ML BASED PLANT LEAF DISEASE PREDICTION

A PROJECT REPORT

Submitted by

ARUN V DAS

REG NO: TKM20MCA-2012

In partial fulfilment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

JULY 2022

DECLARATION

I undersigned hereby declare that the project report on "***ML BASED PLANT LEAF DISEASE PREDICTION***", submitted for partial fulfillment of the requirements for the award of the degree of M.C.A in APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. **ALSHAINA S**. This submission represents my ideas in my own words and where ideas or words of others have been included; I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma, or similar title of any other University.

Place: Kollam **ARUN V DAS**

Date: 18-JULY-2022

THANGAL KUNJU MUSALIAR COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER APPLICATION

2020-2022

CERTIFICATE

This is to certify that, this report titled "***ML BASED PLANT LEAF DISEASE PREDICTION***" is a bonafide record of the **Project Work** presented by **ARUN V DAS (TKM20MCA-2012)**, under our guidance and supervision, in partial fulfilment of the requirements for the award of the degree, **MASTER OF COMPUTER APPLICATIONS in APJ Abdul Kalam Technological University.**

Internal Supervisor Head of the Department External Examiner

ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. Firstly I would like to thank the almighty for giving me the wisdom and grace for making my project a memorable one. I thank him for steering me to the shore of fulfillment under his protective wings.

I express my sincere gratitude to **Dr. T A SHAHUL HAMEED**, Principal of T.K.M College of Engineering for giving me an opportunity to present my project. I would like to thank **Dr. FOUSIA M SHAMSUDEEN**, Assistant Professor and Head of the Department, MCA, TKMCE, for her constant support and encouragement throughout the project work.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my guide **Prof. ALSHAINA S**, Assistant Professor, MCA, TKMCE, for her expert guidance, cooperation, and immense encouragement. I also extend my thanks to the entire faculty and staff of the Department of Computer Applications, TKMCE, who has encouraged me throughout this work.

I also express my thanks to my loving parents, brother, and friends, for their support and encouragement in the successful completion of this project work.

ARUN V DAS

ABSTRACT

Agriculture is one among the major sectors of the Indian economy. As we know farmers detect plant disease through their naked eye. But it requires more efforts to discover in large number of plants and also it is time consuming process. In such issues to enhance the accuracy rate and make it more favourable suggested techniques are implemented where plant disease detection help to make process cheaper and easier. In this project, various machine learning methods like Random Forest, Support Vector Machine (SVM), XGBoost, Long Short-Term Memory (LSTM) etc., have been utilize for recognition, discovery, and categorization of plant diseases. This method will improves productivity of crops .The Proposed method compares the accuracy of above mentioned machine Learning methods for plant disease prediction and also find out the type of disease. This work passed through the steps such as Image acquisition, image pre-processing, features extraction and AI based classification etc.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In India, it is extremely important to bring about technological advancement in areas related to crop yield. Agriculture shows a critical role in all country's economy. The improvement in the agricultural sector is generally aimed at meeting the increasing needs of the people. The agricultural sector needs to be modernized to survive in the current situation. Mostly Crops are affected by both fungal and bacterial infections. This causes great loss to the efficiency of farmers. Detecting disease in the bare eye will always be a difficult process. To do this, regular farm observing is essential. It's a dull procedure. It is similarly very expensive when the farm dimension is very large. Because of this difficulty, even agricultural experts cannot easily diagnose the disease and to find the solution of problem. Automated system would be great help to farmers for detect plant diseases.

This system serves as a tool to notify farmers in time and take necessary precautions. Diseases of various plants can damage parts of the plant such as leaves, fruits and seeds. These infections are endemic to certain parts of the plant. The plant leaves are the most vital part of the plant. When leaves are infected with pathogens, it directly disrupts the plant's life cycle. Diseases that commonly affect plant leaves include fungal and bacterial diseases. Therefore, early detection of plant diseases is very important. Leaf disease in machine learning appears to be a superior choice to solve this task. Various machine learning (ML) algorithms have been estimated to automatically predict plant leaf diseases from digital crop images. Plants are affected by certain diseases in multiple parts of the plant such as leaves, stems, seeds and fruits. The disease occurs only in certain parts of the plant. The leaves can be considered the main part of the plant, but photosynthesis takes place with the help of the leaves. If the leaves are susceptible to disease, it directly affects the plant's life cycle.

To face these type diseases, it is essential to have an automated system of diseases classification. Over the last few years, Server based and mobile based techniques are used for disease prediction. Some benefits of these two techniques like efficient processing, high resolution camera and substantial built in accessories leads in automatic disease identification.

Now a day's deep learning algorithm and Machine Learning (ML) algorithm has been utilised in order to increase the accuracy and the detection rate of the outcomes. Recently, numerous machine learning algorithms that may be used for recognising and classifying plant diseases based on images of plants have been developed. These techniques can be used to analyse images of plants. While these automated tools have solved the problem, the bigger challenge is the consistency and robustness of the test results obtained. The proposed work utilizes the Machine Learning algorithms such as Random Forest, Support Vector Machine, XGBoost, LSTM etc., for recognition, discovery, and categorization of plant diseases.

1.1 Problem Statement

To develop an automatic prediction system that uses a leaf from an apple plant as the input and then uses an image pre-processing technique to improve the image. Later, a variety of image features are extracted for the enhanced image. On the basis of these features, the image of the leaf can be classified as either healthy or diseased. Finally, the project is also aimed to provide remedy for the disease that has been identified.

1.2 Objective

The aim of the project is to design, implement and evaluate an image-processing based software solution for automatic prediction and classification of plant disease

- The key objective is to detect healthy leaves and diseased leaves.
- Also provide a treatment for the disease that has been identified.

CHAPTER 2

LITERATURE REVIEW

The literature review is a complete analysis and interpretation of the relevant literature on a certain issue. When conducting a literature review, research questions are first identified, after which one attempts to provide a solution by looking up and assessing pertinent material. The ability to re-examine the study's findings might lead to the development of fresh insights, which is one benefit of literature reviews. A literature review summarises and explains the whole and most recent body of information about a subject that may be found in academic books and journal articles. There are two different kinds of literary critique that you might write in college: one that students are required to write as a stand-alone component of a course, and one that is written as part of an introduction or as a preparation for a longer project, typically a thesis or research report. The type of review you write will affect the review's perspective, topic, and the kind of hypothesis or argument you present. Reading published literature reviews or the introductory sections of theses and dissertations in your own subject is one approach to learn how the two differ from one another. Examines the manner they address the issues and the organisation of their arguments.

2.1 Purpose of the Literature Review

- It facilitates readers' access to research on a specific issue by choosing articles or research that is pertinent, significant, meaningful, and valid and summarising it in a thorough report.
- It offers a great place to start for researchers who are commencing their research in a new field. By requiring them to assess, assess, and compare relevant original research,
- It guarantees that researchers won't rework already completed work.
- It can suggest areas of interest or offer hints about potential future study directions.
- It highlights results.

- It points out holes and contradictions in the literature.
- It offers a constructive critique of the methods and strategies used by other researchers.

2.2 Related work

A technique for picture segmentation has been presented by Vijai Singh et al. [1], which can be utilised for the automatic detection and classification of plant leaf diseases. In order to determine which section of the leaf is affected, a digital image of several distinct plant leaves is examined. These images are then processed by applying different types of image processing methods to obtain different and useful features that are required for further analysis. After pre-processing the input leaf image, green colour pixels are covered by computing a threshold value. Eliminate the masked cells and use genetic algorithms to obtain useful segments for classifying leaf diseases. The texture as well as the colour of the image are both taken into consideration by the colour co-occurrence technique, which is a method for the feature extraction process. The values of the extracted features are kept in the dataset, and then the classification process is carried out. The classification process makes use of the support vector machine approach.

Image processing, a support vector machine (SVM) classifier, and k-means clustering are utilised in the identification of plant leaf diseases by Sujatha and et al. [2]. This method is suggested for application in the detection of plant leaf diseases. Image acquisition, image pre-processing, segmentation, feature extraction, and classification are the five phases involved in identifying leaf diseases. Image acquisition comes first. Gather some leaves off the tree, take a picture of the plant's leaves, and then enter the image of the leaves into the system. Then, digital image is divided into multiple sections. Then, contrast enhancement is done on original image and converts it to HIS (Hue, Saturation, and Intensity) image. The objects are then clustered/split into k groups based on the leaf function using the k-means clustering

algorithm. This is done through the use of the Euclidean distance measure. Finally, classification is done by using SVM classifier. By using this concept to identify diseases for all leaf types, users can also correctly understand the percentage of leaf area affected by disease recognition.

Vinoth Kumar and et al [3], coined a novel methodology based on a distrot model to control the segmentation for plant disease dectecting. Transform the RGB color space so that the color information of the image can be effectively distinguished and divided using the k-means algorithm. In this work, the proposed system can be classified into - Image Processing Segment and Pattern Recognition Segment. Transform an image into an exceptionally uniform color space. Create a color space representation that models the color information for a block of pixels. The patch representations are clustered into k clusters using the k-means clustering algorithm. Measure the detection contrast using k representative colors of the leaf image and the feature image.

Random Forest is utilised by Shima Ramesh and et al [4] in order to differentiate between healthy and unhealthy leaf samples taken from the datasets developed. The papaya leaf picture dataset was used for the analysis. The original leaf image in RGB is turned into a grayscale image. This is done for the main reason that the Hu moments shape descriptor as well as the Haralick features may be computed on a single channel. Therefore, RGB to grayscale conversion is required before the Hu moments and Haralick feature are calculated.. The proposed paper includes the different stages of implementation, namely dataset generation, feature extraction, classifier training, and classification. A random forest was trained with the datasets that were generated for diseased and healthy images. Image feature extraction using a HOG (Histogram of an Oriented Gradient). Three feature descriptions are used here: Haralick texture, Color Histogram and Hu moments. The random forest model made better predictions on the papaya dataset.

Shruthi and et al [5], present the phases of common systems for detecting plant diseases and a proper review of machine learning methods for this purpose image acquisition, a dataset, image processing, feature extraction, and classification are the individual components that make up the plant disease detection system. The image needs to be involved in the pre-processing step in order to improve some essential aspects of the image before moving on to the next stage of processing. The method of segmentation is used to the plant photos in order to split them up into their respective segments. This can be utilised to differentiate the diseased spot in the plant leaf from the background. Using the Gray level Co-occurrence Matrix (GLCM), extract the features of colour, form, and texture. SVM Classifier, ANN Classifier, KNN Classifier, FUZZY Classifier, and Deep Learning were all subjected to an in-depth analysis as part of a comparative research project with the purpose of recognising plant diseases. In comparison to other classifiers, the support vector machine (SVM) classifier is frequently used by authors to categorise diseases. According to the findings, the CNN.

Pushkara Sharma and et al [6] describe an automatic AI-based detection and categorization of plant leaf diseases. This allows for the rapid and simple identification of the disease, as well as the subsequent treatment of it. Image collection, image preprocessing, segmentation, and classification make up the primary components of this paradigm, which can be broken down into their respective phases. In order to clean up the image that has been gathered, filters are applied. The image is converted from RGB to HSV, which is a different colour space. K-means clustering is used to do the segmentation, and there are two cluster cores used. During the training process, various machine learning and deep learning algorithms, such as logistic regression, KNN, SVM, and CNN, are compared based on their levels of accuracy. According to the findings of the experiment, the algorithm that achieves the highest results both in training and testing is called CNN.

Using techniques from machine learning, Ahmed and et al [7] demonstrate a system that can detect diseases in rice leaf samples. This research was able to identify three of the most frequent diseases that affect rice, including brown spot, bacterial leaf spot, and leaf smut. As input data used crisp pictures of rice leaves that had been damaged, and they had a white background. Following any necessary preprocessing, the input is then sent into the module responsible for feature selection. The strategy of selecting features based on correlation is the one that is applied. The dataset is trained with a variety of machine learning techniques including Naive Bayes, Logistic Regression, KNN (K-Nearest Neighbor), and J48 (Decision Tree). After going through 10 rounds of cross-checking, the decision tree algorithm was able to attain an accuracy of more than 97% when it was applied to the test data set.

Meghna and et al [8], present an automated method that uses Machine Learning (ML) and image processing technique to identify and classify diseases that can affect tomato plants. It is important to consider the diseases that might affect tomatoes, including Early Blight, Late Blight, Septoria Leaf spot, Spider mite, Mosaic Virus, Yellow leaf curl virus, and Target spot. When applied to images, the feature extraction approach helps to extract visual features that are then used for training the algorithm. In order to determine which machine learning algorithm is most suited for illness detection, it is necessary to conduct an analysis of the capabilities of a variety of machine learning algorithms using training data. Shape, colour, and texture are the three global feature descriptors that are used in this project respectively. The Haralick Texture, Colour Histogram, and Hu Moments algorithms are utilised in order to extract these properties. For the purpose of categorization, the Random Forest (RF) machine learning algorithm is applied here. In order to validate the efficiency of the system in the identification of plant diseases, the test image is utilised. The accuracy of the method can be summarised as 95 percent overall.

Poojan Panchal and et al [9], clarifies a variety of methods that are utilised for image segmentation and the classification of plant diseases. The dataset includes a variety of leaves affected by a variety of diseases, such as bacterial spot on pepper bell and tomato leaves, early and late blight on potato and tomato leaves, and bacterial spot on pepper bell and tomato leaves. In addition to that, it has pictures of the leaves in their natural state. Image segmentation can be accomplished through the use of the proposed method by either employing K-Means clustering or HSV value Alteration. During the feature extraction process, the segmented images are examined. Classification for recognising diseased parts of the leaf and feature extraction based on GLCM were carried out. The GLCM is used to extract five features, and those features are then saved in the dataset. There are a variety of classifiers that are utilised, including random forest, k-nearest neighbours, decision tree, and SVM. Using Random Forest's classifier, the efficiency of the suggested methodology is able to successfully detect and categorise plant illnesses with a success rate of 98 percent.

CHAPTER 3

METHODOLOGY

The project mainly aims to predict whether leaves are disease or healthy. In this system, a machine learning approach for classifying plant disease is proposed. The analysis of the methodology of the proposed work has been presented using images. The steps used in this project included data collection, image pre-processing, features extraction, and classification.

3.1 Proposed System

The suggested system is based on methods of machine learning to predict whether leaves are disease or not. Image dataset are collected from the kaggle plant village dataset. Images from four classes : Blackroot, Cadre, Scab, and Healthy are used. For the purpose of processing leaf images and extracting information from such images, image processing methods are implemented. Both the Haralick feature and the Scale Invariant Feature Transformation (SIFT) feature are applied during the course of this project. Here, the Support Vector Machine, Random Forest, LSTM, and XgBoost machine learning techniques are utilised. After evaluating the performance of these four classifiers, choose the best one to generate the final model.

Different phases of project implementation include:

- Data collection
- Data pre-processing
- Feature extraction
- Dataset preparation
- Classification and prediction
- Evaluation

3.2 System Architecture

The system architecture design is used to abstract the general structure of the software system as well as the connections, limitations, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system.

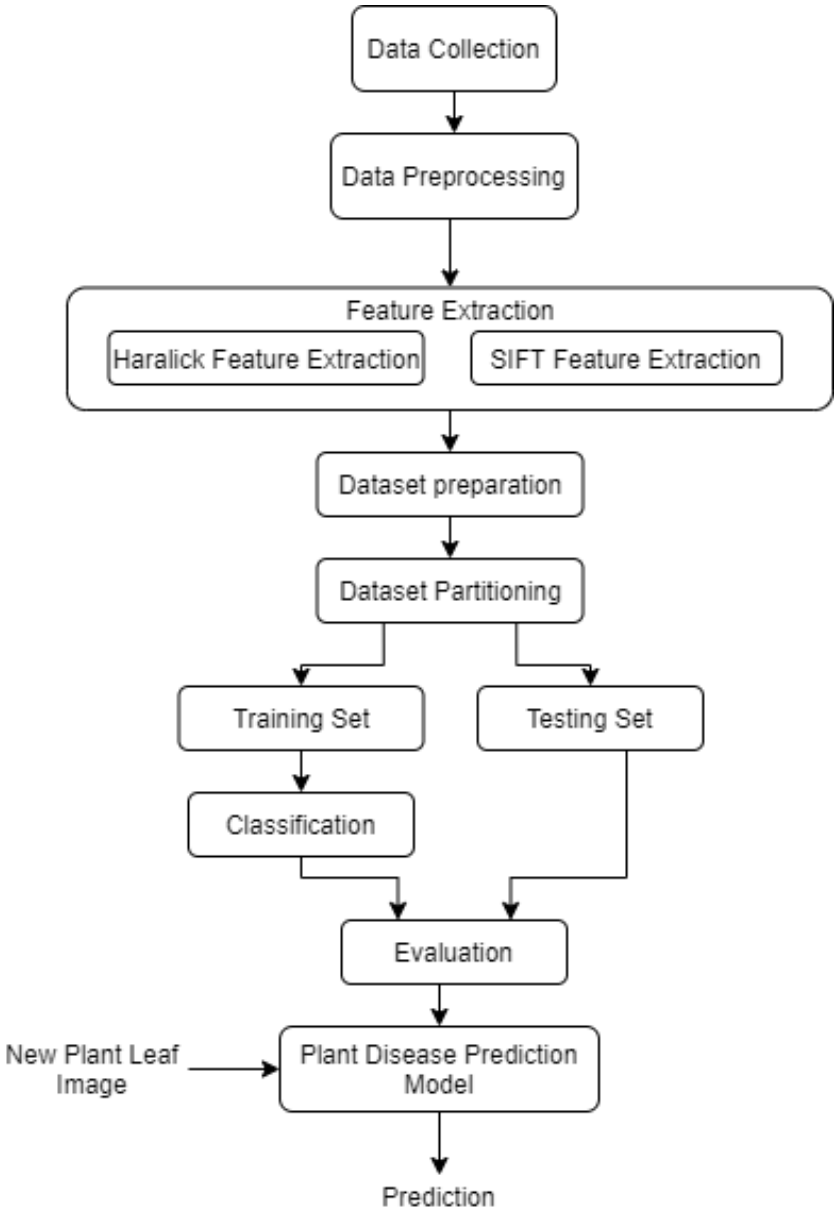

Figure 3.1: System Architecture

3.2.1 Data Collection

The process of collecting and measuring information from a variety of sources is known as data collection. The Plant Village dataset from www.kaggle.com was used to implement the project. The dataset contains a total of 502 images with the file extension .jpg. That Dataset contain images of healthy and diseased leaves. Photo collection includes Scab, Black_rot, Cadar disease of apple's leaves.

3.2.2 Data Pre-processing

Data pre-processing based on the process of arranging raw data to fit the ML model. The primary goal of pre-processing is to enhance the data in preparation for the upcoming processing steps. It contains data cleaning, data reduction, data transformation, missing value processing, etc... Image pre-processing techniques are used here because the dataset comprises images of plants leaves. Remove unfavourable noise from the images by using the Gaussian filter. And also removes the green part from the background of the image.

Dataset Preparation

Data set preparation involves organizing the data set for classification. The dataset was prepared based on the features took through Haralick texture feature extraction and SIFT feature extraction techniques on the input image. The dataset holds fifteen columns, i.e. thirteen columns as Haralick texture feature, one for class values and one for SIFT. The dataset comprises numerical value for the haralick texture feature and SIFT feature.

3.2.3 Feature Extraction

Haralick Texture Feature Extraction

Haralick Texture features are derived to measure images based on texture. Extract global features from the input image. Adjacent pixel values that occur in an image are recorded over the full image. Haralick texture features are derived from a Gray Level Co-occurrence Matrix (GLCM), which is a matrix that counts the co-occurrence of neighbouring grey levels in an image. The GLCM is a square matrix with dimension Ng, where Ng is the number of grey levels in the provided image. Element [i,j] of the GLCM is formed by counting the number of times a pixel with value I is neighbouring a pixel with value j and then dividing the whole matrix by the total number of comparisons performed. Each item represents the probability that a pixel with value I will be discovered adjacent to a pixel with value j. For every image, different texture feature value is obtained. And for each image, hashing is computed on the feature value and the hashed value is saved.

- The 13 Haralick feature extraction are:

  - $Angular\ second\ moment = \sum_i \sum_j p(i,j)^2$
  - $Contrast = \sum_{n=0}^{N_g-1} n^2 \left\{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)\right\}, |i-j| = n$
  - $Correlation = \frac{\sum_i \sum_j (i,j) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$

    $where\ \mu_x, \mu_y, \sigma_x\ and\ \sigma_y\ are\ the\ means\ and\ standard\ deviations$
  - $Sum\ of\ suares: variance = \sum_i \sum_j (i-\mu)^2 p(i,j)$
  - $Inverse\ Difference\ moment = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j)$
  - $Sum\ average = \sum_{i=2}^{2N_g} i p_{x+y}(i)$

    $where\ x\ and\ y\ are\ coordinates\ of\ an\ entry\ in\ GLCM, and\ p_{x+y}(i)\ is\ probability\ ofco - occurance\ matrix\ coordinates\ summing\ to\ x+y$
  - $Sum\ variance = \sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i)$
  - $Sum\ entropy = -\sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\} = f_8$
  - $Entropy = -\sum_i \sum_j p(i,j) \log(p(i,j))$

- $Difference\ variance = \sum_{i=0}^{N_g-1} i^2 p_{x-y}(i)$
- $Difference\ entropy = -\sum_{i=0}^{N_g-1} p_{x-y}(i) \log\{p_{x-y}(i)\}$
- $Info.measure\ of\ correlation\ 1 = \frac{HXY-HXY1}{\max\{HX,HY\}}$
- $Info.measure\ of\ correlation\ 2 = (1 = \exp[-2(HXY2 - HXY)])^{\frac{1}{2}}$

  $where\ HXY =$

  $-\sum_i \sum_j p(i,j) \log(p(i,j))$,

  $HX, HY\ are\ the\ entropies\ of\ p_x\ and\ p_y\ ; HXY1 =$

  $-\sum_i \sum_j p(i,j) \log\{p_x(i)p_y(j)\}; HXY2 =$

  $-\sum_i \sum_j p_x(i)p_y(j) \log\{p_x(i)p_y(j)\}$

- **Scale Invariant Feature Transform**

  A tool known as the Scale Invariant Feature Transform, or SIFT, that may be used to locate local features in an image. These features are frequently referred to as the "key point" of the image collection. The key points are stands for scale and rotation invariant. Both key points and corresponding descriptors are extracted. The descriptor value is mainly used as the feature value.

3.2.4 Dataset Preparation

The data set contains the values of the characteristics of the different images on the sheet as well as the value of the category. Prepare the dataset with Haralick feature values, SIFT features and category values. The dataset contains six columns. The first four columns are Haralick texture feature values, one column for SIFT and the rest are for class values, if essential, apply pre-processing techniques and then provide learning tasks.

3.2.5 Classification and Prediction

For classification, data set are separated into 2 parts:

1. Training set
2. Testing set

Majority of dataset are considered to the testing part which is 80 percentages while the remaining 20 percentage is considered to the testing part. The model trained by using training part. Once the training is completed, run the trained model through the test set and then calculate the accuracy of the model. Adjust the dataset values and retrain the model, if the accuracy shows low, Repeat this process until the desired accuracy is achieved. When training and testing are complete with restructured dataset, both training and testing part are combined into a single dataset and uses updated dataset as training set for training the final model. Later completing the training, a new data can be predicted by using the trained model.

Classification Algorithm

Random Forest

Random forest is a popular machine learning algorithm belonging to supervised learning method. Both the classification and regression problems can be used by random forest. Random forest based on the idea of ensemble learning that combines numerous classifiers in order to enhance the performance of the model and to resolve complex issues.

A random forest is a classification method that requires a set of decision trees that have been applied to various subsets of a given dataset and then combines the results in order to retrieve the accuracy of the dataset's predictions. Instead of depending on decision trees, Random Forest makes predictions based on the majority of the predictions from each tree and then determines what the final conclusion will be based on those.

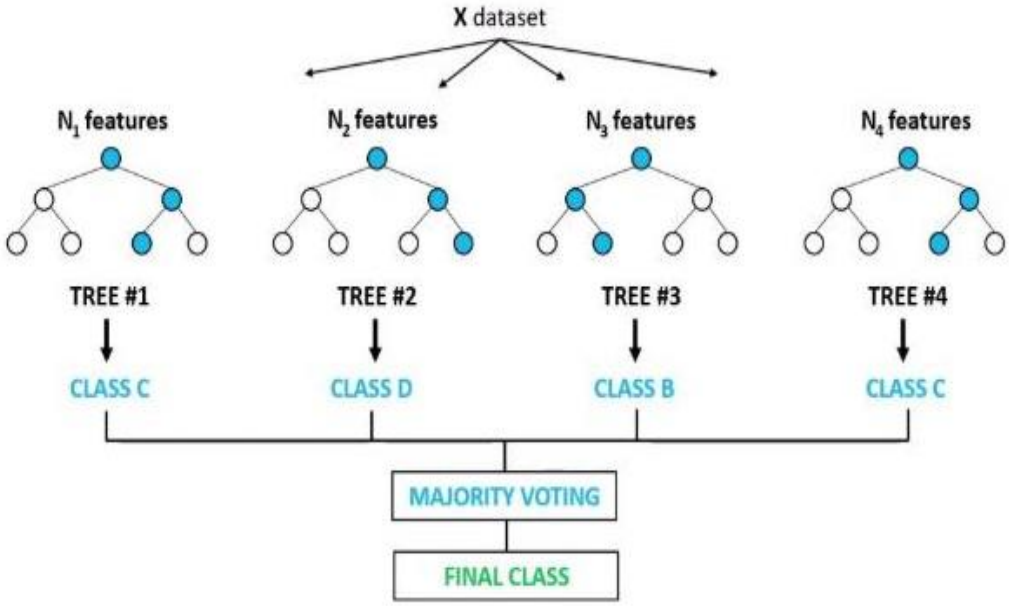

Fig 3.2: Random Forest Model

Random forests use multiple trees to predict the class of a set of data. Some decision trees may predict the right output, while others may not. However, when the information from each tree is combined, a more accurate prediction can be made. Two expectations with regard to the improvement of the random forest model are as follows: The classifier has to have definite actual values for the feature variables of the dataset in order to make accurate predictions as opposed to making educated guesses about the results. The prediction for each tree should have a very low correlation.

The random forest is divided into two phases. First phase, a random forest is built by combining "N" decision trees, and second phase; predictions are made for each tree built in the first phase.

The working process is as follows:

Step 2: First, select K random training data points.

Step 2: Assign the data points chosen as a subset to a decision tree.

Step 3: Determine what value "N" will represent for the decision tree that you intend to construct.

Step 4: Go step 1 and step 2.

Step 5: Find the prediction for each decision tree for each new data point, and then assign it to the highest voted category.

Support Vector Machine Algorithm

SVM, which stands for support vector machines, are one of the most common supervised learning techniques that are used for classification and regression issues. On the other hand, its primary application should be for solving classification problems in machine learning.

The Support Vector Machine (SVM) algorithm's objective is to deliver the best decision path or boundary that can divide the n-dimensional space into classes in order to simplify the future assignment of new data points to the proper categories. As a result, the hyperplane is the term given to the optimal decision boundary.

The extreme points or vectors that contribute to the formation of the hyperplane are defined by the SVM. These extreme situations are known as support vectors, and the method that analyses them is known as a support vector machine. Have a look at the following diagram, which illustrates two distinct categories that can either be classed by decision boundaries or hyperplanes.

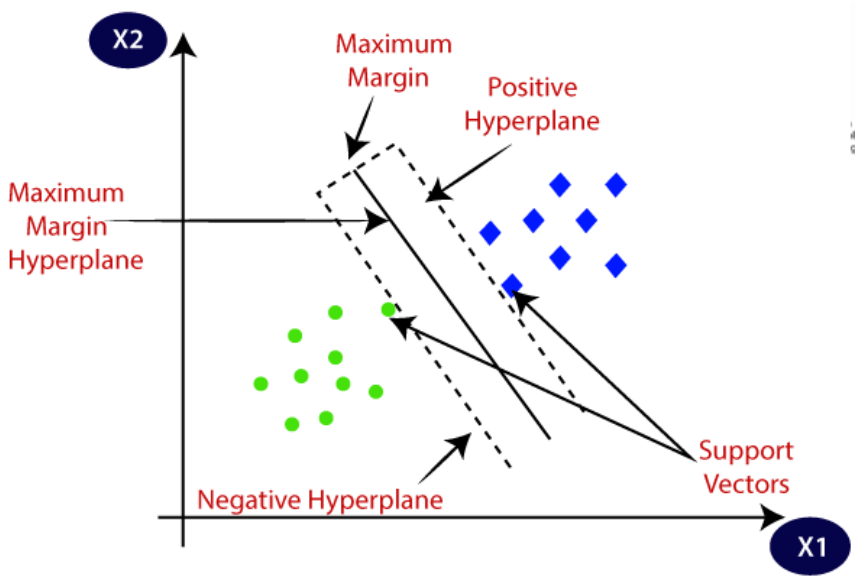

Figure 3.3: SVM Model

Long Short Term Memory

The Long-Short-Term Memory, often known as LSTM, is a form of Recurrent Neural Network (RNN). In an RNN, the input for the step that is now being processed is the output of the previous step. Hochreiter and Schmidhuber are responsible for developing of LSTM. This solved the problem of long-term dependencies that were plaguing RNNs. In this particular situation, RNNs are unable to predict words that are kept in long-term memory; however, they are able to generate more accurate predictions based on information that is now available. RNNs are incapable of providing effective performance when the gap length is increased. LSTM is capable of storing information for an extended period of time by default. For the purpose of processing, forecasting and classifying the data derived from time series. LSTMs have a structure that is similar to a chain, with four distinct neural networks and memory blocks that are referred to as cells.

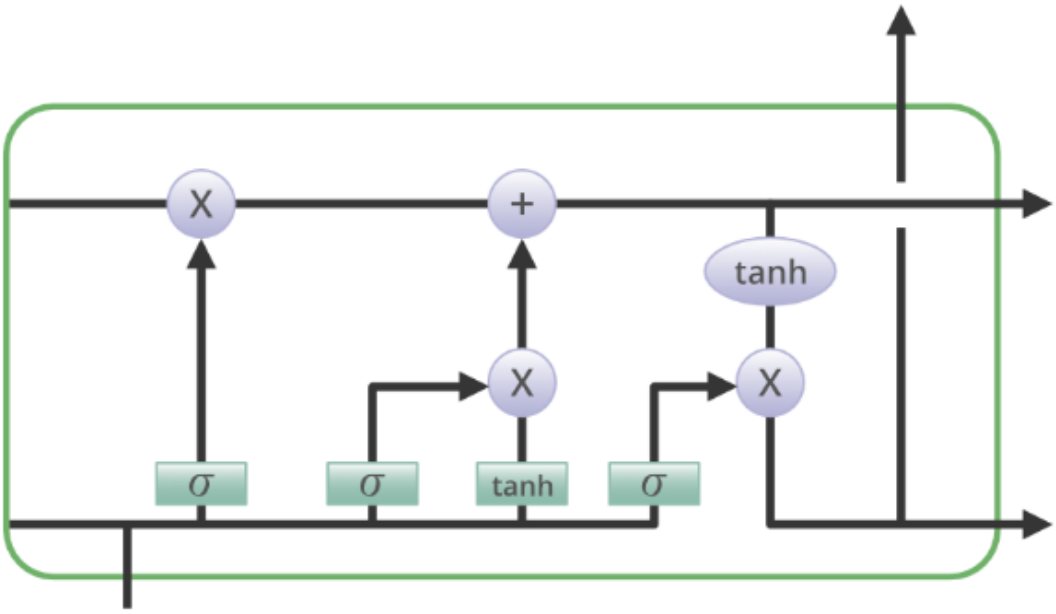

Figure 3.4: LSTM Model

Information is held in cells and memory operations are performed in gates. There are three gates –

- **Forget Gate**: The forget gate eliminates data from the cell state that is no longer required. The gate receives two inputs—xt, the input at a specific time, and ht-1, the output of the cell before it—which are multiplied by a weight matrix before a bias is added. The activation function receives the result and generates a binary output. When a cell state's output is "zero," that information is lost and is replaced with information from output "one," which is retained for later use.

- **Input Gate**: Through input gates, relevant information can be added to the cell state. The information is first regulated by a sigmoid function, and then, similarly to a forget gate, the stored values are filtered by the inputs ht-1 and xt. The function tanh is then used to produce the vector. Every possible value for xt and ht-1 is returned in the output of this function, which ranges from -1 to 1. Finally, multiply the value of the vector by the adjusted value to get useful information.

- **Output gates** Output gates work to extract helpful data about the present state of the cell, which would be displayed as output. By first using the tanh function on the cells, a vector is first created. The data is then filtered using the inputs xt and ht-1 and modulated using a sigmoid function in accordance with the values to be remembered. The values in the vector and the controlled values are then multiplied and sent as outputs and inputs to the following cell.

XGBOOST

The performance and speed of tree-based machine learning algorithms have been improved by the gradient boosting (ensemble) technique known as Extreme Gradient Boosting, or XGBoost. One of the approaches that has contributed to this progress is XGBoost (sequential decision tree). XGBoost was created by Tianqi Chen and was initially controlled by the Distributed (Deep) Machine Learning Community Group (DMLC).). It has quickly become the most popular applied machine learning algorithm due to its ability to provide effective solutions for structured and tabular data. It used to be the case that packages for his XGBoost were exclusively developed in Python and R, but now it has expanded to other languages including as Java, Scala, and Julia.

XGBoost belongs to the group learning enhancement technologies category. To improve prediction accuracy, collective learning uses a set of predictors, which are different models. By adding weights to the model, the boosting strategy tries to fix errors produced by earlier models.

3.2.6 Evaluation

To quantify model performance, system needs a model evaluation metric. Choosing an evaluation metric depends on specific machine learning task

- The test set is applied to the model that has been trained in order to evaluate the model.
- It's a way of measuring how many of a classifier's predictions were accurate, how many were wrong, and where the classifier confused.
- In order to assess how well the model performs, the accuracy of the model is determined.
- The percentage of correctly predicted outcomes for a set of test data is the definition of accuracy.

$$accuracy = \frac{correct\ predictions}{all\ predictions}$$

3.2 Software Requirements and Specifications

Python 3

Python is a high-level, interpretative, object-oriented, general-purpose programming language. It was created by Guido van Rossum between 1985 and 1990. The GNU General Public License also applies to the source code of the Python programming language, just like it does for Perl (GPL). Python is the title of the television show "Monty Python's Flying Circus," not a snake.

In 2008, Python 3.0 was released. Even though it claims to be backward compatible, many of this version's crucial features have been backported to work with version 2.7.

Python has been created to be simple to read. While other languages include punctuation and have less syntactic structure than English, English keywords are frequently used.

Python is necessary for both professionals and students to become competent software developers. in particular when working in the field of web development.

Listed below are some of the most important advantages of becoming proficient in Python.

Python code must be interpreted: The interpreter is responsible for Python processing during runtime. It is not necessary to compile the programme in order for you to be able to run it. This is comparable to the programming language known as PERL or PHP.

Python allows for interaction: In point of fact, all you have to do to write a Python programme is to sit down at the Python prompt and speak to the interpreter immediately.

Python is an object-oriented language: Given that Python is an object-oriented programming language, it supports programming techniques that enclose code in objects.

Python is a good language to learn for beginners: Python is a great programming language for beginners to learn since it enables the building of a diverse array of programmes, ranging from straightforward word processing to web browsers and games.

Important features of Python are:

- To create large programmes, it can be used as a scripting language or compiled to bytecode.
- It supports both structured and functional programming methods, as well as object-oriented programming.
- Supports dynamic type checking and offers very high-level dynamic data typing.
- Support automatic garbage collection.
- Easy integration with C++, COM, ActiveX, CORBA, Java. Browsers and games.

Python Libraries

A library is usually a collection of books or a room or place where many books are kept for later use. Similarly, in the world of programming, a library is a collection of precompiled code that may be afterwards utilised programmatically for well-defined operations. Libraries can include documents, message templates, configuration information, values, classes, and more in addition to executable code.

The Python library is a collection of modules that share a common characteristic. It includes bundled chunks of code that are adaptable for use in a variety of different applications. Programmers will find that using Python is simplified and made more convenient as a result of this. You won't have to keep writing the same code over and over again for different apps. The use of Python libraries is extremely significant in many different fields, including machine learning, data science, and the visualisation of data.

Python Standard Library

The Python standard library includes the precise syntax, semantics, and tokens that are used by python. It features a number of built-in modules, including I/O and other core modules, which provide access to the basic system functions. The majority of Python's libraries were developed using the programming language C. Over two hundred fundamental modules are contained within the Python standard library. Python is able to achieve its status as a high-level programming language because to the combined efforts of all of these factors. The Python standard library is responsible for a fairly significant part of the programme. Without them, programmers will not have access to the features that Python has to offer. Aside from that, Python is equipped with a number of libraries that make the life of the coder simpler. Some of the most popular libraries are as follows:

1. Matplotlib

The plotting of numerical data can be done with the help of this library. As a result, it is used for of data analysis. In addition to that, it is a library that is freely accessible to the public and can plot high-definition numerical information in the form of graphs, pie charts, histograms, and scatterplots. Pandas is a library that is necessary for data scientists and is known by its acronym Pandas. It is an open-source machine learning library that offers customisable high-level data structures and a variety of analytical tools. Facilitates data analysis, data manipulation and data purification. Pandas enables users to do operations such as sorting, reindexing, iterating, concatenating, concatenating, manipulating data, aggregating, and displaying data.

2. Pandas

Data scientists cannot do their jobs without the Pandas library. It is an open-source machine learning software that offers a selection of analytical tools along with high-level data structures that can be customised. Simplify the processes of data analysis, data manipulation, and data cleaning. Pandas enables users to do operations

such as sorting, reindexing, iterating, concatenating, concatenating, manipulating data, aggregating, and displaying data.

3. Numpy

Numpy is short for "Numeric Python." This is the most popular library. Large arrays and multidimensional data are supported by this well-liked machine learning package. It includes built-in math operations to simplify calculations. Even libraries like TensorFlow internally use Numpy to carry out different tensor operations. One of this library's key features is the matrix interface.

4. SciPy

SciPy is short for "Scientific Python." It is a powerful scientific computing open source library. The foundation of this library is the Numpy extensions. Use Numpy to manage complicated mathematical computations. Numeric data codes are saved in SciPy while sorting and indexing of array data is supported by Numpy. Application developers and engineers utilise it a lot.

5. Scikit-learning

A well-known Python package for handling complicated data is this one. A free machine learning library is called Scikit-learning. It supports a number of supervised and unsupervised methods, including clustering, linear regression, and classifiers. SciPy and Numpy both integrate well with this package.

MySQL

When it comes to relational database management, MySQL is among the most widely used database management system programmes. Oracle has provided support for the MySQL open-source database software. When compared to Oracle and Microsoft SQL Server Database, this database management solution is far more efficient, user-friendly, and scalable. It is frequently combined with PHP scripts in order to create enterprise applications that are strong and dynamic, and these applications are typically based on the web.

Written in the programming languages C and C, MySQL is a database management system that was developed, sold, and supported by Swedish company MySQL AB. MySQL is not to be pronounced like "My Sequel"; that is not the correct pronunciation. My Ess Que Ell. On the other hand, you can pronounce it whatever you like. MySQL is compatible with numerous operating systems, including Windows, Linux, and MacOS, and it achieves this versatility by utilising the C and Java programming languages.

MySQL is an RDBMS that offers some benefits.

- It allows database operations to be performed on tables, rows, columns, and indexes.
- It establishes relational database structures in the form of tables, which are essentially collections of rows and columns. These structures are also referred to as relations.
- This permits the indexes on the table to be automatically updated, and it also ensures that there is relational integrity across rows or columns in various tables.
- Makes use of a large number of SQL queries in order to compile helpful data for end users from a variety of tables.

HTML

The Hypertext Markup Language (HTML) is a collection of markup symbols or code placed in web-viewable files. Markup instructs a web browser on how to render text and graphics on a website. Each part of the markup code (between the "<" and ">" characters) is called an element, but often also a tag. Some elements are presented in pairs to identify the beginning and completion of the display effect.

Hypertext Markup Language is a computer language that makes it easy to create websites. The language has code words and syntax like any other, is relatively easy to understand, and over time becomes more and more powerful in what someone can create. HTML continues to evolve to meet the needs and demands of the Internet under the guise of the World Wide Web Consortium, the organization that designs and maintains the language. Like the transition to Web 2.0.

HyperText is how Internet users navigate the Web. By Clicking on special text called a hyperlink takes the user to a new page. Using Hyper means it's non-linear, the user can simply click on any available link and he can go anywhere on the web. Markup is what you do to text that contains HTML tags. They marked it as some kind of text. For example, markup text can be bold or italic to draw special attention to a word or phrase.

CSS

The language known as Cascading Style Sheets, or CSS for short, is a style sheet language that is used to specify the display of a page that was produced in a mark-up language such as HTML. Along with HTML and JavaScript, CSS is one of the three fundamental technologies that underpin the World Wide Web.

The Cascading Style Sheets (CSS) standard was created to enable the separation of presentation and content in areas like as colours, fonts, and layout. This separation has the potential to increase content accessibility, provide more flexibility and independence in establishing layout attributes, and enable multiple websites to share

formatting by defining CSS in a single file. css files independently, which eliminates unnecessary complexity and redundancy in organised information and makes it possible to cached it. css files improve the speed at which pages load across a variety of file formats and file sharing websites.

By separating the format from the content of a page, it is possible to show the same mark-up page in different styles for different perspectives, such as on screen, in print, by voice (using an audio browser or screen reader), and on touch devices in Braille. If the material is accessed from a mobile device, CSS also contains various rules for formatting it to be shown. The use of a particular precedence scheme to determine which style rule should be applied in the event that many style rules match a given element is where name nesting originates.

JavaScript

JavaScript is a text-based programming language that is used on both the client and server sides to allow web pages to interact with users. JavaScript was developed by Sun Microsystems. HTML and CSS are programming languages that give web pages with structure and style, respectively. JavaScript, on the other hand, provides the interactive features that bring users to web pages. The search boxes on Amazon, the embedded movies on the New York Times website that summarise the news, and the updates to the Twitter feed are all examples of common uses of JavaScript. The user experience provided by your website is enhanced when JavaScript is included by converting static pages to interactive pages. In summary, JavaScript adds behaviour to web pages.

VS CODE – IDE

Visual Studio Code combines the simplicity of a source code editor with powerful development tools, such as IntelliSense debug and code completion. Best of all, it's a backward-bending editor. The smooth edit-build-debug cycle means more time implementing your ideas.

In its most basic form, Visual Studio Code is a source code editor that is incredibly quick and definitely works for regular use. Syntax highlighting, bracket matching, auto-indent, field selection, and extraction are just some of the time-saving features that come standard with Visual Studio Code, which also supports a large number of programming languages. You can traverse your code with ease due to the intuitive keyboard shortcuts, the ease with which customisation is possible, and the keyboard shortcut mappings contributed by the community.

When you are doing real coding, you will almost always gain more from using tools that comprehend code as opposed to those that just read blocks of text. IntelliSense code completion, semantically rich code understanding and navigation, and code refactoring are all included into Visual Studio Code as standard features. And when coding gets difficult, debugging becomes even more difficult. Because debugging is frequently the feature that developers miss the most in the experience of lean coding the most, we decided to build it ourselves. The interactive debugger that comes with Visual Studio Code allows you to explore the source code, inspect variables, see the call stack, and run commands in the console.

Additionally, VS Code interfaces with authoring and scripting tools to execute typical operations, hence accelerating your everyday workflow. VS Code supports Git, allowing you to use source control and view pending changes without leaving the editor.

DJANGO

Django is a high-level web framework written in python that promotes speedy development as well as design that is clean and pragmatic. It is built by seasoned developers, and it takes care of a significant portion of the difficulty that will be associated with web development. As a result, you can concentrate on building your application without having to invent the wheel. It is completely free to use and open source.

Incredible acceleration: Django was created to assist developers in bringing applications from the concept stage all the way through to a finished state as rapidly as feasible.

Confidently safe and sound: Django places a high priority on security and provides developers with tools to help them avoid making many of the most common security problems.

Extremely adaptable to changing needs: Django's capacity to quickly and flexibly scale is being placed to use on some of the largest websites on the web.

Django is a web framework that is written in Python and is free and open-source. It utilises the model–template–views (MTV) architectural structure. The Django Software Foundation (DSF), an American impartial organisation that was founded as a non-profit, is the group responsible for its maintenance.

The fundamental objective of Django is to simplify the process of developing database driven websites with complicated structures. The framework places an emphasis on the reusability and "pluggability" of components, as well as on the reduction of code, the low coupling of components, rapid development, and the "don't repeat yourself" philosophy. Python is utilised in every aspect, including the configuration settings, file management, and data modelling. Additionally, Django comes with an optional administrative create, read, update, and delete interface. This interface is generated dynamically via introspection and is managed by administrative models.

CHAPTER 4

RESULT AND DISCUSSION

Kaggle's Plants Village Apple Leaf image is used as a training dataset. Some images are from the Apple dataset and features are extracted from the images. The dataset consists of four classes, class 0 representing scab disease, class 1 representing black rot disease, class 2 representing cedar disease, and class 3 representing healthy leaves. The results obtained after applying preprocessing steps, that shown in Fig 4.1 and fig. 4.2

Figure 4.1 Input image

Figure 4.2 After contrast enhancement

After classifying the disease, the results are presented in Fig. .3.1 The input image of the leaf with scab disease is displayed. The classifier correctly classified in class “0” that is Scab disease. It also shows treatment and prevention of the diseased plant leaf.

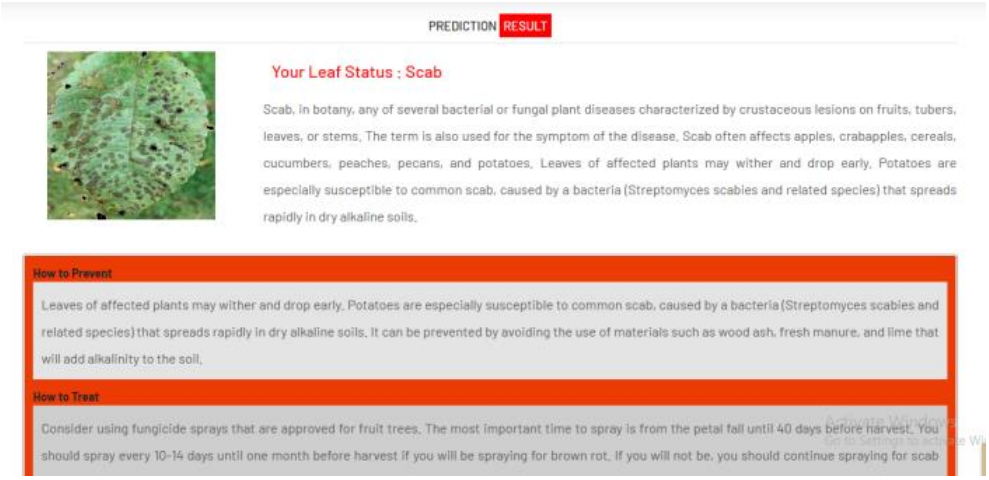

Figure 4.3: Output of Classifier

Black rot disease pictures are given. The classifier correctly classified in class 1 i.e. shown in fig. 4.4.

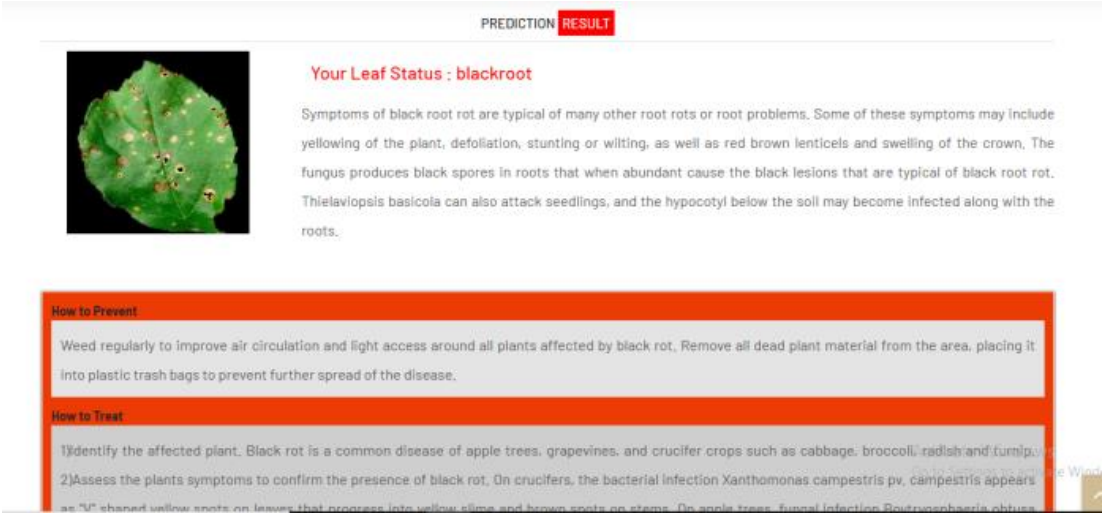

Figure 4.4 Output of Classifier

DISCUSSION

The evaluation of 4 classifiers is done using the test set, which contains features of 150 images. The test set is applied to 4 classifiers and the prediction result is used for performance evaluation. The below figures show the classification report and confusion matrix of these 4 classifiers. From the classification report, it is clear that out of 4 classifiers, Random Forest shows better performance and LSTM shows least performance. Thus, random forest classifier is used for the final model creation and prediction.

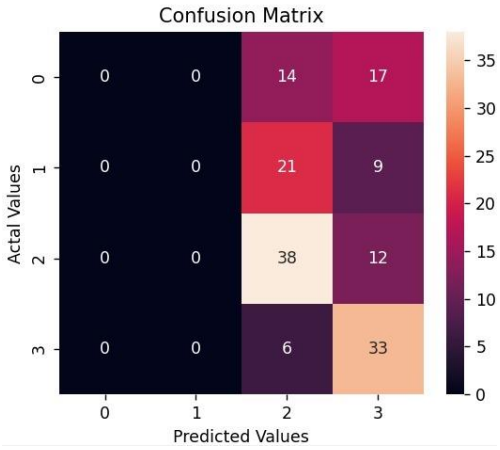

Figure 4.5 Confusion Matrix of SVM

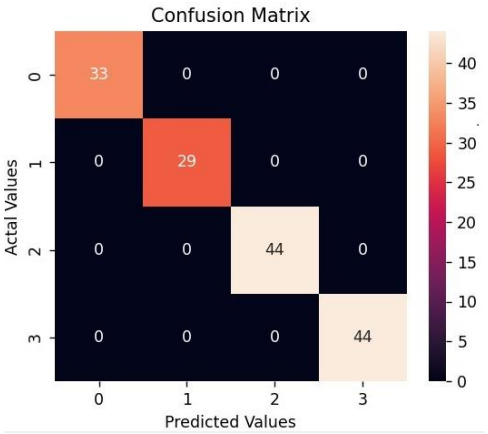

Figure 4.6 Confusion Matrix of Random Forest

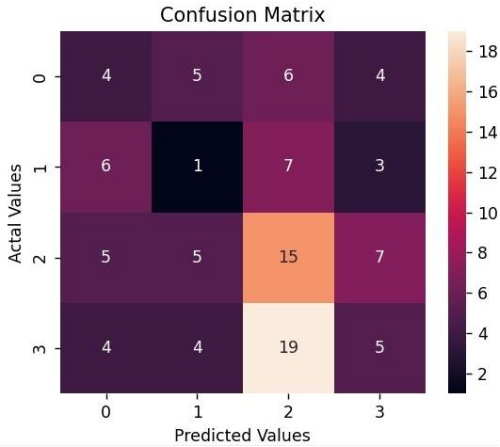

Figure 4.7 Confusion Matrix of LSTM

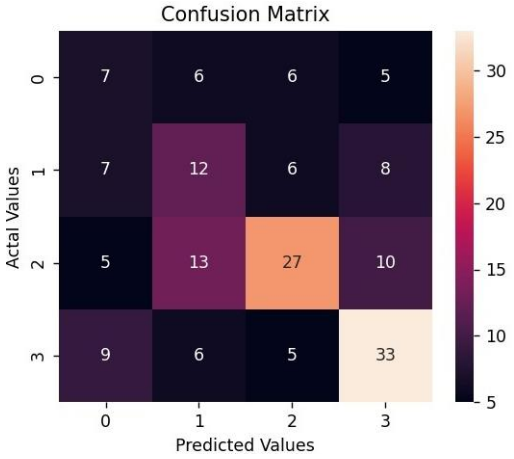

Figure 4.8 Confusion Matrix of XGBOOST

CHAPTER 5

CONCLUSION

This project presents a highly accurate solution for classifying and predicting leaf diseases of various plants using machine learning algorithms for classification purposes. The system predicts plant diseases at the early stages of disease and helps farmers apply appropriate disease elimination methods. Early prediction of disease can help increase crop yields. Treatments and preventions of classified diseases can also be included in the model. To extract the features of the image from the dataset, uses the Haralick feature extraction method and the SIFT feature extraction method. Random forest, SVM, XgBoost, LSTM machine learning is used for classification.

5.1 FUTURE ENHANCEMENT

In future, it is possible to add sensors to monitor temperature, humidity etc. and process these sensed data to support precision farming.

REFERENCES

[1] Vijai Singh , A.K. Misra , 2017, Detection Of Plant Leaf Diseases Using Image Segmentation And Soft Computing Techniques.

[2] Sujatha R, Y Sravan Kumar and Garine Uma Akhil, 2017, "Leaf Disease Detection Using Image Processing", Journal of Chemical and Pharmaceutical Sciences.

[3] K. Vinoth Kumar and T. Jayasankar, 2019, "An Identification of Crop Disease Using Image Segmentation", IJPSR.

[4] Shima Ramesh, Mr. Ramachandra Hebbar, Niveditha M, Pooja R, Prasad Bhat N, Shashank N, 2018, "Plant Disease Detection Using Machine Learning", International Research Journal of Engineering and Technology (IRJET)

[5] Mrs. Shruthi U, Dr. Nagaveni V, Dr. Raghavendra B K, 2019, "A Review On Machine Learning Classification Techniques For Plant Disease Detection", 5th International Conference on Advanced Computing & Communication Systems (ICACCS),IEEE.

[6] Pushkara Sharma, Pankaj Hans, Subhash Chand Gupta, 2020, " Classification Of Plant Leaf Diseases Using Machine Learning And Image Preprocessing Techniques", 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE

[7] Kawcher Ahmed, Tasmia Rahman Shahidi, Syed Md. Irfanul Alam and Sifat Momen, 2019, "Rice Leaf Disease Detection Using Machine Learning Techniques", International Conference on Sustainable Technologies for Industry 4.0 (STI), IEEE

[8] Meghana Govardhan, Veena M B, 2019, "Diagnosis of Tomato Plant Diseases using Random Forest", Global Conference for Advancement in Technology (GCAT), IEEE

[9] Poojan Panchal, Vignesh Charan Raman, Shamla Mantri, 2019, "Plant Diseases Detection and Classification using Machine Learning Models", 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), IEEE.

[10] Ghaiwat Savita N, Arora Parul. Detection and classification of plant leaf diseases using image processing techniques: a review. Int J Recent Adv Eng Technol 2014;2(3):2347–812. ISSN (Online).

[11] Arivazhagan, S., Newlin S., Ananthi, S., Vishnu V (2013) “Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features”, Agric Eng. Int CIGR.

[12] S. D.M., Akhilesh, S. A. Kumar, R. M.G. and P. C., "Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight," 2019 International Conference on Communication and Signal Processing (ICCSP), 2019.

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

[14] F. T. Pinki, N. Khatun and S. M. M. Islam, “Content based paddy leaf disease recognition and remedy prediction using support vector machine,” 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, 2017, pp. 1–5.

APPENDIX

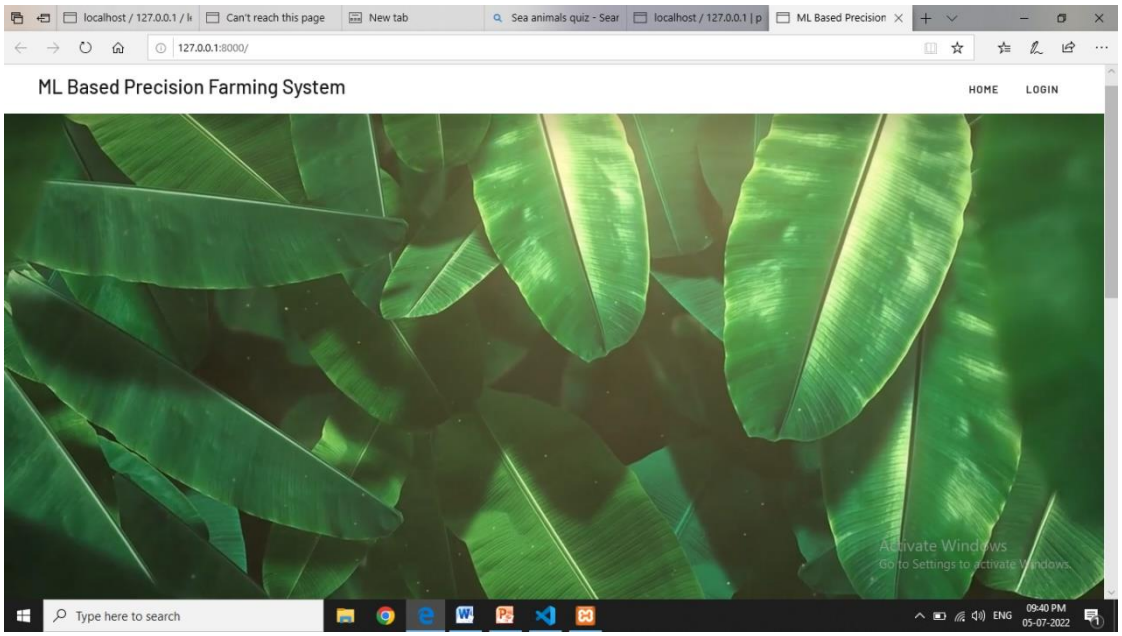

Fig. A.1 Homepage

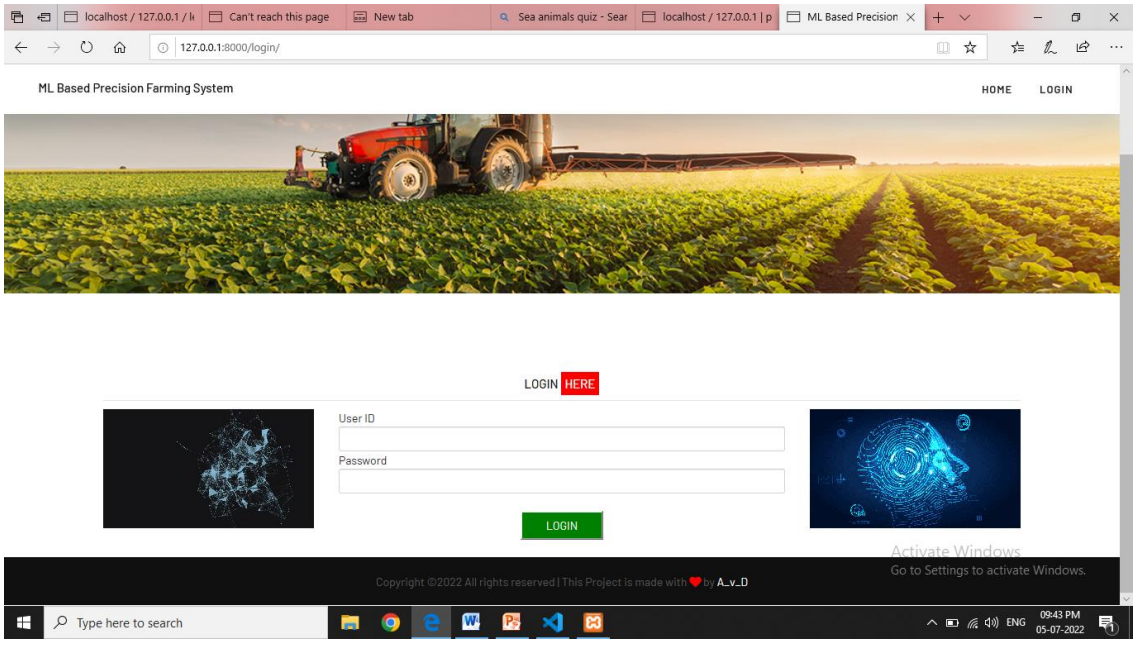

Fig. A.2 Login page

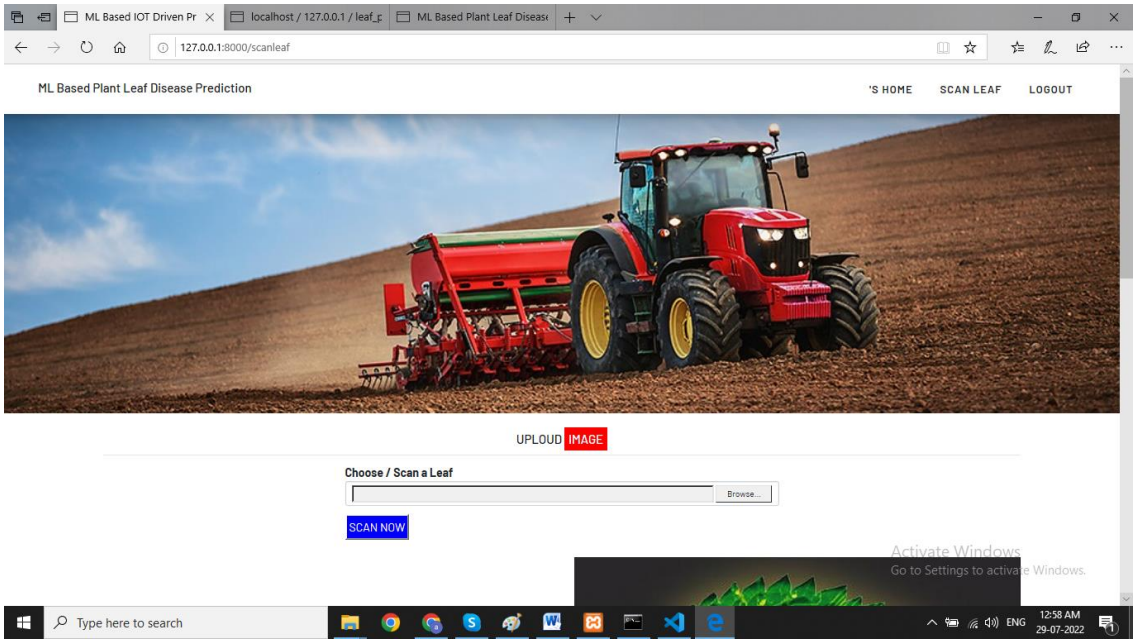

Fig. A.3 Scan leaf

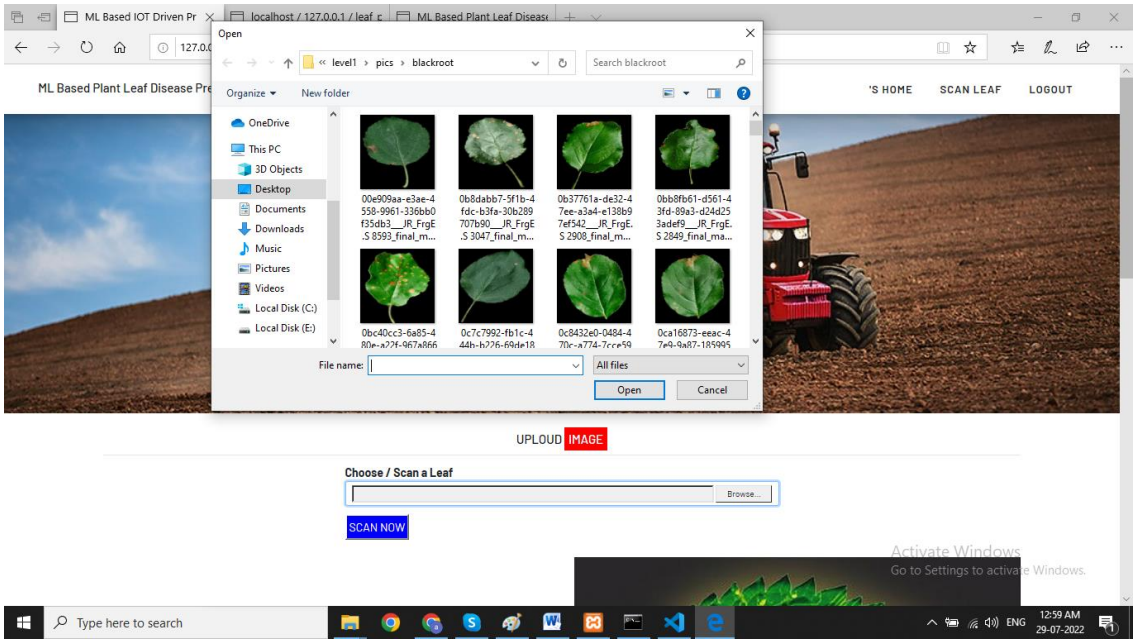

Fig. A.4 Upload Image

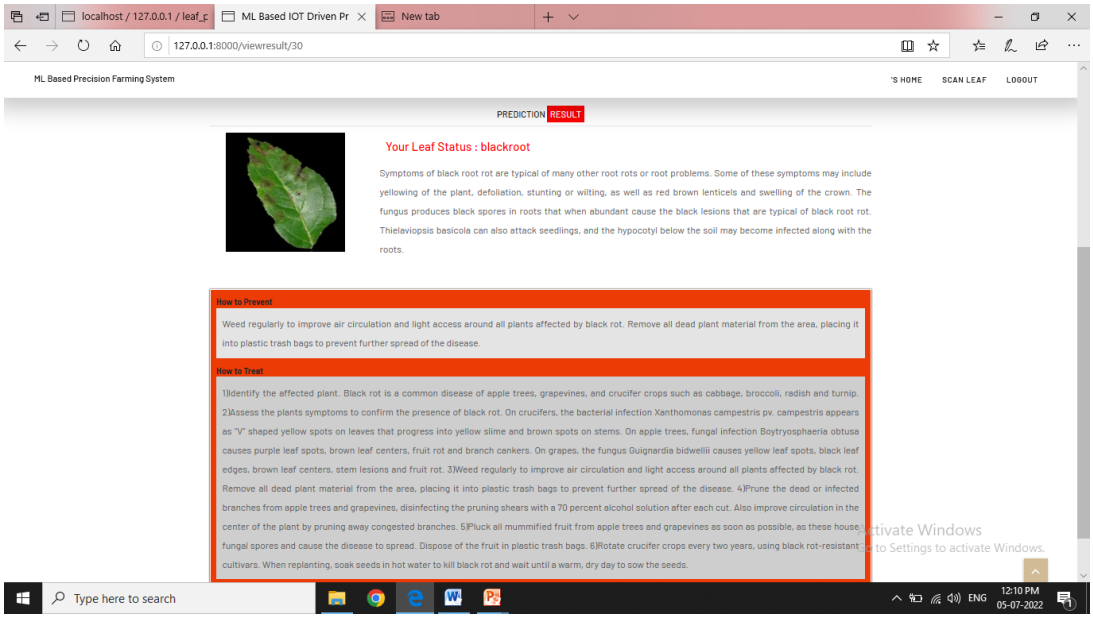

Fig. A.5 Disease prediction