Classification:** The type of leaf disease can be classified with least error.
3. **Computational Time:** The proposed system reduces the computational time effectively by using hybrid features.
4. **Graphical User Interface:** The system provides an easy to use GUI for the user to carry out the operations.

CHAPTER 5

SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design mainly focuses on detailed specification of elements like process, data, input, user and output of the system. It uses a list of requirements to design a model. It also describes how proposedSystem is to be built.

5.1 System Architecture

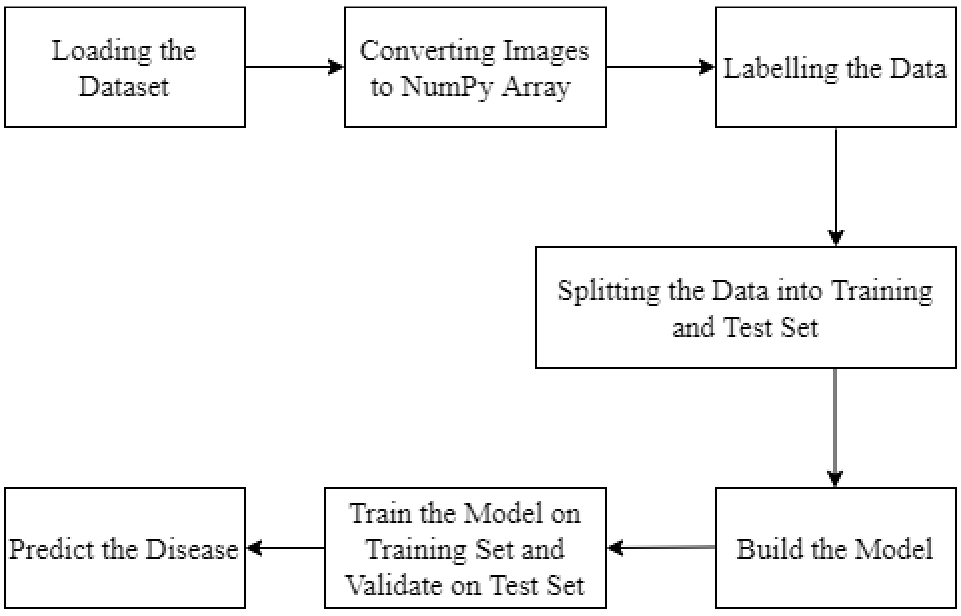

Loading the Dataset
Converting Images to NumPy Array
Labelling the Data
Splitting the Data into Training and Test Set
Build the Model
Train the Model on Training Set and Validate on Test Set
Predict the Disease

Figure 5.1 : Block Diagram of the Proposed Scheme

The proposed plant disease prediction method takes input from the plant's leaves images. Figure 5.1 represents the block diagram of the proposed method. Firstly the data is preprocessed by resizing the input images and further a NumPy array is created for the same. Next the dataset and label of all the images are segregated. The model has been trained on a specific data set consisting of images of the different diseased plant leaves which are considered for this study. The labeled data is now stored in pickle files which are again extracted during the training period of the model. For the model, the convolution layers are declared followed by max-pooling layers. After that, 25% of the whole data is dropped out. The output is flattened to feed the dense network. The last layer has a softmax activation to predict the disease of the given leaf. To reduce the loss function Adam optimizer is utilized. The framework consequently distinguishes the picture of leaf given and pre-processes the picture further for prediction. The model will produce 15 distinctive probability values for 15 labels respectively among which the probability value with highest score to the relating name will be the anticipated disease or result for that particular image.

A. Input Layer

In this layer input is fed to the model. At this beginning stage of the neural network, the no of neurons and number of features are equal. Considering an image the number of pixels in it is equivalent to the total number of features. The input data is divided into two parts which are used for training and testing the model. The major part of data is used for training and the minor part of it is used for testing.

B. Hidden Layer

This layer receives the output from the input layer. It is dependent upon both the model and size of data as well. Number of neurons may vary in each of the hidden layer.

C. Output Layer

A logistic function receives the data from hidden layer as input. The probability score is obtained for each class by converting the output of each class by a logistic function. It coverts each class output into an equivalent probability score for the same.

5.2 Dataflow Diagram for the Proposed Model

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel unlike a flowchart which also shows this information.

The Figure 5.2 shows the level 0 data flow diagram of the proposed model. Here, the image will be read from the user interface. The image is pre-processed and region of interest is detected and further the classification is done based on the detected diseased sites.

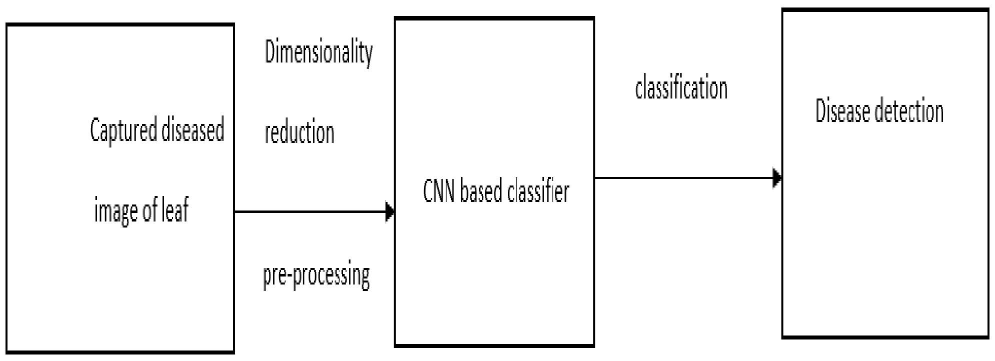


Figure 5.2: Level 0 Data Flow Diagram of the Proposed Model

Figure 5.3 shows the level 1 data flow diagram of the proposed model. The diagramconsists of 2 phases: Training phase and testing phase. In the training phase, the dataset isgiven along with the labels of what type of disease is present in the image. The image is pre-processed and the required features are extracted from the image. Then later the type of disease is classified using the RELU and the disease is predicted. In the Testing phase, the dataset is given as an input to the system, which is further pre-processed and the ROI is extracted from the image. Finally, using RELU the type of leaf disease is detected and the results are displayed on the screen.

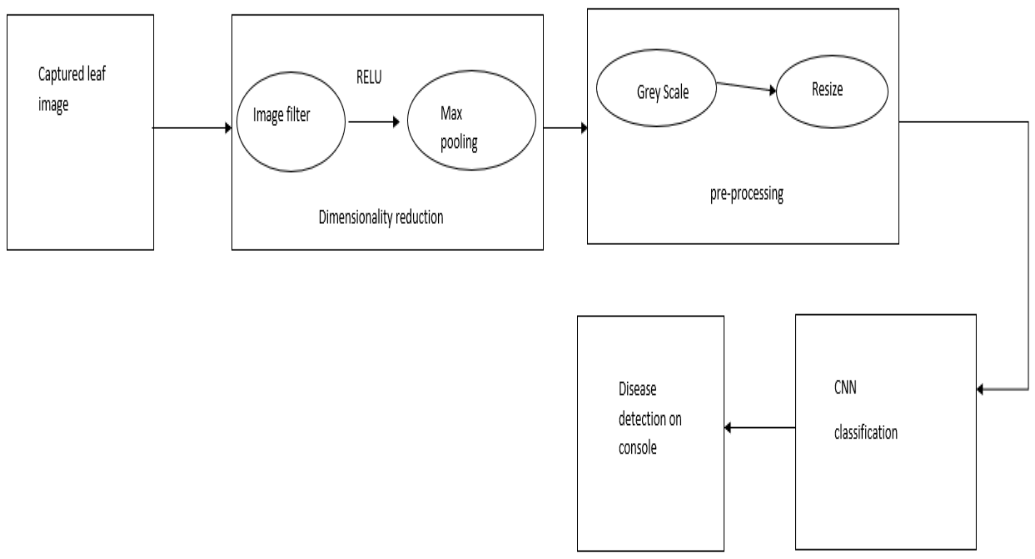


Figure 5.3: Level 1 Data Flow Diagram of the Proposed Model

Figure 5.4 shows the level 2 data flow diagram of the proposed model. The leaf disease image that needs to be classified is given as input for the user interface. This image further undergoes pre-processing like enhancement and soothing of the images which makes it easier for feature extraction further. Then the required features are extracted. Later the identified disease sites are classified using the RELU as normal or diseased leaf.

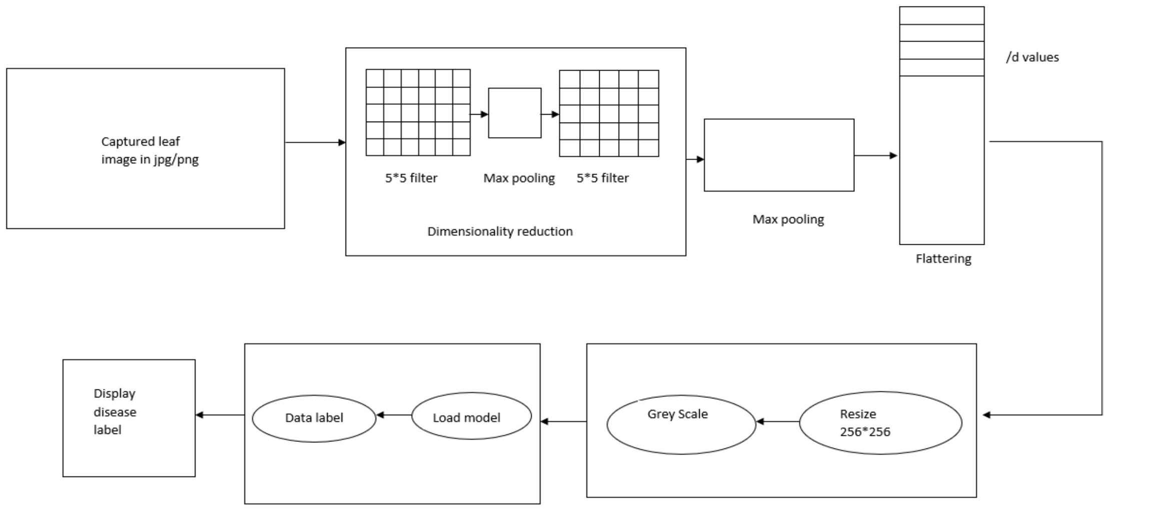


Figure 5.4: Level 2 Data Flow Diagram of the Proposed Model

CHAPTER 6

IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage, the main workload and the major impact on the existing system shift to user department. If the implementation is not carefully planned and controlled, it can cause confusions.

The name “Convolutional Neural Network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

A Convolutional Neural Network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. The Figure 6.1 shows applied methodology of CNN architecture

Though the layers are colloquially referred to as convolutions, this is only by convention. Mathematically, it is technically a sliding dot product or cross-correlation. This has significance for the indices in the matrix, in that it affects how weight is determined at a specific index point.

When programming a CNN, the input is a tensor with shape (number of images) x (image height) x (image width) x (image depth). Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map height) x (feature map width) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

- Convolutional kernels defined by a width and height (hyper-parameters).
- The number of input channels and output channels (hyper-parameter).

- The depth of the convolution filter (the input channels) must be equal to the number channels (depth) of the input feature map.

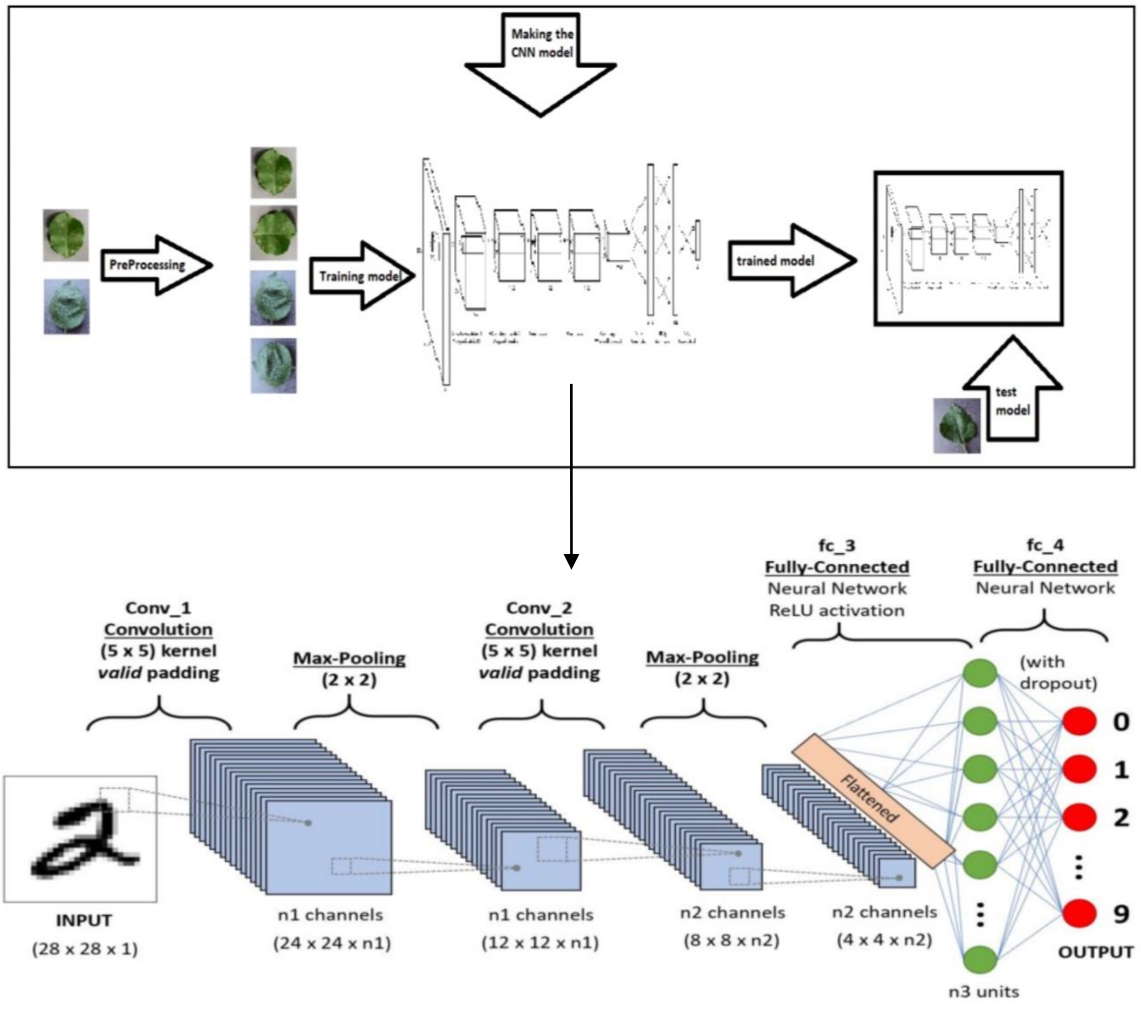


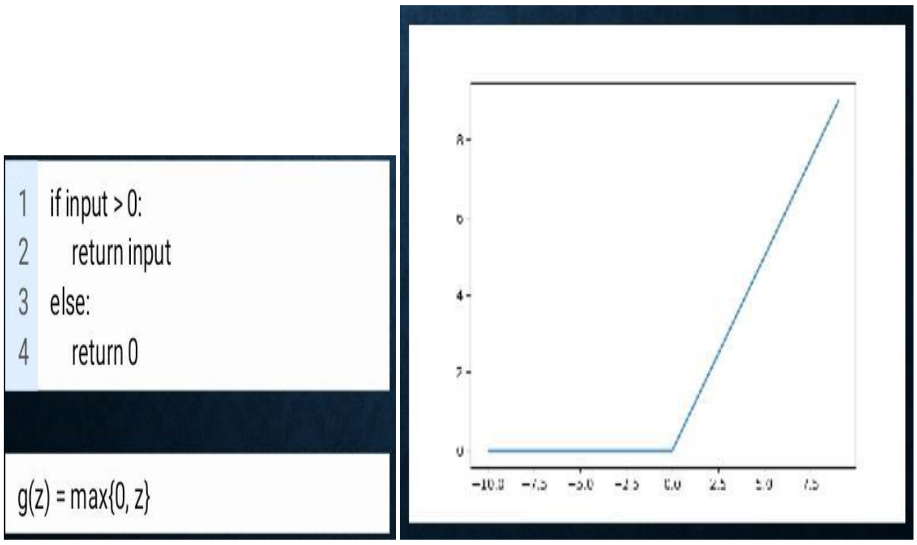
Figure 6.1 : Applied Methodology of CNN Architecture

Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feed-forward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for each neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. By using regularized weights over fewer parameters, the vanishing gradient and exploding gradient problems seen during back-propagation in traditional neural networks are avoided.

1. Activation Function

Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

ReLU : The Figure 6.2 shows the rectified linear activation function which is a simple calculation that returns the value provided as input directly, or the value 0.0 if the input is 0.0 or less.



```
if input > 0:
    return input
else:
    return 0
```

$g(z) = \max\{0, z\}$

Figure 6.2 : Rectified Linear Activation Layer (ReLU)

2. Max Pooling

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer. In addition, pooling may compute a max or an average. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer. Average pooling uses the average value from each of a cluster of neurons at the prior layer.

3. Flatten

After finishing the previous two steps, we are supposed to have a pooled feature map by now. As the name of this step implies, we are literally going to flatten our pooled feature map into a column as shown in Figure 6.3.

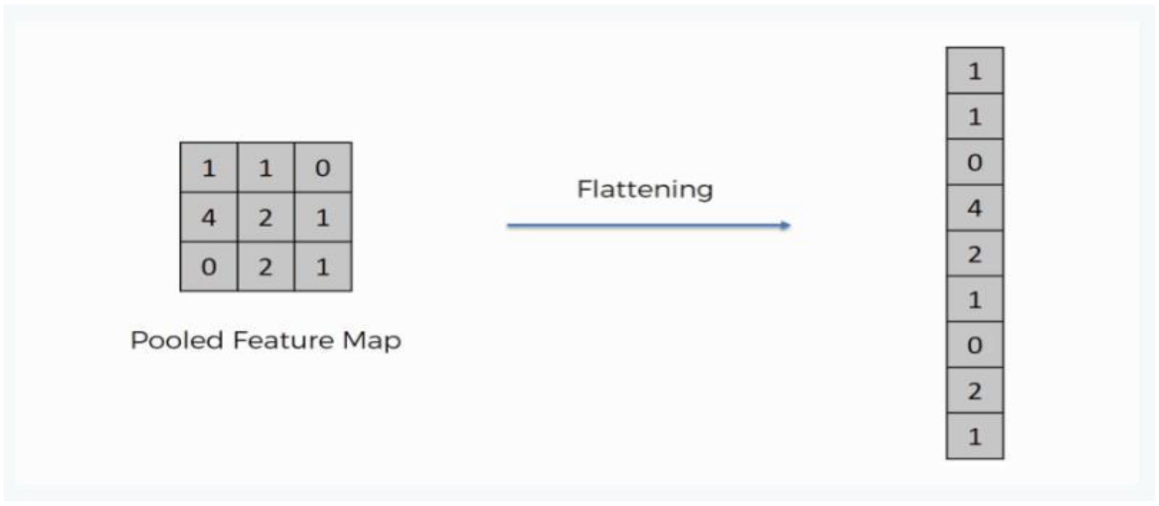


Figure 6.3 : Flatten of Pooled Feature Map

4. Full Connection

Adding a Fully-Connected layer is a cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.

5. LeNet Architecture

The LeNet architecture consists of two sets of convolutional, activation, and pooling layers, followed by a fully-connected layer, activation, another fully-connected, and finally a softmax classifier.

The original LeNet architecture used TANH activation functions rather than RELU. The reason we use RELU here is because it tends to give much better classification accuracy due to a number of nice, desirable properties.

The LeNet architecture consists of the following layers:

INPUT => CONV => RELU => POOL => CONV => RELU => POOL => FC => RELU => FC

In the first set of CONV => RELU => POOL layer sets.

- Our CONV layer will learn 20 convolution filters, where each filter is of size 5 x 5. The input dimensions of this value are the same width, height, and depth as our input images, so we will have 28 x 28 inputs with a single channel for depth (grayscale).
- We will then apply the ReLU activation function followed by 2 x 2 max-pooling in both the x and y direction with a stride of 2.

In our second set of CONV => RELU => POOL layers

- We will be learning 50 convolutional filters rather than the 20 convolutional filters as in the previous layer set. It’s common to see the number of CONV filters learned increase in deeper layers of the network.
- We will apply the same ReLU activation function and 2 x 2 maxpooling as the first set.

Next, we come to the fully-connected layers of the LeNet architecture:

- We take the output of the preceding MaxPooling2D layer and flatten it into a single vector, allowing us to apply dense layers. We know that a dense/fully-connected layer is a “standard” type of layer in a network, where every node in the preceding layer connects to every node in the next layer.
- Our fully-connected layer will contain 500 units which we pass through another nonlinear ReLU activation.
- Finally, we apply a soft max classifier that will return a list of probabilities, one for each of the class labels. The class label with the largest probability will be chosen as the final classification from the network.
- With this LeNet Model we use Adam optimizer. The Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum.

6.1 Pseudo Code for Training Module

// This module is used to train the CNN algorithm, the dataset used here has four types of leaf diseases which are Cercospora, Miner, Phoma and Rust and also to obtain trained model and accuracy graph.

Input: Image in folder with names of folder as disease names.

Output: Trained model .h5 file obtained and also accuracy of training graph obtained.

Steps 1: Image preprocessing

```
image = cv2.resize(image, (28, 28))

image = image.astype("float") / 255.0

image = img_to_array(image)

image = np.expand_dims(image, axis=0)
```

Step 2: Select the folder directory for training.

Step 3: Name the following based on disease names.

Step 4: Load CNN model with epochs and batch size.

Step 5: Obtain trained model and accuracy graph.

6.2 Pseudo Code for Testing Module

//This module is used to test whether the features present are in the authorized manner and to check whether the diseased leaf image label is in text format.

Input: Image to be tested for disease in jpg or png format.

Output: The diseased leaf image label in text format.

Step 1: Select the leaf image directory for image to be tested.

Step 2: Perform dimensionality reduction and pre process on image.

Step 3: Load the trained model .h5file and detect label of disease.

Step 4: Display disease label.

6.3 Pseudo Code for Automated Leaf Disease Classification Module

//This module is used to automatically detect the leaf disease, further classify it and also to display the image with disease printed on it.

Input: Use UI to browse the image location.

Output: Image displayed with disease printed on it.

Step 1: Select image from UI.

Step 2: Click on detect button to run CNN, load model and detect.

Step 3: View the image file to find the disease printed on image.

CHAPTER 7

TESTING

The aim of testing stage is to discover defects or errors by testing individual program components. These components may be functions, objects or modules. During system testing, these components are integrated to form the complete system. At this stage, testing should focus on establishing that the system meets its functional requirements, anddoes not behave in an unexpected way. Test cases are inputs to test the system and the outputs are predicted from these inputs if the system operates according to its specification.This is to examine the behavior in a cohesive system. The test cases are selected to ensurethat the system behavior can be examined in all possible combinations of conditions. Accordingly, expected behavior of the system under different combination is given.Therefore test cases are selected which have set of input and its expected output. Inputs that are not valid, for which suitable messages must be given and inputs that do not occur frequently which can be regarded as special cases.

7.1 Unit Testing

The purpose is to validate that each unit of the software performs as designed. A unit is thesmallest testable part of any software. It usually has one or a few inputs and usually a single output.

Table 7.1 : Unit Testing of Analyzing the Image for Satisfactory/Unsatisfactory Results

Test Case No.	**Input**	**Expected Output**	**Actual Output**	**Satisfactory/ Unsatisfactory**
TC1	Unit Testing for 256*420 Px image for classified leaf	Analyse successful. [disease predicted]	Analyse successful. [disease predicted]	Satisfactory
TC 2	Unit Testing for 256*300 Px image for healthy leaf	Analyse successful. [Healthy detected]	Analyse successful. [Healthy detected]	Satisfactory

TC3	Unit Testing for image other than leaf	No disease detected	Disease detected by noisy closest match	Unsatisfactory
TC4	Unit Testing for Invalid Format(.doc,.docx, pdf)	Invalid Format	Invalid formats not shown during selection. Program error	Unsatisfactory

The Table 7.1 is unit testing of analyzing the image for satisfactory condition, if the input given is a valid image, the expected output is analyze successful and the actual output is also analyze successful. Hence this condition is satisfactory.

7.2 Integration Testing

The purpose of this level of testing is to expose faults in the interaction between integrated units. In the Table 7.2, the input is click on the “Know more” link, the expected output is type of leaf disease with its description and actual output is also the disease type and its description. Hence this condition is satisfactory.

Table 7.2 : Integration Testing of Analyzing the Image for Satisfactory/Unsatisfactory Results

Test Case No.	**Input**	**Expected Output**	**Actual Output**	**Satisfactory/ Unsatisfactory**
TC1	Integration Testing for click on the “detect button” of diseased image	Disease detection and label of disease	Disease detection and label of disease	Satisfactory
TC2	Integration Testing for healthy leaf	Healthy label to be displayed	Healthy label to be displayed	Satisfactory
TC3	Integration Testing for Other plant leaf	Invalid input	Noisy match	Unsatisfactory

7.3 System Testing

The purpose of this test is to evaluate the system's compliance with the specified requirements. In the Table 7.3, GUI testing (part of System testing) is shown. When the application is running, the GUI provides easy to use interface and the alignment of text and buttons is in a proper way. Hence the condition is satisfactory.

If the input is a valid image and run the application using a 4GB ram system, the response time is more when it is compared to response time taken when the application is run in 8GB ram system. Hence the test case fails and gives unsatisfactory results.

Table 7.3 : System Testing of Analyzing the Image for Satisfactory/Unsatisfactory Results

Test Case No.	**Input**	**Expected Output**	**Actual Output**	**Satisfactory/ Unsatisfactory**
TC1	System Testing for GUI (dialogue) box opened when application is running.	Alignment of buttons and text is proper. Usability of application is easier.	Alignment of buttons and text is proper. Usability of application is easier.	Satisfactory
TC2	Running the application with 4GB ram system by giving input as a diseased image of size 256*420 px.	Less Response time taken to give the results.	More response time taken to obtain results.	Unsatisfactory

CHAPTER 8

RESULTS AND DISCUSSION

This project shows the importance of plant disease detection in these days. This model was developed using Deep Learning in python. 20% (14,059) images from PlantVillage dataset were used to test the accuracy of this model. These images are from 38 different classes, 20% of each class randomly selected for testing. Some real time images were also used. Those images were captured from local environment. They do not belong to any classes which are present in dataset. But model give us more than 95% accuracy on those images as well by telling either leaf is healthy of unhealthy. Total 100 images were used and 96 were classified correctly. Some images were captures at night with the help of flash light and some images have dirt upon it so that they were misclassified. Some of the images were captured from local environment. Testing dataset gives accuracy more than 98%. It means 1379 images from 14,059 images were classified correctly by model. Figure 8.1 shows the training and validation accuracy graph generated by our model on testing dataset.

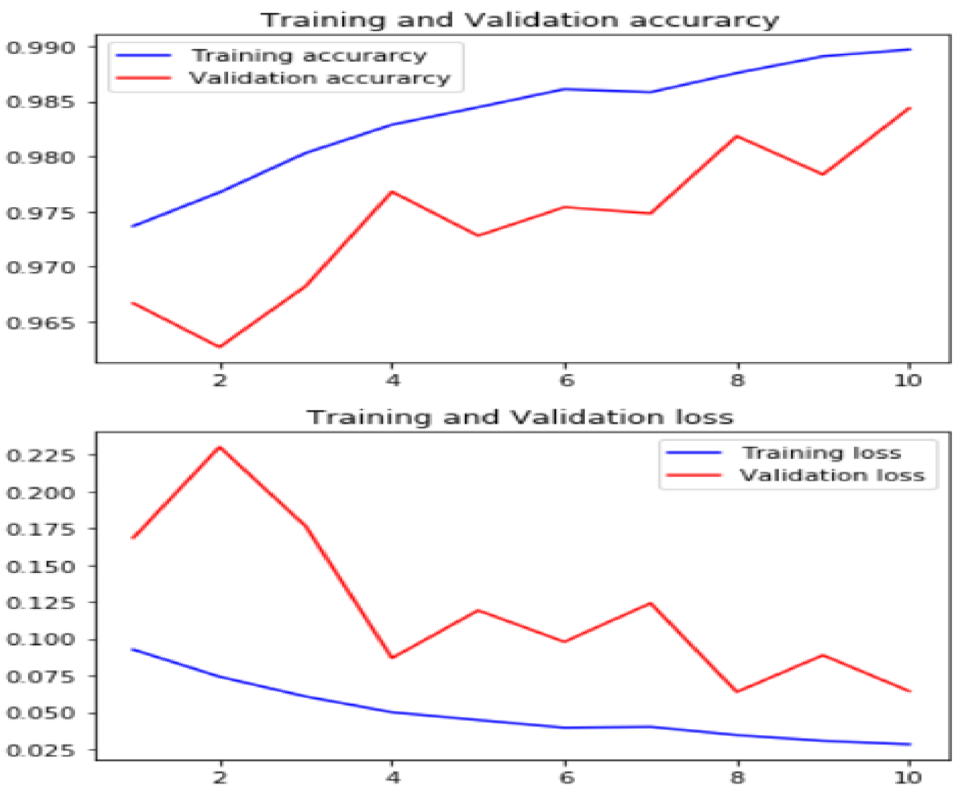


Figure 8.1 : Training and Validation Accuracy on Testing Dataset

The Table 8.1 shows the final model performance.

Table 8.1 : Final Model Performance

Model	**Dataset for Training**	**Dataset for Testing**	**Training Accuracy**	**Testing Accuracy**
CNN	PlantVillage (80%)	PlantVillage (20%)	99%	98%+
CNN	PlantVillage (80%)	Actual Environment (100 Images)	99%	95%+

The proposed LeNet architecture successfully classifies and detects the diseases of coffee plants with an accuracy of 94.24%. Furthermore, accuracy of detecting and classifying four diseases is summarized in Table 8.2.

Table 8.2 : Accuracy Graph

Diseases	**Accuracy**
Cercospara	93.06
Rust	99.92
Phoma	99.60
Miner	94.80

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

9.1 Conclusion

It is important to detect whether a leaf is healthy or diseased. Once detected, the disease needs to be identified. Four different coffee disease classes are detected in this study, i.e. Cercospara, Phoma, Miner and Rust. Here, we have proposed a leaf disease detection approach that is based on Convolutional Neural Networks. The deeplearning-based approach can automatically extract the discriminative features of the diseased leaf images and detect the diseases with high accuracy. Finally, a CNN model, LeNet is developed for classification stage using the extracted features.

9.2 Future Enhancement

In future, we want to target multiple plants with multiple diseases. This will make them capable of adopting the decision to improve yield by taking necessary precautions, preventions and correct measures to improve the health of citrus orchards. The accuracy of our results can be improved using more training example. Our proposed methodology can be utilized in detection of diseases in other plants with the collaboration of respective domain experts. We intend to develop a mobile application that is based on proposed technique for direct benefits of the farmer. There is still a need of standard publically available datasets to improve the overall performance of such systems and making the computer aided diagnosis systems more wide-ranging, which are capable of detecting and classifying different diseases more accurately.

BIBLIOGRAPHY

[1] Peng Jiang, Yuehan Chen, Bin Liu, Dongjian HE, and Chunquan Liang, “Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks”, IEEE ACCESS, vol. 7, pp. 59069-59080, May 2019.

[2] Tan Nhat Pham , Ly Van Tran, And Son Vu Truong Dao, “Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection”, IEEE ACCESS, vol. 8, pp. 189960-189973, October 2020.

[3] Zhongqi Lin, Shaomin Mu, Feng Huang, Khattak Abdul Matten, Minjuan Wang, Wanlin Gao, AND Jingdun Jia “A Unified Matrix-Based Convolutional Neural Network for Fine-Grained Image Classification of Wheat Leaf Diseases”, IEEE ACCESS, vol. 7, pp. 11570-11590, October 2020.

[4] Yue Li, Jingdun Jia, Li Zhang, Abdul Matten Khattak, Shi Sun, Wanlin Gao, And Minjun Wang “Soybean Seed Counting Based on Pod Image Using Two-Column Convolution Neural Network” , IEEE ACCESS, vol. 7, pp. 64177-64185, May 2019.

[5] Malvika Ranjan, Manasi Rajiv Weginwar, NehaJoshi, Prof. A B Ingole, “Detection and Classification of Leaf Disease Using Artificial Neural Network”, International Journal of Technical Research and Applications, vol. 6, pp. 331-333, June 2015.

[6] Y Sanjana, Ashwath Sivasamy , Sri Jayanth , “Plant Disease Detection Using Image Processing Techniques”, International Journal of Innovative Research in Science, Engineering and Technology, vol. 4, pp. 19-20, May 2015.

[7] Ferentinos K P,“Deep Learning Models for Plant Disease Detection and Diagnosis”, Computers and Electronics in Agriculture, vol. 4, pp. 311-318, May 2018.

[8] Prof. Sanjay B Dhaygude, Nitin P Kumbhar, “Agricultural Plant Leaf Disease Detection Using Image Processing”, International Journal of Advanced Research in

Electrical, Electronics and Instrumentation Engineering, vol. 2, pp. 26-27, January 2013.

[9] S Dubey, M Dixit, “Facial Expression Recognition using Deep Convolutional Neural Network”, International Symposium on Advanced Intelligent Informatics (SAIN) , vol. 8, pp. 96-101, August 2018.

[10] Sandesh Raut, Amit Fulsunge J, “Plant Disease Detection in Image Processing Using Matlab”, International Journal of Innovative Research in Science, Engineering and Technology, vol. 6, pp. 10373-10380, June 2017.

[11] Smita Naikwadi, NiketAmoda, “Advances in Image Processing for Detection of Plant Diseases”, International Journal of Application or Innovation in Engineering and Management (IJAIEM), vol. 3, pp. 168-174, November 2015.

[12] Arpita Patel, Barkha Joshi, “A Survey on the Plant Leaf Disease Detection Techniques”, International Journal of Advanced Research in Computer and Communication Engineering, vol. 6, pp. 229-231, January 2017.

[13] Arti N Rathod, Bhavesh A Tanawala, Vatsal H Shah, “Leaf Disease Detection Using Image Processing and Neural Network,” International Journal of Advance Engineering and Research Development (IJAERD), vol. 1, pp. 2348-4470, June 2014.

[14] Belal A ,M Ashqar, Samy S Abu-Naser, “Image-Based Tomato Leaves Diseases Detection Using Deep Learning”, International Journal of Academic Engineering Research (IJAER) , vol. 2, pp. 10-16, December 2018.

[15] Sushma S Patil, Suhas K C, “Identification and Classification of Cotton Leaf Spot Diseases using SVM Classifier”, International Journal of Engineering Research and Technology (IJERT), vol. 3, pp. 1511-1513, April 2014.

APPENDIX A

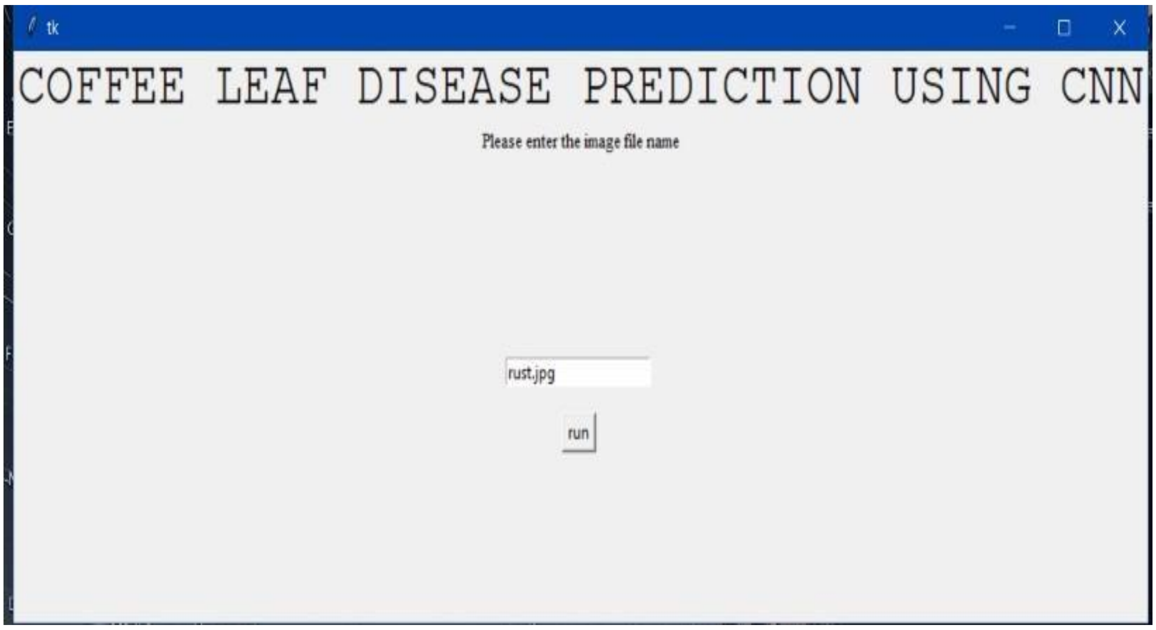
ACRONYMS

ALDD	Apple Leaf Disease Dataset
ANN	Artificial Neural Network
CNN	Convolution Neural Network
DFD	Data Flow Diagram
FC LAYER	Fully-Connected Layer
FPS	Frames Per Second
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HSV	Hue, Lightness and Saturation
INAR	Inception Module and Rainbow Concatenation
LAN	Local-Area Network
MAE	Mean Absolute Error
M-bCNN	Matrix Based Convolution Neural Network
MSE	Mean Square Error
RELU	Rectified Linear Unit (Activation Function)
RFE	Recursive Feature Elimination
RGB	Red, Green and Blue
RMSprop	Root Mean Square Propagation

ROI	Region of Interest
SGD	Stochastic Gradient Descent
SSD	Single Shot Detector
SVM	Support Vector Machine
TANH	Hyperbolic Tangent Activation Function
TC	Test Case
TCNN	Two-Column Neural Network
UI	User Interface
VGGNet	Visual Geometry Group Network

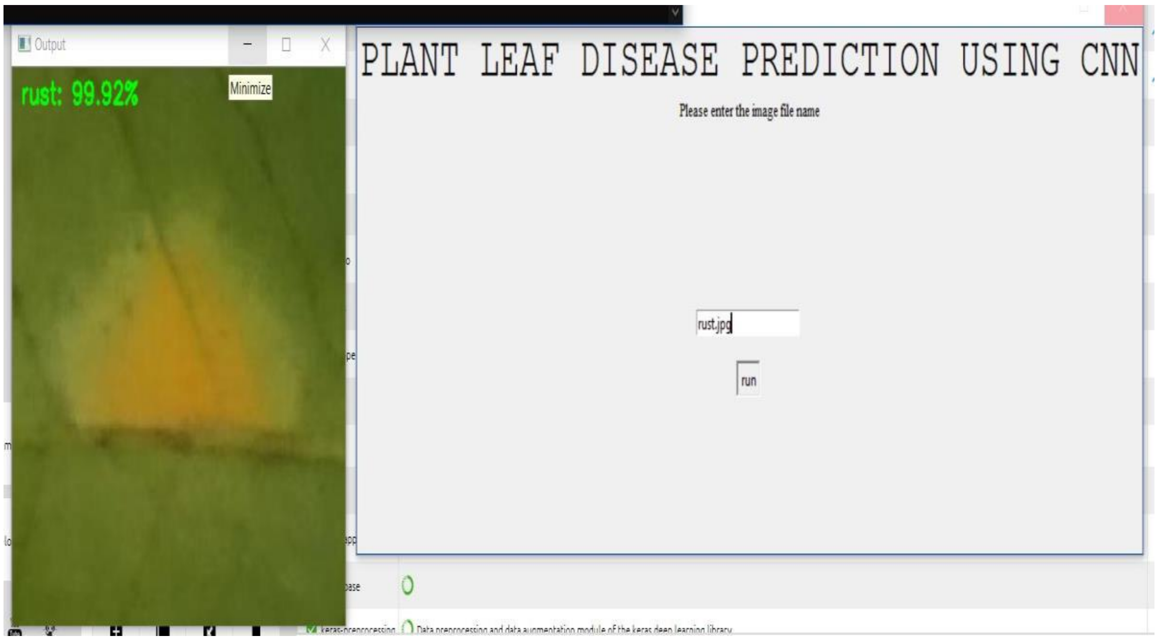
APPENDIX B

SNAPSHOTS


tk
COFFEE LEAF DISEASE PREDICTION USING CNN
Please enter the image file name
rust.jpg
run

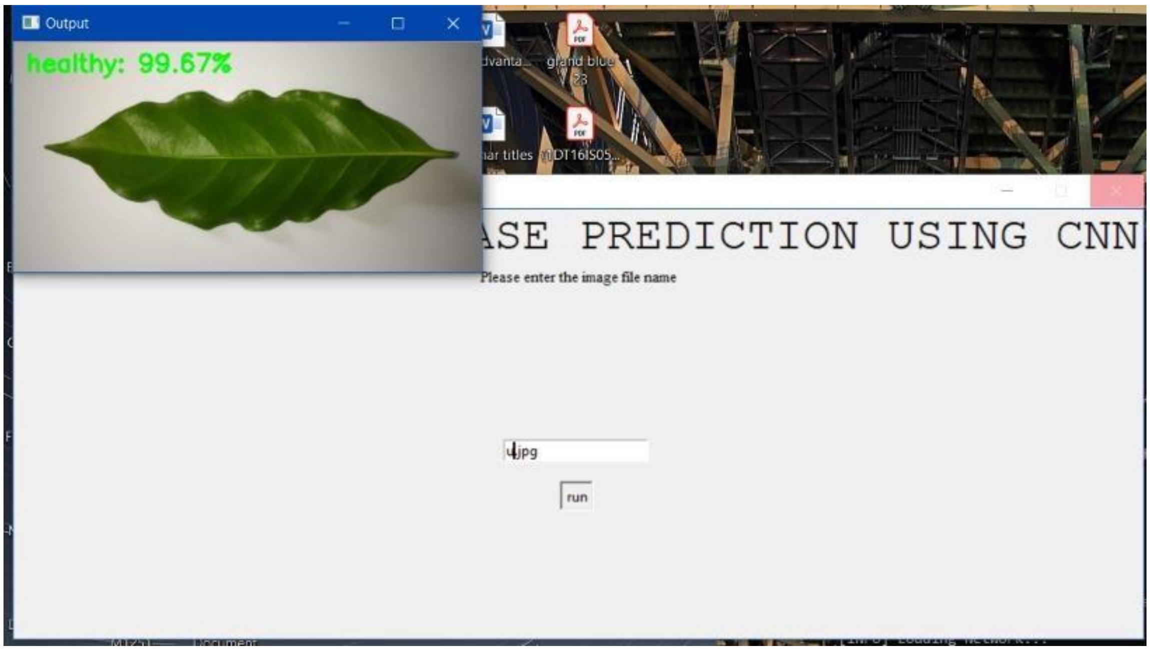
B.1 : Home Page

B.1 : This is the Home page where we can enter the filename of the image.


Output
rust: 99.92%
Minimize
PLANT LEAF DISEASE PREDICTION USING CNN
Please enter the image file name
rust.jpg
run

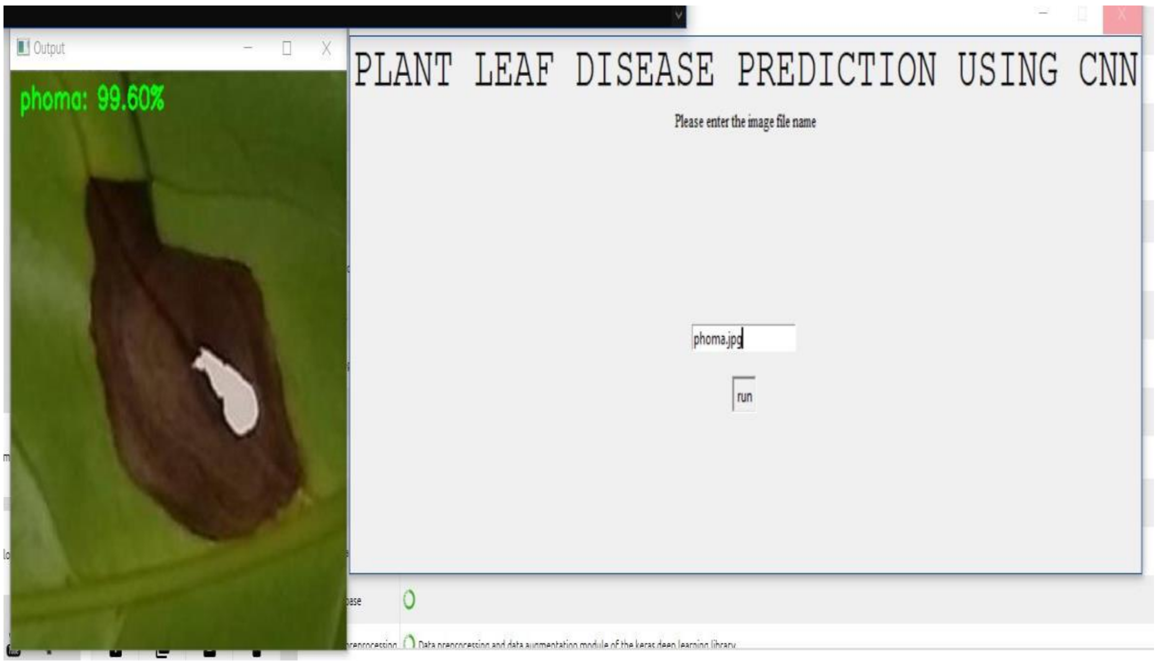
B.2 : Result of Rust Image

B.2 : This page shows the result of rust leaf disease prediction when we input the filename and click on the run button.



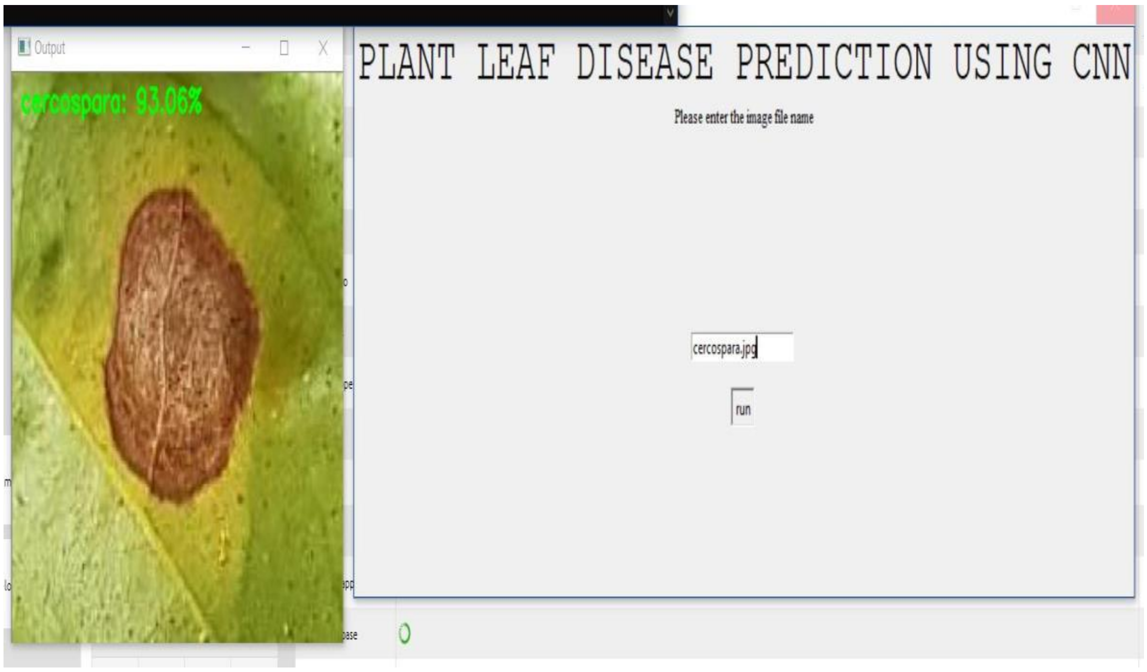
B.3 : Result of Healthy Image

B.3 : This page shows the result of healthy leaf disease prediction when we input the filename and click on the run button.


Output
phoma: 99.60%
PLANT LEAF DISEASE PREDICTION USING CNN
Please enter the image file name
phoma.jpg
run

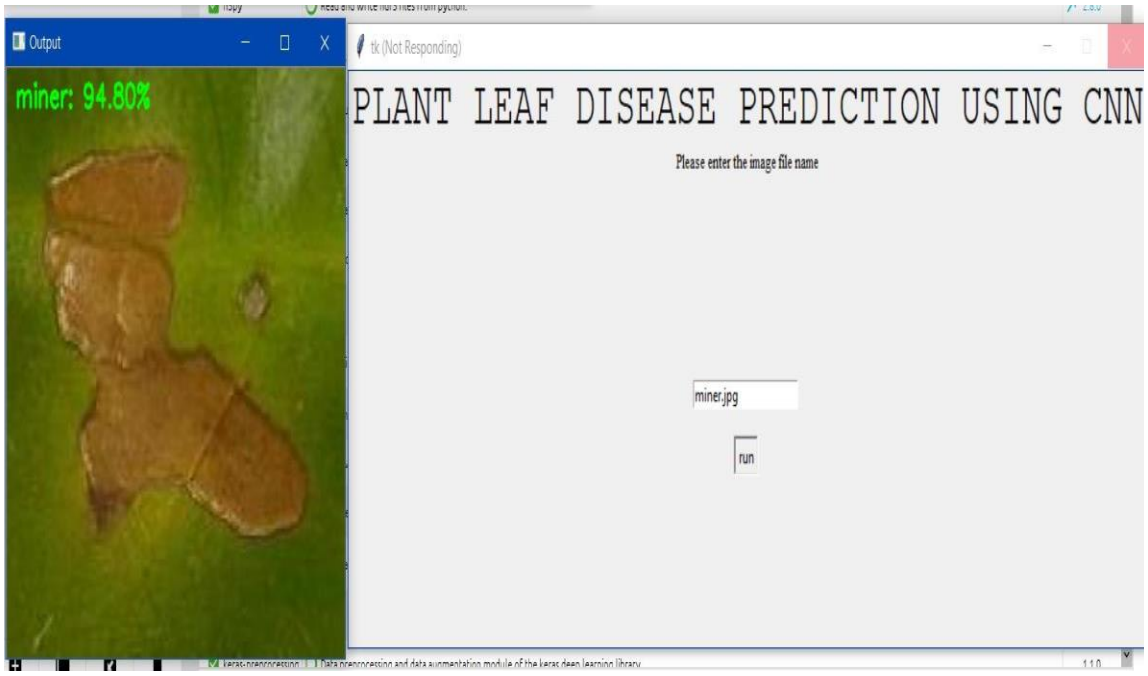
B.4 : Result of Phoma Image

B.4 : This page shows the result of phoma leaf disease prediction when we input the filename and click on the run button.


Output
cercospora: 93.06%
PLANT LEAF DISEASE PREDICTION USING CNN
Please enter the image file name
cercospara.jpg
run

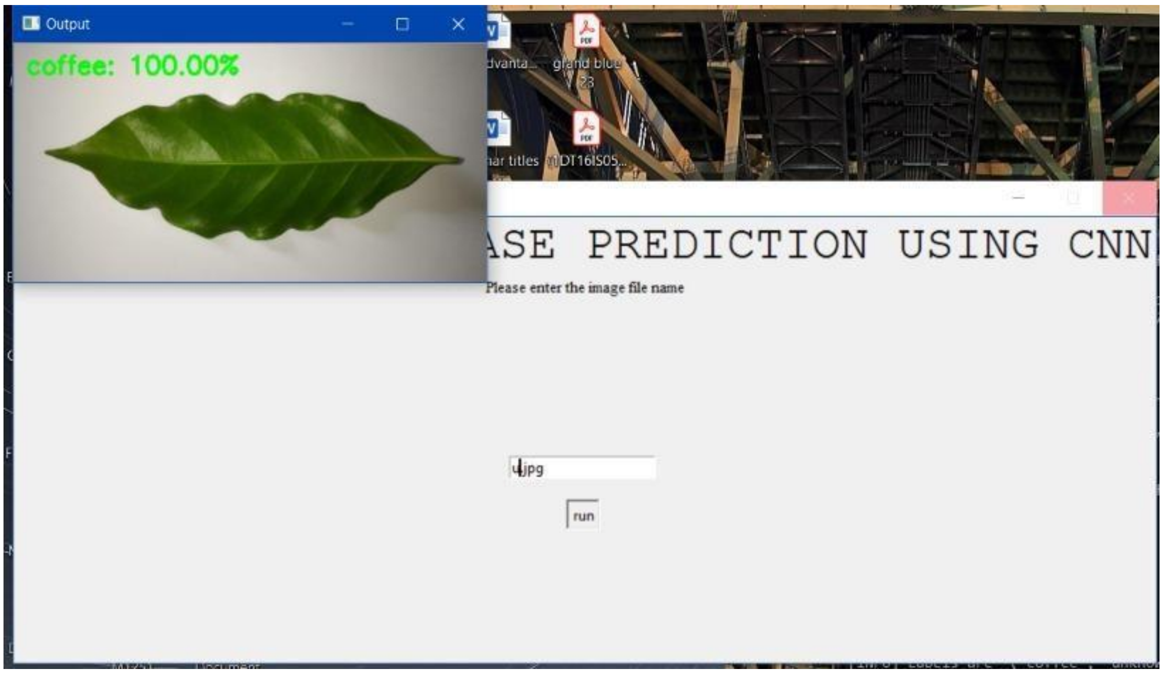
B.5 : Result of Cercospora Image

B.5 : This page shows the result of cercospora leaf disease prediction when we input the filename and click on the run button.


Output
miner: 94.80%
tk (Not Responding)
PLANT LEAF DISEASE PREDICTION USING CNN
Please enter the image file name
miner.jpg
run

B.6 : Result of Miner Image

B.6 : This page shows the result of miner leaf disease prediction when we input the filename and click on the run button.



B.7 : Result of Healthy Image Showing 100% Accuracy

B.7 : This page shows the result of healthy leaf disease prediction showing 100% accuracy when we input the filename and click on the run button.