# MUNGBEAN LEAF DISEASE DETECTION USING K-NEAREST NEIGHBOR ALGORITHM

**HNIN PWINT ZAW**

**M.C.Tech.** **JANUARY 2023**

# MUNGBEAN LEAF DISEASE DETECTION USING K-NEAREST NEIGHBOR ALGORITHM

**BY**

**HNIN PWINT ZAW**

**B.C.Tech.**

**A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of**

**Master of Computer Technology**

**(M.C.Tech.)**

**University of Computer Studies, Yangon**

**JANUARY 2023**

# ACKNOWLEDGEMENTS

I would like to thank the Minister, Ministry of Science and Technology for full facilities support during the Master course at the University of Computer Studies, Yangon.

First and foremost, I would like to express my deepest gratitude and my sincere thanks to Dr. Mie Mie Khin, Rector, University of Computer Studies, Yangon, for her kind permission to submit this thesis.

I would like to express my gratitude to Dr. Thandar Thein, Rector of the University of Computer Studies, Maubin and Dr. Khin Swe Swe Myint, Professor and Head of Department of Faculty of Computer System and Technologies of the University of Computer Studies, Maubin, for helps suggestions and administrative support.

I would like to thank and regards to Dr. Htar Htar Lwin, Pro-rector and Head of Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her kind management throughout the completion of this thesis.

I would like to express my deeply thanks to Dr. Amy Tun, Professor and Course Coordinator of Master (CT), Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her painstaking suggestion and encouragement throughout the development of the thesis.

I would like to express my deepest and respectful gratitude to my supervisor, Dr. Khant Kyawt Kyawt Theint, Lecturer , Faculty of Computer System and Technologies of the University of Computer Studies, Yangon , for her patient supervision, tenderness, encouragement and providing me with excellent ideas throughout the study of this thesis. I will always remember her for being a mentor to me.

I also would like to express my respectful gratitude to Daw Hnin Yee Aung, Lecturer, Department of English, University of Computer Studies, Yangon, for her advice, editing and suggestion from the language point of view.

I am very grateful to all of my teachers from the University of Computer Studies, Yangon and the University of Computer Studies (Maubin) who had been helping me from beginning to end of my thesis. I really appreciate for their valuable comments, suggestions, helpful hints, and fullest cooperation during the seminars of my thesis.

# ABSTRACT

Disease detection is a very important part to protect loss of crop in agriculture. Symptoms of the plant diseases can be detected by using machine learning techniques. Machine learning technique can solve for classification and regression problems. This proposed system presented that mungbean leaf disease detection by using digital image processing and machine learning techniques. Image preprocessing state used image enhancement technique to improve the quality of images. This enhanced image is segmented by using k-means clustering techniques. This technique is used to segment region of interest in leaf area. Gray Level Co-occurrence Matrix (GLCM) is used to extract features from preprocessing and cluster images. And also, mungbean leaf diseases are classified using the k-nearest neighbor algorithm (k-NN). According to the results of the experiments, the system can successfully detect and classify healthy and unhealthy or infected leaf areas. In this system, the k-NN algorithm can classify disease types with 96.7% accuracy and the support vector machine (SVM) algorithm with 86.7%.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF EQUATIONS

# LIST OF TABLES

**TABLES** **PAGES**

# CHAPTER 1
# INTRODUCTION

Myanmar is an agricultural country because agriculture is the primary source of income for the Burmese people. One factor influencing agricultural productivity is a disease outbreak. Farmers must therefore accurately identify the type of disease and treat it as soon as possible. The leaf is the most important part of the plant to inspect for plant diseases. It is critical to accurately detect and classify leaf diseases in order to prevent agricultural losses. Mungbean is also one of the most important plants and seeds in the world, whether dried or fresh. Mungbeans are a high-protein food with numerous health benefits. Furthermore, this is the most economically important bean on the planet, providing protein to millions of people as well as a variety of other products. Mungbean diseases, such as angular leaf spot and bean rust, stifle production. To solve the problem at an early stage, an accurate classification of leaf diseases is required. Using the Plant Village dataset of leaf images, a machine learning approach is proposed to identify and classify bean leaf diseases. In machine learning, there are numerous classification approaches. This system uses the k-nearest neighbors (KNN) algorithm to determine whether a leaf is healthy or unhealthy. This algorithm can accurately determine the suffer area of a leaf by analyzing the symptoms of the image.

## 1.1 Objectives of the Thesis

The main objectives of the thesis are:

- To avoid agricultural product yield and quantity losses
- To reduce pesticides and soil damage
- To efficiently support and assist farmers.
- To widely apply agricultural technologies.
- To assist mungbean farmers by technically.
- To correctly and precisely classify diseases.

## 1.2 Related Work

The thesis book contains some references to previous proposal papers. Machine learning (ML) is the study of computer algorithms that automatically improve themselves based on experience and data. Machine learning algorithms are classified into four types: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. In the proposed system, the k-NN Algorithm is used to classify the various types of leaf diseases.

S.Agustin and R.Dijaya gave a presentation titled "Beef Image Classification using the k-NN Algorithm for Identification Quality and Freshness" in 2019. The experiment results in the ability of the system that detects meat quality based on color and texture to detect the type of beef [18].

G. Geetha, S. Samundeswari, G. Saranya, K. Meenakshi, and M. Nithya proposed "Plant Leaf Disease Classification and Detection System Using Machine Learning", a tomato plant leaf disease detection system that generated an advanced and efficient system that makes the process of producing high yield of tomato much easier for farmers[10].

Malti K.Singh, Subrat Chetia, Malti Kri Singh presented that Detection and Classification of Leaf Disease using K-means-based Segmentation and Neural-networks-based Classification. As a result, by employing these techniques, which perform well in all types of leaf diseases sampled and can successfully detect and classify the examined diseases with a precision of around 93%[14].

## 1.3 Motivation of the Thesis

When plant diseases occur, various pesticides are used. Later, many of these cases affected the soil, making it weak to develop crops. It would result in a significant loss of productivity and would have an impact on the economy. If the type of disease is correctly identified, it is possible to reduce soil damage and save money and the environment by using appropriate pesticides.

## 1.4 Organization of the Thesis

There are five chapters in the dissertation. Introduction, Thesis Objective, Related Work, and Research Motivation are all included in Chapter 1 and Chapter 2 discusses the theoretical foundations of Machine Learning. Image Processing, as well as image enhancement using the contrast enhancement technique and k-means Clustering based on region of interest, feature extraction, the k-NN classification method, and the Support Vector Machine are described in Chapter 3. Chapter 4 discusses the system's design and implementation. The conclusion, advantages and limitations of the proposed system are discussed in Chapter 5.

# CHAPTER 2
# THEORETICAL BACKGROUND

Machine learning algorithms are classified into four types: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. It is gaining popularity in a wide range of fields, including disease diagnosis in health care. Traditional diagnostic methods are costly, time-consuming, and frequently require human intervention. Traditional diagnosis techniques are limited by the individual's ability, whereas machine learning-based systems are not, and machines do not exhaust themselves as humans do. As a result, in health care, a method for diagnosing disease in the presence of an unexpectedly large number of patients may be developed. Machine learning is an artificial intelligence subset that employs data as an input resource.

The application of predefined mathematical functions to produce a result (classification or regression) that is often difficult for humans to achieve. Based on historical data, this algorithm predicts new output values. The learning result from predicting new data should be an accurate prediction rule. Machine learning algorithms estimate updated output values utilising the input of historical data. Similar to "programming by example," machine learning uses examples.. The k-NN algorithm is used for classification and regression that is a supervised machine learning algorithm. It is an object classification method that employs nearby learning data as well as the number of nearby neighbors also referred to as k values.

The development of practical and effective Machine learning research's primary goal is to create general-purpose algorithms. In terms of learning, time and space efficiency as well as the quantity of information needed by the learning algorithm should be taken into account. The solution to problem 3 should be generic enough to be used for a variety of learning issues, including those mentioned above. Making a prediction rule that can correctly forecast fresh data is the primary goal of learning. The learning-generated prediction rules can be simpled to interpret. Because machine learning algorithms are data driven and can examine large amounts of data, the results of machine learning are often more accurate than static programming results. A static program written by a human expert, on the other hand, is likely to be guided by imprecise impressions or an examination of only a small number of

examples or data. The general process of a typical machine learning model is depicted in Figure 2.1.

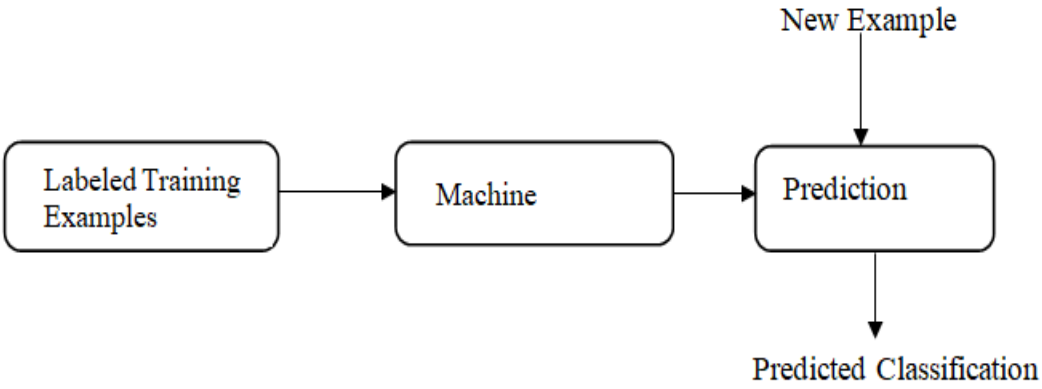


**Figure 2.1 A general Machine Learning Process Diagram**

## 2.1 The Fundamentals of Machine Learning

One of the most promising approaches for dealing with difficult decision and regression problems in the face of uncertainty is machine learning. A domain expert provides example data that demonstrates the desired behavior on representative problem instances rather than explicitly modeling a solution. A suitable machine learning algorithm is then trained on these examples in order to accurately replicate the expert's solutions and generalize them to new, previously unseen data. Over the last two decades, tremendous progress toward ever more powerful algorithms has been made. For example, using ML, it is frequently easier to locate malignant cells in a microscopic image, which is typically difficult to do just by looking at the images. Machine learning algorithms' emergence in disease diagnosis domains demonstrates the technology's utility in medical fields. This algorithm predicts new output values by using historical data as input. When forecasting based on new information, the learning result must be accurate in order to follow the prediction rule. Machine learning is analogous to "programming by example," as previously stated. The primary distinction between machine learning and static programming is that static programming produces more accurate results.

## 2.2 Machine Learning Algorithm Types

Machine Learning Algorithms can be defined in a variety of ways, but they can generally be divided into categories based on their purpose, with the following being the most important. Learning can be classified into four types:

- Supervised learning
- Unsupervised learning
- Semi supervised learning and
- Reinforcement learning

Supervised learning is used to label datasets in these learning methods. These datasets are intended to train or "supervise" algorithms in accurately classifying data or predicting outcomes. Unsupervised methods aid in the discovery of features useful for categorization. It occurs in real time, allowing for the analysis and labeling of all input data while learners are present. Unlabeled data can be obtained from a computer more easily than labeled data, which requires human intervention. The algorithm can learn from a small number of labeled text documents while classifying a large number of unlabeled text documents in the training data using semi-supervised learning. Reinforcement learning aids in determining whether an algorithm produces a correct right answer or a reward indicating a good decision. RL is founded on interactions between an AI system and its surroundings.

### 2.2.1 Supervised Learning

Supervised learning refers to the presence of a teacher in the form of a supervisor. Essentially, supervised learning is a type of learning in which the machine is taught using information that has already been labeled with the correct answer. This algorithm recognizes the input pattern and produces the expected output. Output can be expected from our input data. Consider biometric attendance as an example: It takes a fingerprint as input and predicts whether it will be an output. Regression and classification problems are the two most common types of supervised learning problems. There are six common algorithms in supervised learning. They are:

- K-Neighborhood
- Bayesian Inference
- Determination Trees

- Regression Linear
- Support Vector Machines (SVM) and
- Random Forest

### 2.2.2 Unsupervised Learning

The teaching procedure of a machine to learn from unlabeled information and then enabling the algorithm to take action on that data without supervision is known as unsupervised learning. Without any prior data training, the machine's task in this case is to group unsorted data based on similarities, patterns, and differences. These are the most common pattern recognition and descriptive modeling machine learning algorithms. However, the algorithm has no output labels on that to model relationships. By using techniques on the input data, these algorithms attempt to look for rules for rules, discover forms, summarize and group record, derive meaningful insights, and provide users with a better description of the information. For examples**:**

- **Banking sector:** Customers can be segmented based on behavioral characteristics by creating multiple segments using clustering.
- **Retail sector:** Make a collaborative filtering model based on past purchases to recommend products to customers.

There are three common algorithms in unsupervised learning. They are:

- k-means clustering
- Hierarchical clustering and
- Apriori algorithm

### 2.2.3 Semi-Supervised Learning

In the previous two types, either without labels or with labels exists for all of the observations in the dataset. Semi-supervised learning occupies a middle ground. Labeling is costly in many practical situations because it necessitates the use of skilled human experts. As a result, this algorithm is the most qualified candidates for model construction when labels are absent in the majority of observations but present in a few. These methods take advantage of the fact that, while the unlabeled data's group memberships are unknown, it contains critical data about the Organizational constraint.

### 2.2.4 Reinforcement Learning

This method seeks to maximize reward or minimize risk by using observations gleaned from interactions with the environment. The agent is an iterative reinforcement learning algorithm that is constantly learning from its surroundings. As it progresses, the agent learns from its environment experiences until it has explored every possible state.

Reinforcement Learning is a subset of Machine Learning, and thus of Artificial Intelligence. It enables software agents to automatically determine the best way to behave in a given context in order to maximize their performance. To teach the agent how to behave Simple reward feedback is required this is known as the reinforcement signal.

There are numerous algorithms that address this issue. In fact, Reinforcement Learning is defined by a specific type of problem, and all solutions to that problem are classified as Reinforcement Learning algorithms. In the problem, an agent must decide which action to take based on his current state. Reinforcement learning goes through the following steps to create intelligent software:

- The agent observes the input state.
- The decision-making function is used to direct the agent's actions.
- The environment provides a reward or reinforcement to the agent after completing the action. Correct answers are rewarded, while incorrect answers are penalized.
- The reward information is stored in the state-action pair.

There are three common algorithms in unsupervised learning. They are:

- Q-Learning
- Temporal Distinction and
- Adversarial Networks at a Deep Level

## 2.3 Data Mining

The process of analyzing large amounts of data in order to discover patterns, identify trends, and gain insight into how that data can be used is known as data mining. Data miners can then apply what they've learned to make decisions or forecast outcomes. It is a computer science and statistics interdisciplinary subfield that aims to extract data from a data set and convert it into a comprehensible structure to be used in

the future (via intelligent methods). The analysis phase of the "knowledge discovery in databases" process is known as data mining. In addition to the preliminary examination, it addresses problems with databases and data management, as well as Pre-processing of data, modeling, and inference issues, interestingness metrics, complexity issues, discovered structure post-processing, visualization, and online updating. In recent data mining projects, major data mining techniques such as association, classification, clustering, prediction, sequential patterns, and regression have been developed and used.

Data is classified using the k-NN algorithm based on its proximity to other data. It is predicated on the assumption that data points close to each other are more similar than other bits of data. Based on individual data points, this non-parametric, supervised technique predicts group features. Data is processed by neural networks using nodes. These nodes have inputs, weights, and an output. The data is mapped using supervised learning. This model can be fitted to provide threshold values for determining the accuracy of a model. Predictive analysis aims to use historical data to create graphical or mathematical models that forecast future outcomes. This data mining technique, which overlaps with regression analysis, aims to support an unknown figure in the future based on current data.

# CHAPTER 3
# METHODOLOGY OF THE PROPOSED SYSTEM

The methods used to develop the proposed mungbean leaf disease classification system are described in detail in this chapter. Firstly, image processing and image segmentation using K-means clustering methods are presented. Secondly, features are extracted by using GLCM and k-NN algorithm is used for classification the type of diseases. This system classifies healthy and unhealthy or infected mungbean leaves.

## 3.1 Image Processing

Digital image processing is the process of processing digital images using a digital computer. It can be said that the use of computer algorithms improves an image or extract useful information.

Image processing consists primarily of the following steps:

- Importing the image using image acquisition software
- Image analysis and manipulation and
- Output, which can be an altered image or a report based on the analysis of that image.

In addition, Image processing phases:

- Acquisition- It could be as easy as receiving a digital image.
- Image enhancement- It is a basic and appealing aspect of image processing that is used subjectively to extract some hidden details from an image.
- Image restoration- It considers image appeal and is based on a mathematical or probabilistic model of image degradation.
- Color Image Processing- It deals with the processing of both pseudo-color and full-color images. Digital image processing can make use of color models.
- Wavelets and multi-resolution processing- It is the foundation of image representation at various levels.
- Image Compression- This operation necessitates the creation of some functions. It is primarily concerned with image resolution or size.

- Morphological Processing- t is concerned with image component extraction tools that can be used to represent and describe shapes.
- Segmentation Procedure- It is the process of dividing an image into its constituent objects. The most difficult task in image processing is autonomous segmentation.
- Representation and Description- It is the outcome of the segmentation stage, and selecting a representation is only one component of the solution for converting raw data to processed data.
- Detection and recognition of objects-This is the process of labeling an object based on its descriptor.

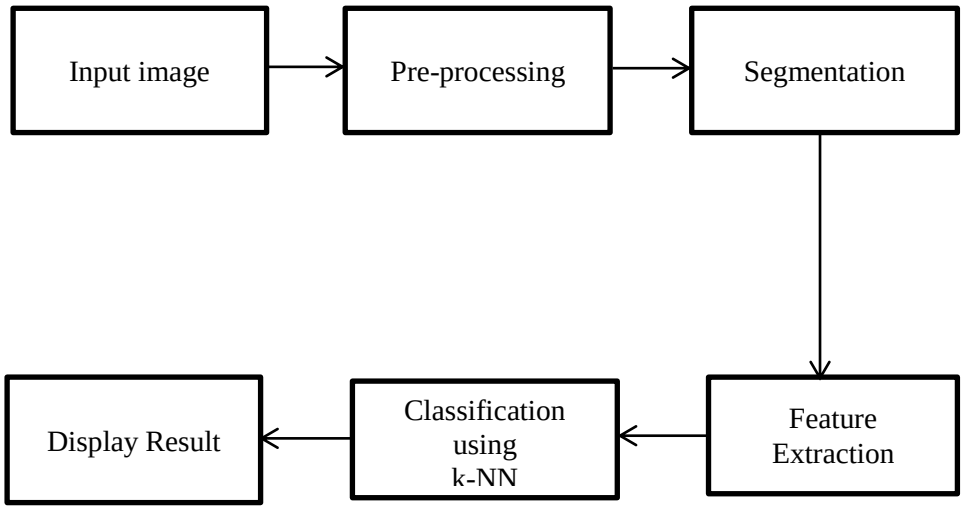


**Figure 3.1 Block Diagram of the Proposed System**

Figure 3.1 depicts the proposed system's block diagram. In this system, the input image is first preprocessed, and then the image is segmented and the k-NN algorithm is applied based on feature extracted values. Finally, display the image's output result.

### 3.1.1 Image Preprocessing

The image preprocessing stage in this proposed system includes image enhancement via contrast enhancement. Image enhancement is one of the most straightforward, interesting, and visually appealing areas of image processing. It can be used to improve the quality of specific image features. The goal of image enhancement is to improve the interpretability of images or to provide better input

for other automated image processing systems. Image enhancement techniques can be divided into two categories: Spatial domain methods rely on direct manipulation of pixels in an image. The fourier transform of an image is modified in frequency domain methods. The visual results indicate the visual interpretability of an image. The quantitative results are used to select the most appropriate processing techniques. One of the most difficult problems in low-level image processing is contrast enhancement. Contrast enhancement techniques are used to improve the visual perception and color reproduction of low contrast images.

One of the most important issues in image processing is contrast enhancement. By increasing the brightness difference between objects and their backgrounds, contrast enhancements improve object visibility in a scene. Typically, contrast enhancements are achieved through the contrast stretching process. This technique is used in photography, medical applications, and display gadgets. It is needed to improve visual perception and color reproduction as the demand for high-quality images grows. By increasing the brightness difference between objects and their backgrounds, contrast enhancements improve object visibility in a scene. Contrast enhancements are usually done in two steps: a contrast stretch followed by a tonal enhancement, though they can be done in the same step. A contrast stretch uniformly improves the brightness differences across the image's dynamic range. The contrast enhancement technique is used to preprocess the input image in Figure 3.2. The relative darkness and brightness of objects in a scene are adjusted using this technique to improve image quality. The following is the contrast enhancement equation:

$$Y = aX + b \tag{3.1}$$

Where,

Y =output value

X= input vale

a = contrast value and

b = the brightness level

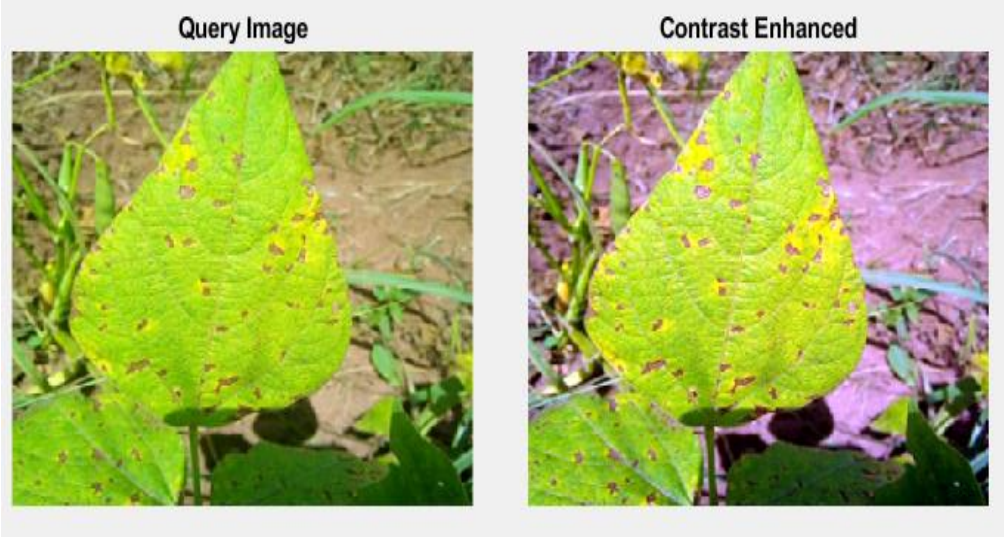


**Figure 3.2 Original Image and Contrast enhancement Image**

Figure 3.2 depicted a contrast enhanced image created from an original image during the image preprocessing stage.

### 3.1.2 k-means clustering

The enhanced image is segmented by using k-means clustering techniques. It is used to create teams of observations with similar characteristics. In this proposed system, three parameters for k values are used to identify the area of interest in the leaf's affect area. Image clustering is the process of grouping images into clusters so that images within the same clusters are similar and those in n different clusters differ. Clustering seeks to identify distinct groups or "clusters" within a data set. The tool creates groups using a machine language algorithm, where items in a similar group will have similar characteristics to each other in general.

This is a signal processing vector quantization method that intends to partition n observations into k clusters, with each observation belonging to the cluster with the closest mean (cluster centers or cluster centroid), which serves as the prototype for the cluster. As a result of this, the data space is divided into Voronoi cells. Within-cluster variances are minimized by k-means clustering, but not regular Euclidean distances, which would be the more difficult Weber problem that optimizes squared errors, but only the geometric median minimizes Euclidean distances. Better Euclidean solutions, for example, can be found using k-medians and k-medoids. Both k-means and Gaussian mixture modeling use an iterative refinement approach that is similar to the expectation-maximization algorithm for mixtures of Gaussian distributions. They both

use cluster centers to model the data; however, k-means clustering finds clusters with comparable spatial extent, whereas the Gaussian mixture model allows for different cluster shapes.

The k-means algorithm, which uses an explicit distance measure to partition the data set into clusters, is the most widely used clustering algorithm. The main idea behind the k-means algorithm is that each cluster is represented by a vector of mean attribute values from all training instances for numeric attributes and a vector of modal values for nominal attributes assigned to that cluster. Cluster center is the name given to this cluster representation.

The k-NN classifier, a popular supervised machine learning technique for classification that is frequently confused with k-means due to the name, is related to the unsupervised k-means algorithm. New data is classified into existing clusters using the k-nearest neighbor classifier on the k-means cluster centers. For this, the Rocchio algorithm, also known as the nearest centroid classifier, is used. The k-means algorithm, which uses an explicit distance measure to partition the data set into clusters, is the most widely used clustering algorithm. Cluster center is the name given to this cluster representation.

In digital image processing, this algorithm is used to find groups that have not been explicitly labeled in the data. This can be used to validate business assumptions about the types of groups that exist or to identify unknown groups in large data sets. It is used to create groups of observations with similar characteristics. If customer data, groups of similar customers can be created and then target each group with different types of marketing. Clustering is a well-known machine learning algorithm. Furthermore, k-means clustering is applicable in nearly every domain, from banking to recommendation engines, cyber security, document clustering, and image segmentation. This algorithm will do the following:

Level 1: Determine clusters number.

Level 2: Pick k points at random.

Level 3: Create k Clusters.

Level 4: Determine each cluster has a new centroid.

Level 5: Evaluate the effectiveness of each cluster.

Figure 3.3 depicts how the improved image is segmented using k-means clustering techniques. K denotes the use of clustering to create groups of observations with similar characteristics. Three parameters for k values are used in this proposed

system to identify the area of interest in the leaf's affect area. K-means clustering is described by the following equation:

$$J=\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} \|x_n - m_n\|^2 \quad (3.2)$$

Where,

N= total number of data points

K= clusters number

$x_n$= n measurement vector

$m_n$= average k cluster

$r_{nk}$ = a variable that indicates whether $x_n$ should be assigned to k.

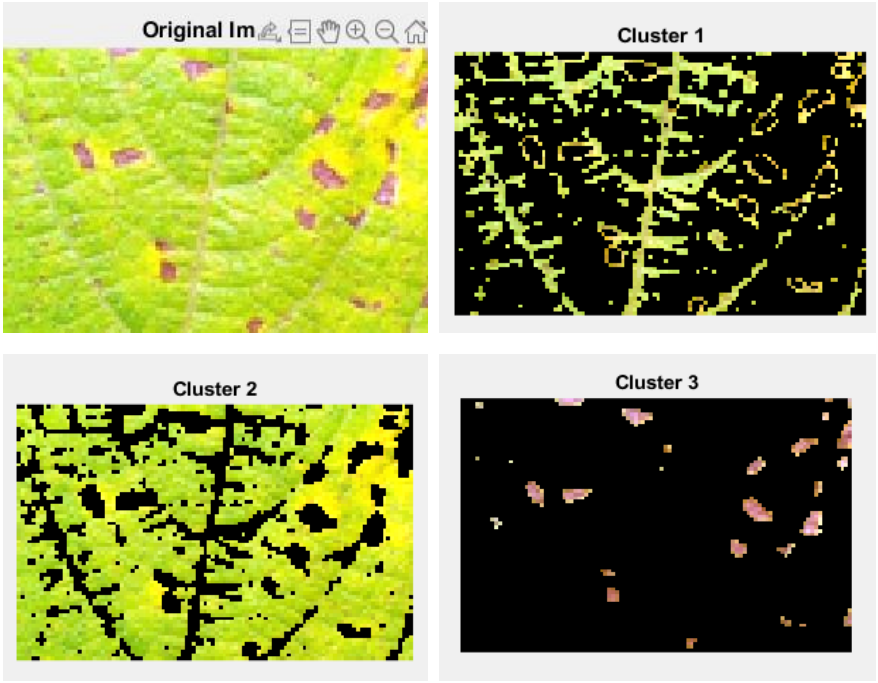


**Figure 3.3 Clustering Image using K-means Clustering**

Figure 3.3 depicts clustering images are obtained by using k-means clustering to identify regions of interest in an affected area of the leaf.

## 3.2 Feature Extraction

The procedure for converting raw data into numerical features that can be processed while preserving the information in the original data set is known as feature extraction. It outperforms applying machine learning directly to raw data in terms of results. A step in the dimensionality reduction process that divides and reduces an initial set of raw data to more manageable groups is feature extraction. As a result,

processing will be more straightforward. The most significant feature of these large data sets is the large number of variables. It is also a dimensionality reduction method for representing the interesting parts of an image in a compact feature vector. Domain-specific image features include color, texture, and shape extraction.

In this system, the Gray Level Co-occurrence Matrix is used to extract texture features (GLCM). It computes an image's gray level dependence. The pixel relationship is computed vertically to the right (0). A GLCM employs the texture classification concept. A homogeneity value is assigned to each pixel in the image. Following the calculation of the homogeneity values, a value matrix is generated. The GLCM value is calculated whenever the homogeneity value of a specific pixel changes. The X-ray tumor distinguishes itself from the gray mass in the brain. The texture of the gray mass differs from that of the tumor. GLCM is the best option at that point. When the matrix value abruptly changes, the likelihood of developing the tumor rises. GLCM is the best method for classifying pixels based on their values.

The GLCM matrix computes the probability value of a relationship between two pixels in an image that have the same intensity at the same distance and orientation at the same angle. It is one of the techniques used to extract texture analysis features. d and angled separate the two-pixel coordinates. Distances are represented by pixels, while angles are represented by degrees. The angular orientation will be divided into four directions with a one-pixel distance between pixels: 0°, 45°, 90°, and 135°.

Step 1: The initial GLCM matrix is made up of two-pixel pairs that line up at 0°, 45°, 90°, or 135°.

Step 2: Creating a matrix by combining the GLCM's initial matrix;

Step 3: Subtract the number of pixel pairs from each GLCM element's probability value.

Step 4: Determine the total number of extracted features for each formed direction and namely.

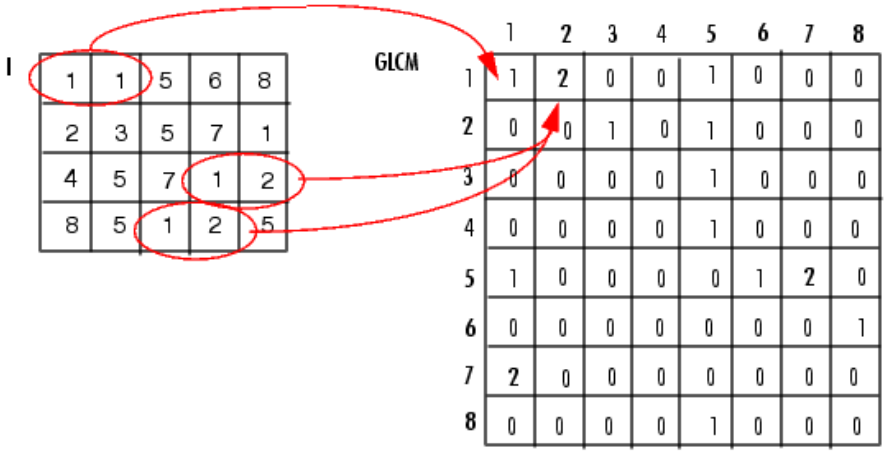


**Figure 3.4 Feature extract using GLCM**

Figure 3.4 depicts the extracted feature by using a gray level co-occurrence matrix (GLCM), which represents the relationship between the reference pixel (i) and the neighbor pixel (j) in various orientations.

In this proposed system, texture features are calculated by using Contrast, Correlation, Mean, Variance, Energy, Entropy and Homogeneity and color features are calculated by using RMS, Smoothness, Kurtosis , S.D , IDM and Skewness.

**1. Contrast**

The contrast of an image represents a different GLCM moment and measures its spatial frequency. It is calculated by subtracting the highest and lowest values of the pixels in the adjacent set. The contrast texture quantifies the image's local variations. A low-contrast image is distinguished by low spatial frequencies and a GLCM concentration term around the principal diagonal.

$$\text{Contrast} = \sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2 \qquad (3.3)$$

**2. Homogeneity**

The inverse difference moment is another name for this statistical metric. It measures image homogeneity by assuming larger values for smaller differences in grey tone between pairs of elements. The GLCM's homogeneity is more sensitive to the presence of near diagonal elements. The value of homogeneity is maximized when all of the elements in an image are the same. Contrast and homogeneity in the GLCM are strongly but inversely related, which means that as contrast increases while energy remains constant, homogeneity decreases.

$$\text{Homogeneity} = \sum_{i,j=0}^{N-1} \frac{Pi,j}{1+(i-j)^2} \quad (3.4)$$

**3. Correlation**

It quantifies the image's linear dependencies. A high correlation means that the image contains a lot of linear structures. Correlation can be defined as

$$\text{Correlation} = \sum_{i,j=0}^{N-1} Pi,j \left[\frac{(i-\mu i)(j-\mu j)}{\sqrt{(\theta i^2)(\theta j^2)}}\right] \quad (3.5)$$

**4. Energy**

It is a measure of image uniformity and is also known as the angular second moment. As a result, it is inversely related to entropy. An image with a high energy value is uniform. Energy is expressed as

$$\text{Energy} = \sum_{i,j=0}^{N-1} Pi,j^2 \quad (3.6)$$

**5**. **Entropy**

It quantifies the disorder in a grayscale image. The disorder in the image increases as the entropy increases. Entropy is calculated as

$$\text{Entropy} = \sum_{i,j=0}^{N-1} Pii1nPij^2 \quad (3.7)$$

**6**. **Mean**

The arithmetic mean of squares of a given set of numbers is known as mean value. For a complex-valued signal set with discrete sampled values.

$$\text{Mean} = \frac{\sum_{i=1}^{N} xi}{N} \quad (3.8)$$

**7. Variance**

The variance threshold is an easy place to start when selecting features. It eliminates all features whose variance is less than a certain threshold. By default, it removes all zero-variance features, that is, features that have the same value in all samples.

$$\text{Variance} = \frac{\sum(xi-\bar{x})^2}{N} \quad (3.9)$$

**8. RMS**

The RMS value of a signal is calculated as the square root of the signal's average squared value. For a complex-valued signal set with N discrete samples.

$$\text{RMS} = \sqrt{1/T \int f(x)^2 \, dx} \quad (3.10)$$

**9. Smoothness**

It is determined by the quantity of intermittent movement, which is directly related to coordination.

$$\text{Smoothness} = \rho\left(\frac{\partial v}{\partial t} + v.\nabla v\right) = \nabla\rho + \nabla.T + f \quad (3.11)$$

**10. Kurtosis**

Kurtosis can be used to define an investment's risk. The calculated Kurtosis value can also be used to predict the nature of the investment in order to maximize returns. The deviation from the mean of any investment data set is proportional to its excess.

$$\text{Kurtosis} = n^* \frac{\sum_i^n (Yi-\bar{Y})^4}{\sum_i^n (Yi-\bar{y}^2)^2} \quad (3.12)$$

**11. Standard Deviation**

A set of values' standard deviation is a measure of their variation. A low standard deviation suggests that the values in the set are close to the mean.

$$\text{Standard Deviation} = \frac{\sqrt{\sum_{i=1}^{n}(xi-\bar{x})^2}}{n-1} \quad (3.13)$$

**12. IDM**

Inverse Difference Moment Inverse Difference Moment (IDM) is the local homogeneity. It is high when local gray level is uniform and inverse GLCM is high. IDM weight value is the inverse of the Contrast weight.

$$IDM=\sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\frac{P(i,j)}{1+(i-j)^2} \quad (3.14)$$

**13. Skewness**

Skewness is a distribution term. A distribution is said to be asymmetrical if its left and right sides are not mirror images. A distribution's skewness can be positive, negative, or zero.

$$Skewness= \frac{[(N-1)(N-2)}{N}\sum_{j=1}^{N}\frac{(xj-\bar{x})^3}{{\sigma_x}^3} \quad (3.15)$$

| FEATURES | |
|---|---|
| Mean | 7.92473 |
| S.D | 37.4301 |
| Entropy | 0.599137 |
| RMS | 2.59946 |
| Variance | 1344.77 |
| Smoothness | 0.999993 |
| Kurtosis | 25.3541 |
| Skewness | 4.80758 |
| IDM | 196 |
| Contrast | 0.63754 |
| Correlation | 0.713664 |
| Energy | 0.885981 |
| Homogeneity | 0.969907 |

**Figure 3.5 Extracted texture and color feature value**

Figure 3.5 depicts the extracted texture and color feature values for an angular leaf spot image.

## 3.3 Classification

Classification works for both structured and unstructured data. The first step in the process is to predict the class of given data points. Classes are also referred to as the target, label, or categories. The focus of classification predictive modeling is to the mapping function that discrete variables transitioning from input to output. The primary goal is to identify that category the new information belongs to.

Classification in data administration allows for data separation and classification based on predefined criteria for various business or personal goals.

The systematic grouping or categorizing of objects based on predetermined criteria is referred to as classification. It is a fundamental concept in early childhood education. Comparing items based on similarities and differences is part of classification. Classification can be used to teach children about a wide range of topics. The three most important factors are as follows:

- It can be used to identify objects or living organisms.
- It helps with understanding and studying the characteristics, similarities, and differences of various objects.
- It explains how objects are classified and grouped into various categories.

Objects can be classified using a variety of parameters. It could be of various colors, sizes, and so on. Let's look at some different classification methods and examples. Classification based on shape.

- Classification based on size.
- Number classification.
- Classification based on color.

Classification is used in machine learning (ML) predictive modeling to give a class label to input data. A spam-detecting email security program, for example, might use natural language processing to classify emails as "spam" or "not spam."

It is a machine learning predictive modeling problem in which the class label for a given example of input data is predicted. Training data with a numerous input and output datasets is required to determine handwriting characters, identify spam, and so on. The Classification algorithm is a Supervised Learning technique that classifies new observations using training data. Classification programs use a given dataset or set of observations to learn how to classify new observations into one of several classes or groups.

It is a supervised learning strategy in that a computer software learns from information and makes new classifications in machine learning. There are numerous classification algorithms in machine learning.

In this many algorithm, there are two type of useful classification algorithm:

- k-Nearest Neighbors and
- Support Vector Machine

### 3.3.1 k-Nearest Neighbor

The k-NN algorithm for classification and regression is a simple supervised machine learning algorithm. It is a classification method for objects based on nearby learning data and the number of their nearest neighbors, also known as k values. It is a non-parametric supervised learning classifier that it uses to classify or predict the grouping of a single record. The machine-learning method k-NN is well-known and has been used in large-scale data mining projects. The concept is to use a large amount of training data, with each data point defined by a unique set of variables.

The k-NN algorithm is a non-parametric search algorithm, supervised learning classifier that uses proximity to classify or predict record grouping. While it can be applied to either regression or classification problems, it is most commonly applied to classification, with the assumption that similar points are nearby.

To assign a class label to a classification problem, a majority vote is used, and the label most commonly associated with a given record is used. While technically "plurality voting" is used, the term "majority vote" is more commonly used in literature. The distinction between these terms is that "majority voting" technically necessitates a majority of more than 50%, which is only possible when there are only two options. When there are multiple classes—for example, four categories—a decision about a class does not always require 50% of the vote; a class label could be assigned with a vote of more than 25%.

Regression problems are similar to classification problems in that they make a classification prediction using the average of the k nearest neighbors. The main difference is that classification is used on discrete values, whereas regression is used on continuous values. However, before making a classification, the distance must be defined. The Euclidean distance is the most commonly used distance. k-NN algorithm belongs to the "lazy learning" model family that it skips the training stage and instead stores a training dataset. It implies that the entire computation occurs when a classification is made. It is referred to as memory-based learning method because it stores all of its training data in memory.

### 3.3.1.1 Advantages of k-NN

It can be used for classification in a supervised setting with a dataset containing target labels. It finds the k nearest data points in the training set for classification, and the target label is computed as the mode of the target label of these k nearest neighbors. It has numerous advantages. They are as follows:

- Simple to implement: Because of its ease of use and precision, the algorithm is one of the first classifiers that a new data scientist will learn.
- When used for classification and regression: it can learn non-linear decision boundaries. Can devised a highly flexible decision boundary by varying the value of K.
- There has been no preparation time for classification/regression: The k-NN algorithm has no explicit training step, and all work is done during prediction.
- Constantly evolves with new data: Because there is no explicit training step, the prediction is adjusted as we add new data to the dataset without having to retrain a new model.
- Single Hyperparameter: The value of K is the only hyperparameter. This facilitates hyper parameter tuning.
- Distance metric selection: There are numerous distance metrics to choose from. Euclidean, Manhattan, Minkowski, hamming distance, and other popular distance metrics are listed below.
- Easily adapts: Because all training data is stored in memory, the algorithm adjusts as new training samples are added to account for any new data.
- Several hyperparameters: When compared to other machine learning algorithms, k-NN requires only a k value and a distance metric.

### 3.3.1.2 Disadvantages of k-NN

It also has some disadvantages. They are as follows:

- Works poorly with large datasets: In the case of large datasets, the cost of calculating the distance between the new point and each existing point is enormous, reducing the algorithm's performance.

- Feature scaling is required: Prior to applying the k-NN algorithm to regardless of dataset. If it must not be performed, k-NN may make incorrect predictions.
- Because calculating each dimension's distance becomes difficult as the quantity of dimensions grows, this algorithm does not perform well in high dimensions.
- Large datasets with high prediction complexity: Because the entire training data is processed for each prediction, it is not suitable for large datasets. For each prediction, the time complexity where M is the data dimension and N denotes the number of instances in the training data.. It should be noted that there are specialized ways of organizing data to address this issue and speed up k-NN.
- Greater prediction complexity with higher dimensions: In supervised learning, prediction complexity increases with higher dimensional data.
- All features are weighted equally: k-NN may reject points that are extremely close in one dimension but far apart in another because it expects points to be close in ALL dimensions. This can be changed by selecting an appropriate distance measurement. It is also sensitive if the ranges of different features vary. Scaling features after appropriate pre-processing can address this.
- Vulnerable to outliers: A single incorrectly labeled example can shift class boundaries. If there is an outlier in one dimension, the average separation in higher dimensions tends to be higher, so outliers can have a greater impact in higher dimensions (curse of dimensionality).

### 3.3.2 Background of k-NN

The k-NN collects data from a training data set and uses it to predict new records in the future. For each new record, the k-closest records from the training data set are determined. The value of the target attribute of the closest records is used to make a prediction for the new record. A prediction for the new record is made based on the value of the target attribute of the closest records. This algorithm employs a majority voting mechanism. It gathers data from a training data set and uses it to predict new records later on.

The k-closest records from the training data set are determined for each new record. A prediction for the new record is made based on the value of the target attribute of the closest records. For any given instance, the basic nearest neighbor (NN) algorithm predicts classification or regression. For this purpose, the k-NN algorithm identifies a training instance that is closest to the arbitrary instance. This algorithm then returns the predicted class label or target function value for the arbitrary instance as the class label or target function value from the training instance. Instead of just one training instance, the k-NN algorithm employs a predetermined number k1 of the closest training instances. The typical value range is from one to several dozens of dollars. The outcome is determined by whether this algorithm is used for classification or regression. The predicted class label in k-NN classification is determined by voting for the nearest neighbors. In k-NN regression, the predicted value is the average of the target function values of the nearest neighbors. By specifying k1, control the tradeoff between prevention and resolution of overfitting. For noisy data, overfitting prevention may be required. In order to obtain different predictions for similar instances, resolution may be required.

### 3.3.3 Working of k-NN Algorithm

This algorithm would have a training dataset and a test dataset. Because of the discussion of single label classification, each instance of training data will contain multiple features but only one label. As a result, it is possible to determine which labels each piece of data belongs to.

It calculates the distances between a query and all of the examples in the data, then selects the number of examples (K) closest to the query and votes for the most frequent label (in the case of classification) or averaging the labels (in the case of regression). This algorithm selects a number k as the nearest Neighbor to the data point to be classified. If k is 3, it will look for the 3 nearest Neighbors to that data point. In k-NN algorithm have a training dataset and a test dataset. Because we are only talking about single label classification, each instance of training data will have several features and only one label. The k-NN working can be explained on the basis of the below algorithm:

Level 1: Count the number of neighbors (k).

Level 2: Calculate the distance in Euclidean terms each of the K neighbors.

Level 3: Determine the k closest neighbors using the calculated Euclidean distance.

Level 4: Count the number of information sources in each of these k categories.

Level 5: Assign the new information within the realm of the greatest number of neighbors.

Level 6: Our model is finished.

Figure 3.6 shows how the k-nearest neighbor algorithm is used to extract texture and color features from images. The supervised algorithm employs the k-Nearest Neighbor algorithm as a classification method. It is a straightforward algorithm that can be easily integrated into machine learning algorithms to solve classification and regression problems. This algorithm is used in classification to find the value of group k on the object in the training data that is closest (similar) to the object in the testing data. In general, this algorithm uses the Euclidean distance statistical formula to calculate the distance between two objects x and y. The following is the Euclidean distance equation:

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3.16)$$

Where,

d (x,y) : the separation between testing and training data

x: testing data

y: training data and

n: the number of features.

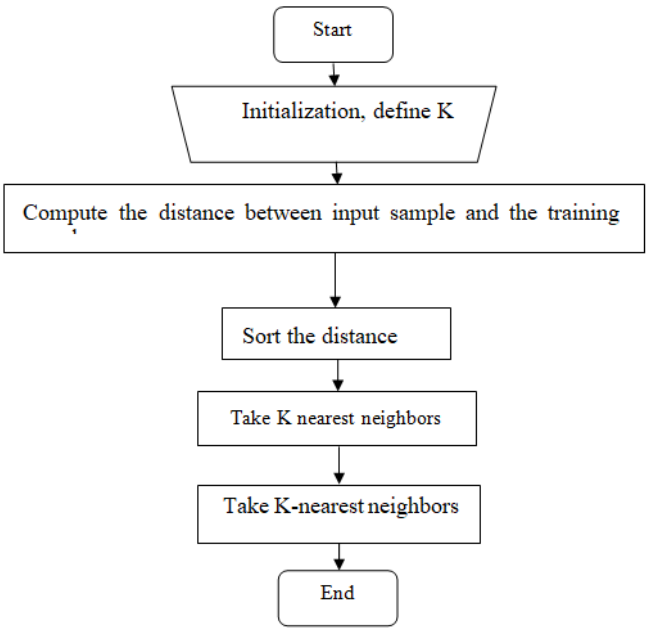


**Figure 3.6 k-Nearest Neighbor Workflow Diagram by using Euclidean distance**

Figure 3.6 depicts the operation of k-Nearest Neighbor, which defines k and computes the distance between test data and each row of training data using euclidean distance.

## 3.4 Support Vector Machine

A support vector machine (SVM) is a supervised machine learning (SML) algorithm that divides data into two categories to perform classification or regression tasks. It is capable of solving linear and nonlinear problems and has a wide range of practical applications. The SVM concept is straightforward: To divide the data into classes, the algorithm draws a line or a hyperplane. The goal of the SVM algorithm is to find the best line or decision boundary for categorizing n-dimensional space so that new data points can be easily placed in the correct category in the future. The best decision boundary is a hyperplane. The best divider is one that is equally distant from each group's boundaries. The algorithm is used for classification rather than regression. Even when the data is not otherwise linearly separable, the Support Vector Machine (SVM) categorizes it by mapping it to a high-dimensional feature space. The algorithm is called SVM because the separating hyperplane is supported (defined) by the vectors (data points) closest to the margin. SVM has been shown to perform well in a wide range of real-world learning problems and is widely regarded as one of the best "out-of-the-box" classifiers. A separator is discovered between the categories, and the data are transformed so that the separator can be drawn as a hyperplane.

The support vector machine algorithm seeks a hyperplane that distinguishes between data points (N — the number of features). A variety of hyperplanes could be used to separate the two types of data points. Our objective is to find the plane with the greatest margin, or distance, between data points from both classes. Increasing the margin distance provides some reinforcement, allowing future data points to be classified more confidently.

SVM (Support Vector Machine) uses a hyperplane to classify data, which acts as a decision boundary between different classes. Support Vectors are extreme data points from each class. SVM attempts to find the best and most optimal hyperplane with the highest margin from each Support Vector. Non-linear data is classified using kernel functions or tricks. It converts non-linear data to linear data before drawing a hyperplane.

It is also determined by the number of features. When there are only two input features, the hyperplane is simply a line. The hyperplane transforms into a two-dimensional plane when the number of input features reaches three. It becomes difficult to imagine when the number of features exceeds three. Support vectors are data points that are closer to the hyperplane and have an effect on its position and orientation. Maximize the classifier's margin using these support vectors. The position of the hyperplane will change if the support vectors are removed. These are the factors that will shape SVM.

### 3.4.1 Advantages of SVM

Support vector machine has the advantage of avoiding the difficulties associated with using linear functions in high-dimensional feature space, and the optimization problem is transformed into dual convex quadratic programs.

- SVM is the most effective in high-dimensional spaces with more dimensions than samples.
- SVM also saves memory and performs well on smaller, cleaner datasets. Because it only uses a subset of training points, it may be more efficient.
- SVM works reasonably well when there is a clear margin of separation between classes. SVM outperforms in high-dimensional spaces. When the number of dimensions exceeds the number of samples, SVM is effective. SVM consumes very little memory.
- SVM is a great method to use when the data does not know as much as it should. It is capable of storing data such as images, text, and audio. It can be used for data that is not regularly distributed and has an unknown distribution.
- The SVM contains a very useful technique known as kernel, and by utilizing the associated kernel function, any complex problem can be solved. Kernel selects a function that is not necessarily linear and can take different forms depending on the data it operates on, making it a non-parametric function.
- Regularization capabilities: L2 Regularization is included in SVM. As a result, it has good generalization abilities and avoids over-fitting.
- Handles non-linear data efficiently: SVM can handle non-linear data efficiently by employing the Kernel trick.

- Classification and regression problems can be solved: SVM can solve classification and regression problems. SVR (Support Vector Regression) is used for regression problems, whereas SVM (Support Vector Regression) is used for classification.
- Stability: Minor changes to the data have little impact on the hyperplane and thus the SVM. As a result, the SVM model is consistent.

### 3.4.2 Disadvantages of SVM

It also has some disadvantages. They are as follows:

- The SVM algorithm is unsuitable for large data sets. SVM does not perform well when the data set contains more noise, i.e. target classes overlap.
- The SVM will underperform when the number of features for each data point exceeds the number of training data samples.
- There is no probabilistic explanation for the classification because the support vector classifier works by placing data points above and below the classifying hyperplane.
- It is sensitive to noise - A small number of mislabeled examples can significantly reduce performance.
- Choosing the right kernel is a difficult task.
- Slower as dataset size increases.
- It classifies using geometry, whereas for many classification problems, probability produces better results.
- It is difficult to select an appropriate Kernel function: Choosing an appropriate Kernel function (to deal with non-linear data) is a difficult task. It could be difficult and complicated. Many support vectors can be created when using a high dimension Kernel, which significantly reduces training speed.
- Extensive memory requirements: SVM's algorithmic complexity and memory requirements are extremely high. A large amount of memory is required because all of the support vectors must be stored in memory, and this number grows dramatically with the size of the training dataset.
- Requires Feature Scaling: Before using SVM, variables must be feature scaled.

- Long training time: SVM training takes a long time on large datasets.
- Difficult to interpret: Unlike Decision Trees, the SVM model is difficult for humans to understand and interpret.

### 3.4.3 Background of SVM

SVM is a supervised classification method derived from statistical learning theory that consistently produces good classification results from complex and noisy data. The classes are divided using a decision surface that maximizes the class margin. The surface is known as the optimal hyperplane, and the data points closest to it are known as support vectors. The most important training set's components are the support vectors. Using nonlinear kernels, it can be transformed into a nonlinear classifier. While SVM is primarily a binary classifier, by combining several binary SVM classifiers, it can be used as a multiclass classifier (creating a binary classifier for each possible pair of classes). This is especially important for non-separable training sets in multiclass classification. The penalty parameter controls the trade-off between allowing for training errors and requiring precise margin. It produces a soft margin, which allows for some misclassifications, such as training points on the incorrect side of the hyperplane. Increasing the penalty parameter's value increases the cost of incorrectly classifying points and forces the development of a more accurate model, which may or may not generalize well. Choose the Kernel Type from the drop-down menu. Additional fields may appear depending on the option you choose. All of these options are different mathematical representations of a kernel function, which is a function that provides weights for nearby data points when estimating target classes. The lowest and highest values are 1 and 6, respectively. The default value is 2. The boundary between classes is more precisely defined by increasing this parameter.

It works well when the classes are very distinct. However, in most cases, a high degree of variation and mixed pixels will be working with imagery. Increasing the polynomial value causes the algorithm to more accurately follow the contours between classes, but the classification is suitable for risky noise. Set the SVM algorithm's Penalty Parameter. This is a floating-point number that is not zero. The default value is 100.0. The penalty parameter allows for some misclassification, which comes in handy when working with non-separable training sets. It gives the option of allowing training errors or imposing strict margins. Increasing this value reduces the cost of

misclassifying points while producing a more accurate model that may or may not generalize well. Use the threshold slider to indicate the degree of certainty that the closest segments of any given segment (in the segmentation image) belong to the same class. Only the most closely related segments will be classified because higher values indicate greater certainty. Lower values indicate that unsure whether the closest neighbors belong to the same class. As a result, classify segments that are further apart.

# CHAPTER 4
# SYSTEM DESIGN AND IMPLEMENTATION

The Mungbean Leaf Disease Detection Using K-Nearest Neighbor Algorithm is implemented with four steps. They are Image preprocessing, Image segmentation, Feature extraction and KNN classification. This chapter describes detail of the system implementation. This system is simulated by using the MatLab programming language. Finally, Experimental results and performance evaluations are presented for this system.

## 4.1 Proposed System Design

The proposed system includes preprocessing, segmentation, texture feature extraction, and classification diseases of mungbean leaves according to these extracted features using k-NN as a classifier. Figure 4.1 depicts the overflow design of the proposed mungbean leaf disease detection system. The proposed system is implemented using k-Nearest Neighbor.

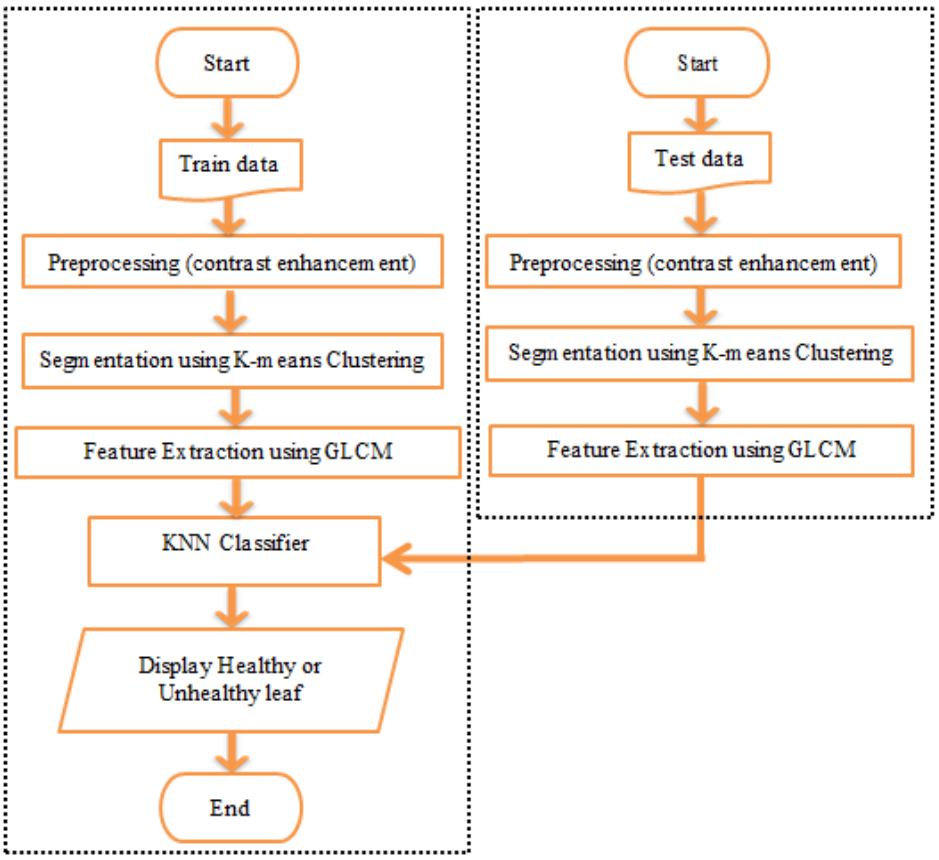


**Figure 4.1 Flowchart of the Proposed System**

In the first stage, the input image is preprocessed using the contrast enhancement technique. This technique is used to enhance image quality by adjusting the relative darkness and brightness of the scene's objects to improve visibility. The mungbean images are 500×500 pixels in size.

In the second stage, the enhanced image is segmented by using k-means clustering techniques. It is used to create groups of observations with similar characteristics. In this proposed system, three parameters for k values are used to determine the region of interest in the leaf's affect area.

In the third stage, the extracted texture and color features from the mungbean leaf images are extracted, and the features dataset is constructed using the extracted feature vectors as described in the previous chapter.

In the final stage, a testing image is inserted into the system, and its features are extracted. Using KNN as a classifier, the tested image is classified based on distances between the features vector of the trained image.

## 4.2 Data Description

In this proposed system, uses plant village dataset for training and testing data. Training data contains number of leaf image nearly seven hundreds (689) and testing data is about three hundreds (313). Healthy leaf contains number of leaves nearly three hundreds and unhealthy leaf contain over seven hundreds. Therefore, in this proposed system used number of mungbean leaf dataset totally one thousands. Plant Village dataset is used and it was split into 80% for training and 20% for testing.


**Figure 4.2 Sample Dataset of Angular Leaf Spot**

Figure 4.2 depicts a sample dataset for angular leaf spot with unhealthy leaf.


**Figure 4.3 Sample Dataset of Bean Rust**

Figure 4.2 depicts a sample dataset for bean rust with unhealthy leaf.


**Figure 4.4 Sample Dataset of Healthy**

Figure 4.2 depicts a sample dataset for healthy leaf.

## 4.3 Implementation of the system

Mungbean Leaf Disease Detection Using k-NN is implemented in accordance with the overview design shown in Figure 4.1. The MatLab programming language is

used in the system's implementation. The system's main graphical user interface (GUI) consists of two key buttons: features extraction and disease classification as shown in Figure 4.5.

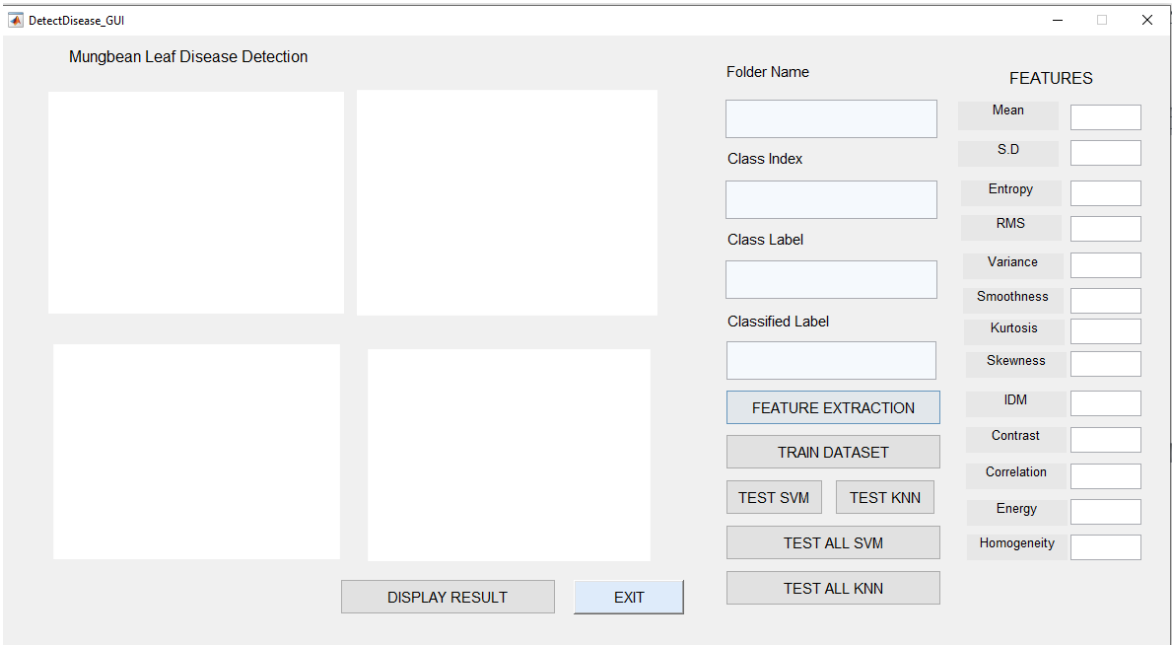


**Figure 4.5 Main GUI of the System**

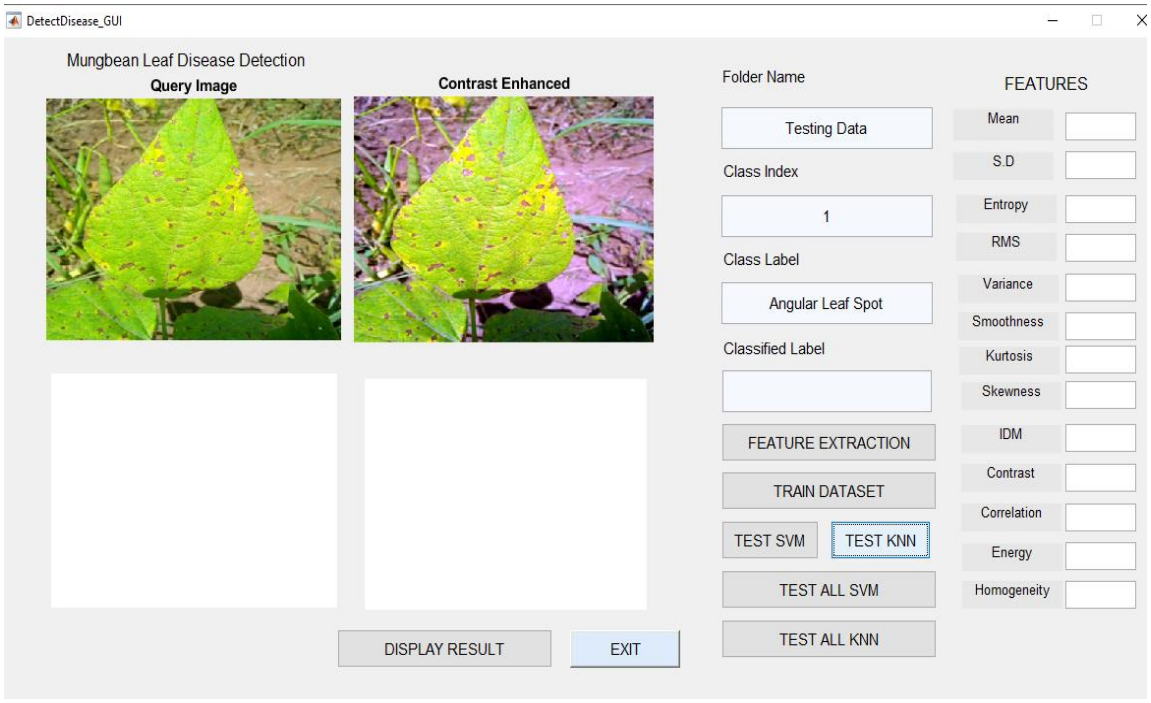


**Figure 4.6 Preprocessing image for angular leaf spot test with k-NN**

Figure 4.6 show that an input image is preprocessed by using the contrast enhancement technique for angular leaf spot.

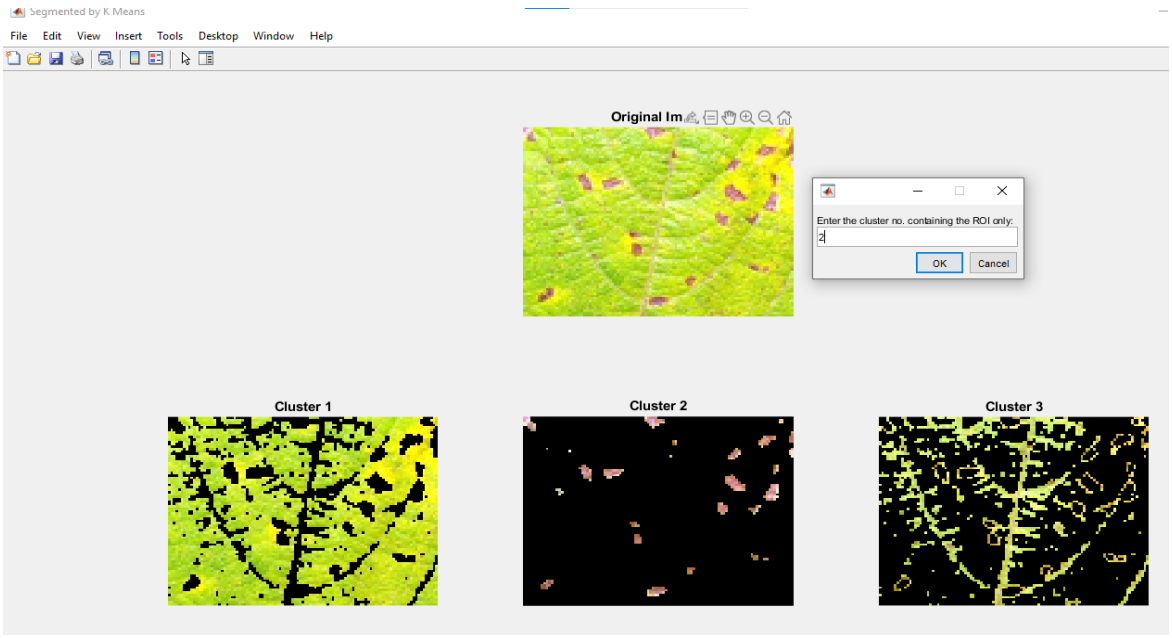


**Figure 4.7 Clustering image result for angular leaf spot test with k-NN**

Figure 4.7 shows an contrast enhancement image is segmented by using the k-means clustering technique that it specify the region of interest in the leaf's affect area.

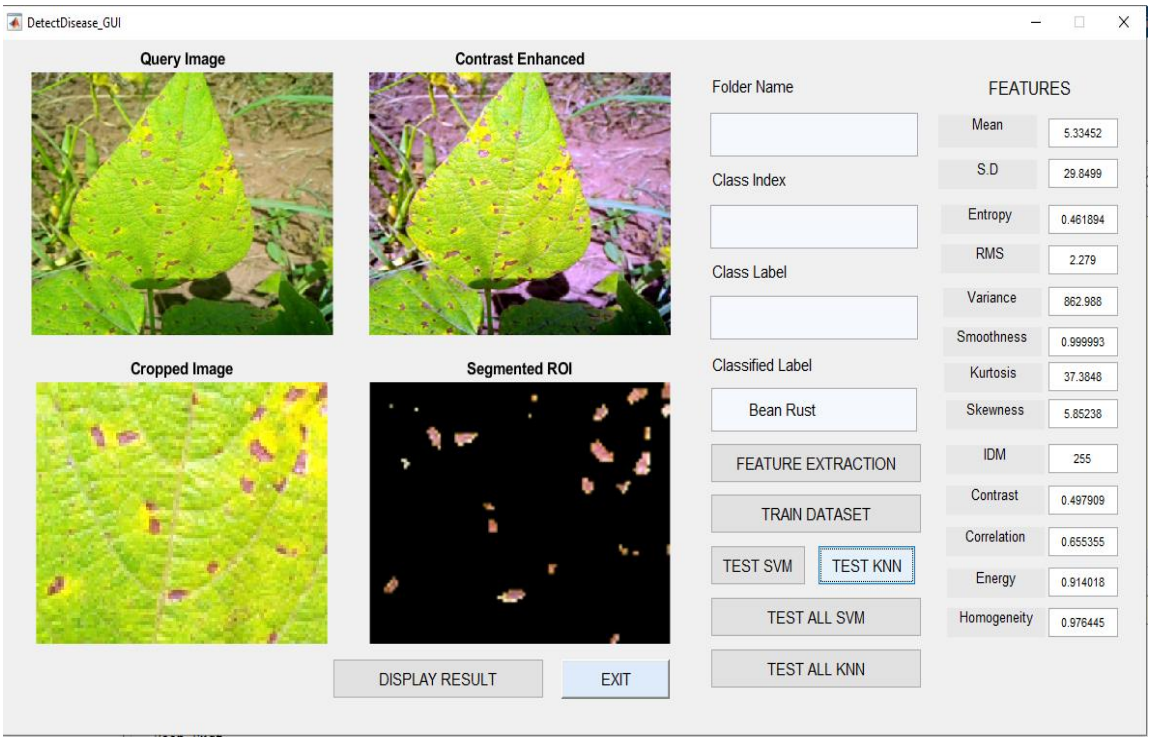


**Figure 4.8 Feature output result for angular leaf spot test with k-NN**

Figure 4.8 show that extracted features result from preprocessing and segmented images.

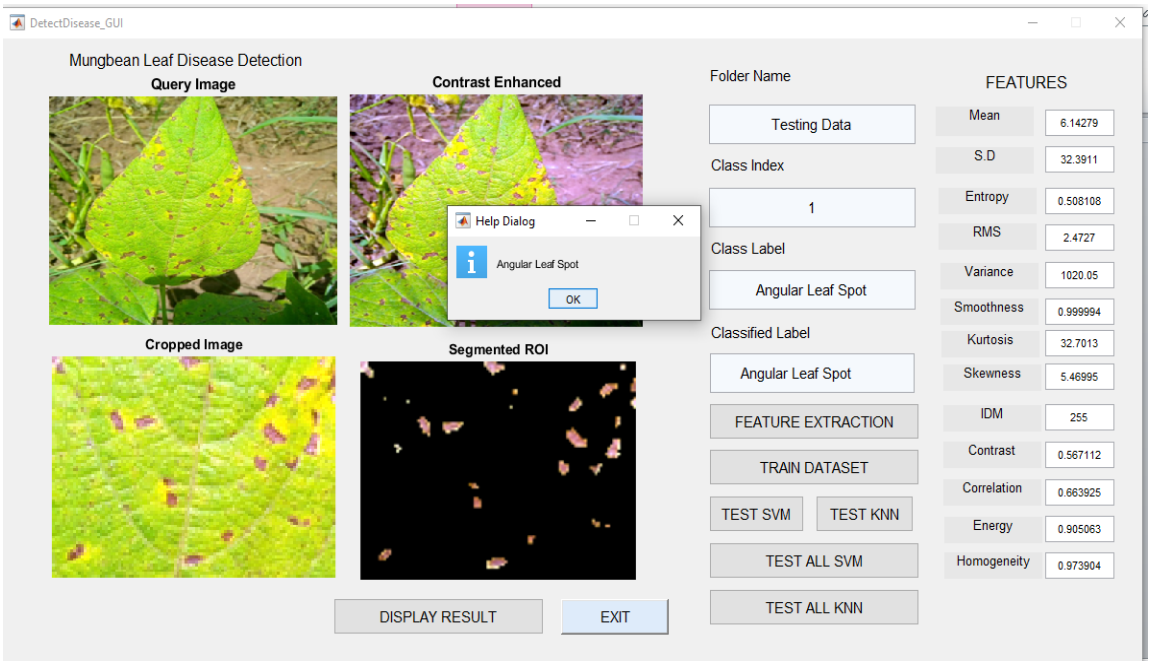


**Figure 4.9 Classification Result for angular leaf spot test with k-NN**

Figure 4.9 shows a classification result image generated by the k-nearest neighbor algorithm based on the extracted texture and color features of the images. As a result, it is an angular leaf spot.

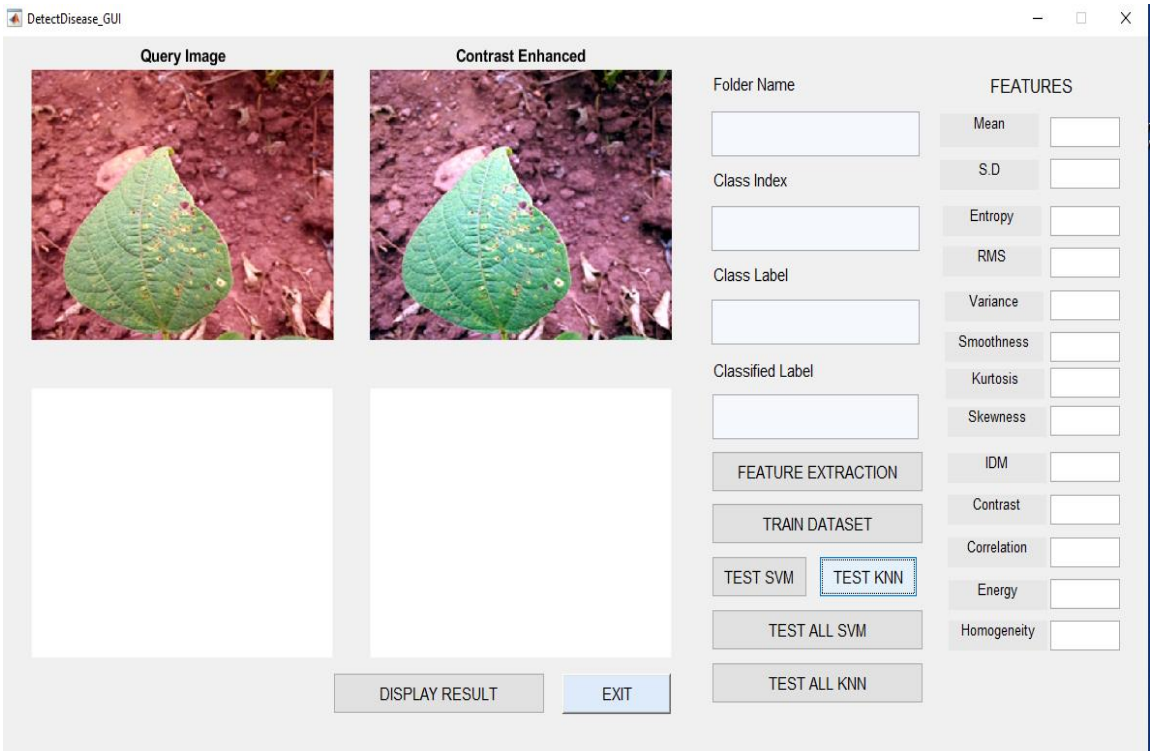


**Figure 4.10 Preprocessing input image for bean rust test with k-NN**

Figure 4.10 depicts that an input image is preprocessed by using the contrast enhancement technique for bean rust.

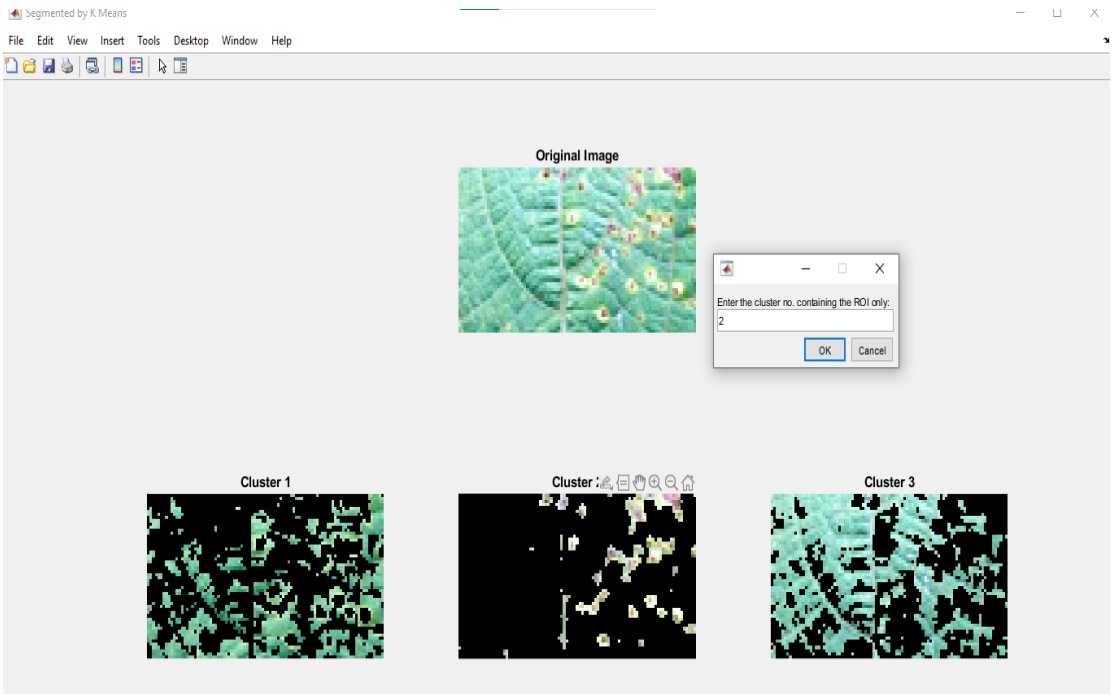


**Figure 4.11 Clustering image result for bean rust test with k-NN**

Figure 4.11 shows a contrast enhancement image is segmented by using the k-means clustering technique that it specify the region of interest in the leaf's affect area.

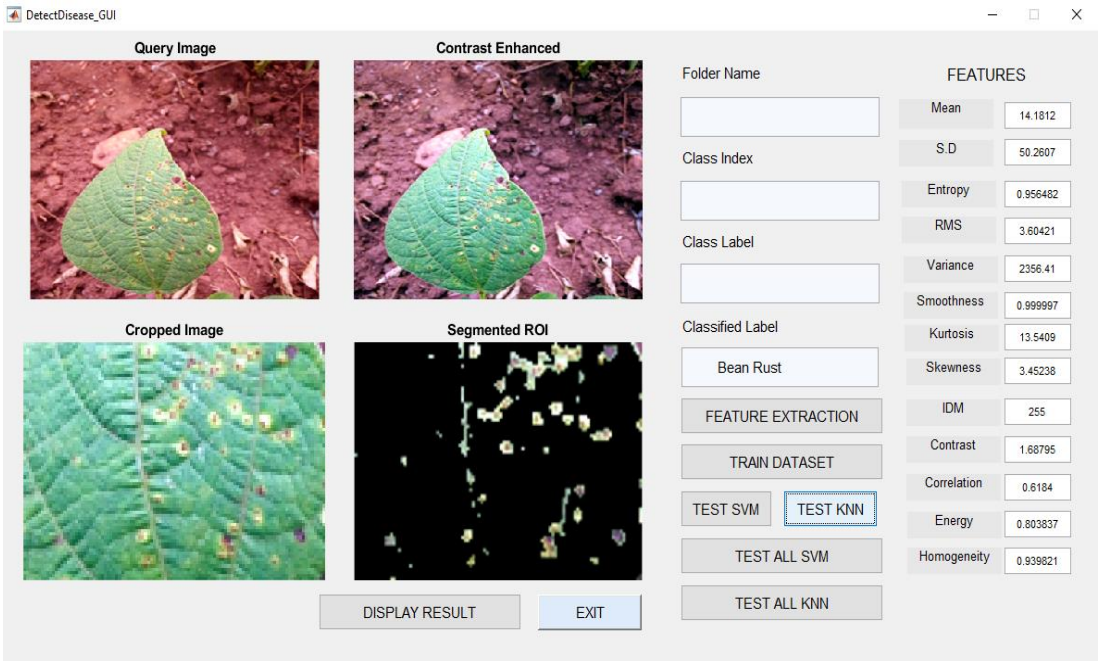


**Figure 4.12 Feature output result for bean rust test with k-NN**

Figure 4.12 show that extracted features result from preprocessing and segmented images.

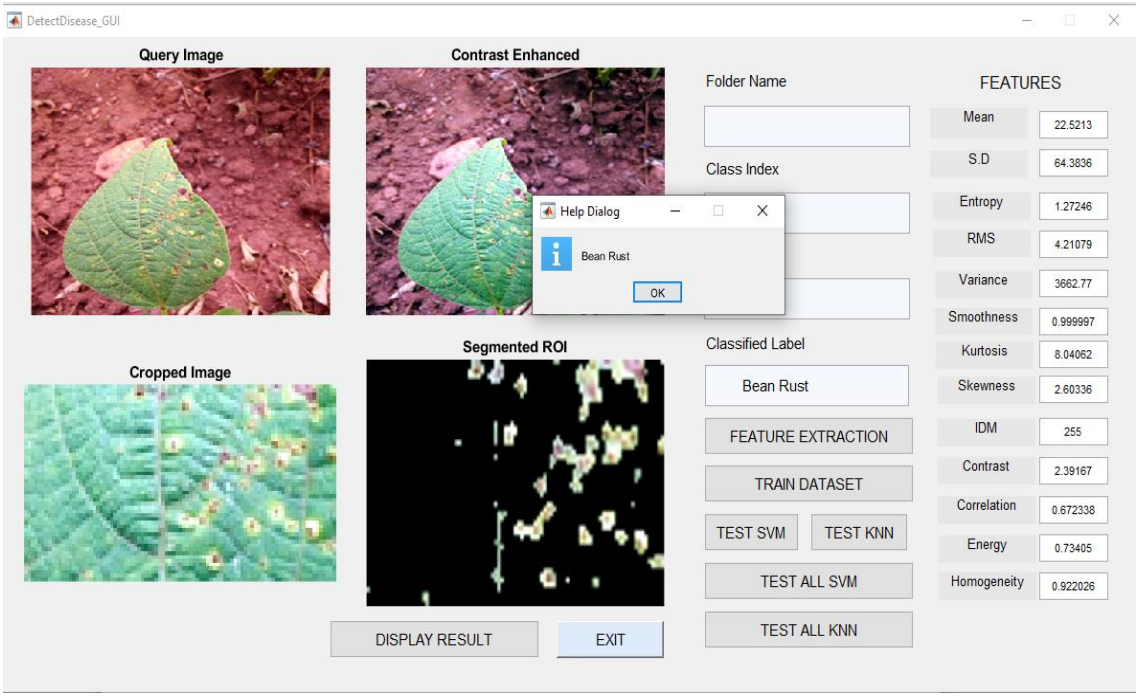


**Figure 4.13 Classification result for bean rust test with k-NN**

Figure 4.13 shows a classification result image generated by the k-nearest neighbor algorithm based on the extracted texture and color features of the images. As a result, it is an bean rust.

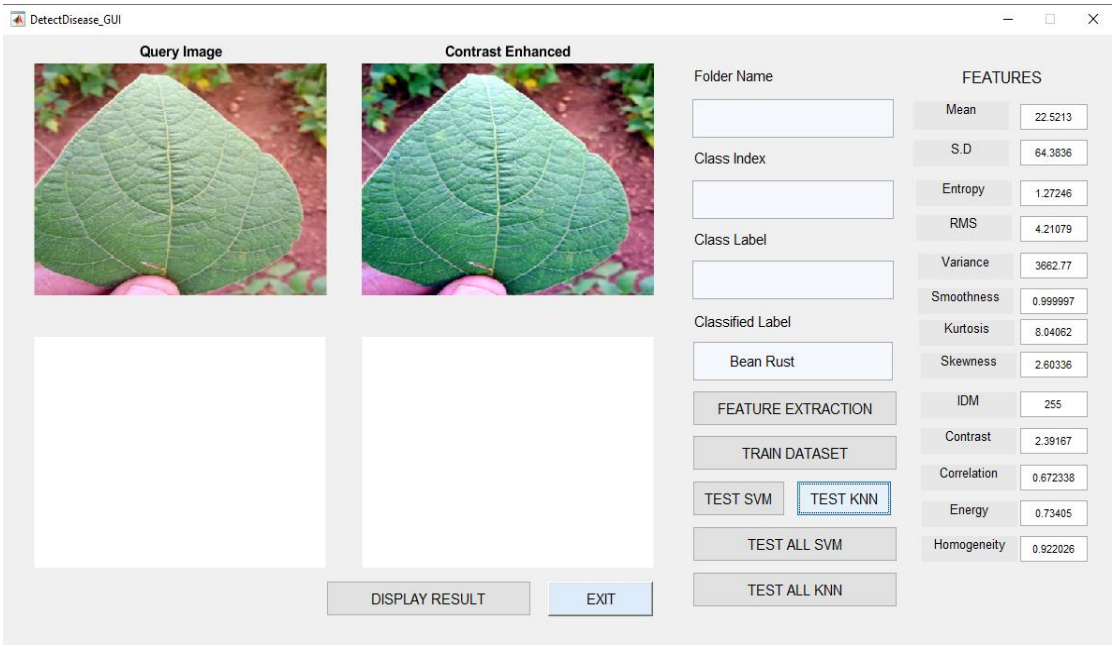


**Figure 4.14 Preprocessing input image for healthy test with k-NN**

Figure 4.14 depicts that an input image is preprocessed by using the contrast enhancement technique for healthy.

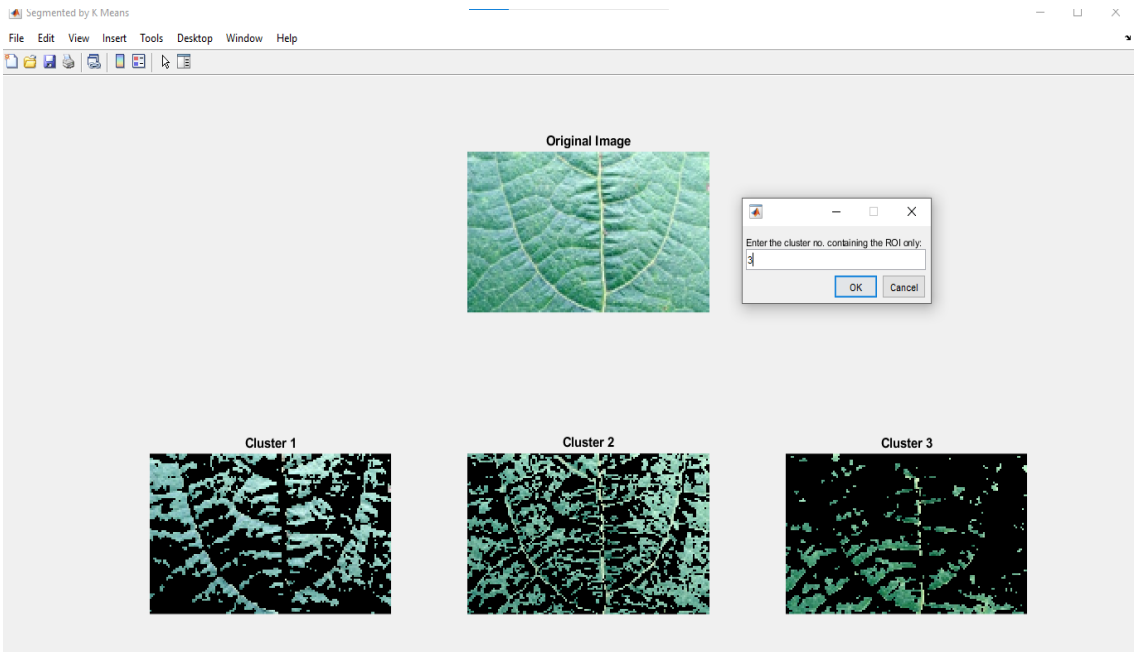


**Figure 4.15 Clustering image result for healthy test with k-NN**

Figure 4.15 shows an contrast enhancement image is segmented by using the k-means clustering technique that it specify the region of interest in the leaf's affect area.

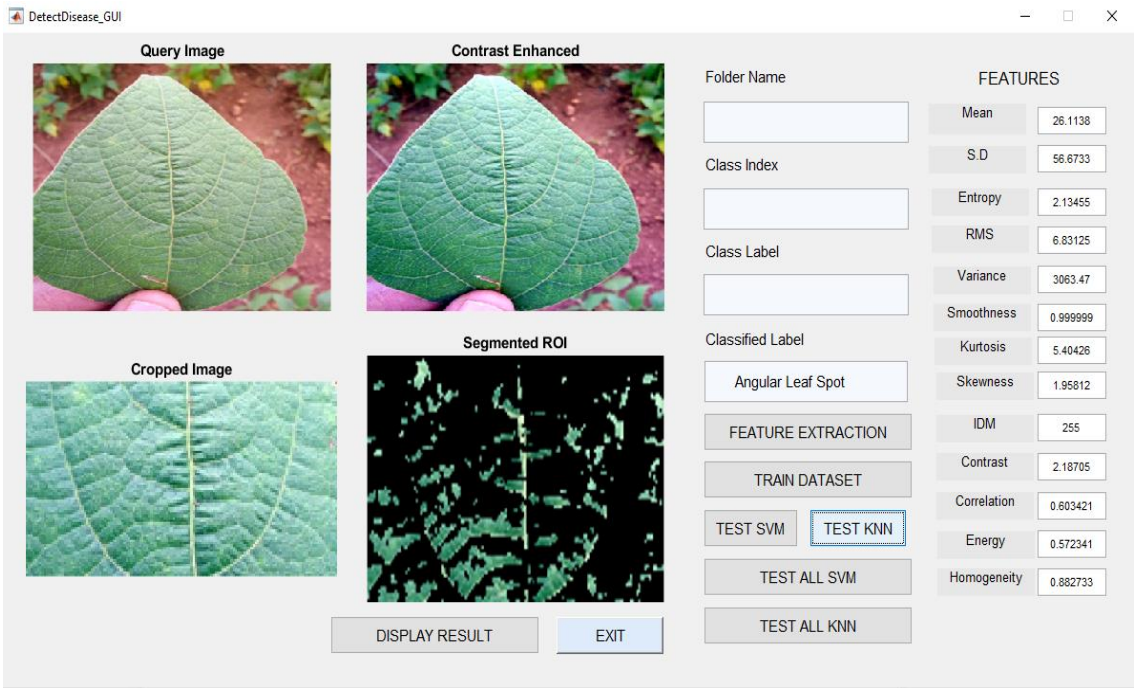


**Figure 4.16 Feature output result for healthy test with k-NN**

Figure 4.12 shows that extracted features result from preprocessing and segmented images.

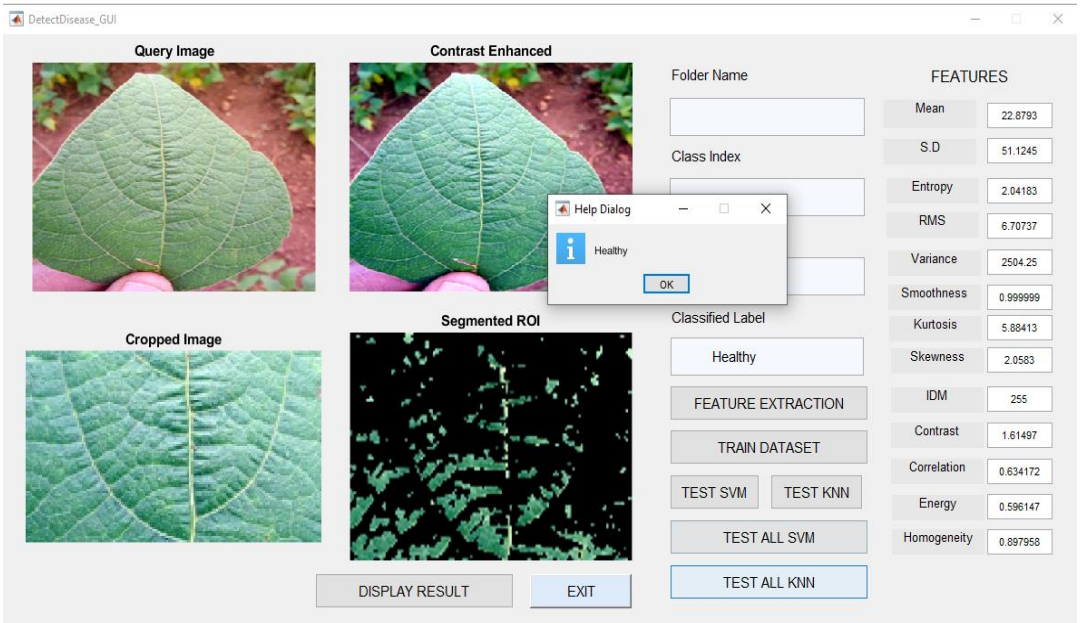


**Figure 4.17 Classification result for healthy test with k-NN**

Figure 4.17 shows a classification result image generated by the k-nearest neighbor algorithm based on the extracted texture and color features of the images. As a result, it is an healthy.

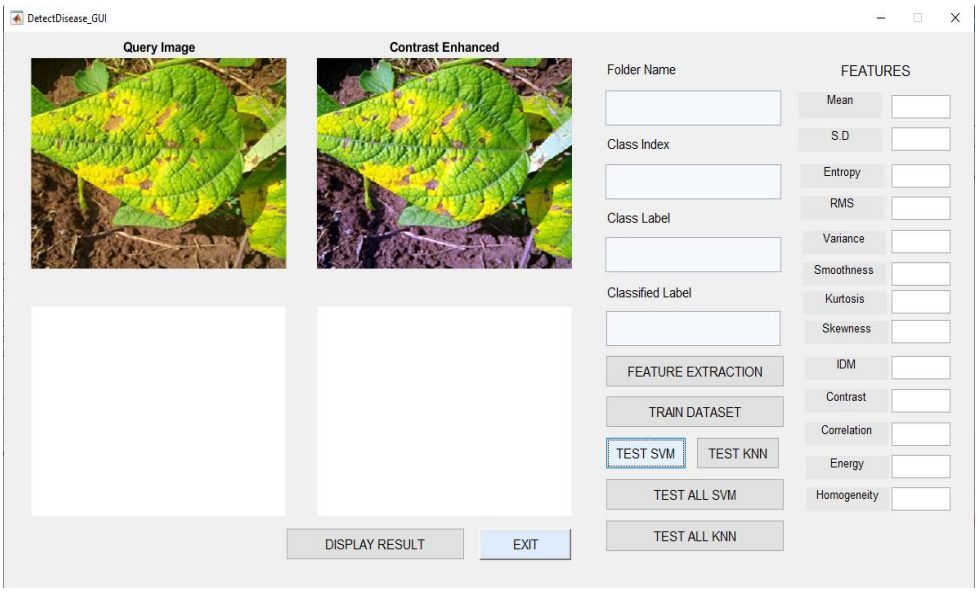


**Figure 4.18 Preprocessing input image for angular leaf spot test with SVM**

Figure 4.18 show that an input image is preprocessed by using the contrast enhancement technique for angular leaf spot.

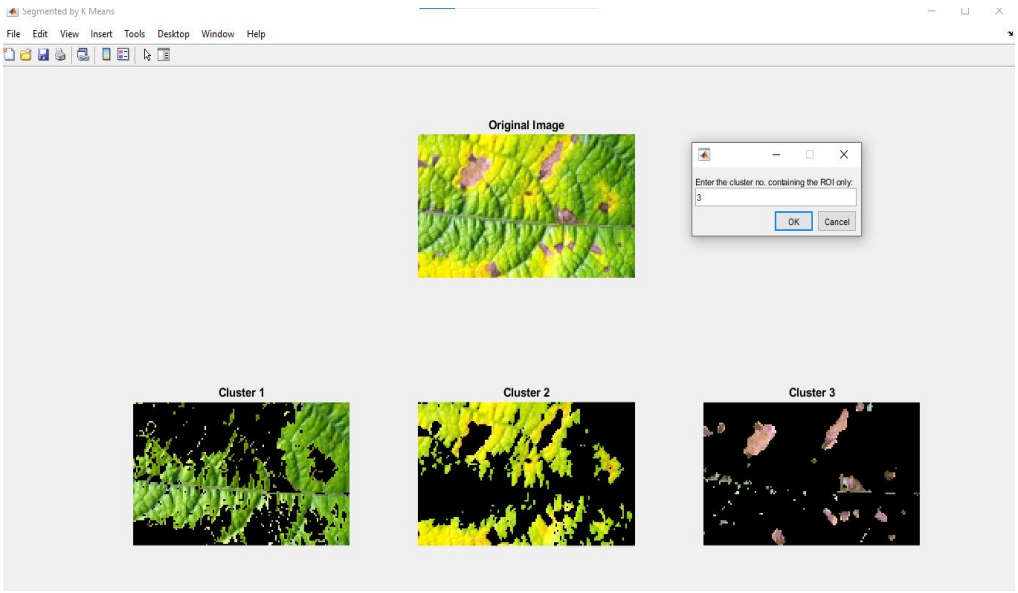


**Figure 4.19 Clustering image result for angular leaf spot test with SVM**

Figure 4.19 shows a contrast enhancement image is segmented by using the k-means clustering technique that it specify the region of interest in the leaf's affect area.

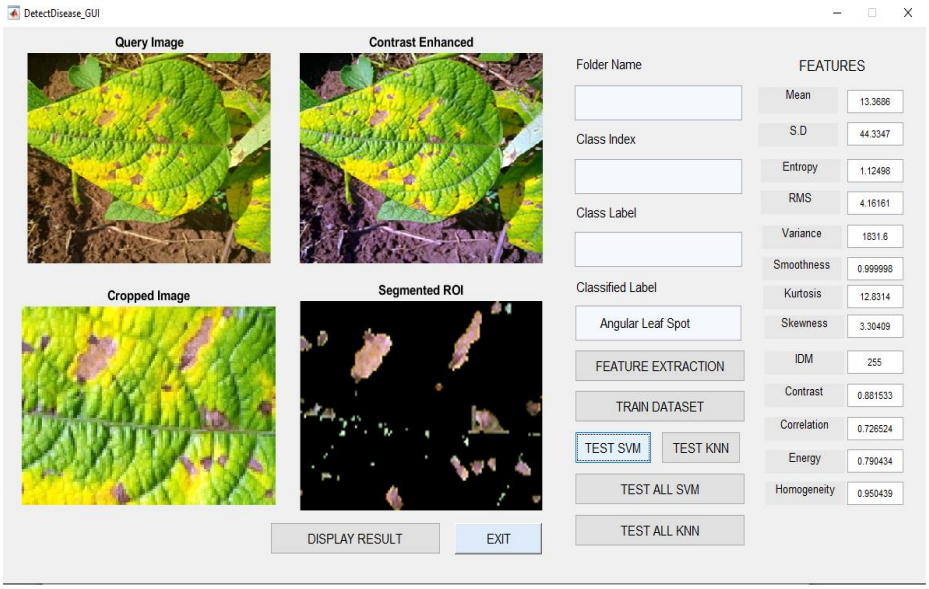


**Figure 4.20 Feature output result for angular leaf spot test with SVM**

Figure 4.20 shows that extracted features result from preprocessing and segmented images.

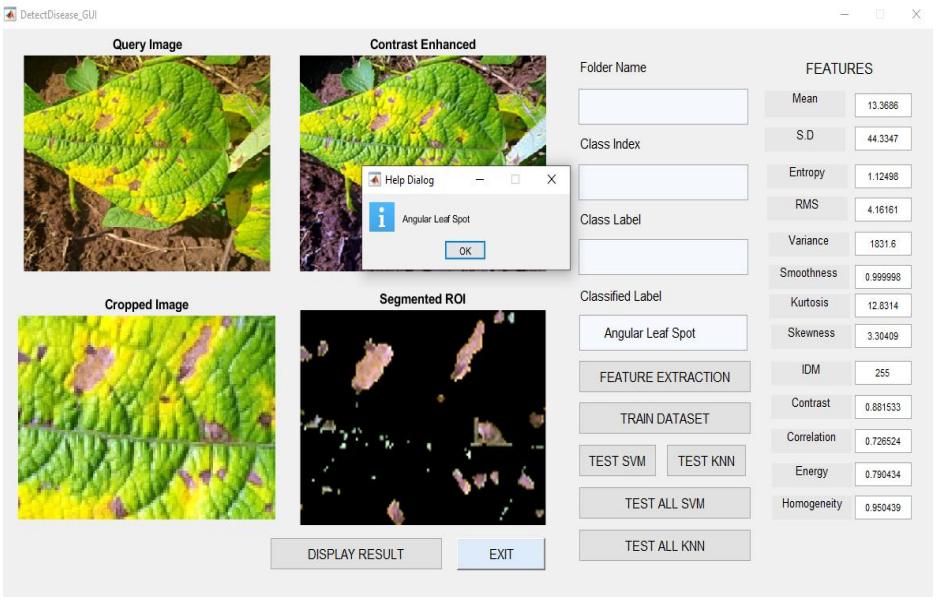


**Figure 4.21 Classification result for angular leaf spot test with SVM**

Figure 4.21 shows a classification result image generated by the support vector machine based on the extracted texture and color features of the images. As a result, it is an angular leaf spot.

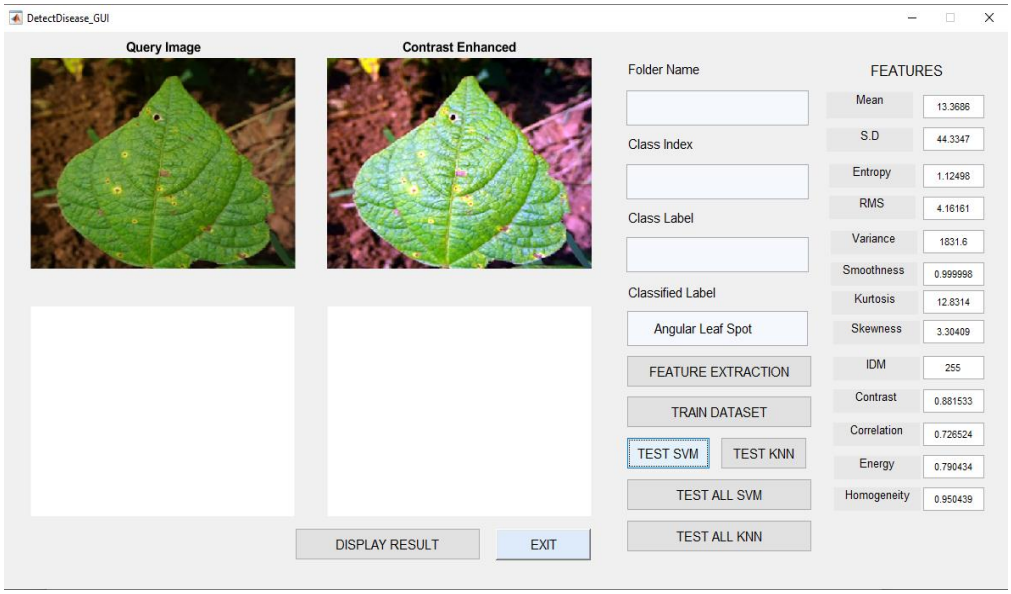


**Figure 4.22 Preprocessing input image for bean rust test with SVM**

Figure 4.18 show that an input image is preprocessed by using the contrast enhancement technique for bean rust.

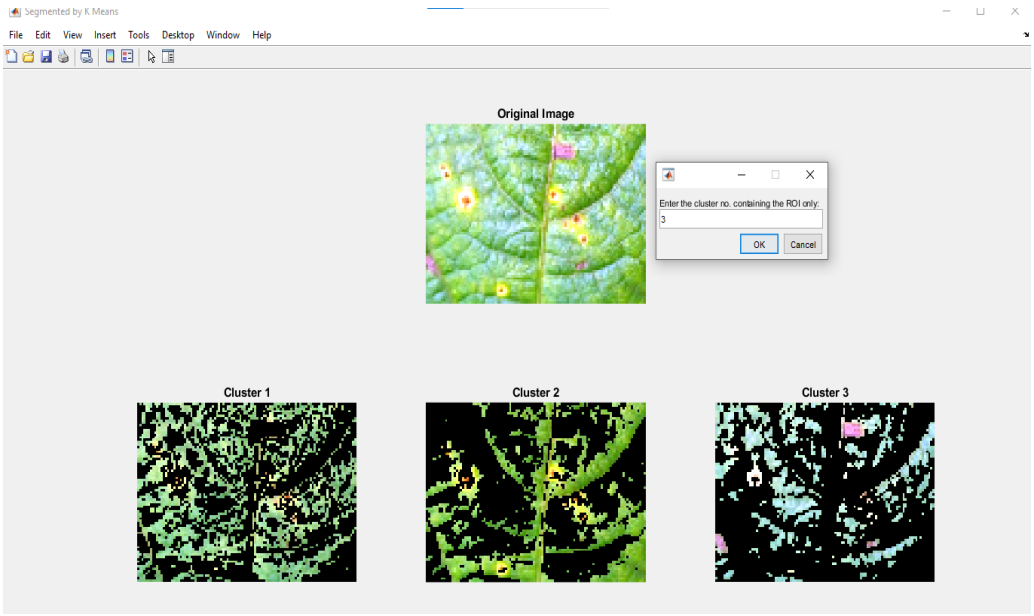


**Figure 4.23 Clustering image result for bean rust test with SVM**

Figure 4.23 shows an contrast enhancement image is segmented by using the k-means clustering technique that it specify the region of interest in the leaf's affect area.

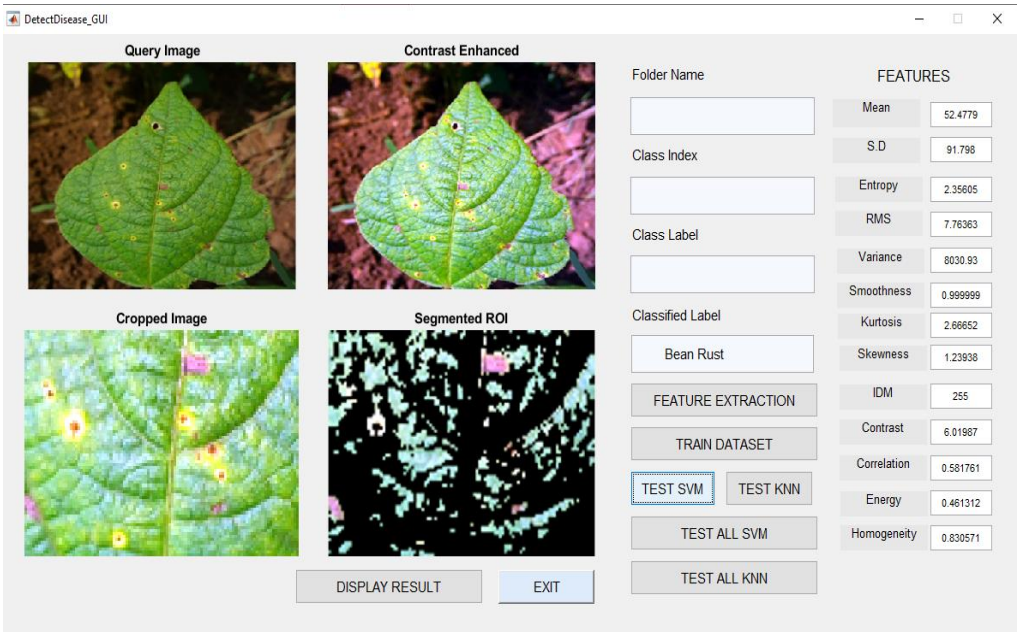


**Figure 4.24 Feature output result for bean rust test with SVM**

Figure 4.24 shows that extracted features result from preprocessing and segmented images.

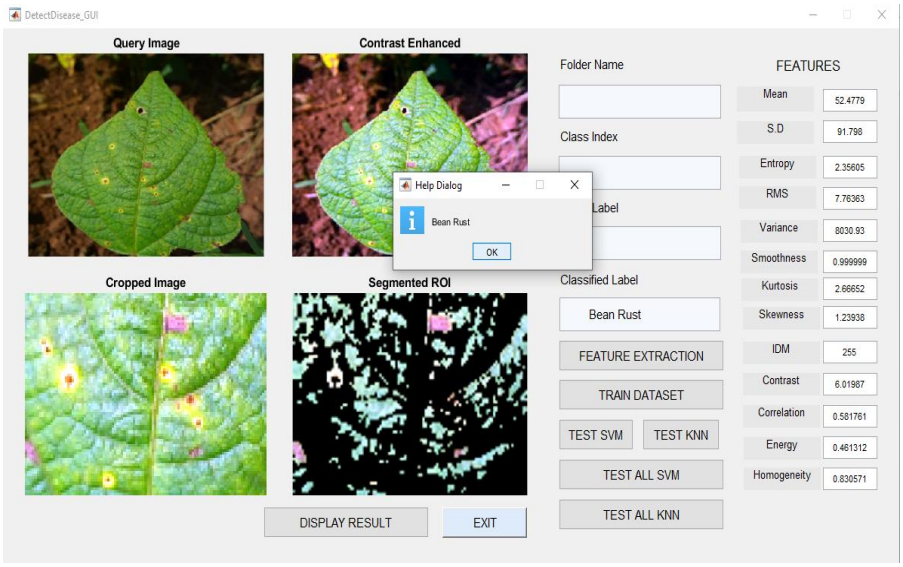


**Figure 4.25 Classification result for bean rust test with SVM**

Figure 4.25 shows a classification result image generated by the support vector machine based on the extracted texture and color features of the images. As a result, it is a  bean rust.

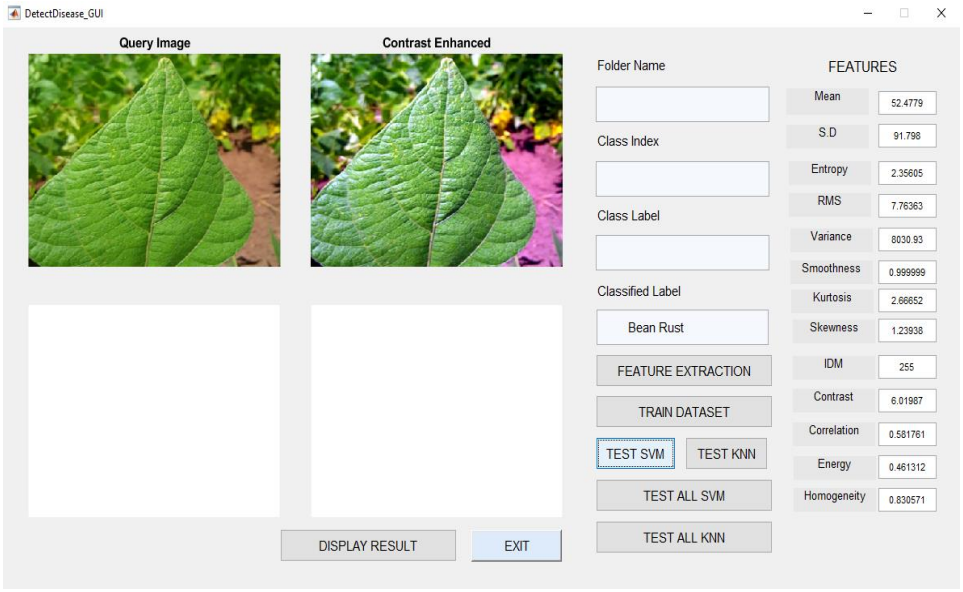


**Figure 4.26 Preprocessing input image for healthy test with SVM**

Figure 4.26 show that an input image is preprocessed by using the contrast enhancement technique for bean rust.

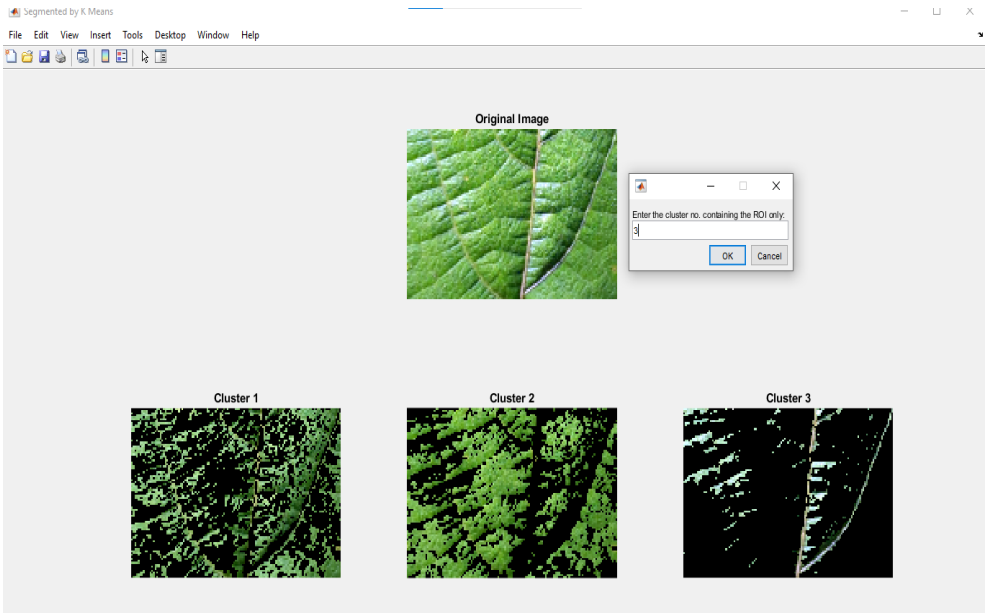


**Figure 4.27 Clustering image result for healthy test with SVM**

Figure 4.27 shows an contrast enhancement image is segmented by using the k-means clustering technique that it specify the region of interest in the leaf's affect area.

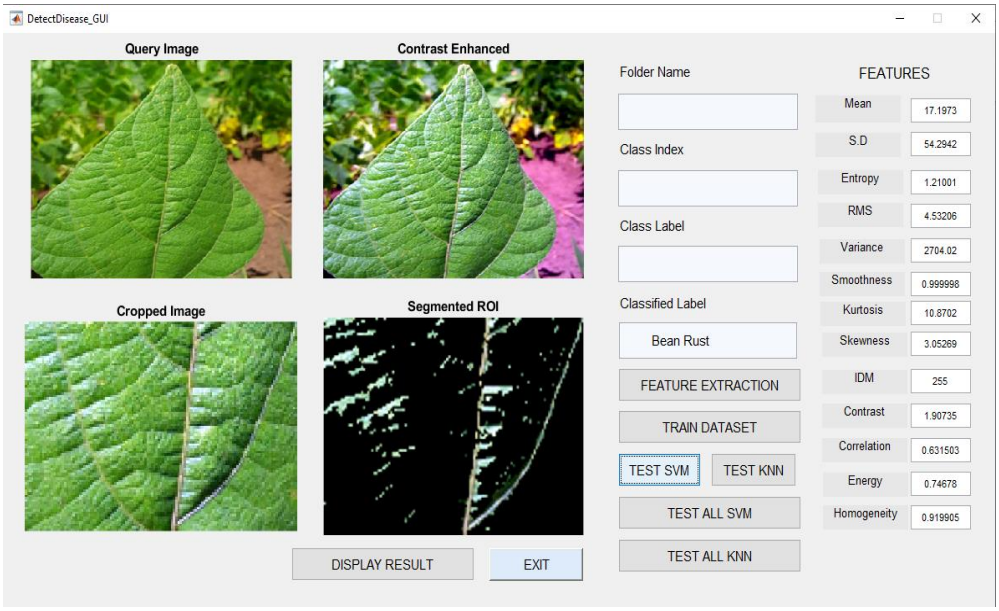


**Figure 4.28 Feature output result for healthy test with SVM**

Figure 4.24 shows that extracted features result from preprocessing and segmented images.

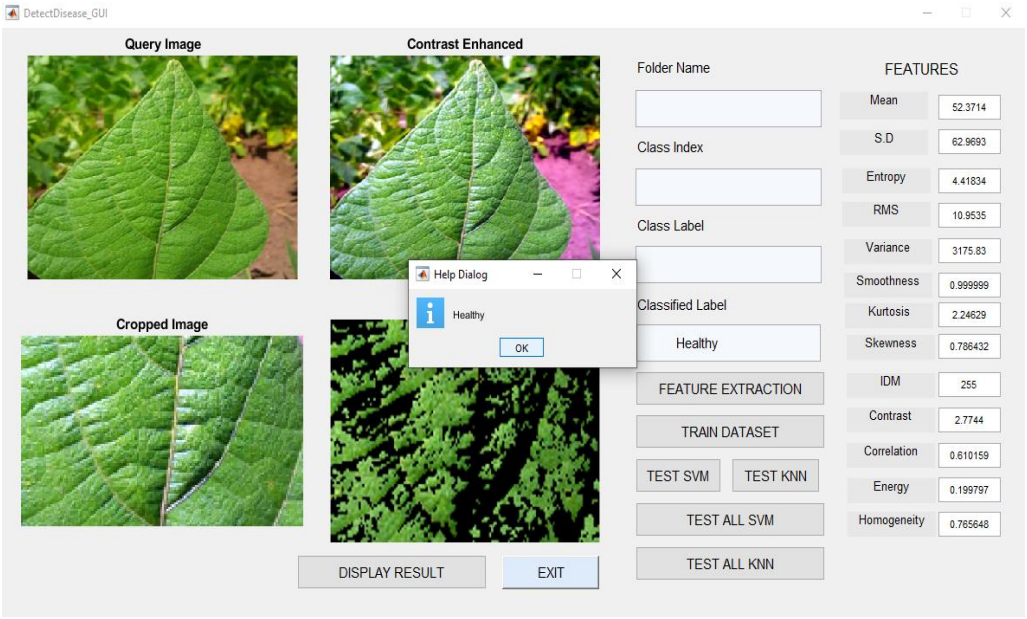


**Figure 4.29 Classification result for healthy test with SVM**

Figure 4.29 shows a classification result image generated by the support vector machine based on the extracted texture and color features of the images. As a result, it is a healthy.

## 4.4 Experimental Result

The proposed system's performance is evaluated using two factors: Accuracy and Precision.

### 4.4.1 Accuracy

A classifier's accuracy is expressed as a percentage of total correct predictions divided by total number of instances. If the classifier's accuracy is deemed acceptable, it can be used to classify future data tuples for which the class label is unknown. One metric for evaluating classification models is accuracy. Accuracy is defined as the percentage of correct predictions made by model. Accuracy is defined formally as follows:

- Accuracy is a critical component in data mining and machine learning module performance because the success of a module is dependent on its accuracy because measurement accuracy shows how close it is to its true value.

- A test's accuracy is defined as its ability to correctly distinguish between healthy and unhealthy cases. To estimate a test's accuracy, the proportion of true positive and true negative results must be computed in all evaluated cases.

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

Where,

- True positive= the number of positive cases correctly identified for disease
- False positive= the number of cases misidentified as positive for disease
- True negative=the number of negative cases correctly identified for disease
- False negative=the number of cases misidentified as negative for disease

### 4.4.2 Precision

Precision is defined as "the quality of being exact," and it refers to how close two or more measurements are to each other, regardless of accuracy. Precision measurements can be accurate or inaccurate. Precision is defined mathematically as the number of true positives divided by the number of true positives plus the number of false positives. Precision is distinct from accuracy. It is required to achieve the highest possible quality measurement. It is not necessary for a set of measurements to be completely accurate in order for them to be precise. This happens because a series of measurements are precise as long as they are grouped by value.

$$\text{Precision} = (TP)/(TP+FP)$$

**Table 4.1 Calculation Confusion Matrix**

| Actual Class | Predicted Class | |
|---|---|---|
| | True | False |
| True | True Positive(TP) | False Negative(FN) |
| False | False Positive(FP) | True Negative(TN) |

Table 4.1 shows a confusion equation matrix with two labels, True and False, and four different predictive and actual value combinations.

**Table 4.2 Calculation Precision in confusion matrix for K-NN**

| Predicted Class | | | | |
|---|---|---|---|---|
| Angular leaf spot | Bean Rust | Healthy | | |
| 29 | 1 | - | Angular leaf spot | Actual Class |
| 6 | 23 | 1 | Bean Rust | |
| 2 | 3 | 25 | Healthy | |

**Table 4.3 Calculation Precision in confusion matrix for SVM**

| Predicted Class | | | | |
|---|---|---|---|---|
| Angular leaf spot | Bean Rust | Healthy | | |
| 26 | 3 | 1 | Angular leaf spot | Actual Class |
| 5 | 20 | 5 | Bean Rust | |
| 3 | 4 | 23 | Healthy | |

Tables 4.2 and 4.3 show the precision values for angular leaf spot, bean rust and healthy in a k-NN and SVM confusion matrix. The first row of the confusion matrix shows the total number of testing times, as do the second and third rows. Diagonal values are the number of times the actual class matches the predicted class. Other values represent the number of incorrect times.

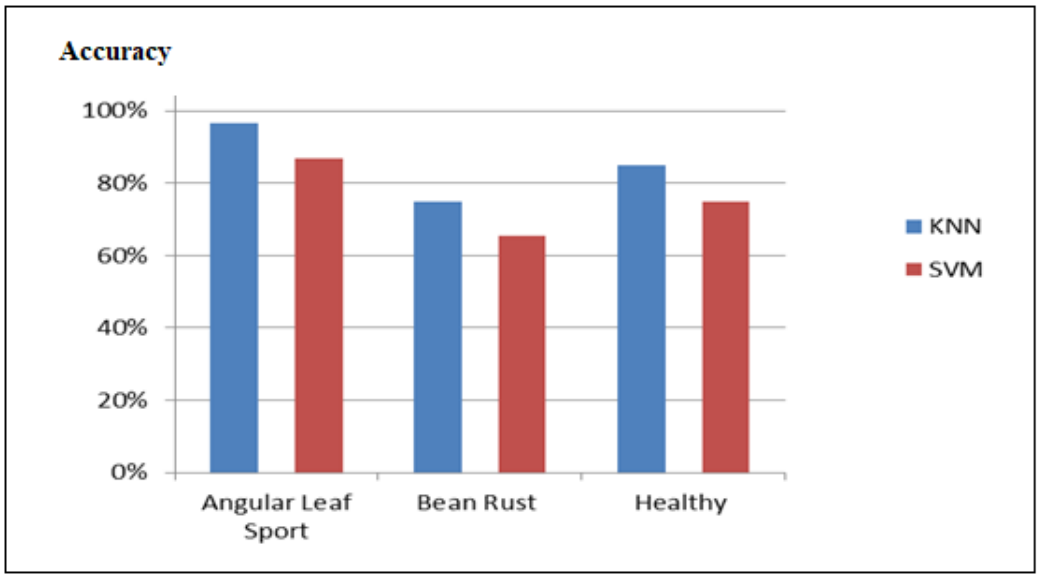


**Figure 4.30 Evaluation Result**

Figure 4.30 depicts the accuracy values for k-NN and SVM using barchart. According to the results of the tests, the k-NN algorithm is more accurate than the SVM algorithm. In this system, the k-NN algorithm can classify disease types with 96.7% accuracy and the SVM algorithm with 86.7%.

# CHAPTER 5
# CONCLUSION

Agriculture is vital to the economy, and many more modern technologies are used than traditional technologies. In this system, disease detection is accomplished through the use of image processing and machine learning algorithms. The k-NN algorithm can improve classification results for mungbean while being simpler to implement. The image classification of mungbean leaf disease using k-NN and GLMC for feature extraction is extremely accurate. To achieve high accuracy, this method can be developed using the optimization method of attribute selection or bagging techniques, and it can be applied to applications to make it easier for farmers to prevent crop failure and provide automatic detection of mungbean leaf disease, making it efficient for the agricultural sector.

## 5.1 Advantages and Limitations of the System

One of the most important aspects of a plant pathologist's education is diagnosis. Disease control measures can be a waste of time and money if the disease and the disease-causing agent are not properly identified. This can lead to additional plant losses. As a result, accurate disease diagnosis is critical. In this system, k-NN classifier will classify the diseases like mungbean leaf diseases such as angular leaf spot, bean rust and healthy. The proposed approach can successfully detect and recognize the selected diseases with 96.76 % accuracy and this classifier also classify other various plant species.

This research can be expanded in a variety of ways. But there are also numerous classifications methods can be applied leaf disease detection. Thus, any other extended version of algorithm can be used for classification. Other improved methods should be used to improve the performance of accuracy.

# REFERENCES

[1]Alvarez A.M, Integrated approaches for detection of plant pathogenic bacteria and diagnosis of bacterial diseases Annu. Rev. Phytopathol. 42 339-366,2004.

[2]A.Bashish, Dheeb, Malik Braik, and Sulieman Bani-Ahmad. “A framework nference on signal and image processing.. IEEE, 2010.

[3]Al.Bashish D, Braik M, Bani-Ahmad S (2010) A Framework for Detection and Classification of Plant Leaf and Stem Diseases. Int Conf on Signal and Image Processing IEEE.

[4]Adi K, Pujiyanto S, Nurhayati O D and Pamungkas A 2015 Beef Quality Identification using Color Analysis and K-Nearest Neighbor Classification International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME) pp 180–4.

[5]Dhanachandra, Nameirakpam, Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm /India/December 2015.

[6]Dheeb Albashish, Detection and Classification of Leaf Disease using K-means-based Segmentation and Neural-networks-based Classification /India/February 2011.

[7]Dixit A. and Nema S, Wheat Leaf Disease Detection Using Machine Learning Method-A Review Int. J. Comput. Sci. Mob. Comput. 7 124-129,2018.

[8]Elangovan K, Plant Disease Classification Using Image Segmentation and SVM Techniques Int. J. Comput. Intell. Res. 13 1821-1828, 2017.

[9]Fuentes Alvaro, Lee Yujeong, Hong Youngki, Yoon Sook and Park Dong, Characteristics of Tomato Plant Diseases - A study for tomato plant disease identification A genetic algorithm-based feature selection ed Oluleye H. Babatunde et al 2014.

[10]G.Geetha ,S.Samundeswari , G.Saranya ,K.Meenakshi and M. Nithya, “Plant Leaf Disease Classification and Detection System Using Machine Learning”, ICCPET, 2020.

[11]H.Al-Hiary, "Fast and accurate detection and classification of plant diseases",Int J.Comput. Appl., vol. 17, no. 1, pp. 31-38, 2011.

[12]Indriani O. R., Kusuma E. J., Sari C. A., Rachmawanto E. H. and Setiadi D. R. I. M, Tomatoes classification using K-NN based on GLCM and HSV color

space 1-6, . 2018 Proc. - 2017 Int. Conf Innov. Creat. Inf. Technol. Comput. Intell. IoT, ICITech 2017 .

[13]Kartika D. S. Y., Herumurti D. and Yuniarti A, Butterfly Image Classification Using Color Quantization Method on HSV Color Space and Local Binary Pattern IPTEK J. Proc. Ser. 4 78,2018.

[14]Malti K. Singh, Detection and Classification of Plant Leaf Diseases in Image Processing using MATLAB /India/ December 2017.

[15]P.M. Mainkar, S. Ghorpade, and M. Adawadkar, "Plant leaf disease detection and classification using image processing techniques", International Journal of Innovative and Emerging Research in Engineering, vol. 2, no. 4, pp. 139-144,2015.

[16]Puspha L. S., Annabel, Annapoorani T. and Deepalakshmi P, Machine Learning for Plant Leaf Disease Detection and Classification, International Conference on Communication and Signal Processing (ICCSP) (Chennai, India), 2019.

[17]Pawar P., Turkar V. and Patil P, Coimbatore Cucumber disease detection using artificial neural network , International Conference on Inventive Computation Technologies (ICICT), 2016.

[18]S.Agustin and R.Dijaya gave a presentation titled "Beef Image Classification using the k-NN Algorithm for Identification Quality and Freshness".Computer Science. Journal of Physics: Conference Series, 1,July, 2019.

[19]U.P.Singh, S.S.Chouhan and S.JAIN."Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease" IEEE Access journal, v.7, March 2019.

[20]Vagisha Sharma, Amandeep Verma, Neelam Goel ,“Classification Techniques for Plant Disease Detection”, IJRTE, 2020.

[21]Weizheng S, Yachun W, Zhanliang C, Hongda3 W (2008) Grading Method of Leaf Spot Disease Based on Image Processing. Int Conf on Comp Sc and Soft Eng, IEEE 491-494.

[22]XE.Pantazi , D.Moshou and AA.Tamouridou "Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers." Comput Electron Agric v.156, pp.96–104, 2019.

# AUTHOR'S PUBLICATION

[1] Hnin Pwint Zaw, Khant Kyawt Kyawt Theint, "Mungbean Leaf Disease Detection Using K-Nearest Neighbor Algorithm", the Proceedings of the Conference on Parallel and Soft Computing (PSC 2023), Yangon, Myanmar, 2023.