Dissertation on
Detection of Plant Leaf Diseases by Utilizing Multiclass Support Vector Machine

*Thesis submitted towards partial fulfilment
of the requirements for the degree of*

Master of Technology in IT (Courseware Engineering)

Submitted by
Sampa Biswas

EXAMINATION ROLL NO.: *M4CWE23009*
UNIVERSITY REGISTRATION NO.: *160376 of 2021-22*

Under the guidance of
Mr. Joydeep Mukherjee

School of Education Technology
Jadavpur University

Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata-700032
India
2023

M.Tech. IT (Courseware Engineering)
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled **"Detection of Plant Leaf Diseases by Utilizing Multiclass Support Vector Machine"** is a bonafide work carried out by Sampa Biswas under our supervision and guidance for partial fulfilment of the requirements for the degree of **Master of Technology in IT (Courseware Engineering**) in School of Education Technology, during the academic session 2022-2023.

SUPERVISOR
School of Education Technology
Jadavpur University,
Kolkata-700 032

DIRECTOR
School of Education Technology
Jadavpur University,
Kolkata-700 032

DEAN - FISLM
Jadavpur University,
Kolkata-700 032

M.Tech. IT (Courseware Engineering)
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF APPROVAL**

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactory to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made or opinion expressed or conclusion drawn therein but approve the thesis only for purpose for which it has been submitted.

	--
Committee of final examination for evaluation of Thesis	--
	--
	--

** Only in case the thesis is approved.

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of her **Master of Technology in IT (Courseware Engineering)** studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by this rule and conduct, I have fully cited and referenced all materials and results that are not original to this work.

NAME: SAMPA BISWAS

EXAMINATION ROLL NUMBER: M4CWE23009

THESIS TITLE**: Detection of Plant Leaf Diseases by Utilizing Multiclass Support Vector Machine.**

SIGNATURE: DATE:

ACKNOWLEDGEMENT

I feel extremely grateful in presenting this thesis at **School of Education Technology, Jadavpur University, Kolkata**, in partial fulfilment of the requirements for the degree of **Master of Technology in IT (Courseware Engineering)**.

It is a great pleasure to me to express my sincere appreciation and gratitude to my Supervisor **Mr. Joydeep Mukherjee**, who provided me with invaluable guidance, support, and encouragement throughout my M.Tech journey. I am deeply grateful for the intellectual freedom he gave me and for his constant encouragement to successfully complete my research work.

I extend my heartfelt appreciation **to Prof. (Dr.) Matangini Chattopadhyay, Director of the School of Education Technology**, and **Dr**. **Saswati Mukherjee** for their generous help, support, and encouragement during the entire course of my thesis work. I would also like to thank my batchmates for their support, motivation, and insightful discussions to complete my thesis successfully.

Date: Thanks & Regards,

………..……………………………..
SAMPA BISWAS
Examination Roll No: M4CWE23009
Registration No: *160376 of 2021-22*
M. Tech. IT (Courseware Engineering)
School of Education Technology
Jadavpur University, Kolkata- 700032

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF PUBLICATON

S. Biswas and Mr. J Mukherjee, "Utilizing Multi-Class Support Vector Machine to Detect Plant Leaf Diseases" International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 11, Issue VIII, Aug 2023

Link-https://doi.org/10.22214/ijraset.2023.55343

EXECUTIVE SUMMARY

Agriculture is an essential component of our economy and food security, but it is threatened by various factors, including plant diseases. Leaf diseases are a common problem in agriculture, and they can have a significant impact on crop yields and quality. Leaf disease detection and management are crucial for sustainable agriculture, and there are various methods to achieve this. One of the most promising methods is leaf disease detection using image processing. This technique involves capturing digital images of plant leaves and analysing them to identify disease symptoms. Plant leaf disease detection is an important task in agriculture to prevent the spread of diseases and increase crop yield. With the advancement of machine learning techniques, automated systems have been developed to detect and diagnose plant leaf diseases.

Leaf disease detection using image processing and machine learning algorithms to identify the symptoms of diseases on the leaves of plants. The technique involves acquiring an image of a plant leaf, processing the image to extract features, and classifying the image based on the features. The input images are pre-processed, and the features are extracted using various techniques such as convolutional neural networks (CNN), support vector machines (SVM), and decision trees. The extracted features are then used to train the machine learning model, which can classify new images as healthy or diseased. The accuracy of these models varies depending on the quality of the input images and the size of the dataset. Additionally, the technique can help farmers make informed decisions about when and how to treat their crops, leading to improved yields and reduced costs.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Agriculture is the backbone of many economies worldwide, providing food, fibre, and other essential resources to the population. However, plant diseases pose a significant threat to the agricultural industry, leading to substantial crop losses and reduced yields. Plant leaf diseases are caused by pathogens such as fungi, bacteria, and viruses, as well as environmental factors such as temperature, humidity, and soil conditions.

The identification and diagnosis of plant leaf diseases are crucial in preventing the spread of disease and minimizing the economic impact on farmers. Traditional methods of diagnosis involve visual inspection by trained experts, which can be time-consuming and often inaccurate. However, advances in technology, particularly in image processing and machine learning, have led to the development of more efficient and accurate techniques for plant leaf disease detection. Automated detection systems using machine learning algorithms can quickly and accurately identify plant leaf diseases, enabling early disease diagnosis and timely intervention. This technology has the potential to transform the agricultural industry by improving crop yields and reducing the use of pesticides and other chemicals that are harmful to the environment.

Plant leaf disease detection technology can provide farmers with the tools they need to make informed decisions and reduce the economic impact of plant diseases on their crops, thereby increasing food security for the population. Plant leaf disease detection is a rapidly growing field that involves the use of technology to identify and diagnose diseases that affect plants. With the increasing demand for food production and the challenges posed by climate change, the need for accurate and timely diagnosis of plant diseases has become more critical than ever before. Plant leaf disease detection is achieved through various techniques such as image processing, machine learning, and deep learning algorithms. These techniques enable the automated and efficient detection of plant diseases, which can aid in early disease detection and control, preventing significant crop loss. This technology has the potential to revolutionize the agriculture industry by enabling farmers to diagnose and treat plant diseases promptly, leading to increased crop yields and improved food security.

1.2 PROBLEM STATEMENT

The purpose of this research is to study and observe common diseases among plant leaves and to develop a reliable and accurate method for identifying and categorizing those diseases that are present in plant leaves.

1.3 OBJECTIVES

The project aims to contribute to sustainable agriculture by providing an effective, accessible, and reliable tool for early disease detection and improving crop yields and minimizing economic losses. The following are the main goals of the initiative to identify plant leaf diseases via image processing:

1. Disease Identification: Create a system that can correctly identify and distinguish between different kinds of plant leaf diseases using leaf image analysis.

2. Accuracy Improvement: Using image processing techniques, feature extraction techniques, and machine learning algorithms, increase the accuracy of disease recognition.

3. Multi-Class Classification: Enable the system to categorize plant leaves into various disease classes, enabling a thorough evaluation of the health status of the plant.

1.4 BACKGROUND STUDY

Plant leaf disease detection is an important topic in agricultural research, as it plays a crucial role in ensuring food security by detecting and controlling diseases that can affect crop yield and quality. Traditionally, plant diseases were identified by visual inspection, but this method is often time-consuming and prone to errors. With the advent of modern technology, automated systems for plant disease detection have been developed, which can provide faster and more accurate results. One of the most common methods for automated plant disease detection is image analysis, which involves capturing images of plant leaves and using software algorithms to analyse the images for signs of disease. Image analysis can detect a wide range of plant diseases, including fungal, viral, and bacterial diseases, as well as nutrient deficiencies. In recent years, the development of computer vision techniques has allowed for more efficient and accurate detection of plant diseases. This has been made possible by the availability of large datasets of plant images and the advancement of machine learning algorithms, particularly deep learning. Deep learning models, such as convolutional neural networks (CNNs), have shown promising results in detecting plant leaf diseases from images. These models can learn to identify specific patterns and features of diseased leaves and classify them into different disease categories. There are also different approaches to plant leaf disease detection, such as using hyperspectral imaging, which involves capturing images of leaves at multiple wavelengths. This allows for the detection of subtle changes in leaf colour and texture that may not be visible to the naked eye. Overall, plant leaf disease detection is an important and rapidly evolving field, with many exciting research opportunities. By developing more accurate and efficient methods for disease detection, researchers can help to ensure the health and productivity of crops and contribute to global food security.

1.5 ORGANISATION OF THESIS

This thesis presents methods for maximizing the benefits of image processing technology. Discussions on the problems, existing remedies, and suggested approaches are held. In addition, the suggested methods provided that raise the level of accuracy and increase quality.

The structure of the thesis is as follows:

Chapter 1: Introduction

This chapter depicts the overview of plant diseases and their impact on agriculture. It also represents the problem statement, objectives, and the background information of plant disease identification.

Chapter 2: Literature Survey

This chapter presents the traditional methods of plant disease detection Image processing techniques Summary of existing methods and their limitations. Literature survey will be about the concepts and references of previous work done so far about the proposed approach or related algorithms.

Chapter 3: Methodology

This represents the methods of dataset collection techniques of Pre-processing, segmentation using **K-means Clustering**, feature extraction using **Grey Level Co-occurrence Matrix (GLCM)**, and **SVM** for image classification using image processing algorithms.

Chapter 4: Proposed approach

Description of proposed approach, block diagram, and working principles of proposed methods are discussed in this chapter.

Chapter 5: Experimentations and Results

Evaluation of the proposed method and discussion of results and analysis are discussed. The accuracy of proposed algorithm is descripted here.

Chapter 6: Comparative analysis

In this section comparison of proposed algorithm with existing methods are shown. The methodologies were used and their accuracy values are compared here.

Chapter 7: Conclusion and Future Scope

Summary of this research work, their limitations and recommendations for future work are described here.

Chapter 8. References

List of references cited in the thesis are given here. Total 30 references are given.

Appendix

The Code snippets and software used in the research are descripted in this section.

CHAPTER 2

LITERATURE SURVEY

Plant leaf disease detection using image processing is a crucial area of research in agriculture and computer vision. It involves the use of digital image processing techniques to identify and classify plant diseases based on the visual symptoms present on the plant leaves. Plant leaf disease detection is an ongoing research area that has gotten a lot of interest in recent years because of the growing demand for efficient and reliable disease detection systems in agriculture.

R. Meena Prakash et al. [1] employ digital image processing to identify illness in citrus leaves. They employed the K-means clustering approach for picture segmentation, the Grey Level Co-occurrence Matrix (GLCM) method for feature extraction, and the SVM method for classification. This suggested technique extracts four features: contrast, energy, homogeneity, and correlation. The dataset is made up of 60 photos of leaves, 35 of which are sick and 25 of which are healthy.

Raut and Fulsunge proposed an algorithm to detect plant leaf and fruit disease detection [2]. Image acquisition, pre-processing i.e., image resizing, filtering of noise, contrast enhancement, and morphological operations are done. Using k-means segmentation is done and the input image is partitioned into three cluster. Different statistical texture features i.e., energy entropy, correlation, contrast, covariance is extracted using GLCM.

Multilevel SVM is used as classifier. Total 73 leaf images and21 fruit images are used for disease detection. And they found the k-means and SVM algorithm provides maximum accuracy and consumes very less time.

In paper [3] Khirade and Patil consider colour, texture, and morphological features for detection of diseases. In their paper they have discussed technique for pre-processing, noise removing, various method for image segmentation like Boundary and spot detection, k-means, and Otsu Threshold algorithm. Features extraction using colour co-occurrence method and H&B components and classified using both Artificial Neural Network (ANN) and back propagation. Using this model various plant disease can be identified.

Singh and Misra proposed an automatic method for plant disease detection using genetic algorithm [4]. They have worked on little leaf disease in pine trees and classified into five classes after features extraction using genetic algorithm and minimum distance criterion and then classification.

A. Devaraj et al. designed their system to determine the illness of leaves, which is more time efficient [5]. Image acquisition, pre-processing i.e., filtering, RGB to HIS (Hue, Intensity and Saturation) conversion, morphological operations, l*a*b conversion, classification using Random Forest classifier are the main steps involved in their system.

Authors deployed a model [6, 14] which involves image pre-processing, segmentation using Lloyd's or k-means algorithm, features extraction using Grey Level Co-occurrence Matrix (GLCM), classifying using multiclass SVM. Throughout their project, they convert RGB to greyscale, remove noise, compute the centroid, and Kernal functions.

P. Kulkarni et al. developed a machine learning model for plant diseases detection. The model is consisting of Gaussian filter, Otsu's thresholding algorithm, morphological transform, bitwise AND operation, HSV (Hue, Saturation and Value) colour space conversion and Random Forest classifier. Their system is able to detect 20 different diseases with 93% accuracy [7].

Nishant Shelar et al. demonstrated the application of CNN to detect plant diseases. The PlantVillage dataset is used. Pre-processing and augmentation are done using image-data generator API by Keras. The Convolutional Neural Network (VGG19) consisted of a convolutional layer, a pooling layer, and fully connected layer. Their model accuracy was 95.6% [8].

Menukaewjinda et al. [9] suggested a Back Propagation Neural Network (BPNN) technique for effective grape leaf colour extraction. They evaluated Modified Self Organizing Feature Map (MSOFM) and genetic algorithms (GA) and discovered that these systems provide automated parameter modification for grape leaf disease colour extraction. Support vector machine has also been proven to be highly promising for efficiently classifying leaf diseases.

N. Chourasia and colleagues investigate the use of convolutional neural networks for illness identification. The architectures AlexNet and GoogLeNet, as well as the Transfer Learning and Training from Scratch training mechanisms, are used throughout the project [10].

Sahu and Pandey in their research offers a novel Hybrid Random Forest Multiclass SVM (HRF-MCSVM) model for detecting plant foliar diseases using automated diagnosis from leaf pictures. By adjusting MSVM hyperparameters, the TDO optimizer improves true positive rates. HRF-MCSVM outperforms previous approaches in terms of accuracy (97.9%), specificity, sensitivity, and recall. Future aims include broadening the method to identify other illnesses in a sustainable manner [11].

In [12], a review has been suggested which is useful for researchers looking for effective Machine Learning (ML) and Deep Learning (DL) based classifiers for leaf disease identification, as measured by metrics such as F1 score, precision, and accuracy. Javidan et al. [13] proposes a new technique using SVM with a linear kernel. Using Principal Component analysis (PCA) the outperformed deep learning methods has accuracies of 86.82% and 94.05%, respectively, demonstrating efficient and accurate detection of grape leaf diseases.

Kiran R. Gavhale, and U. Gawande suggested a technique using Back Propagation Neural Network (BPNN), SVM, K-nearest neighbour, and Stochastic Gradient Descent with Momentum (SGDM), which are applied to analyse healthy and diseased plant leaves [15]. A hybrid approach utilizing image processing and machine learning techniques is employed for disease detection in crops. Initially, a dataset from Kaggle containing over 12,949 images of healthy and unhealthy crop leaves is used. Segmentation and feature extraction via grey level co-occurrence matrix (GLCM) are followed by SVM-based recognition achieving 80% accuracy. Implementing a Convolutional Neural Network (CNN) enhances accuracy to 97.71% [16].

Authors using Convolutional Neural Networks (CNNs) to address agricultural plant disease challenges. A ResNet34 model is explored in this study, achieving 97.2% accuracy and over 96.5% F1 score on a dataset of 8,685 controlled environment leaf images, detecting 7 diseases and distinguishing healthy tissue. The success underscores CNNs' potential in aiding smallholder farmers via AI-driven solutions for accurate plant disease classification [17].

Authors aiming to mitigate tomato crop losses in rural India, this project employs image processing techniques across four stages—pre-processing, leaf segmentation, feature extraction, and classification—to identify diseases affecting tomato plant leaves. The k-nearest neighbours (KNN) algorithm, a supervised machine learning approach, is utilized for classification and regression tasks. The system offers quick and dependable disease detection based on colour, boundary, and texture, guiding farmers towards effective treatment, and addressing crucial challenges in Indian agriculture [18].

After parameter fine-tuning, a convolutional network achieved an 88% accuracy when trained solely on original images, with individual class testing and comparison to other results. In this paper authors introduces a novel approach using deep learning for automatic plant disease classification and detection from leaf images, detailing data collection, preprocessing, augmentation, and CNN training. The method lacks comparisons with similar techniques but demonstrates promise in disease recognition, exemplified by a test image of a tomato leaf with Septoria leaf spot [19].

K. Pawar et al. [20] in their article described surveys of diverse image processing techniques employed for early leaf disease detection and classification, often utilizing artificial neural networks (ANN), Support Vector Machine (SVM), and other classifiers to enhance recognition. The proposed approach includes fuzzification in a Fruit Detection System, employing fuzzy curves and surfaces for rapid feature identification and effective pattern recognition. K-means clustering excels in segmentation, while SVM proves optimal for accurate classification via high-dimensional feature space mapping; for low-quality images, the Intent Search Technique improves image quality effectively.

G. Singh et al. [21] explores various CNN models (VGG16, Inception Net, ResNet, NasNet, MobileNet, VGG19) for effective leaf disease image classification, focusing on potato plants. The goal is a rapid, accurate, and automated system aiding disease identification and classification, utilizing transfer learning. The developed solution achieves impressive 99.62% classification accuracy on test datasets, providing farmers with early disease detection to boost yields, outperforming existing techniques.

Kolli et al. [22] proposed a model achieves a 94.87% accuracy in detecting plant diseases using OpenCV image processing, allowing deployment on IoT devices, phones, drones, and cameras, offering large-scale disease control. A. Taslim et al. [23] proposes a CNN-based leaf identification system to classify five Malaysian leaf types using deep learning. Utilizing ResNet-50 architecture, the network is trained on mobile-captured leaf images. Through multiple steps including preprocessing, feature extraction, plant identification, and testing, the MATLAB-based system achieves over 98% accuracy across different training sets and successfully produces an interface using MATLAB app designer.

N. Vijay [24] Comparing Convolutional Neural Networks (CNN) and K-nearest Neighbours (KNN) for tomato leaf disease detection, the study employs metrics like Accuracy, Precision, Recall, and F1-Score, revealing CNN's superiority in all metrics. Local Interpretable Model-agnostic Explanations (LIME) technique is applied for model explain ability. Despite user feedback indicating distrust in AI and XAI models for disease detection, farmers provide suggestions for potential improvements based on their experience.

Debasish Das et al. proposes an automated leaf disease detection model to address this issue, aiding early disease identification for increased production. Utilizing various feature extraction techniques, Support Vector Machine (SVM) stands out among classifiers (Random Forest, Logistic Regression), demonstrating its potential for real-world implementation in disease identification [25].

Kowshik B. et al [26] proposed a system employs Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN) algorithms to detect crop diseases early through regular field monitoring. Machine learning trains the model, guiding disease management decisions. The review highlights various strategies for automated crop disease detection, including an image segmentation algorithm, demonstrating promising results across multiple plant species. The approach emphasizes early detection using Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN) for enhanced disease recognition rates.

Joen at al. [27] introduces a novel leaf classification approach utilizing CNN models and modifying GoogleNet's network depth. The method achieves over 94% recognition accuracy for leaves, even when damaged up to 30%. Wani et al. [28] study focuses on extracting various leaf features for plant classification, training the classifier to accurately identify plant classes. The Convolutional Neural Networks (CNN) model aims to achieve efficient and effective leaf classification with reduced computational complexity. ResNet-50 a pre trained CNN that has been used. Total 180 images for testing and more than 177 images have been classified correctly, the recognition rate is 98.33%.

Authors introduces a CNN-based leaf identification system that distinguishes five Malaysian leaf types (acacia, papaya, cherry, mango, rambutan). Utilizing mobile-captured leaf images, the Convolutional Neural

Networks (CNN), specifically ResNet-50, undergoes image pre-processing, feature extraction, identification, matching, and testing steps, with MATLAB facilitating results extraction. Testing includes various image types. The MATLAB app designer creates interfaces for the system, achieving over 98% accuracy across different training sets for the five leaf classes, confirming the successful implementation of the recognition process [29].

Yadav et al. [30] investigates the factors influencing network performance, addressing challenges in plant species identification from field observations, which traditionally requires botanical specialists. The Convolutional Neural Networks (CNN) model, significantly improving accuracy, achieves over 88.22% recognition accuracy for leaves from the Flavia leaf set, demonstrating its effectiveness in leaf recognition.

CHAPTER 3

METHODOLOGY

Plant diseases are conditions that harm the growth, health and productivity of plants. In this article, digital image processing has been used to detect healthy leaves and four types of diseased plant leaves, those are:

1. Anthracnose
2. Alternaria Alternata
3. Cercospora Leaf Spot
4. Bacterial Blight

1. **Anthracnose:** Anthracnose is a plant disease caused by various species of fungi in the genus Colletotrichum and other related genera. It affects a wide range of plants, including trees, shrubs, fruits, and vegetables. Anthracnose is characterized by dark, sunken lesions or spots on leaves, stems, flowers, and fruit, which can eventually lead to defoliation, reduced crop yields, and in severe cases, plant death. Symptoms of anthracnose vary depending on the host plant but often include small, irregularly shaped lesions that are initially water-soaked and later turn dark brown or black. These lesions can merge to form larger areas of infected tissue. In some cases, the disease can cause premature leaf drop, dieback of twigs and branches, and fruit rot.

Managing anthracnose typically involves a combination of cultural practices and chemical treatments. These may include pruning and removing infected plant parts, promoting good air circulation around plants, avoiding overhead irrigation, using disease-resistant plant varieties, and applying fungicides when necessary.

2. **Alternaria Alternata:** Alternaria alternata is a fungal species within the Alternaria genus. It is a well-known plant pathogen and is responsible for causing a variety of diseases in plants, particularly in agricultural and horticultural settings. Infection by Alternaria alternata typically results in the development of dark, concentric rings or lesions on leaves, stems, fruits, and other plant parts. These lesions often have a characteristic "bull's-eye" appearance and can lead to leaf wilting, yellowing, and defoliation. On fruits, it can cause rotting and blemishes.

   Controlling Alternaria alternata often involves a combination of cultural, biological, and chemical strategies. These may include crop rotation, proper spacing between plants to improve air circulation, using disease-resistant plant varieties, applying fungicides when necessary, and practicing good sanitation by removing and disposing of infected plant debris.

3. **Cercospora Leaf Spot:** Cercospora leaf spot, also known as Cercospora leaf spot disease, is a common fungal disease that affects a wide range of plants, including many agricultural crops, ornamental plants, and trees. This disease is caused by various species of fungi in the genus Cercospora. Cercospora fungi are known for their ability to cause leaf spots and can be particularly damaging in conditions with high humidity and moderate temperatures. Symptoms of Cercospora leaf spot vary depending on the host plant, but they typically manifest as small, circular to irregularly shaped spots or lesions on the leaves. These spots

are often tan to gray with dark margins and a lighter center. Over time, the spots may enlarge and coalesce, leading to extensive leaf damage. Severe infections can result in premature defoliation and reduced crop yields.

Managing Cercospora leaf spot typically involves a combination of cultural, chemical, and biological control measures. These may include crop rotation, pruning and disposing of infected plant material, using disease-resistant plant varieties, applying fungicides when necessary, and optimizing irrigation practices to minimize leaf wetness. Fungicides can be effective in controlling Cercospora leaf spot, especially in commercial agriculture. Fungicide applications are often timed to coincide with periods of high disease pressure, and it's important to follow label instructions and rotate among different fungicide classes to prevent resistance development.

4. **Bacterial Blight:** Bacterial blight is a plant disease caused by various species of bacteria that can infect a wide range of plant species, leading to significant damage to crops, ornamental plants, and trees. The specific bacteria responsible for bacterial blight can vary depending on the host plant, but common culprits include Xanthomonas spp. and Pseudomonas spp. Symptoms of bacterial blight can vary, but they often include the appearance of water-soaked lesions on leaves, stems, and other plant parts. As the disease progresses, these lesions may turn brown or black and develop a slimy or oozy appearance. Infected leaves can wilt, curl, and die, leading to defoliation. In some cases, bacterial blight can also cause wilting and dieback of branches.

   Managing bacterial blight can be challenging, and control measures may vary depending on the specific host plant and bacterial species involved.

Copper-based fungicides are often used to manage bacterial blight, particularly in organic farming. These products can help reduce bacterial populations on plant surfaces. Planting disease-resistant varieties can be an effective long-term strategy for managing bacterial blight. Practices such as crop rotation, avoiding overhead irrigation, and improving air circulation around plants can help reduce disease pressure. Some beneficial microorganisms can antagonize the growth of pathogenic bacteria and may be used as part of an integrated pest management (IPM) strategy.

Here are some common types of plant diseases:

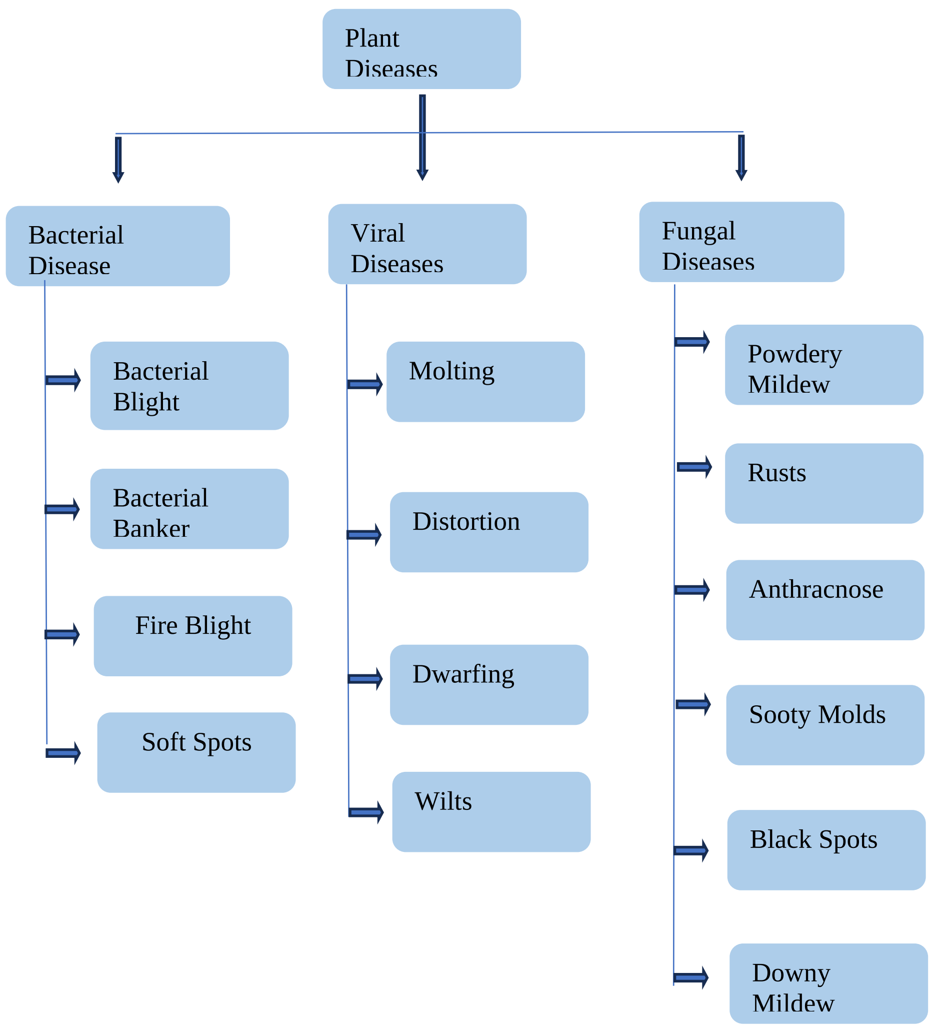

Figure 1- Various types of plant diseases

Source- Self-made using MS Word

3.1 K-MEANS CLUSTERING

K-Means clustering is an unsupervised machine learning algorithm used to partition a dataset into distinct, non-overlapping clusters. It assigns each data point to the cluster whose mean (centroid) is closest to it. The algorithm aims to minimize the variance within each cluster while maximizing the variance between clusters. K-Means is widely used for data exploration, customer segmentation, image compression, and more.

Algorithm Steps:

1. Initialization:
- Choose the number of clusters, K.
- Initialize K cluster centroids randomly or using a specific initialization technique.
2. Assignment:
- For each data point, calculate the distance (usually Euclidean distance) to each centroid.
- Assign the data point to the cluster corresponding to the nearest centroid.
3. Update Centroids:
- Recalculate the centroids of each cluster by taking the mean of all data points assigned to that cluster.
4. Repeat Assignment and Update:
- Repeat the assignment and centroid update steps iteratively until convergence.
- Convergence occurs when the centroids no longer change significantly or a maximum number of iterations is reached.

3.2 Gray-Level Co-Occurrence Matrix (GLCM)

Gray-Level Co-occurrence Matrix (GLCM) is a texture analysis method used in image processing and computer vision to characterize the spatial relationships between pixel intensities in an image. GLCM is often used in grayscale images, where each pixel contains an intensity value that represents its brightness. The goal of GLCM is to quantify the distribution patterns of various pixel intensities within a local neighbourhood of a picture. The standard form of the GLCM is a square matrix, where each entry (i, j) in the matrix denotes the frequency with which a pixel with intensity value i is followed by a pixel with intensity value j at a particular relative position and distance. This matrix can then be used to generate various statistical measures to extract different texture aspects that define the qualities of the image.

Following are a few typical texture properties derived using GLCM:

1. Contrast: Calculates how much an image's local intensity fluctuations vary.

2. Energy: Indicative of the homogeneity of the image, it represents the GLCM's total of squared elements.

3. Homogeneity: Evaluates how closely the elements in the GLCM are distributed to its diagonal.

4. Correlation: Determines the degree of linear dependence between the intensities in the GLCM.

5. Entropy: Describes the randomness or complexity of the texture. GLCM-based texture analysis has a wide range of applications, including image segmentation, object recognition, medical image analysis, remote sensing.

3.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. It is particularly effective for problems where the data points are not linearly separable, as SVM can handle both linear and non-linear decision boundaries. SVMs are widely used in various domains, including image classification, text classification, and bioinformatics.

SVM working principles:

1. Linear SVM: In the case of linearly separable data, the hyperplane is determined by finding the weights and bias that define it. The goal is to find the weights that minimize a certain cost function while still correctly classifying the data.
2. Soft Margin SVM: Real-world data is often noisy and not perfectly separable. In such cases, SVM allows for some misclassification by introducing a "soft margin." This means that some data points can fall within the margin or even on the wrong side of the hyperplane, but a trade-off between margin width and misclassification is considered.
3. Kernel Functions: Commonly used kernel functions include the linear kernel, polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel. The choice of kernel function depends on the data and the complexity of the decision boundary.
4. Training and Optimization: Training an SVM involves finding the hyperplane parameters that minimize the cost function while considering the margins and misclassification. This is typically solved as a convex optimization problem. SVMs have several advantages, such as their ability to handle high-dimensional data, their effectiveness in both linear and non-

linear scenarios, and their ability to generalize well to new, unseen data. However, they can be sensitive to the choice of hyperparameters like the kernel type and regularization parameter.

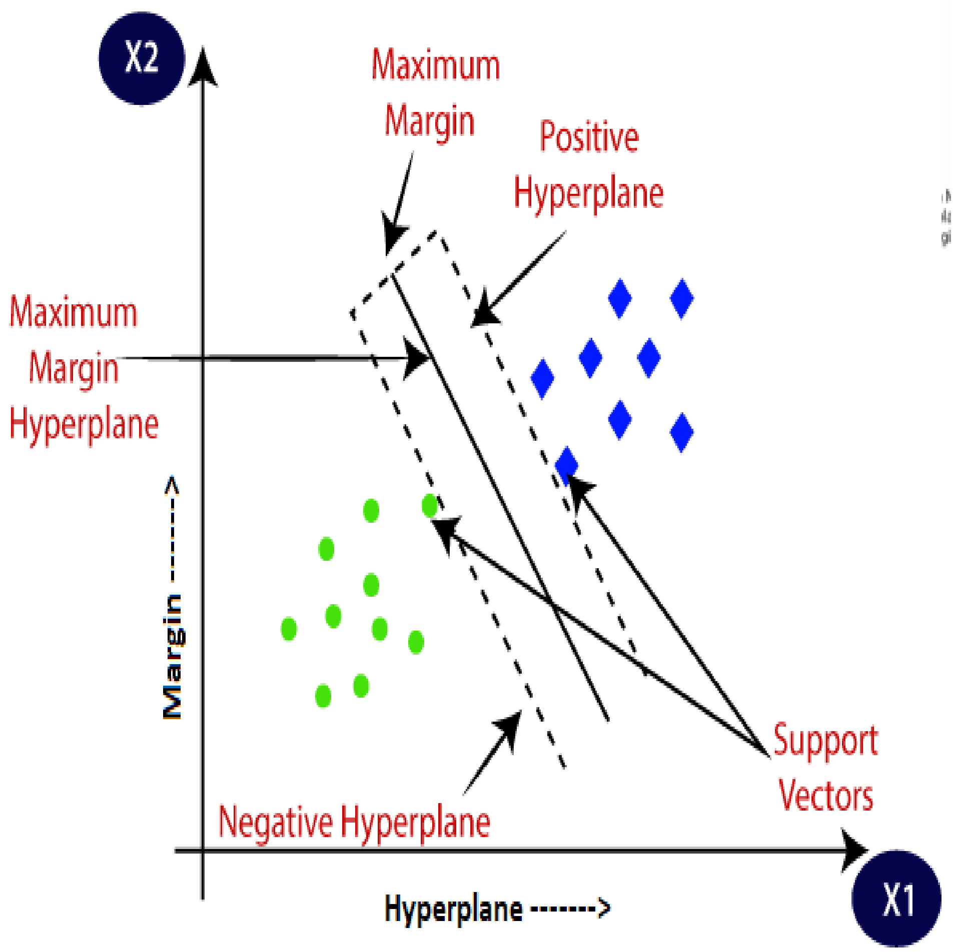

Figure 2: Classification using Multiclass SVM

Source- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

CHAPTER 4

PROPOSED ALGORITHM

The basic steps of the proposed algorithm using image processing is image acquisition, pre-processing, segmentation, feature extraction and classification. Total 120 leaf images (100 diseased leaves and 20 healthy leaves) used as dataset. At first those data images are pre-processed i.e., image resizing, cropping, noise removal, image smoothing, image enhancement is done. Then pre-processed images are segmented based on the Region of Interest (ROI) using k-means clustering then all the features including Contrast, Correlation, Energy, Homogeneity, Mean, Standard Deviation, Euclidean Distance, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness are extracted using Grey Level Co-occurrence Matrix (GLCM). The colour feature and morphological features gives the best results for plant disease classification. Finally, the input image is classified with the help of Support Vector Machine classifier based of the extracted features.

4.1. SYSTEM ARCHITECTURE

The system architecture for plant leaf disease detection using Support Vector Machines (SVM) provides a structured framework for implementing a robust and accurate disease detection system. The quantity and quality of the training data, and the fine-tuning of the SVM parameters are all crucial to the system's performance.

The architecture of suggested algorithm step-by-step procedure is depicted below.

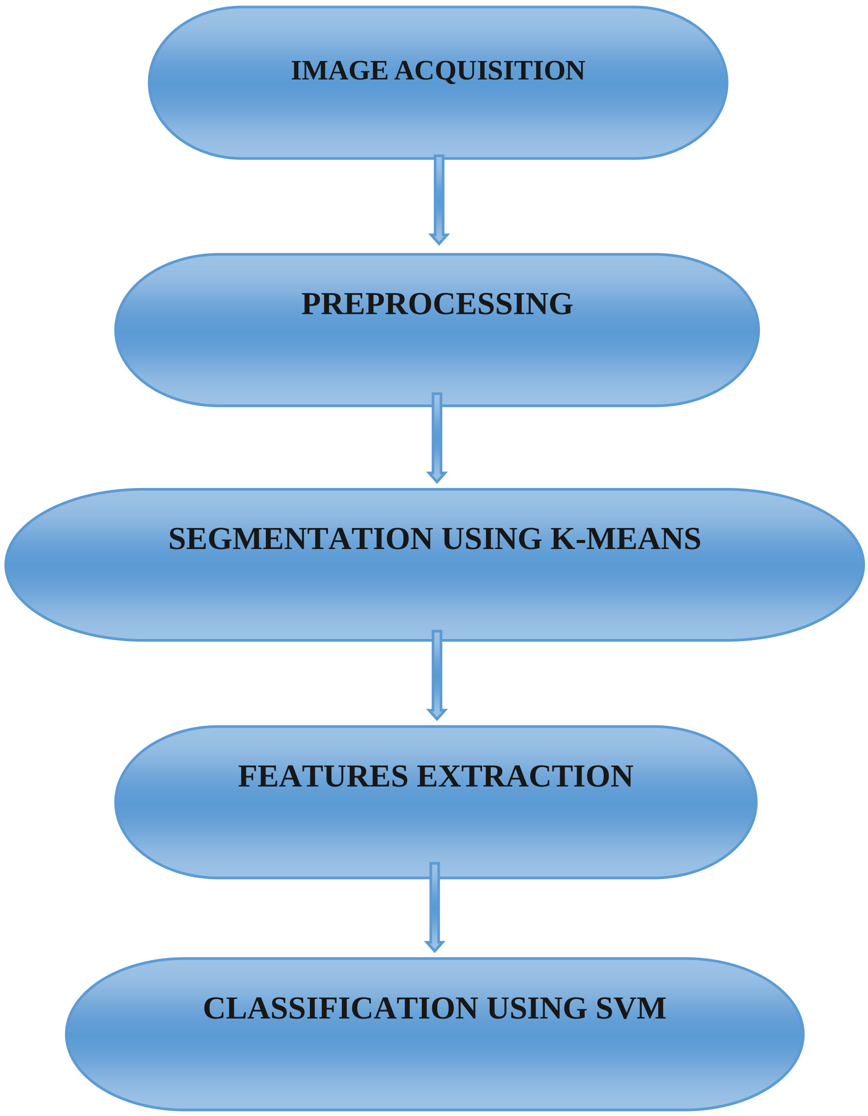

Figure 3- Architecture of Proposed Algorithm

Source- Self-made using MS Word

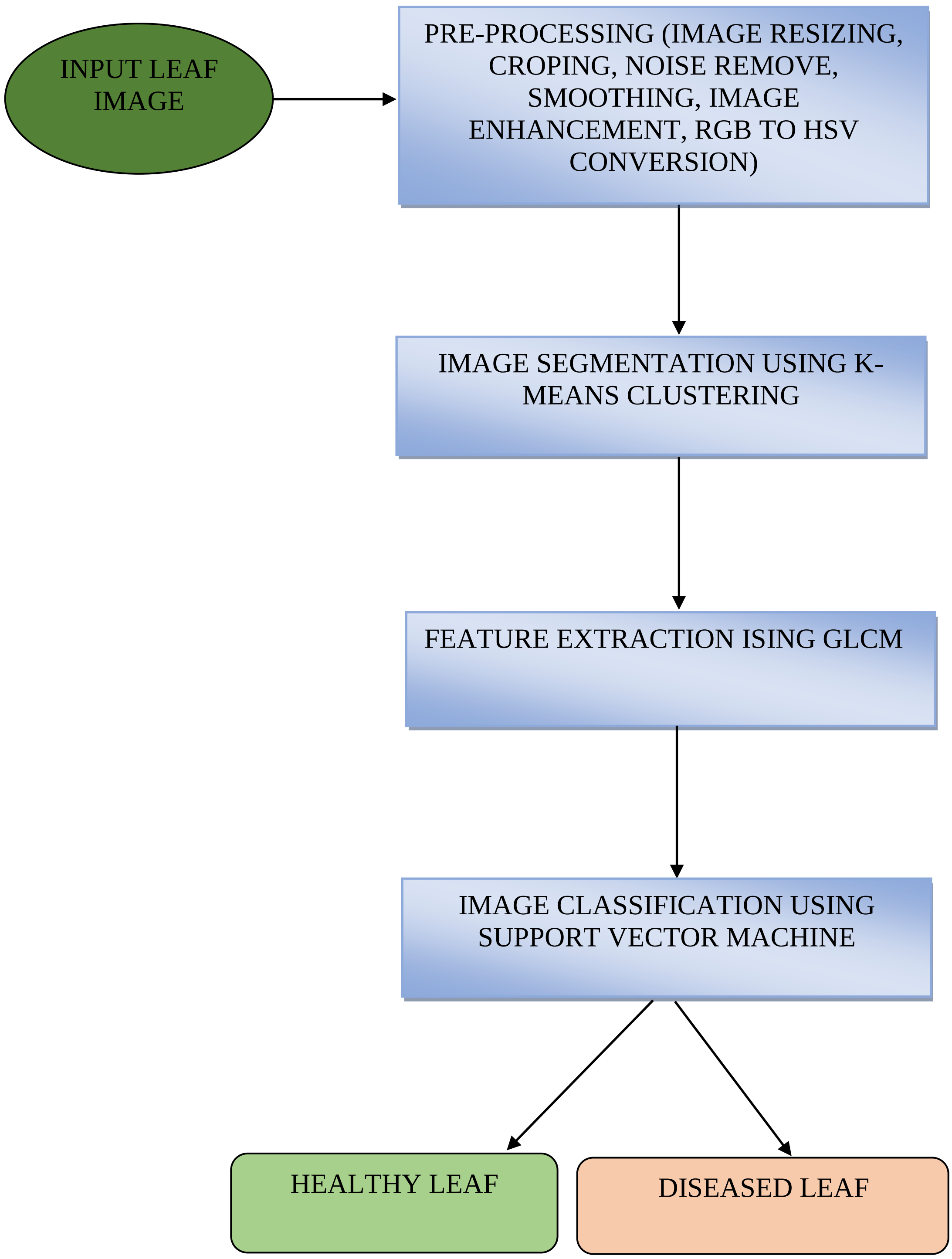

Figure 4- Block Diagram of Proposed Algorithm

Source- Self-made using MS Word

4.2. IMAGE ACQUISITION

Image acquisition refers to the process of capturing visual information from the real world and converting it into a digital format that can be processed, stored, and manipulated using computers or other electronic devices. In this project firstly, collect the dataset or capture images using digital camera and ready them to feed the proposed algorithm. Resized all the images to 256x256 and the dataset is given as the input to the acquisition process.

Here are some essential ideas regarding photo acquisition:

1. Sensors and cameras: Depending on the particular application and needs, many types of sensors and cameras are utilized to gather images. These include satellite sensors, digital cameras, smartphone cameras, and imaging equipment for medical use.

2. Resolution: Resolution, which refers to the number of pixels (picture components) that make up an image, is frequently used to define the quality of a captured image. Image acquisition is the process of obtaining an image. Images with higher resolutions have more detail.

3. Colour Information: Colour information can be captured during image collection using a variety of methods, including the use of several sensors to record the red, green, and blue (RGB) light components or the use of specialized sensors for certain wavelengths.

4. Image Formats: Depending on the application and the required amount of compression and quality, acquired photos are typically stored in a variety of file formats such JPEG, PNG, TIFF, or RAW.

4.3. IMAGE PRE-PROCESSING

Pre-processing refers to a set of techniques and operations applied to raw images to improve their quality, enhance relevant information, and make them suitable for further analysis. Pre-processing aims to remove noise, contrast enhancement, colour correction, RGB to grey scale conversion morphological operations to ensure accurate and effective image analysis using various MATLAB code.

Here is a general outline of preprocessing steps for plant disease detection:

1. Image Resizing: Resize the image to a specific resolution, making it easier to process and reducing computational requirements. Resizing is especially important when dealing with images of varying sizes.

2. Image Enhancement: Apply techniques to improve the visual quality of images, such as adjusting contrast, brightness, and sharpness. Histogram equalization and contrast stretching are examples of enhancement techniques.

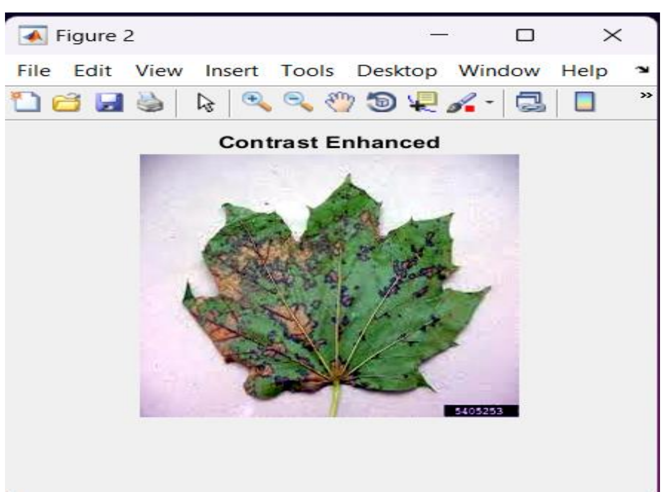

Figure 5: Pre-processed leaf image

Source- Output Image has been obtained using MATLAB R2018a Version

3. Noise Reduction: Apply filters to reduce noise and artifacts that might have been introduced during image acquisition or transmission. Common noise reduction filters include Gaussian, median, and bilateral filters.

4. Colour Space Conversion: Convert the image from one colour space to another (e.g., RGB to grayscale, RGB to HSI). Different colour spaces can highlight different image features and simplify certain types of analyses.

RGB to grayscale conversion:

$$GreyScale = 0.299R + 0.587G + 0.114B \qquad \text{Eq. (1)}$$

RGB to HSI conversion:

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\left(\left[(R-G)^2+\frac{(R-B)(G-B)^{1/2}}{2}\right]\right)}\right\} \qquad \text{Eq. (2)}$$

$$H = \begin{cases}\theta & if\ B \leq G \\ 360-\theta & if\ B > G\end{cases} \qquad \text{Eq. (3)}$$

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \qquad \text{Eq. (4)}$$

$$I = \frac{1}{3}(R+G+B) \qquad \text{Eq. (5)}$$

5. Background Subtraction: Remove or reduce the background from the image to focus on the foreground objects. This is particularly useful in applications like object detection.

6. Image Thresholding: Convert a grayscale image to a binary image by applying a threshold. Pixels with values above the threshold are set to one colour, and those below are set to another, helping to segment objects from the background.

7. Image Smoothing: Apply filters to reduce high-frequency noise and fine details while preserving important structural information. Smoothing can be useful for removing small artifacts.

8. Edge Detection: Identify and highlight edges or boundaries in the image. Edge detection is often a precursor to more complex analysis tasks.

9. Morphological Operations: Use operations like dilation and erosion to modify the shape and structure of objects in the image. These operations are particularly useful for processing binary images.

10. Data Augmentation: Generate variations of the image by applying transformations like rotation, scaling, and flipping. Data augmentation is common in machine learning to increase the diversity of the training dataset.

11. Resampling: Adjust the pixel spacing of the image, which is particularly important in medical imaging to ensure accurate measurements and consistent spatial resolution.

4.4. SEGMENTATION

Image segmentation is a fundamental concept in digital image processing that involves dividing an image into meaningful and semantically coherent segments. It helps to isolate and identify the regions of interest (ROI) within the leaf image, allowing for more accurate and targeted analysis. Segmentation can be done using various methods However, k-means clustering is a popular tool for detecting plant diseases. K-means clustering is an unsupervised learning approach that divides data into K clusters based on similarities. Each pixel's colour information is considered as a data point, and the algorithm partitions the pixels into K clusters, aiming to minimize the within-cluster variance. Using l*a*b colour space conversion image can be regenerated. Here is the step-by-step process how image segmentation using the K-means algorithm perform:

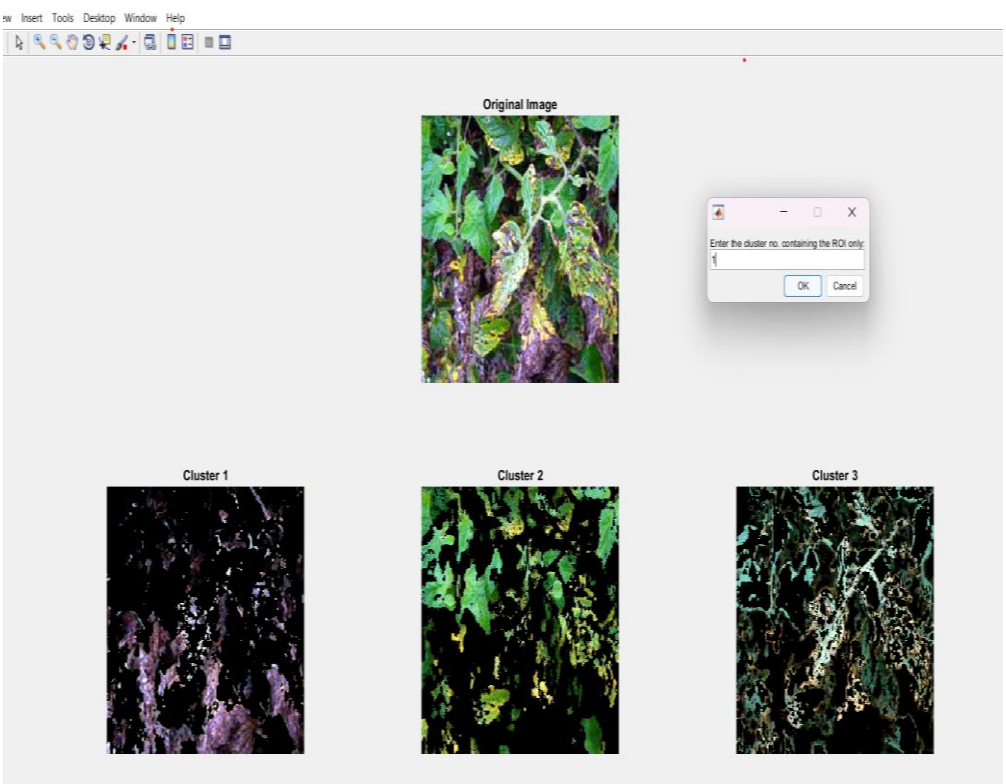

Figure 6: Three different clusters with Region of Interest

Source- Output Image has been obtained using MATLAB R2018a Version

1. Select the Number of Clusters (K): Determine how many segments or clusters you want to divide the image into. This choice depends on the specific characteristics of the image and the objects you want to segment.

2. Vectorization: Represent each pixel in the image as a data point in a high-dimensional space. For colour images, this typically involves representing each pixel by its colour values (e.g., RGB values).

3. Initialization: Randomly select K initial cluster centers (centroids). These centroids will serve as the starting points for the algorithm.

4. Assign Pixels to Clusters: For each pixel in the image, calculate its distance to each centroid and assign the pixel to the cluster with the nearest centroid. This step creates initial clusters based on proximity to centroids.

5. Update Cluster Centers: Calculate the mean of all pixels assigned to each cluster and update the cluster centroids accordingly.

6. Repeat Assigning and Updating: Repeat the assignment and centroid update steps iteratively until convergence. Convergence occurs when the centroids no longer change significantly or a predetermined number of iterations is reached.

7. Image Reconstruction: Once convergence is reached, assign a representative colour or intensity value to each cluster. You can use the centroid values as these representative values.

4.5. FEATURES EXTRACTION

Feature extraction in image processing is the process of capturing relevant and distinctive information from images to represent them in a more compact and meaningful form. To detect the plant disease colour, texture and morphological features are considered. Feature extraction using GLCM is a widely used method to characterize the texture patterns in an image. GLCM calculates the frequency of pixel intensity value pairs at a specific distance and orientation within the image.

For each pixel in the image, calculate the GLCM using the defined neighbourhood window and offset. To calculate the GLCM, follow these steps:

a. Select a pixel (p1) in the image.
b. Identify the pixel (p2) that is at the specified offset from p1.
c. Record the pair of intensity values (grey levels) at P1 and p2.
d. Move on to the next pixel and repeat the process until all pixels in the image have been considered.
e. Accumulate the frequency of each intensity value pair in a matrix, which is the GLCM.

Once the GLCM is normalized, you can compute various statistics as features from the GLCM. Common statistics include: Contrast, Correlation, Energy, Homogeneity, Mean, Standard Deviation, Euclidean Distance, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness.

Formula of Contrast:

$$Contrast = \sum_{i,j=0}^{N-1} (P_{ij})(i-j)^2$$

Eq. (6)

Formula of Energy:

$$Energy = \sum_{i,j=0}^{N-1} (P_{ij})^2$$

Eq. (7)

Formula of Homogeneity:

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{(P_{ij})^2}{[1 + (i-j)^2]}$$

Eq. (8)

Formula of Correlation:

$$Correlation = \sum_{i,j=0}^{N-1} P_{ij} \left(\frac{(i-\mu)(j-\mu)}{\sigma^2} \right) \ (\mu \text{ is mean value and } \sigma \text{ is variance})$$

Eq. (9)

Formula of Mean value:

$$\mu = \frac{1}{n} \sum_{1}^{n} x_i$$

Eq. (10)

Formula of Standard Deviation:

$$\sigma^2 = \frac{1}{n} \sum_{1}^{n} x_i - \mu^2$$

Eq. (11)

Formula of Euclidean Distance:

$$Euclidean\ Distance = \sqrt{\sum_{i=1}^{i=n} (x_i - y_i)^2}$$

Eq. (12)

Here is an overview of feature extraction in image processing:

4.5.1. TYPES OF FEATURES

Different types of features can be extracted from images, depending on the specific task and the nature of the data. Common types of features include:

1. **Texture Features:** Describe the patterns and variations in texture, such as coarseness, smoothness, and regularity.
2. **Colour Features:** Capture colour distribution and properties, including colour histograms, colour moments, and colour spaces.
3. **Shape Features:** Characterize the shape and geometry of objects, including aspects like perimeter, area, and compactness.
4. **Intensity Features:** Represent the intensity or brightness of pixels in an image.
5. **Statistical Features:** Compute statistical properties of image regions, such as mean, variance, and entropy.
6. **Frequency Domain Features:** Extract information related to the frequency content of the image, often through techniques like Fourier Transform or Wavelet Transform.

4.5.2. FEATURE EXTRACTION PROCESS

1. **Image Preprocessing:** Clean, enhance, and normalize the image using preprocessing techniques such as resizing, denoising, and contrast adjustment.
2. **Region of Interest (ROI) Selection:** Focus on relevant regions or objects within the image to extract features from.
3. **Feature Computation:** Compute the desired features using mathematical formulas or algorithms specific to each type of feature.

4.6. CLASSIFICATION:

SVM classification is a supervised learning technique used for binary and multi-class classification tasks. The pre-processed data used as input into the SVM model for training. The SVM method will determine the hyperplane that optimally separates the samples of distinct classes while maximizing the difference between the two classes. Multiclass SVM has some hyperparameters that need to be set before training, such as the kernel type, regularization parameter (C), and kernel-specific parameters. Once the model is trained and tested, it can be used to predict the classes. Here is how SVM is used for image classification:

1. SVM Training:

Kernel Selection: Choose a suitable kernel function. Common kernels include linear, polynomial, and radial basis function (RBF) kernels.

Training: Feed the training feature matrix and their corresponding labels into the SVM algorithm. The algorithm learns a decision boundary that best separates the data into different classes while maximizing the margin between them.

2. Model Evaluation:

Testing: Use the trained SVM to predict the class labels of the images in the testing set.

Performance Metrics: Evaluate the SVM's performance using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

3. Hyperparameter Tuning:

Regularization Parameter (C): Adjust the C parameter to control the trade-off between maximizing the margin and minimizing the classification error on the training set.

Kernel Parameters: If using kernels like polynomial or RBF, adjust the kernel-specific parameters.

4. Prediction:

Prediction: Use the trained SVM model to predict the class labels of new, unseen images.

Advantages of SVM for image classification:

- They are effective in handling high-dimensional data, which is common in image data.
- SVMs find a hyperplane (decision boundary) that maximizes the margin between classes, leading to good generalization to unseen data.
- With appropriate kernel functions, SVMs can capture complex relationships between features.

CHAPTER 5

EXPERIMENTATIONS AND RESULTS

The quality of the dataset, feature selection, hyperparameter tuning, and the difficulty of the classification problem all influence the outcomes of plant leaf disease detection using multi-Class SVM. GLCM and k-means are particularly effective in detecting plant diseases. SVM can diagnose plant leaf diseases with excellent accuracy, particularly when the dataset is well-balanced and the features are reflective of disease traits. However, the performance is heavily influenced by the quality of the extracted features as well as the hyperparameters specified.

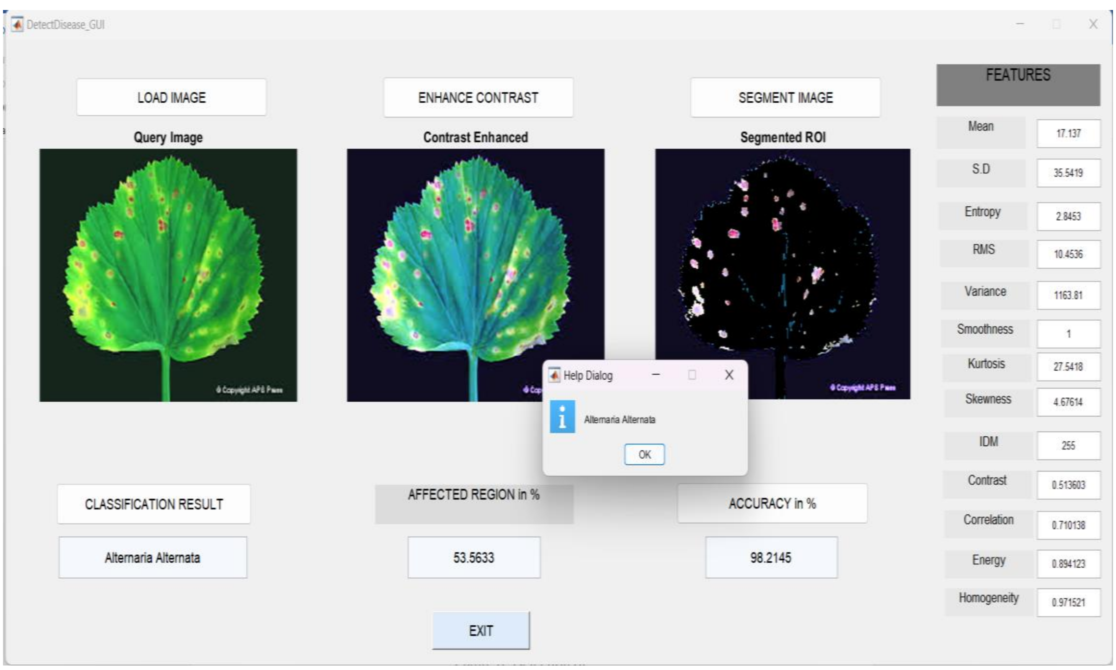

Figure 7: Detection of Alternaria Alternata

Source- Output image has been obtained using MATLAB R2018a Version

Figure 7 is showing the classification result of Alternaria Alternata disease. Affected region of the leaf image is 53.56% and the calculated accuracy value is 98.21%.

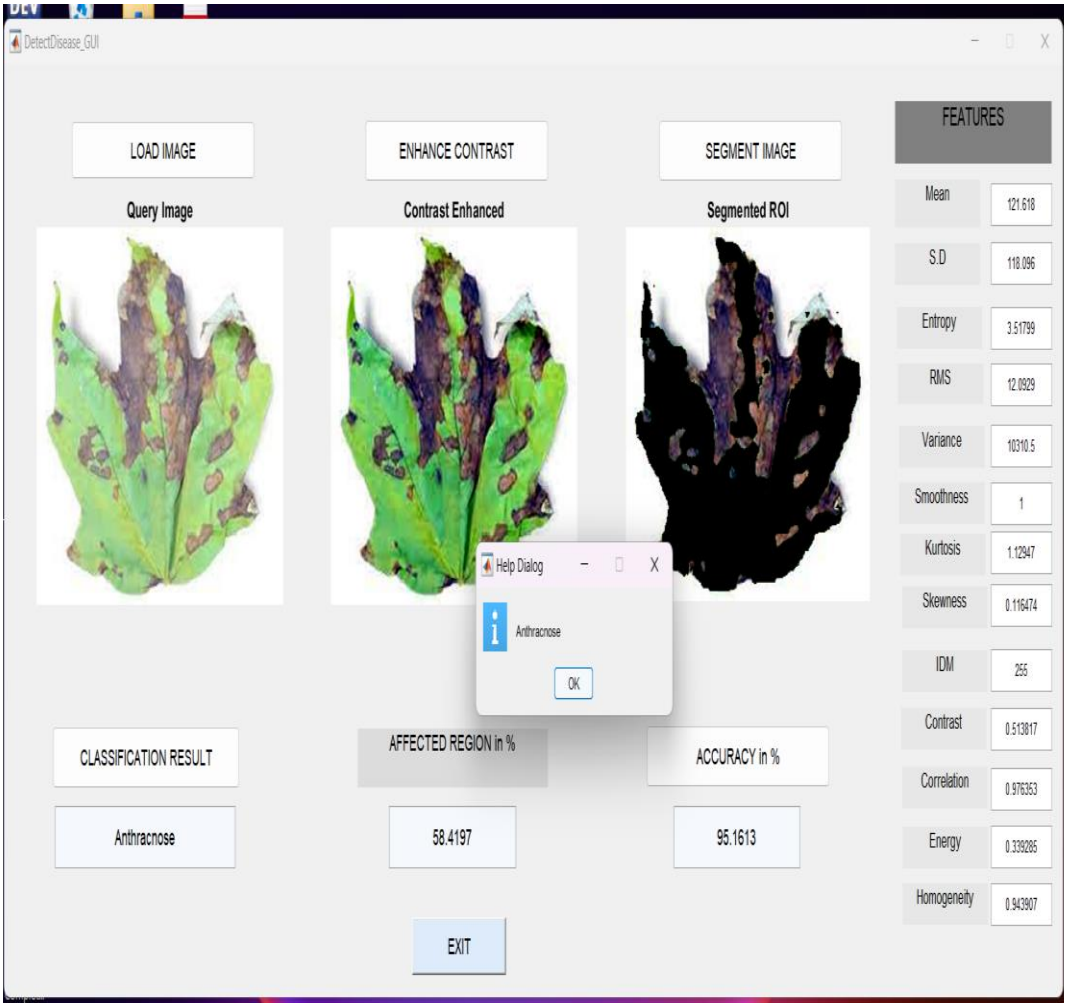

Figure 8: Detection of Anthracnose

Source- Output image has been obtained using MATLAB R2018a Version

Figure 8 is showing the classification result of anthracnose disease. Affected region of the leaf image is 58.41% and the calculated accuracy value is 95.16%.

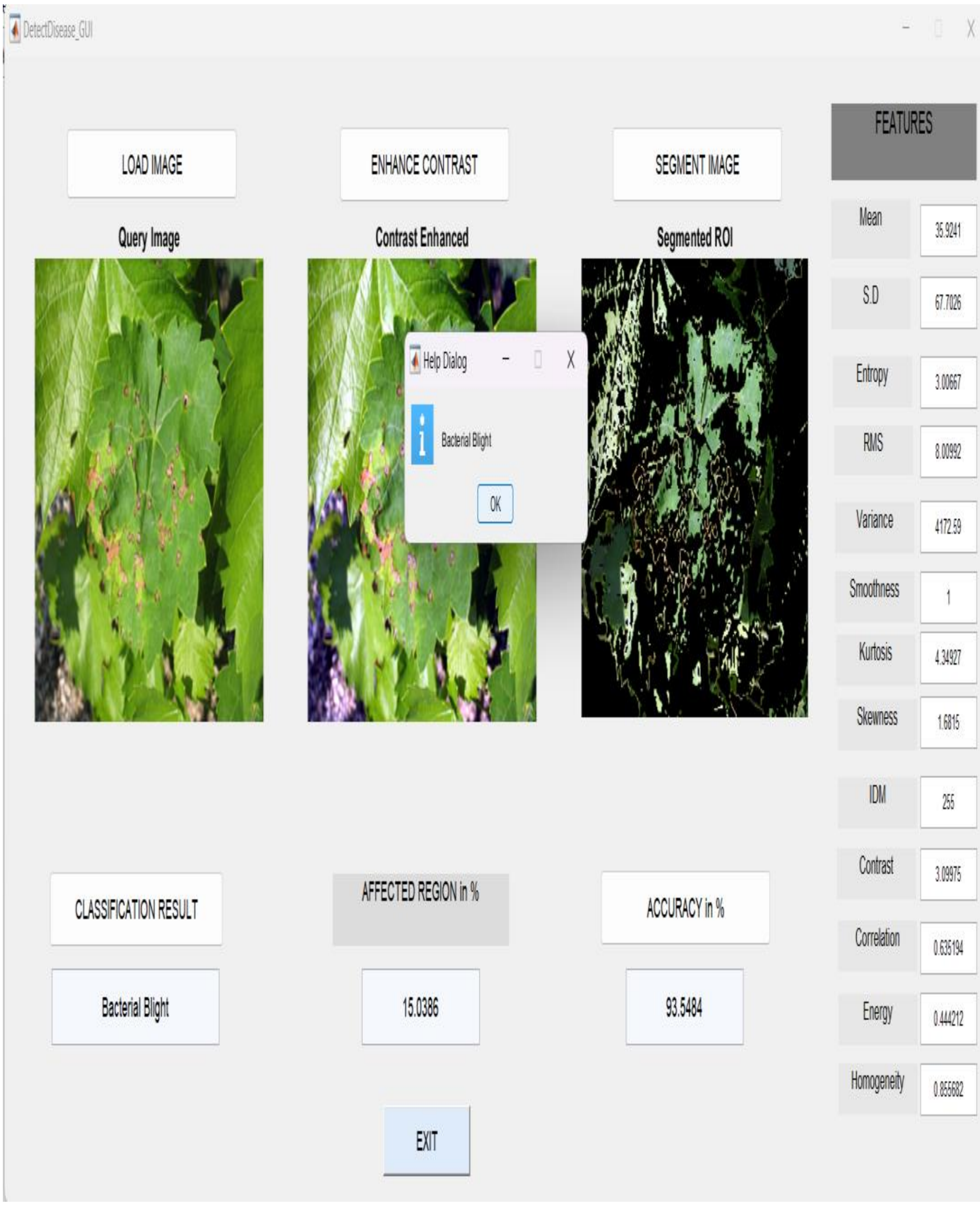

Figure 9: Detection of Bacterial Blight

Source- Output Image has been obtained using MATLAB R2018a Version

Figure 9 is showing the classification result of a bacterial blight diseased leaf. Affected region of the leaf image is 15.03% and the calculated accuracy value is 93.54%.

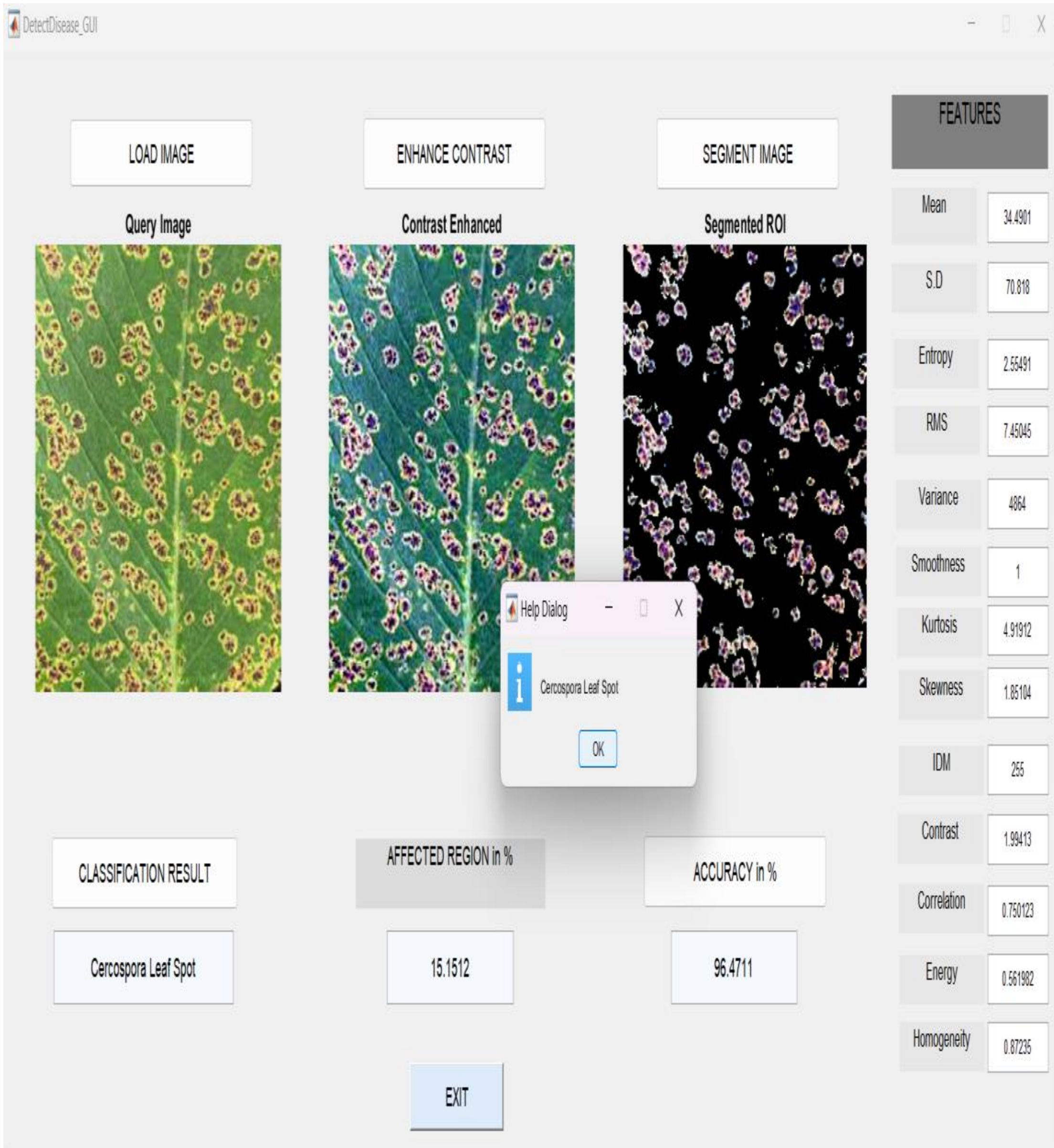

Figure 10: Detection of Cercospora Leaf Spot

Source- Output Image has been obtained using MATLAB R2018a Version

Figure 10 is showing the classification result of cercospora leaf spot disease. Affected region of the leaf image is 15.15% and the calculated accuracy value is 96.47%.

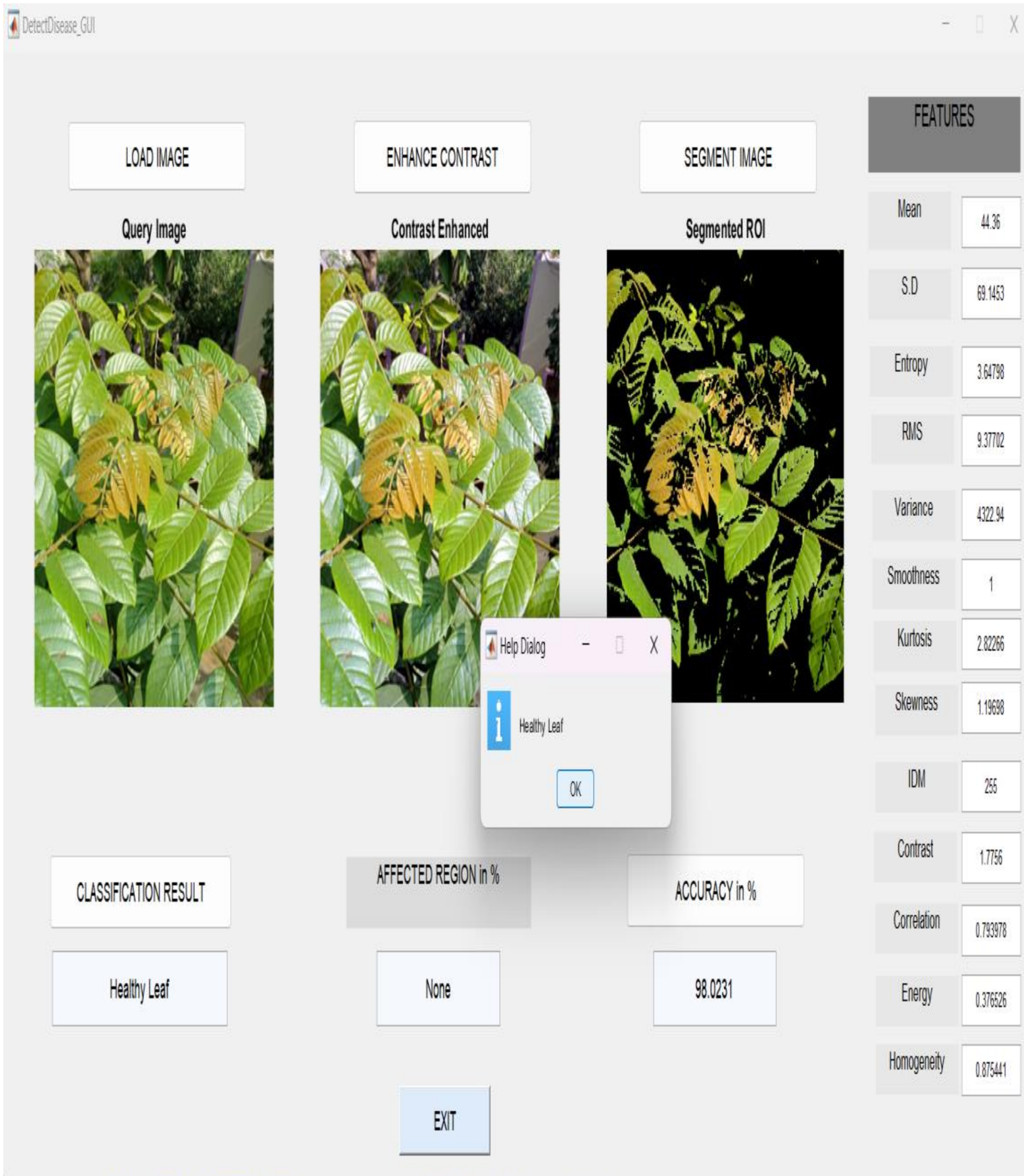

Figure 11: Detection of Healthy Leaf

Source- Output Image has been obtained using MATLAB R2018a Version

Figure 11 is showing the classification result of a healthy leaf disease. Affected region of the leaf image is none and the calculated accuracy value is 98.02%.

5.1. ACCURACY CALCULATION

In image processing, accuracy is often used to evaluate the performance of various algorithms, models, or techniques. The concept of accuracy is closely related to correctly classifying or identifying objects within an image. In the proposed algorithm accuracy is calculated after 100 iteration and each in iteration accuracy of Linear Kernel is calculated. The Maximum value of linear kernel of each iteration is accepted as final accuracy.

$$Accuracy = \frac{Number\ od\ correct\ predictions\ (A1)}{Total\ number\ of\ predicyions\ (A2)} \quad \text{Eq. (13)}$$

$$Accuracy\ Percentage = (Accuracy * 100) \quad \text{Eq. (14)}$$

Table 1- Sample of Iteration Values

Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	BoxConstrain-t	KernelScale
1	Accept	NaN	7.6896	NaN	NaN	261.08	0.0017674
2	Accept	0.19048	2.9826	NaN	0.19048	81.113	106.3
3	Accept	0.20635	0.16494	0.19048	0.19149	0.0078963	70.449
4	Accept	0.44444	6.5548	0.19048	0.19057	0.47716	0.19602
5	Best	0.15873	5.272	0.15873	0.15875	0.015947	0.6908
6	Accept	0.20635	0.3335	0.15873	0.15877	0.029447	70.402
7	Accept	0.20635	0.087331	0.15873	0.16747	0.0010057	77.331
8	Accept	0.20635	0.078066	0.15873	0.15876	0.0023132	90.268
9	Accept	0.20635	1.7761	0.15873	0.15876	702.35	997.84
10	Accept	0.39474	7.6168	0.15873	0.15874	0.0059934	0.014408
11	Accept	0.20635	0.083636	0.15873	0.15875	0.0010055	264.83
12	Accept	0.20635	1.4117	0.15873	0.15875	0.0010002	12.552
13	Accept	0.2381	6.6944	0.15873	0.15874	200.34	1.8695
14	Accept	0.20635	0.077965	0.15873	0.15876	0.0012976	162.7
15	Accept	0.20635	0.092142	0.15873	0.15876	0.39877	999.53
16	Accept	0.20635	0.33674	0.15873	0.15875	6.8915	956.23

```
|   17 | Accept |     0.20635 |     0.15255 |     0.15873 |     0.15877 |    0.0010524 |       27.075 |
|   18 | Accept |     0.20635 |    0.077224 |     0.15873 |     0.15877 |     0.018793 |       989.92 |
|   19 | Accept |     0.20635 |      2.1564 |     0.15873 |     0.15875 |     0.010147 |       3.2946 |
|   20 | Accept |     0.20635 |    0.077817 |     0.15873 |     0.15876 |    0.0011543 |       474.49 |
|   21 | Accept |     0.22222 |      5.9124 |     0.15873 |     0.15878 |     0.001031 |       0.8582 |
|   22 | Accept |     0.20635 |     0.82055 |     0.15873 |     0.15878 |      0.98529 |       161.88 |
|   23 | Accept |     0.20635 |      1.0058 |     0.15873 |     0.15878 |       67.852 |       975.07 |
|   24 | Accept |     0.22222 |      6.4292 |     0.15873 |     0.15878 |       948.94 |       45.197 |
|   25 | Accept |         NaN |      7.3912 |     0.15873 |     0.15878 |       979.63 |    0.0048103 |
|   26 | Accept |      0.1746 |      4.8282 |     0.15873 |      0.1588 |    0.0086105 |      0.63872 |
|   27 | Accept |     0.20635 |     0.10993 |     0.15873 |     0.15881 |     0.093444 |       395.29 |
|   28 | Accept |     0.20635 |      1.9025 |     0.15873 |     0.15877 |       10.035 |       117.13 |
|   29 | Accept |      0.1746 |         6.2 |     0.15873 |     0.15888 |      0.25714 |       1.0934 |
|   30 | Accept |     0.25397 |      6.6941 |     0.15873 |      0.1589 |       822.66 |      0.68787 |
____________________________________________________________________________

Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 113.9044 seconds.
Total objective function evaluation time: 85.0101
Best observed feasible point:
    BoxConstraint    KernelScale
    _____________    ___________

      0.015947         0.6908

Observed objective function value = 0.15873
Estimated objective function value = 0.1589
Function evaluation time = 5.272
Best estimated feasible point (according to models):
    BoxConstraint    KernelScale
    _____________    ___________

      0.015947         0.6908

Estimated objective function value = 0.1589
Estimated function evaluation time = 5.3856
ans =
    'Accuracy of Linear Kernel is: 98.2145%'
ans =
    'Accuracy of Linear Kernel with 100 iterations is: 98.2145%'
```

CHAPTER 6

COMPARATIVE ANALYSIS

Comparative analysis for plant disease detection using image processing involves evaluating and comparing different techniques, algorithms, or approaches to determine their effectiveness in accurately detecting and diagnosing plant diseases from images.

Table 2- Comparison of proposed algorithm and existing model:

AUTHOR	ALGORITHM	ACCURACY
Proposed Algorithm	K-Means Clustering, GLCM, SVM	98.2%
R. Meena Prakash et al. (2017) [1]	OTSU Model, SVM	90%
S. D. Khirade et al. (1015) [3]	ANN and back propagation	91.6%
A. Devaraj et al. (2019) [5]	Random Forest classifier	94.5%

A.Santhosh et al. (2019) [6]	multiclass SVM	97%
P. Kulkarni et al. (2021) [7]	HSV color space conversion and Random Forest classifier	93 %
N. Shelar et al. (2022) [8]	Convolutional Neural Network (VGG19)	95.6%
N. Chourasia et al. (2018) [10]	AlexNet and GoogLeNet, as well as the Transfer Learning	96.2%
S. K. Sahu et al. (2023) [11]	HRF-MCSVM	97.4%
S. M. Javidan et al. (2023) [13]	SVM, PCA	86.82% and 94.05%
Kiran R. Gavhale et al. (2023) [15]	BPNN, SVM, K-nearest neighbor, and SGDM	93.2%

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

In this chapter the conclusion has been given based on study and observations, which can be utilized for designing more effective models in future.

7.1. CONCLUSION

In this thesis work plant leaf disease identification based on image characteristics has been utilized by using **K-means clustering, Gray-Level Co-occurrence Matrix, and Support Vector Machine (SVM).** Segmentation is done to identify regions of interest using K-means clustering, extracting important features to define the diseased areas of leaves using GLCM and lastly training the dataset using multiclass SVM classifier to appropriately classify the diseased and healthy leaves. The suggested method has been achieved maximum accuracy of 98.2% in contrast with the earlier accuracy obtained in paper [1], which was 90%. The algorithm's performance is strongly dependent on the quality of image preprocessing, feature extraction, and the selection of SVM hyperparameters. Proper hyperparameter tuning and feature selection can have a considerable influence on the model's accuracy and resilience. After detection of plant leaf diseases all the data has been loaded into AccuracyData.mat and TrainingData.mat folder in MATLAB software for further analysis.

7.2. FUTURE SCOPE

Plant disease detection will probably take on a more interdisciplinary approach in the future, merging different technologies and fields of knowledge to produce practical and long-lasting solutions for guaranteeing food security around the world. Genomic Methods, DNA-based techniques such as polymerase chain reaction (PCR) and next-generation sequencing (NGS) can be used for disease detection. Artificial intelligence (AI) and Deep learning techniques, such as convolutional neural networks (CNNs), may extract intricate information from images, enhancing the precision in detection of disease. Identification of viral diseases of leaves has been kept as future work.

CHAPTER 8

REFERENCES

[1] M. Prakash, G. Saraswathy, G. Ramalakshmi, K. Mangaleswari and T. Kaviya, "Detection of leaf diseases and classification using digital image processing" 2017 International conference on innovations in information, embedded and communication systems. IEEE 2017.

[2] S. Raut and A. Fulsunge, "Plant disease detection in image processing using matlab" International journal of innovative research in science, engineering and technology volume 6, issue 6, june 2017.

[3] S. D. Khirade and S. B. Patil, "Plant disease detection using image processing" 2015 International conference on computing communication control and automation. IEEE 2015.

[4] V. Singh and A. K. Misra, "Detection of plant diseases using image segmentation and soft computing techniques" Information processing in agriculture 4 (2017) 41-49. Elsevier b.v. 2017.

[5] A. Devaraj, K. Rathan, S. Jaahnavi and K. Indira, "Identification od plant disease using image processing technique" International conference on communication and signal processing. IEEE 2019.

[6] A. Santhosh, K. Saravanakumar, P. Saravanakumar, S.vimalraj and R. Saranya, "Detection of leaf diseases and classifying them using multiclass SVM" International research journal of engineering and technology (irjet) volume 6, issue 3, march 2019.

[7] P. Kulkarni, A. Karwande, T. Kolhe, S. Kamble, A. Joshi and M. Wyawahare, “Plant disease detection using image processing and machine learning” IEEE 2021.

[8] N. Shelar, S. Shinde, S. Sawant, S. Dhumal, and K. Fakir, “Plant disease detection using cnn” itm web of conferences 44, icacc 2022.

[9] A. Meunkaewjinda, P. Kumsawat, K. Attakitmongcol and A. Srikaew, "Grape leaf disease detection from color imagery using hybrid intelligent system." 5th international conference on electrical engineering/electronics, computer, telecommunications and information technology, vol. 1, IEEE, 2008.

[10] N. Chourasia, S. Lanjewar, S. Daduriya, M. Mahalle and U. Giri, “Leaf disease detection using image processing” International journal of creative research thoughts, volume 6, issue 1, march 2018.

[11] S. K. Sahu and M. Pandey, “An optimal hybrid multiclass svm for plant leaf disease detection using spatial fuzzy c-means model” expert systems with applications 214, 2023.

[12] C. Sarkar, D. Gupta, U. Gupta and B. B. Hazarika, "Leaf disease detection using machine learning and deep learning: review and challenges." applied soft computing, 2023.

[13] S. M. Javidan, A. Banakar, K. A. Vakilian and Y. Ampatzidis, "Diagnosis of grape leaf diseases using automatic k-means clustering and machine learning." smart agricultural technology 3: 100081, 2023.

[14] S. U. Rahman, F. Alam, N. Ahmad and S. Arshad, "Image processing based system for the detection, identification and treatment of tomato leaf diseases." multimedia tools and applications 82, no. 6, 2023.

[15] Kiran R. Gavhale and U. Gawande, "An overview of the research on plant leaves disease detection using image processing techniques." iosr journal of computer engineering 16, no. 1, 2014.

[16] U. N. Fulari, R. K. Shastri and A. N. Fulari. "Leaf disease detection using machine learning." j. Seybold rep 1533 (2020): 9211.

[17] Emma Harte, “Plant disease detection using CNN” researchgate, september 2020.

[18] G. Geetha, S. Samundeswari, G. Saranya, K. Meenakshi and M. Nithya. "Plant leaf disease classification and detection system using machine learning." in journal of physics: conference series, vol. 1712, no. 1, p. 012012. Iop publishing, 2020.

[19] J. A. Pandian and V. D. Kumar, oana geman, mihaela hnatiuc, muhammad arif, and k. Kanchanadevi. "Plant disease detection using deep convolutional neural network." applied sciences 12, no. 14 (2022): 6982.

[20] K. Pawar, A. Taneja and Dr. Mukesh Dixit, “A review on fruit & leaf disease detection using various image processing techniques.” International journal for modern trends in science and technology, 8 pp. 127-133, february 2022.

[21] G. Singh, and Kuldeep Kumar Yogi. "Performance evaluation of plant leaf disease detection using deep learning models." archives of phytopathology and plant protection 56, no. 3 (2023): 209-233.

[22] Jahnavi Kolli, Dhara Mohana Vamsi, and V. M. Manikandan. "Plant disease detection using convolutional neural network." in 2021 IEEE bombay section signature conference (ibssc), pp. 1-6. Ieee, 2021.

[23] A. Taslim, S. Saon, M. Muladi and Nur Hidayat. "Plant leaf identification system using convolutional neural network." bulletin of electrical engineering and informatics 10, no. 6 : 3341-3352 , 2021.

[24] Nishanth Vijay "Detection of plant diseases in tomato leaves: with focus on providing explainability and evaluating user trust." diva-portal.org , 2021.

[25] Debasish Das, Mahinderpal Singh, Sarthak Swaroop Mohanty and S. Chakravarty. "Leaf disease detection using support vector machine." in 2020 International conference on communication and signal processing (iccsp), pp. 1036-1040. Ieee, 2020.

[26] Kowshik B., Savitha V., Nimosh Madhav, Karpagam G. and Sangeetha K., “Plant disease detection using deep learning” International research journal on advanced science hub, volume 03 issue 03s, march 2021

[27] Wang-su Jeon and Sang-Yong Rhee. "Plant leaf recognition using a convolution neural network." International journal of fuzzy logic and intelligent systems 17, no. 1 (2017): 26-34.

[28] Shahid Hussain Wani and Er. Shaminder Singh, "Plant leaf recognition using convolutional neural networks" ijrece vol. 7 issue 2 apr.-june 2019.

[29] Amiruzzaki Taslim, Sharifah Saon, Muladi Muladi and Wahyu Nur Hidayat. "Plant leaf identification system using convolutional neural network." bulletin of electrical engineering and informatics 10, no. 6 (2021): 3341-3352.

[30] Yadav, Akash, Smit Viradiya, Milan Vaghasiya and Uma Goradiya. "Automated plant identification with cnn." International research journal of engineering and technology (irjet) volume: 09 issue: 02, feb 2022.

APPENDIX

CODE SNIPPETS:

% Extract the individual component images.

```
rgb = im2double(rgb);
r = rgb(:, :, 1);
g = rgb(:, :, 2);
b = rgb(:, :, 3);
```

% Implement the conversion equations.

```
num = 0.5*((r - g) + (r - b));
den = sqrt((r - g).^2 + (r - b).*(g - b));
theta = acos(num./(den + eps));
H = theta;
H(b > g) = 2*pi - H(b > g);
H = H/(2*pi);
num = min(min(r, g), b);
den = r + g + b;
den(den == 0) = eps;
S = 1 - 3.* num./den;
H(S == 0) = 0;
I = (r + g + b)/3;
```

% Derive Statistics from GLCM

```
stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(seg_img);
Standard_Deviation = std2(seg_img);
Entropy = entropy(seg_img);
RMS = mean2(rms(seg_img));
Variance = mean2(var(double(seg_img)));
```

```
a = sum(double(seg_img(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(seg_img(:)));
Skewness = skewness(double(seg_img(:)));

%Affected_Area

Affected_Area = (A1/A2);
if Affected_Area < 0.1
   Affected_Area = Affected_Area+0.15;
end

% Visualize Results

if result == 0
   helpdlg('Anthracnose ');
   disp(' Anthracnose ');
elseif result == 1
   helpdlg(' Alternaria Alternata ');
   disp('Alternaria Alternata');
elseif result == 2
   helpdlg(' Bacterial Blight ');
   disp(' Bacterial Blight ');
elseif result == 3
   helpdlg(' Cercospora Leaf Spot ');
   disp('Cercospora Leaf Spot');
   elseif result == 4
   helpdlg(' Healthy Leaf ');
   disp('Healthy Leaf ');
end

%% Evaluate Accuracy

load('Accuracy_Data.mat')
Accuracy_Percent= zeros(200,1);
for i = 1:100
data = Train_Feat;
groups = ismember(Train_Label,0);
cp = classperf(groups);
classes = predict(svmStruct,data(test,:));
Accuracy = cp.CorrectRate;
Accuracy_Percent(i) = Accuracy.*100;
end
```