2-1-2017

Automated early plant disease detection and grading system: Development and implementation

Hashem Rizk

THE AMERICAN UNIVERSITY IN CAIRO

School of Sciences and Engineering

Mechanical Engineering Department

Automated Early Plant Disease Detection and Grading System:

Development and Implementation

By

Hashem Rizk

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Under supervision of

Dr. Maki K. Habib

Professor, Mechanical Engineering Department

December 2017

Acknowledgement

I would like to thank my supervisor, Dr. Maki Habib, for his unwavering support, guidance and insight throughout this thesis project. I am extremely grateful for the time and patience he has put in during my time as his student.

I would also like to acknowledge and thank the following people who have supported me, not only during the course of this research project, but throughout my Masters degree.

Firstly, I would like to thank my fellow colleagues, Mohammed Gamal for his tremendous input and expertise during the design stages of the hardware, and Chimson Chukwuemeka for his supportive advice and help during the testing stages. I also would like to thank Joseph and Mina Adel for their knowledge and assistance during the manufacturing of the hardware.

I would also like to thank the staff responsible at the RISE facility for allowing me to undergo my tests in their greenhouse and providing help whenever needed.

A special apperception for my fellow graduate student Khaled Abouelsoud for his friendship and constant motivation during my Masters degree.

Finally, I would like to thank all my close friends and family. Your encouragement has helped me focus on what has been a rewarding and enriching process. This thesis would not be possible without your collective support.

Abstract

As the agriculture industry grows, many attempts have been made to ensure high quality of produce. Diseases and defects found in plants and crops, affect the agriculture industry greatly. Hence, many techniques and technologies have been developed to help solving or reducing the impact of plant diseases. Imagining analysis tools, and gas sensors are becoming more frequently integrated into smart systems for plant disease detection. Many disease detection systems incorporate imaging analysis tools and Volatile Organic Compound (VOC) profiling techniques to detect early symptoms of diseases and defects of plants, fruits and vegetative produce. These disease detection techniques can be further categorized into two main groups; preharvest disease detection and postharvest disease detection techniques. This thesis aims to introduce the available disease detection techniques and to compare it with the latest innovative smart systems that feature visible imaging, hyperspectral imaging, and VOC profiling.

In addition, this thesis incorporates the use of image analysis tools and k-means segmentation to implement a preharvest Offline and Online disease detection system. The Offline system to be used by pathologists and agriculturists to measure plant leaf disease severity levels. K-means segmentation and triangle thresholding techniques are used together to achieve good background segmentation of leaf images. Moreover, a Mamdani-Type Fuzzy Logic classification technique is used to accurately categorize leaf disease severity level. Leaf images taken from a real field with varying resolutions were tested using the implemented system to observe its effect on disease grade classification. Background segmentation using k-means clustering and triangle thresholding proved to be effective, even in non-uniform lighting conditions. Integration of a Fuzzy Logic system for leaf disease severity level classification yielded in classification accuracies of 98%.

Furthermore, a robot is designed and implemented as a robotized Online system to provide field based analysis of plant health using visible and near infrared spectroscopy. Fusion of visible and near infrared images are used to calculate the Normalized Deference Vegetative Index (NDVI) to measure and monitor plant health. The robot is designed to have the functionality of moving across a specified path within an agriculture field and provide health information of leaves as well as position data. The system was tested in a tomato greenhouse under real field conditions. The developed system proved effective in accurately classifying plant health into one of 3 classes; underdeveloped, unhealthy, and healthy with an accuracy of 83%. A map with plant health and locations is produced for farmers and agriculturists to monitor the plant health across different areas. This system has the capability of providing early vital health analysis of plants for immediate action and possible selective pesticide spraying.

Table of Contents

List of Figures

List of Tables

Nomenclature

List of Abbreviations

Acronym	Description
ANN	Artificial Neural Network
BPNN	Back-propagation Neural Network
CCV	Color Coherence Vector
CLBP	Complete Local Binary Pattern
ELISA	Enzyme-linked Immunosorbent Assay
GC	Gas Chromatography
GCH	Global Color Histogram
GMO	Genetically Modified Organisms
GPS	Global Positioning System
GUI	Graphical User Interface
HSV	Hue, Saturation, Value Color Space
I_{WB}	Water Band Index
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LDC	Linear Discriminant Classifier
MCARI	Modified Chlorophyll Absorption in Reflectance Index
MLP	Multilayer Perceptron
MSVM	Multi-class Support Vector Machine
NDVI	Normalized Differential Vegetative Index
NIR	Near Infrared
PCA	Principal Component Analysis
PCR	Polymerase Chain Reaction
PRI	Photochemical Reflectance Index
QDA	Quadratic Discriminant Analysis
RED	Red Channel
RGB	Red, Green, Blue Color Space
RVSI	Red-edge Vegetation Stress Index
SAM	Spectral Angle Mapping

SAVI	Soil-adjusted Vegetation Index
SFF	Spectral Feature Fitting
SID	Spectral Information Divergence
SIFT	Scale-Invariant Feature Transform
SQM	Self-organizing Map
SURF	Speeded up Robust Features
SVM	Support Vector Machine
TMV	Tobacco Mosaic Virus
UART	Universal Asynchronous Receiver-Transmitter
UV	Ultraviolet Light
VARI	Visible Atmospherically Resistance Index
VOC	Volatile Organic Compound
WI	Water Index

List of Symbols

Symbol	Description
b	Brightness
w	Weighted mean
μ	Mean
σ	Variance
σ_W^2	Within Class Variance
σ_B^2	Between Class Variance
p	Probability
$I1$	Chlorophyll Absorption index
$I2$	Chlorophyll Absorption index
I	Florence intensity
R	Reflectance

Chapter 1
Introduction

1.1. Study Background

The agriculture industry is one of the most vital sectors for contribution to the national income in many countries. Throughout the years, many agriculture components and processes have become automated to ensure faster production and to ensure products are of highest quality standards. Because of the increased demand in the agricultural industry, it is vital that agriculture produce is cultivated using an efficient process [1]. Diseases and defects found in plants and crops have a great impact on production in the agriculture industry, and lead to significant economic losses [2]. A loss of an estimated 33 billion dollars every year was the result of plant pathogens found in crops in the United States. Pathogenic species affect plants significantly, introducing diseases such as chestnut blight fungus and huanglongbing citrus disease [3]. Insect infestation along with bacterial, fungal, and viral infections are another main contribution to diseases found in plants [4]. Changes in climate and temperature are also a few factors that may contribute to the increase in diseases found in plants. Once a plant has been infected, symptoms develop on various segments of the plant, ultimately degrading the growth of the subsequent fruit or vegetable [5].

Apple production is a very large industry especially in China with over 17 million tons of produce every year [2]. Apple infections do not only significantly reduce grade and yield, but can also affect the return bloom of the following season [6]. These factors have drastic impact on countries that rely heavily on its agriculture sector as its main method of income. To overcome these losses and issues of plant diseases, farmers tend to look to chemical pesticides as a remedy solution. This solution may be effective in eliminating plant diseases but has drastic drawbacks. As well as being costly, the increase use of pesticides creates dangerous levels of toxic residue levels on agriculture products [7]. This leads to concerns about wholesomeness and healthiness of products raised by the public when pesticides are commonly used in the produce they purchase [8]. Therefore, the use of pesticides must be controlled, and used only when necessary. This controlled or monitored method of pesticide use is known as selective pesticide spraying.

For the purpose to decrease losses found in defective plants many techniques have been introduced. Manual techniques, such as hand inspection and naked eye observation are very common methods used by farmers. Plant diseases are detected and characterized by observation from experts, which can be very expensive and time consuming [2]. Because these methods are very tedious it is prone to sorting errors and judgmental errors from different farmers [6]. Therefore, disease detection systems were introduced that tackle many of the issues faced with labor-intensive techniques.

1.2. Statement of Purpose

Bacterial and fungal viral infections have a significant impact on plant health and introduce diseases that affect growth of produce. In addition, the over reliance on fungicides and pesticides to remedy this issue, is not only costly, but has a considerably negative impact on the environment. Therefore, there is a need to detect and target plant diseases at an early stage to aid farmers to take appropriate precautions to help preserve the defective plant. The purpose of this thesis is to discuss and compare the current plant disease detection techniques, that feature visible imaging, hyperspectral imaging, and VOC profiling methods. An easy to use system to detect plant leaf disease severity is designed for farmers and agriculturists to measure disease severity levels of plants. In addition, an automated approach is designed and implemented for early leaf based plant health monitoring using a robotized system in real field based environments.

1.3. Objectives and Expected Outcomes

The objectives of this thesis are as follows:

1. To study and explore the different tools used for plant disease detection techniques:
    a) Florescence Spectroscopy
    b) Visible and Infrared Spectroscopy
    c) Fluorescence Imaging
    d) Hyperspectral Imaging
    e) Volatile Organic Compound Profiling

2. Categorize and compare the included research into Preharvest and Postharvest disease detection methods.
3. Introduce an automated technique for early plant disease detection and monitoring.

The expected outcomes of the thesis include:

1. The implementation of an automatic plant disease severity level system.
2. The design and implementation of an automated robotized system for plant health monitoring in real field conditions.

1.4. Thesis Structure

The remaining part of the thesis includes:

Chapter 2 introduces the literature review of research activities dealing with plant disease detection techniques. Indirect and Direct methods are explained and different techniques from each method are analyzed and compared, taking into consideration the achievements and challenges of each technique. Chapter 3 introduces two plant disease detection methods according to the research challenges. It also focuses on the outline of the proposed systems and the expected objectives. Chapter 4 discusses the development of the first method and the image processing techniques used to measure disease severity of plant leaves. The implemented GUI is tested with multiple leaf images to evaluate the systems performance and accuracy. Chapter 5 presents the second developed system, in which a robot is used for automated health monitoring of plants. The robot design and image processing techniques are implemented and tested in a greenhouse environment. Chapter 6 covers the conclusions observed from both systems, and the applications in which the systems can be used for. Finally, future work is recommended for the proposed systems for higher disease and health accuracy and broader applications are considered.

Chapter 2

Literature Review and Research Challenges

A disease detection system has the ability to not only detect early symptoms of defective plants, but can also avoid the disease from spreading. Disease detection techniques can be categorized into two methods: direct and indirect methods as shown in Figure 2-1. Direct detection techniques rely on the use of laboratory-based experiments. The most popular and commonly used experiments are enzyme-linked immunosorbent assay (ELISA) and polymerase chain reaction (PCR) [4]. Indirect methods, on the other hand, rely on advanced techniques with an emphasis on the integration of imaging tools. Indirect methods use the integration of sensors and smart systems on site, to provide a more rapid and accurate method for disease detection. Early detection of apparent diseases in plants is of utmost importance [1], as this will aid farmers to take appropriate precautions to help preserve the defective plant. Indirect methods are vast and can be used for disease detection in both preharvest plants and postharvest fruits. If early detection is possible, the percent of defective fruits can be significantly decreased, while maintaining high quality production standards.

2.1. Direct Methods

When a pathogen attacks a plant, the plant DNA is altered, and a specific type of protein molecules are produced and introduced to the plant by the pathogen during infection. Direct methods focus on molecular and serological techniques that test the biological structure of the plant to check for the pathogen DNA or the presence of pathogen produced protein molecules. Commonly known techniques are the Polymerase chain reaction (PCR) and the Enzyme-linked immunosorbent assay (ELISA).

PCR-based disease detection involves genetic material (DNA) extraction of the microorganism causing the disease. The gel electrophoresis is then performed after the DNA has been purified and amplified. If a specific band is present in the gel electrophoresis, then the existence of the plant disease organism is verified [4].

ELISA works by injecting the microbial protein of a specific plant disease into an animal, which in return produces antibodies against that specific disease. The extracted antibodies are used alongside with fluorescence dye and enzymes for disease detection. If the plants were infected, then the sample would fluoresce, verifying the presence of a specific plant disease [4]. Table 2-1 illustrates the difference between each technique and how they compare in disease detection. Because of these techniques, diagnostic kits have been designed to successfully detect diseases in crops such as rice and can also identify genetically modified organisms (GMOs) in shipments of conventional crops.

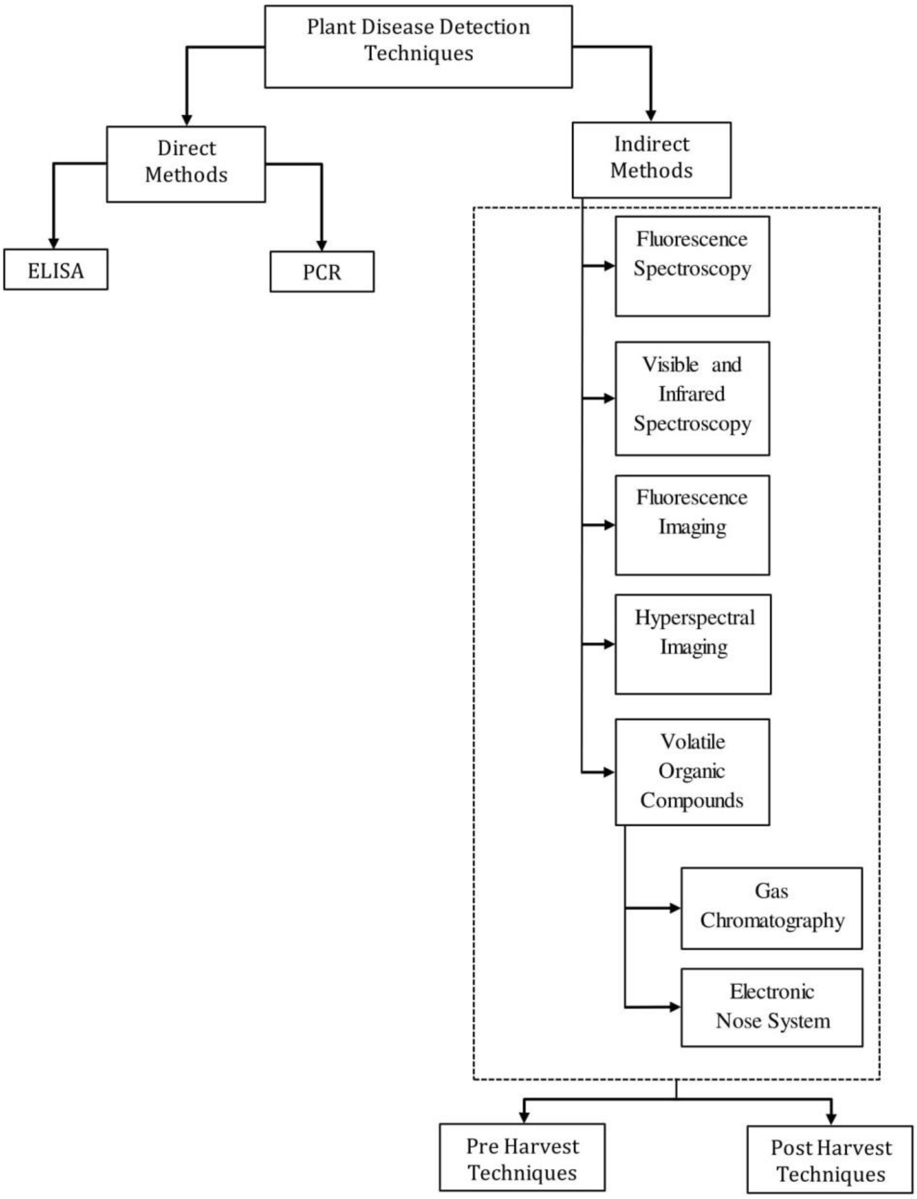


Figure 2-1: Different methods of plant disease detection

Table 2-1: ELISA vs. PCR

	ELISA	PCR
Diagnostic Kit	Protein-Based	DNA-Based
Cost	Simple laboratory equipment No training required	Expensive Costly Equipment
Disease Detection	Root crops Fruits Grains	Bananas Potatoes Cotton

Although these techniques may be robust and very accurate in detecting plant diseases, the drawbacks of these methods are significantly vast. These techniques rely heavily on the use of expensive laboratory equipment and extensive experiments, which can be time-consuming and labor-intensive. Sample preparation consumes a considerable amount of time and effort to ensure reliable and accurate results. These techniques are also very expensive because of the use of consumable reagents that are specifically designed for each pathogen [4]. Therefore, new and more rapid disease detection systems are needed as a preliminary screening tool for processing large numbers of plant samples.

2.2. Indirect Methods

New automated non-destructive methods have been studied to detect plant disease symptoms early and with high sensitivity to specific diseases. These methods should have the ability to detect diseases and stresses in real-time under field conditions. The most common techniques used are spectroscopic and imaging techniques for the detection of symptomatic and asymptomatic plant diseases [4]. The methods that will be studied are fluorescence spectroscopy, visible and infrared spectroscopy, fluorescence imaging, hyperspectral imaging, and volatile organic compounds (VOC) profiling. Not only have these methods been shown to provide successful detection of plant stress and nutrient deficiencies, but also have been useful in monitoring the quality of postharvest fruits and vegetables.

2.2.1. Fluorescence Spectroscopy

Fluorescence spectroscopy works by exciting an object with a beam of light and measuring the fluorescence released. This method has been successfully used to measure stress levels and physiological states of vegetative plants. This information can be utilized to detect different diseases that may be affecting the plant. Fluorescence spectroscopy systems can be used in field-based settings where leaves are still attached, and in laboratory settings where sample leaves are detached. Studies have shown that this method is accurate in discriminating defective plants from non-defective plants [9]. However, studies also showed that this method is inefficient in providing enough information, such as water stress levels [10]. Statistical methods such as principal component analysis (PCA), artificial neural networks (ANNs), cluster analysis, and partial lest square (PLS) regression can be integrated with fluorescence spectroscopy to classify data into multiple classes [4].

2.2.2. Visible and Infrared Spectroscopy

Visible and infrared spectroscopy is a rapid, cost-effective, and non-destructive technique that can be used as a plant disease detection system. Studies have proven that this method can be used to detect stress levels and detect plant diseases accurately, even before symptoms are visible [11]. Using near infrared (NIR) technology alone, on the other hand, has displayed poor ability in identifying defective plants from healthy ones. Therefore, it is proposed to use both visible and infrared spectroscopy for disease detection in plants [12]. A study investigated infections found in grapevines by monitoring each leaf using a portable spectrometer under field conditions. Including vegetative indices and individual indices in the study increased the precision of disease detection, with a maximum accuracy of 75% [13]. Various vegetative indices can be found in Table 2-2. Another study was performed to detect Verticillium wilt in cotton canopy using a portable spectraradiometer. It was concluded that infrared spectra was most effective in predicting the specific disease [14].

Table 2-2: Vegetative indices [4]

Vegetative Index	Estimation	Reference
Disease index (f_D) (specific for individual study)	$f_D = \frac{I_{550nm}}{I_{550nm} + I_{690nm}}$	Moshou et al. (2005)
Normalized difference vegetation index (NDVI)	$NDVI = \frac{R_{NIR} - R_{RED}}{R_{NIR} + R_{RED}}$	Yang and Cheng (2001), Bravo et al. (2004), Yang et al. (2007), Naidu et al. (2009)
Green normalized difference vegetation index (Green NDVI)	$Green\ NDVI = \frac{R_{GREEN} - R_{RED}}{R_{GREEN} + R_{RED}}$	Yang et al.(2007)
Water Band Index (I_{WB})	$I_{WB} = \frac{R_{950nm}}{R_{900nm} +}$	Xu et al. (2007)
Soil-adjusted vegetation index (SAVI)	$SAVI = \frac{(R_{NIR} - R_{RED})(1 + L)}{R_{NIR} + R_{RED} + L}, L = 0.5$	Yang et al.(2007)
Other indices	$(R_{NIR} - R_{RED}), \frac{R_{RED}}{R_{NIR}}, \frac{R_{GREEN}}{R_{RED}}, \frac{R_{NIR}}{R_{RED}}$	Yang et al.(2007)
Photochemical reflectance index (PRI)	$PRI = \frac{R_{531nm} - R_{570nm}}{R_{531nm} + R_{570nm}}$	Huang et al.(2007), Naidu et al.(2009)
Red-edge vegetation stress index (RVSI)	$RVSI = \frac{R_{714nm} + R_{752nm}}{2 - R_{733nm}}$	Naidu et al.(2009)
Modified chlorophyll (a and b) absorption in reflectance index (MCARI)	$MCARI = [(R_{700nm} - R_{670nm}) - 0.2(R_{700nm} - R_{550nm})] \times \frac{R_{700nm}}{R_{670nm}}$	Naidu et al.(2009)
Visible atmospherically resistance index (VARI)	$VARI = \frac{R_{GREEN} - R_{RED}}{R_{GREEN} + R_{RED} - R_{BLUE}}$	Naidu et al.(2009)
Water Index (WI)	$WI = \frac{R_{900nm}}{R_{970nm}}$	Naidu et al.(2009)

I: Fluorescence intensity; R: Reflectance

2.2.3. Fluorescence Imaging

Unlike fluorescence spectroscopy, where only a single spectra is used, fluorescence imaging works by using images obtained from a camera. Wavelengths from an object are recorded on a camera after fluorescence excitation from a UV light. A study confirmed that this method could be used to detect tobacco mosaic virus (TMV) in tobacco plants. Non-infected tobacco plants were successfully differentiated from infected ones in a relatively short period of time

[15]. Another study used fluorescence imaging alongside with quadratic discriminant analysis (QDA) to detect yellow rust in winter wheat. Results showed that QDA was successful in classifying healthy from non-healthy plants with relatively high accuracies. However, QDA was inefficient in distinguishing healthy from mildly infected plants [16]. To improve accuracies, a new study combined multispectral fluorescence imaging with hyperspectral reflectance imaging through sensor fusion, and QDA was used as the classification method. The combination of these methods showed an increase in classification accuracies of healthy plants from 71% to 97%. Using self-organizing map (SQM)-based neural networks further increased classifications accuracies of diseased plants and healthy plants to 98.7% and 99.4% [17]. Because imaging techniques provide three-dimensional spectral information, they give higher accuracies in detecting plant diseases when compared to spectroscopic techniques [4].

2.2.4. Hyperspectral Imaging

Hyperspectral imaging is similar to multispectral imaging but uses a wider range of wavelengths for each pixel. An image is produced consisting of a set of pixel values at each wavelength of the spectra. This method is common in monitoring the quality of food products, such as fruits. The selection of the statistical classification method and disease-specific spectra bands for a particular application is the major challenge faced when using hyperspectral or multispectral imaging. This can be seen in a study where apple bruises were detected using hyperspectral imaging; results were conflicted on which bands were most suitable for identification of bruises in apples [4]. A study developed a system for early detection of yellow rust disease in winter wheat using visble-NIR hyperspectral imaging. Using quadratic discriminant analysis as the classification method yielded an accuracy of 92-98% when classifying diseased plants [18]. However, the use of QDA and multilayer perceptron (MLP) artificial neural networks yielded a classification accuracy of 97.8% and 99.4% for classifying the same disease [17].

In an attempt to detect plant diseases on a large scale, an air-borne hyperspectral imaging detection system was devised to detect ganoderma basal stem rot disease in oil palm plantations. Red edge techniques and multiple vegetative indices were used to classify diseased from healthy plantations. Results indicated accuracies ranging from 73 to 84% depending on the classification method used [19]. Conversely, a study investigating greening of citrus plantation using aerial hyperspectral imaging was not able to identify diseased canopies. Spectral angle mapping (SAM) and spectral feature fitting (SFF) classifications methods were used to classify diseased canopies from healthy ones. This undesirable result can be due to the large variability within the data [20]. Another study used hyperspectral images to detect defects found in Ruby red grapefruit using a spectral information divergence (SID) based classification method. 96% classification accuracy was achieved when using the SID based classification method [21].

2.2.5. Volatile Organic Compounds Profiling

VOCs are released by plants due to factors such as humidity, temperature, light, soil condition, and fertilization [22]. Studies have shown that certain compounds are released when a plant is under stress or is infected by a particular disease [23]. Therefore, VOC profiling can be used under field conditions as a non-destructive disease detection system for plants and postharvest fruits/vegetables. The two commonly used methods for VOC profiling are electronic nose techniques and gas chromatography (GC) [4].

An electronic nose system functions by using multiple gas sensors that can detect a range of organic compounds. The integration of these sensors can be beneficial in the discrimination of different compounds present in the atmosphere around the plant. A study used an array of 32 gas sensors to detect fungal disease in postharvest blueberries in a laboratory setting. Three different fungal diseases were to be classified from one another, Botrytis cinerea, Colletotrichum gloeosporioides, and Alternaria spp. Although berries with C. gloeosporioides could be easily differentiated from other infections, there were overlaps in VOC profiles of berries with B. cinerea and Alternaria spp. Similar VOC profiles can be due to the fact that B. cinerea and Alternaria spp both infect the same region in blueberries [4]. Another study used 10 gas sensors to measure VOCs released from infected wheat using PCA and LDA classification techniques. Results indicated that certain compounds could be identified by the electronic nose system to successfully classify the disease only 6 days after infection [24]. These studies indicate that the profiling of VOCs using an electronic nose system is feasible tool for monitoring plant diseases [4].

Gas Chromatography is a common technique used to analyze volatile metabolites released by plants in different environmental and physiological conditions. Many studies have been performed using GC to classify bacterial and fungal infections in food products by evaluating changes in volatiles. A study used a GC instrument to analyze the change in volatiles released from onions infected with bacterial and fungal species. Although no statistical analysis was used, the study indicated that from the 59 compounds released 25 volatile compounds could be used to identify the disease accurately [25]. In another study, VOC profiling using GC was used to identify infections present in potato tubers. Results indicated that as the disease severity increases, the VOC emissions also increases. An accuracy of 67-75% was achieved when a BPNN model was used to classify the VOCs to a specific disease [26]. In a similar study, rather than use BPNN models a discriminant analysis method was performed to classify the VOCs released from potato tubers. Results indicated an accuracy of 13-100%, which implies a BPNN analysis method yields higher classification accuracies than discriminant analysis models [27]. The low and fluctuating accuracies achieved when using GC for VOC profiling show that further development is needed to make it a more practical method for an early disease detection system [4].

2.1. Direct and Indirect Methods Evaluation

It is clear that direct methods such as ELISA and PCR are outdated solutions for a disease detection system when compared to indirect methods. Indirect methods may be less accurate but do provide a more rapid and field-based detection of plant diseases. Also, indirect methods have the potential in processing a large number of plants in real time, whereas this is not possible in experimental direct methods. However, indirect method solutions such as imaging and spectroscopic techniques provide contradicting results in classification accuracies of different diseases. Similarly, VOC profiling also provides diverse results in its capability of detecting diseases found in plants and postharvest fruits/vegetables. These undesirable results found in VOC profiling can be due to natural variation. These natural variations in VOCs can be due to changes in plant metabolism, environmental changes, and plant age. Because of this natural variation in VOCs it may obscure the changes due to stress and the presence of diseases. Moreover, variations in environmental conditions, such as light greatly affect the accuracy of imaging techniques. The selection of the statistical tool is also very important in accurately detecting diseases, and depends heavily on the plant and disease type. An overview comparison of molecular, imaging, and VOC profiling techniques are shown in Table 2-3.

Table 2-3: Comparison of direct and indirect plant disease detection systems [4]

	Direct Methods	**Indirect Methods**	
Characteristics	Molecular techniques	Imaging and spectroscopic techniques	VOCs profiling-based techniques
Accuracy of method	- Molecular techniques are presently the most accurate method for plant disease detection. - Efforts are ongoing to make molecular methods more reliable and simpler as well as develop molecular detection kits for field applications. However, it is difficult to develop kits for all diseases. They are usually focused towards commonly found and harmful diseases.	- The accuracy of imaging and spectroscopic techniques is plant and disease specific. - Higher the visible symptoms, better is the accuracy of the technique. Nevertheless, the non-visible regions of spectra can be utilized for improving the accuracy of the method.	-Accuracy is currently unknown, as this method is in the developmental stages and has been utilized in recent years. -Identification of disease-specific biomarker volatiles (most challenging step) can improve the accuracy significantly.
Cost	- Moderately expensive. Quite often: is labor intensive and requires specific instrumentation. - Trained personnel are also required for careful handling of samples and results.	- Expensive, especially if techniques as hyperspectral imaging are used. Fewer the wave bands used, cheaper will be the instrument. - Require computers/laptops for data analysis.	- The cost of technique depends on the desired accuracy for VOC profiling. -The cost can range depending on the detector required for biomarker identification.
Applicability for rapid detection	- Speed depends on the samples required to be analyzed, number of personnel, and equipment and materials. - The technique is not fast for a huge amount of samples.	- The focus on the development of this technique is due to its ability for rapid detection.	- The technique shows the potential for rapid plant disease detection.
Applicability for field work/Ruggedness	- Field kits are being developed. However, it is difficult to develop kits for all diseases. - The field kits are rugged, but require good accuracy for reliable results.	- Moderately ruggedness. The ruggedness of the spectrometer or the scanner depends on the base on which the sensor is mounted.	-Moderately rugged, depending on the detector used for sensing VOCs.
Speed of detection	- May require 24 – 48 h for reliable results. - Molecular kits are faster.	- Once this technique is established, may require minutes for disease detection. - The speed depends on the computational speed of the computer as well as speed of the scanner.	-This method may require significantly less time, if proven as an effective method for a particular disease. -Speed would depend on the detector speed and computational speed.

Indirect methods can be used to detect diseases in both preharvest and postharvest fruits/vegetables. To have a better understanding of different disease detection techniques, indirect techniques can be further categorized into two main groups; preharvest and postharvest detection techniques. It is clear that the most suitable method for disease detection in both preharvest and postharvest fruits/vegetables is the use of imaging and spectroscopic techniques. The following section will include more in-depth research on the use of imaging techniques for disease detection in preharvest and postharvest fruits/vegetables. Sufficient research has been done in both categories, each with its own benefits and drawbacks.

2.2. Preharvest Disease Detection Techniques

[28] proposed a method in which leaf disease severity of preharvest sugarcane leaves can be measured using imaging techniques. Fungi diseases are very common in sugarcane leaves, and inhibit their growth immensely. These diseases leave visual spots on leaves, which in turn prevent the vital process of photosynthesis. Photosynthesis is a fundamental process essential for growth and prosperity. Rather than using pesticides, which is not only costly, but also increases toxic residue levels, an early disease detection system can be implemented. Because fungi-caused diseases in sugarcane leaves appear as spots it is applicable to use imaging techniques to detect the severity of the disease [28]. Disease severity is expressed the ratio between the affected area and the leaf area. If the lesion area ratio to leaf area ratio is high, then the leaf is said to have a high disease severity according to Table 2-4.

Table 2-4: Disease Severity Scale Developed by Horsfall and Heuberger [28]

Category	Severity
0	Apparently infected
1	0 – 25% leaf Area infected
2	26 – 50% leaf Area infected
3	51 – 75% leaf Area infected
4	> 75% leaf Area infected

For this study 90 infected sugarcane leaves were used as samples taken from a 12 Mega pixel digital camera. The images were taken in a controlled environment with a white background and light sources to eliminate any reflection for enhanced view and brightness [28]. For improved results, the input leaf image in Figure 2-2 is first transformed from RGB color space to HIS color space. The image is then converted to gray scale as shown in Figure 2-3 and then segmented into two regions; the diseased region and the healthy region shown in Figure 2-4. In order to segment the image a triangle thresholding method was used. After the image has been segmented the leaf area and infected area ratio are calculated by measuring the number of pixels in the white region and in the black region. This experimented showed to be very successful with accuracies of 98.60%.

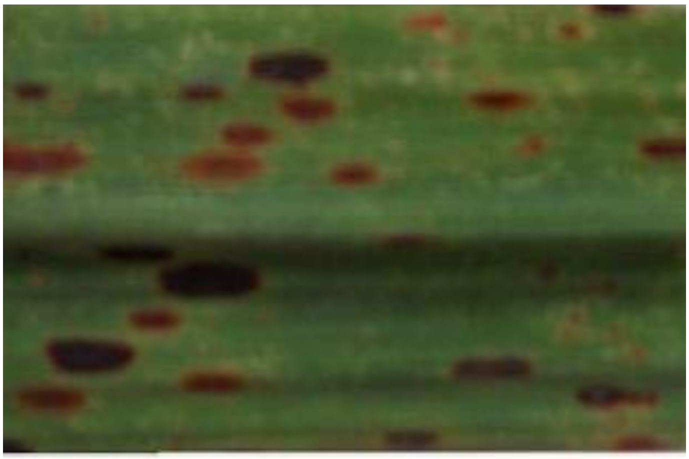


Figure 2-2: Brown spotted diseased sugarcane leaf [28]

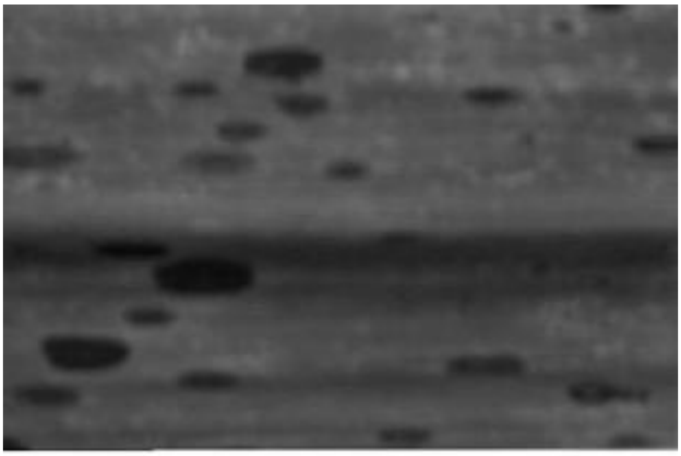


Figure 2-3: Gray scale image [28]

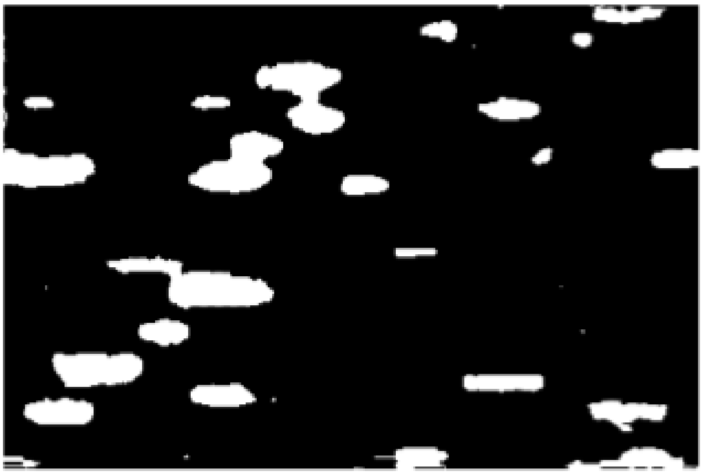


Figure 2-4: Infected Region detection after triangle thresholding [28]

To further improve the efficiency and accuracy of leaf disease grading systems, [7] proposed the use of machine vision and fuzzy logic for disease detection in pomegranate leaves. A table-like percent-infection calculation method is used to illustrate the severity of the disease. A 10-megapixel Nikon Coolpix L20 digital camera was used to take images of disease infected pomegranate leaves at an equal distance of 16 cm. Images were taken from several pomegranate farms with no specific test site indication. A few image restoration techniques are implemented before the image is segmented into two specific regions; disease region and healthy region. This test was conducted under controlled conditions, such as lighting settings and using a uniform background. In addition, no automation is considered in this technique.

Firstly, to reduce the computational complexity of the system, images are first resized to a fixed resolution. Then by the use of a Guassian filter any noise or outer interference in the image is removed or diluted. K-means clustering (K = 10) technique is used here as the segmentation method, and the cluster that contains the diseased spots are saved to calculate the area of the diseased region. After calculating the disease area (A_D) and total leaf area (A_T) it is now possible to calculate the percent of infection regards to the total area. A fuzzy logic system is implemented in order to characterize which disease grade the disease belongs to depending on the percent infection. It is difficult to assess which disease grade the disease belongs to because of the ambiguity and uncertainty of the table a fuzzy logic system can be effective in this case. The segmented regions that are used to calculate the percent-infection index are shown in Figure 2-5 and Figure 2-6. The fuzzy logic system proved to be very effective in accurately grading diseases into their appropriate categories.

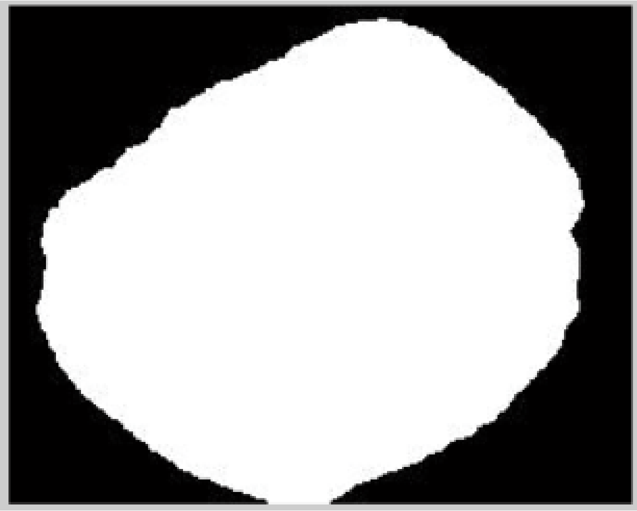


Figure 2-5: Total leaf area (A_T) [7]

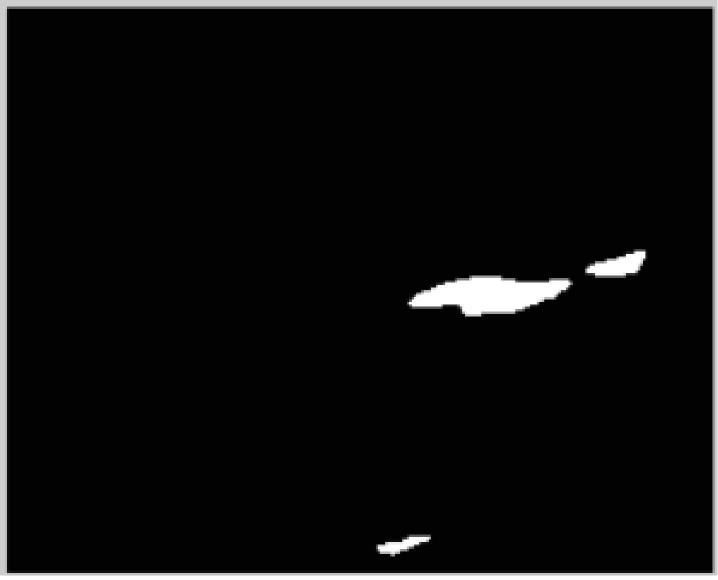


Figure 2-6: Disease area (A_D) [7]

Because it can take up to one week to diagnose plant diseases using traditional chemical analysis, [29] proposed an early plant disease detection technique. During the period of anthesis plants often appear to be nutrient deficient and it is vital to detect these deficiencies early to ensure quality and quantity of plants. Nutrient deficient plants usually leave quite visible symptoms on their leaves, which can be used to diagnose the disease. By extracting features from leaves such as color and texture, plant nutrient deficiency can be diagnosed at an early stage [29].

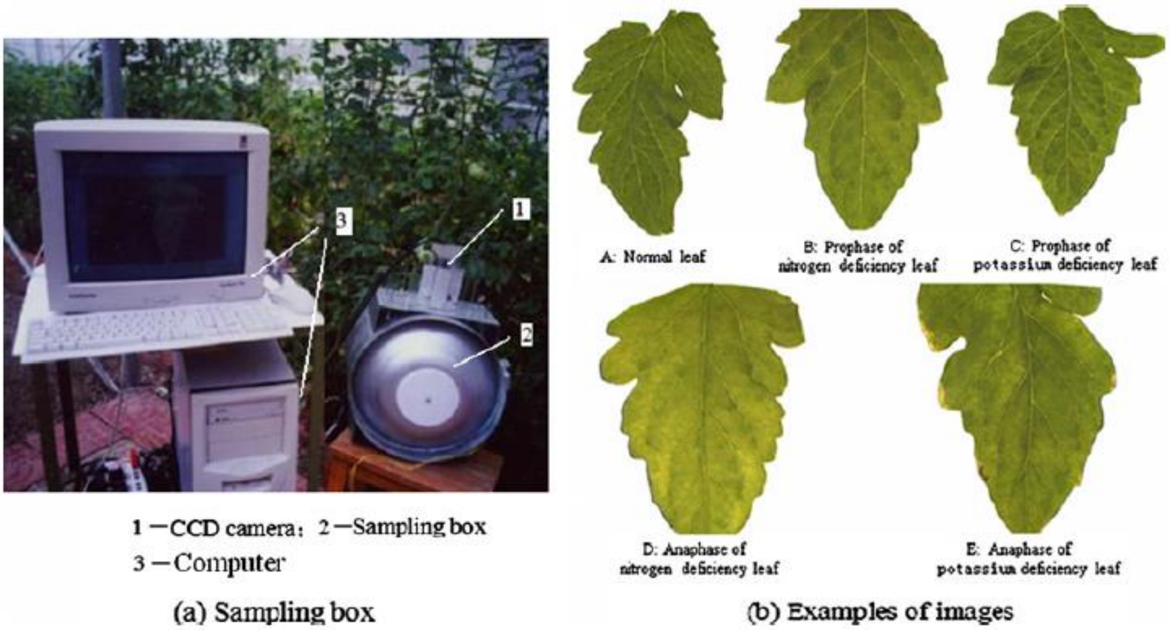

3
1
2
1 —CCD camera; 2—Sampling box
3 —Computer
(a) Sampling box
A: Normal leaf
B: Prophase of nitrogen deficiency leaf
C: Prophase of potassium deficiency leaf
D: Anaphase of nitrogen deficiency leaf
E: Anaphase of potassium deficiency leaf
(b) Examples of images

Figure 2-7: Sampling box and examples of images used in the diagnostics system [29]

Nitrogen and potassium deficient tomato plants were used under a controlled chamber environment to extract color and texture features of leaves. A sampling box shown in Figure 2-7 was used to take images of tomato leaves with a color digital camera. Normal, prophase and anaphase of nitrogen and potassium deficient tomato leaves were used in the proposed system as shown in Figure 2-7.

After images are taken in the chamber they are processed for color and texture feature extraction. Because nitrogen deficient leaves turn yellow with the development of disease, the amount of yellow pixels in the leaf image can reflect the feature of nitrogen deficiency. Therefore, [29] proposed the use of percent histogram to extract feature of color.

[29] tested multiple texture feature extraction techniques such as percent histogram, Fourier transform, and wavelet packet decomposition to ensure the most appropriate method for feature extraction. Experiments showed that the wavelet packet decomposition method proved to result in a high accuracy of distinguishing normal from nitrogen or potassium deficient leaves. To optimize the combination of features, [29] used a Genetic Algorithm to select the most important feature items. The optimized feature set was classified using a fuzzy K nearest neighbor classifier, and proved to be quite accurate in identifying normal, nitrogen deficient, and potassium deficient leaves.

The proposed system was tested using sample collection of tomato leaves, and showed to be effective with an accuracy of identifying normal leaves, nitrogen-deficient leaves and potassium deficient leaves to be 92.5%, 85%, and 82.5% respectively. The diagnostic system can identify potassium deficiency in leaves 10 days earlier than by experts' observation. This gives a significant amount of time for measures to be taken to ensure production quality.

To help aid crop producers and farmers in remote areas for identifying early symptoms of plant disease, [30] used image processing and pattern classification to implement a machine vision system. The system would detect cotton crop damage caused by 3 specific disease; green stink bug, Bacteria angular, and the Ascochyta blight virus as shown in Figure 2-8.

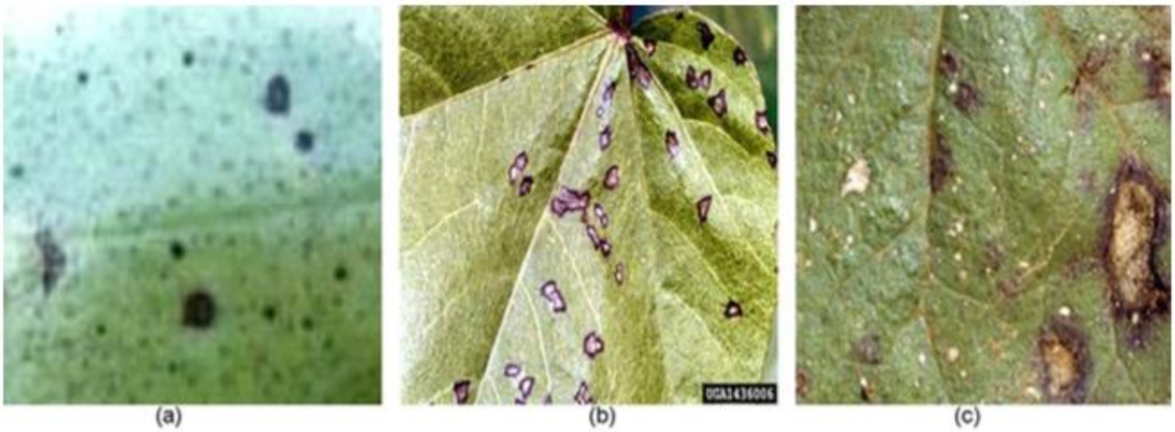


Figure 2-8: Image of cotton crops showing the visual symptopms of damages casued by: (a) Southern green stink but; (b) Cacterial andular; (c) Ascochyta blight. [30]

A set of 117 images were used in the study from different sections of the plant such as the leaf, fruit and stem [30]. Multiple features that characterize shape and appearance of the image were extracted using both image's RGB and HSV color space. For texture extraction, a co-occurrence matrix was used to identity grey levels between a specific position in the image and its neighboring pixel. A box-counting algorithm was used to estimate the dimensions of the image for fractal dimension feature extraction. Gliding Box Algorithm was used to calculate lacunarity feature to measure texture associated with patterns. A Support Vector Machine Classifier was used to identify the best classification model for the different feature sets. Each feature was used individually and grouped to identify the difference in classification accuracy with respect to feature type [30].

Results showed that grouping features of similar type resulted in higher classification accuracies when compared to using individual features. Also, results showed that grouped texture features achieved higher classification rates (83%) when compared to grouped shape features which achieved a classification rate of 55%. When using the total feature set the highest classification rate was achieved of 90%. To further enhance the classification accuracy, one feature was withdrawn at a time until the best feature set was discovered. Using this method, a deducted feature set resulted in a classification accuracy rate of 93.1%.

In order to decrease the cost of uniform and periodical pesticide spraying [8] suggested a selective spraying solution using an agriculture robot. Because primary infections start from a localized discrete foci, a system can be implemented to detect these infected foci points and a target treatment can be established. Eliminating the initial infection point does not only inhibit the spread rate of the infection to other patches, but also significantly decreases the use of pesticides. In this case study, a multispectral camera is used to inspect an entire grapevine canopy for automatic detection of powdery mildew and selective spraying is achieved using a six-degree robotic arm illustrated in Figure 2-9.

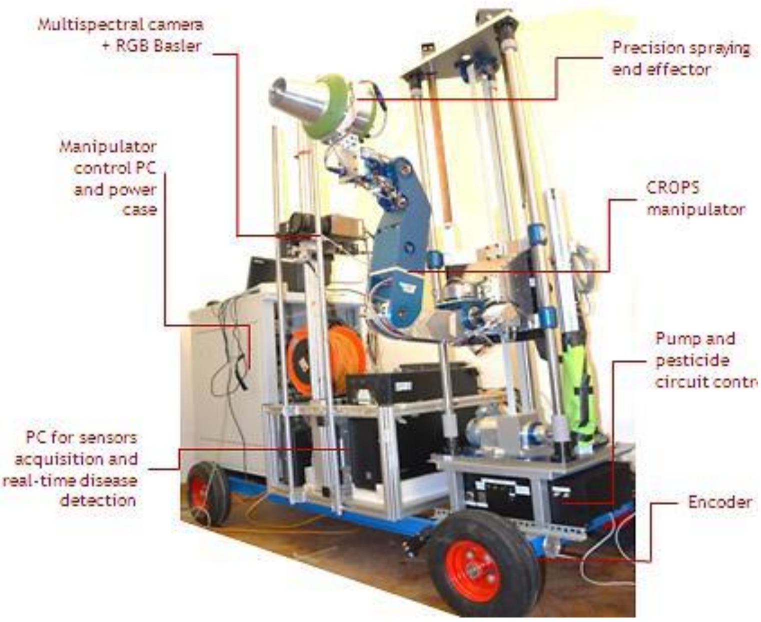

Multispectral camera + RGB Basler
Manipulator control PC and power case
PC for sensors acquisition and real-time disease detection
Precision spraying end effector
CROPS manipulator
Pump and pesticide circuit contr
Encoder

Figure 2-9: Robotic vehicle for disease detection and selective pesticide spraying [8]

The red, green, and NIR channels were the primary channels used for disease detection. An RGB camera was also added to the camera rig for visual documentation of the scene. The cameras were positioned at a constant height of 1.4m while maintaining a 1m distance from the canopy wall as shown in Figure 2-10. Halogen light panels were used to provide diffused illumination of the imaged area for enhanced and more accurate results for indoor conditions.

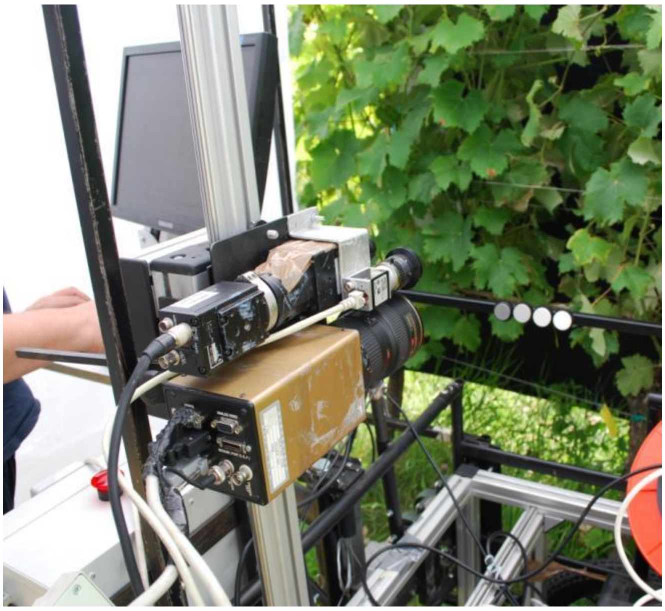


Figure 2-10: Camera rig setup [8]

The disease detection algorithm used in this system was to capture very sharp changes in the reflectance in green and red channels, as this will give a clear indication of the breakdown in chlorophyll content in infected leaves. Specifically, two indices used to measure chlorophyll absorption are calculated using Equation 2.1 and 2.2 respectively [8].

$$I1 = \frac{Red * Green}{NIR^2} \tag{2.1}$$

$$I2 = \frac{Red}{Red + Green + NIR} \tag{2.2}$$

Since healthy regions have high chlorophyll absorption, it is expected that $I1$ and $I2$ will yield higher values for diseased areas and return lower values for healthier regions. To test the proposed system grapevine plants were used to recreate a vineyard canopy wall in a greenhouse as shown in Figure 2-11.

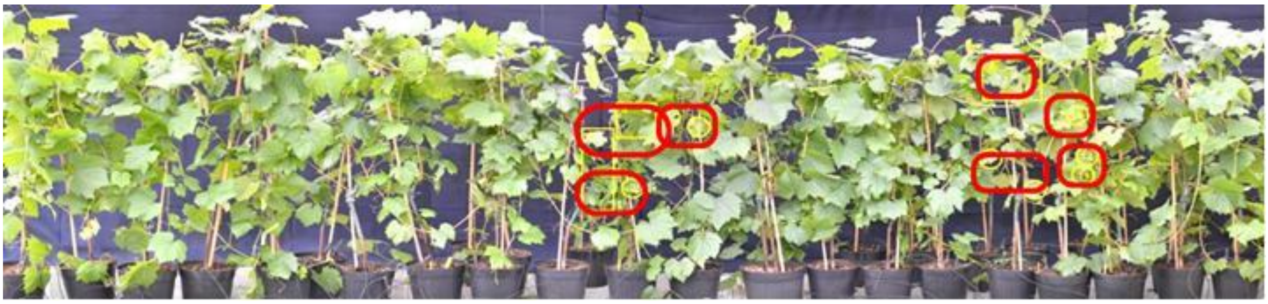


Figure 2-11: Healthy grapevine plants aligned together with infected plants positioned randomly circled in red [8]

Figure 2-12 illustrates the results gathered after the first experimental run. The blue graph shows the disease spots detected by the robot (blue dots) and the robot sprayings (blue circles) operated by the robot. To analyze the results, they are compared against the actual disease spots (red dots) specified by a plant pathologist and computed minimal sprayings (red circles). In this run the robot detected all the diseased areas and successfully covered all the disease foci with pesticide spraying. The selective pesticide spraying achieved here obtained a reduction in pesticide use up to 84% when compared to homogenous spraying of the canopy. However, at different instances the robot did detect disease spots in areas where plants were healthy, this is apparent around the 100 mark. This can be because of illumination changes and shadow effects, which distort the input image and hence the values of $I1$ and $I2$. Also apparent is the increased operated sprayings around detected diseased areas by the robot when compared to minimal sprayings. [8] claims that the surroundings of detected disease areas can be treated anyway by including a conservative safe-border area. This may help to raise the level of acceptance in real world cases, despite the reduction in potential pesticide savings. Overall, the proposed system proved to detect disease foci with an accuracy of 85%, while achieving a reduction of pesticide use close to 90%.

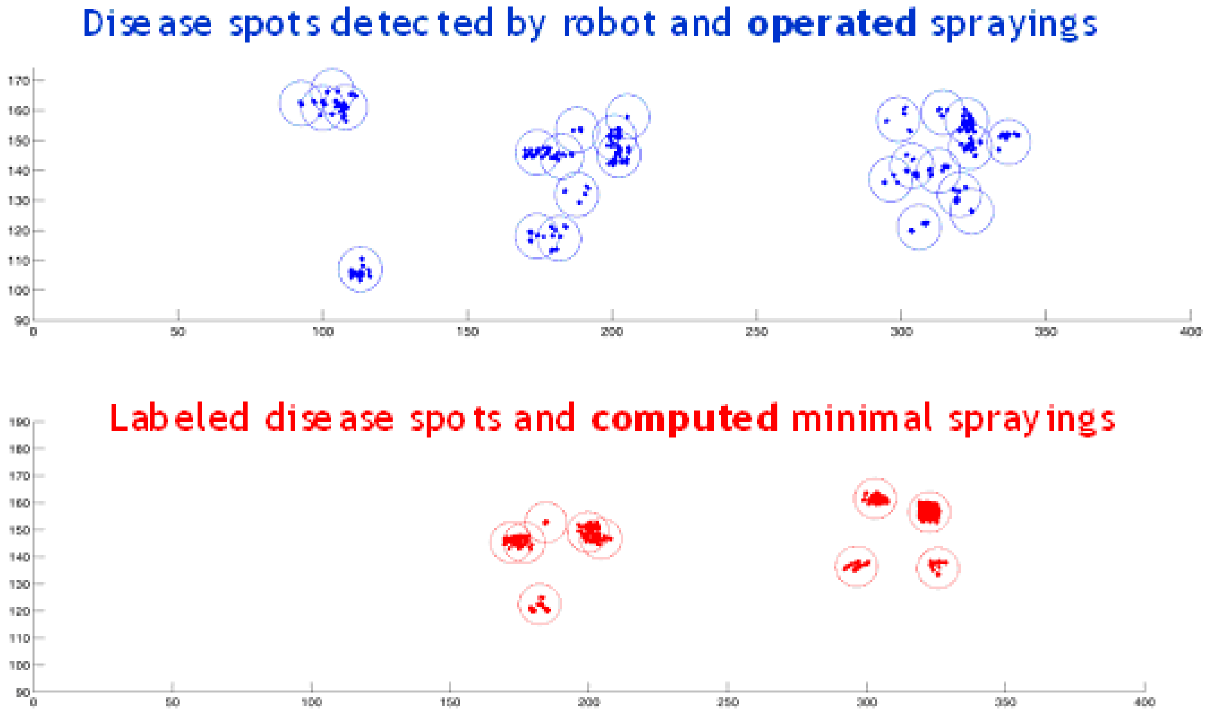


Figure 2-12: Disease spots detected by robot and operated sprayings (blue) against labeled disease spots and computed minimal sprayings (red) [8]

[31] used a semi-autonomous mobile e-nose robot to examine the fertility of soil by using metal oxide gas sensors to detect organic volatile compounds found in soil. A six-wheel robot was designed with an integrated array of gas sensors shown in Table 2-5. This e-nose system was used because of its low cost and high sensitivity to certain target gases [31].

Table 2-5: TGS and MQ gas sensors and their target gases [31]

Sensor	Target gases
TGS 825	Hydrogen sulfide
MQ2	Combustible gases
MQ5	LPG, natural gases
MQ135	NH_3, NO_x, Benzene, CO_2 gas
TGS 2600	Air contaminants (ethanol, iso-butane, hydrogen)
TGS 2602	VOCs and odorous gases

Multiple fans were installed on the robot to ensure air flow towards the gas sensor enclosed chamber as shown in Figure 2-13. The data collected from the gas sensors were recorded in real time and sent to a computer for analyzing and visualizing via Zigbee wireless network. Ultrasonic sensors and accelerometers were also implemented in the robot to ensure obstacle avoidance and smooth navigation. The robot was operated under real conditions in four different locations; floor room, lawn, dry ground, and vineyard row.

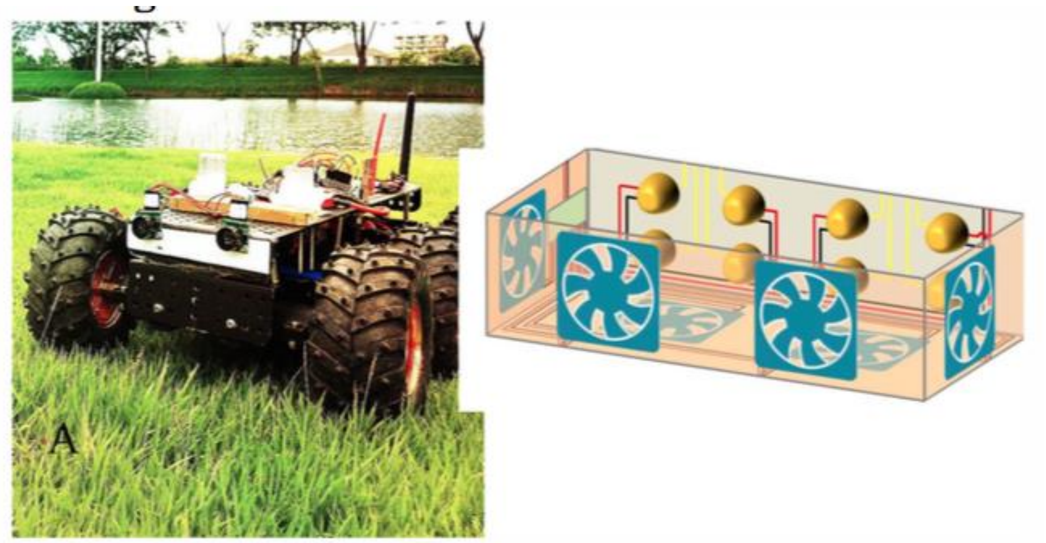


Figure 2-13: A six-wheel robot with e-nose and navigation system and an e-nose chamber [31]

From Figure 2-14 it can be observed that the results collected from the dry ground location yielded high percentage in sensor response to most of the gas sensors. This can be due to the fact that little to none of the volatile gases are absorbed from the surroundings because of the lack of weeds and grass [31]. Most gas sensors yielded quite low responses in the vineyard location except for TGS2602, because of its high sensitivity to odorous gases such as ammonia and hydrogen sulfide. TGS2602 also has a high sensitivity to VOCs such as toluene [31].

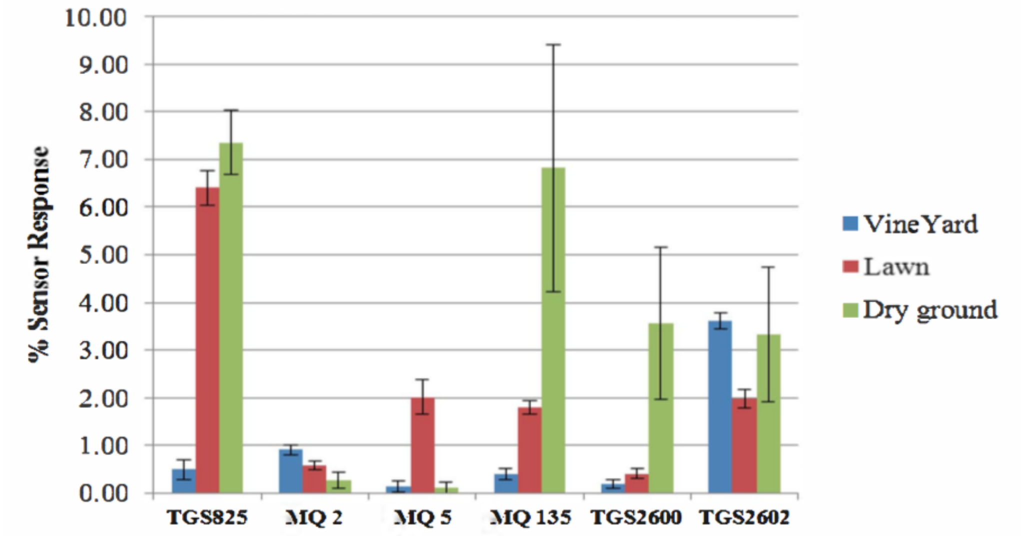


Figure 2-14: Percent sensor responses of six elements used in the e-nose robot system to soil volatiles at different places [31]

The six different sensors show promising results in indicating different volatile gases in dry ground. [31] argue that if enough common odor data from various places is collected and put into a database, this database can later be used to determine irregular events.

2.1. Postharvest Disease Detection Techniques

The method proposed by [1] uses image techniques and artificial neural networks (ANNs) to classify different diseases found in grapes and apples. Image processing was used to extract specific features such as fruit color, texture and morphology. An important factor that may aid or diminish the effectiveness of image processing is the selection of the color space. [1] proposed that the HIS color space is more suitable than RGB as it is less affected by changes in light. A neural network was used to characterize these features into a disease category such as apple scab or apple rot. The neural network was first trained with a data set of various apple diseases. Figure 2-15 illustrates the different steps taken during the training and testing of the neural network.

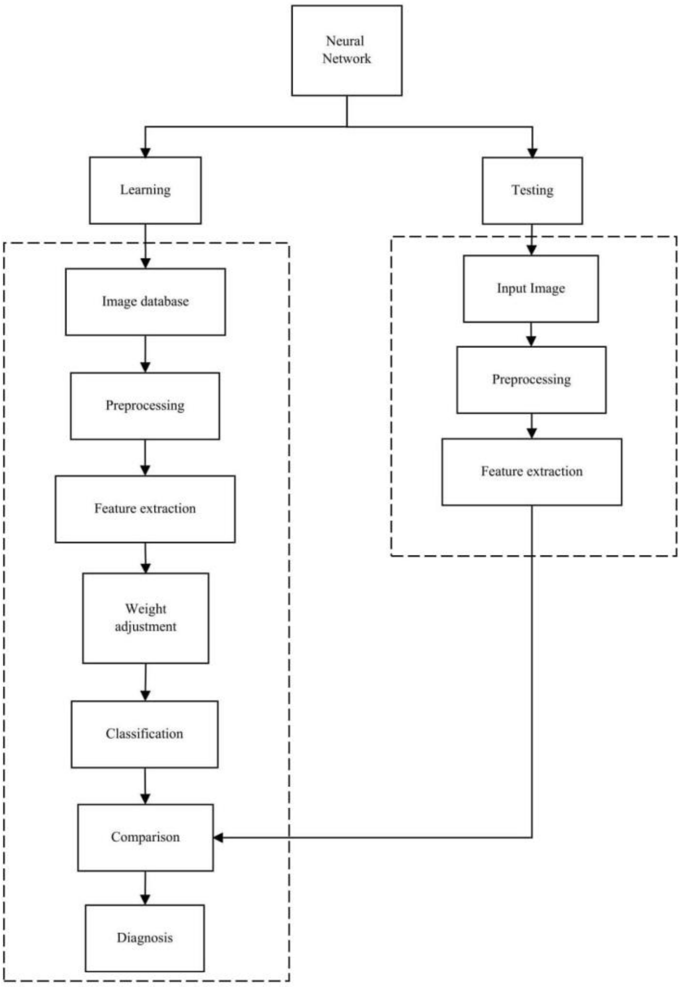


Figure 2-15: Flowchart of proposed system [1]

After being trained, the neural network was capable of characterizing an input apple image into its corresponding disease category. It was concluded that the selection of features plays a vital role in the effectiveness of the neural network. Because diseases are better defined by color and morphology, these features, unlike texture, proved to provide improved results [1].

[2] provided a similar technique with the use of a Multi-class SVM classifier and K-Means image segmentation to detect three diseases found in apples: apple blotch, apple rot and apple scab. Because of the wide variety of skin colors found in different apples, it was proposed that using size and color as features could be challenging in the detection of defects in apples. Therefore, a K-means based image segmentation approach was used in order to extract disease features. In order to ensure shorter processing times for image segmentation, L*a*b* color space was used [2]. K-means clustering was used to segment the apple image into 4 different clusters by categorizing similar pixel densities into their corresponding cluster. [2]

proposed that for improved segmentation results, using 3 to 4 clusters was sufficient. Figure 2-16 demonstrates the results of the K-means clustering for apples with different infections.

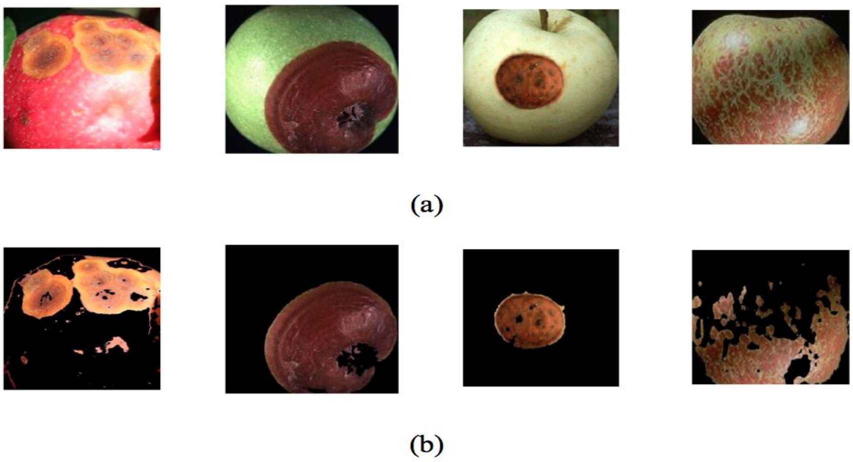


Figure 2-16: Image results before (a) and after (b) K-means clustering segmentation [2]

It is clear that the K-means clustering technique is an effective tool for image segmentation. 4 different techniques were used for feature extraction, Global Color Histogram (GCH), Color Coherence Vector (CCV), Local Binary Pattern (LBP), and Complete Local Binary Pattern (CLBP). These color and texture features were used to validate the accuracy and efficiency of the proposed system [2]. GCH represents the probability of a pixel being a certain distinct color in the form of a histogram, whereas the CCV distinguishes coherent and incoherent pixels into two separate histograms. Coherent pixels are defined as pixels that belong to a large region with homogenous color, and any other pixel is defined as an incoherent pixel. LBP considers the difference of each pixel with respect to its neighbors. CLBP on the other hand, considers signs, magnitude, and original center gray level value of local differences. After the extraction of features, a Multi-class Support Vector Machine (MSVM) was used for the training and classification as shown in Figure 2-17.

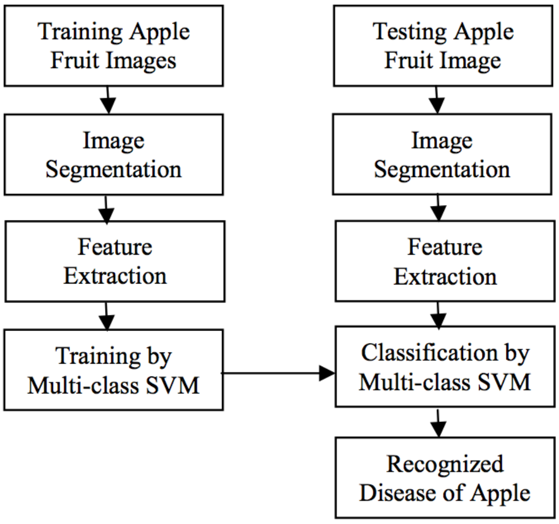
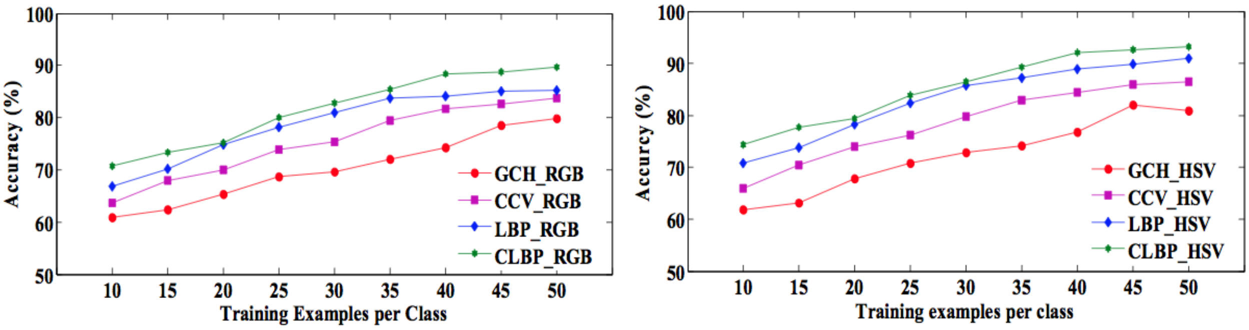

Training Apple Fruit Images
Image Segmentation
Feature Extraction
Training by Multi-class SVM
Testing Apple Fruit Image
Image Segmentation
Feature Extraction
Classification by Multi-class SVM
Recognized Disease of Apple

Figure 2-17: Flowchart of proposed system [2]

Support Vector Machines have significant advantages over ANNs as they are less prone to over fitting and require less computational power. However, since ANNs use a heuristic method, it is easier to develop than an SVM, which involves more theory. A data set of 431 apple images was created with wide variations in apple type and color to insure a more realistic test [2]. The data set is to be categorized into Apple Botch, Apple Rot, Apple Scab, or Normal Apple categories.

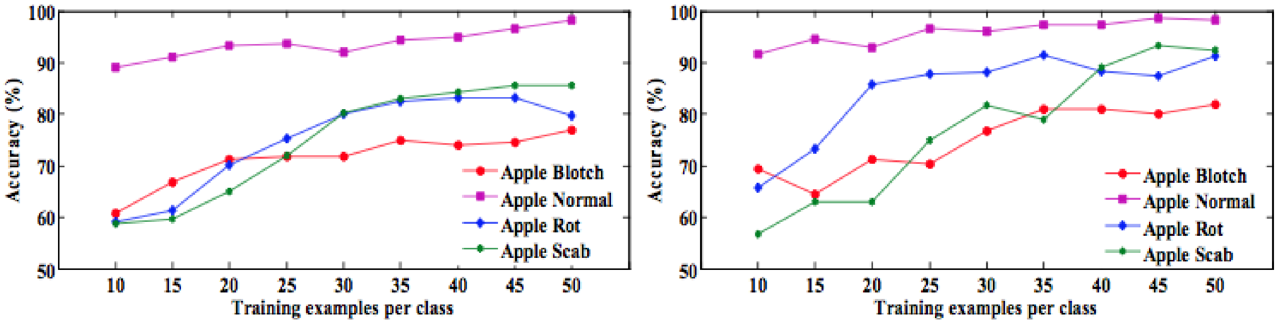
Results proved to coincide with [1] proposal that the use of HSV color space outperforms RGB color space [2]. It is clear from Figure 2-18 that using the HSV color space in every feature extraction technique yields more accurate results. Also apparent from Figure 2-18 is that the most accurate extraction techniques are the CLBP followed by the LBP. Unlike GCH and CCV, both CLBP and LBP use information from neighboring pixels. Because they use local differences, they are more efficient in pattern matching and are less computationally extensive. It can be concluded that it is more effective to use either LBP or CLBP as feature extraction techniques to yield more accurate results. Furthermore, results indicated that the MSVM classifier detection of normal apples and apple scab was significantly easier than the detection of apple blotch and apple rot. Figure 2-19 illustrates this observation, with very high accuracy results for the detection of normal apples and apple scab by using the LBP technique. However, the accuracy rates for the detection of apple blotch and apple rot are significantly lower.



(a) Using RGB colour image.

(b) Using HSV colour image.

Figure 2-18: Accuracy difference of using RGB color space compared to HSV [2]



(a) LBP in RGB color space.

(b) LBP in HSV color space.

Figure 2-19: Accuracy of detecting different apple disease categories in RGB and HSV color space [2]

[6] proposed a more automated technique using on-line experimental system that can simultaneously inspect all four sides of an apple, and sort them accordingly. Two main issues with previous studies were tackled; the first is to acquire the whole surface of an apple at on-line speeds and the second to quickly identify the apple stem and calyx. A description of the system schematic is represented in Figure 2-20.

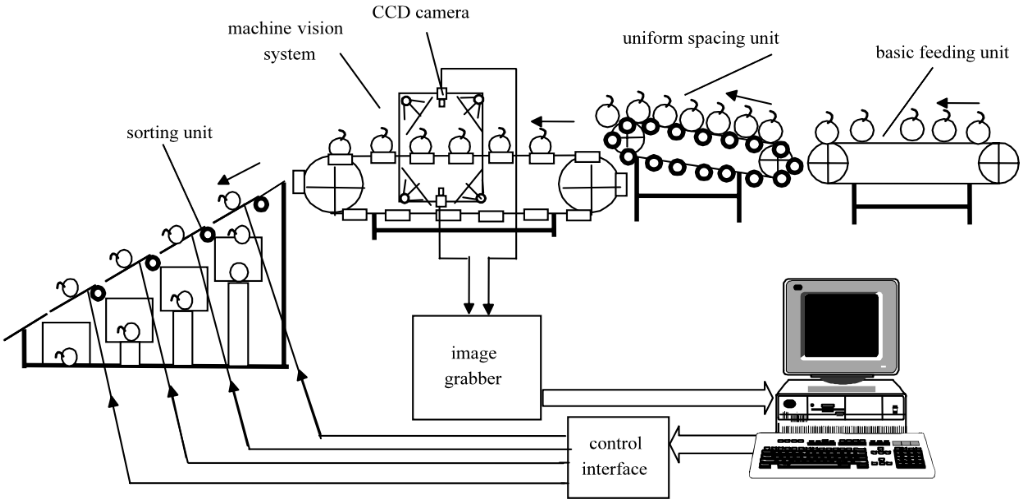


Figure 2-20: Schematic representation of apple defects sorting system [6]

The schematic displays how apples are fed into the machine vision system via conveyors and belts for image acquisition, and how they are sorted accordingly. The feeding conveyor is designed to insure that the stem of the apple is faced upwards for maximum performance. The machine vision system consists of two cameras to provide multiple images of the apple, and a lighting chamber to control the light distribution [6]. By use of mirrors the top camera will cover three side views of the apple; top and two sides. The camera below will take an image of the bottom view of the apple. This setup has the distinctive advantage of inspecting all sides of the apple in one cycle. The setup is illustrated further in Figure 2-21.

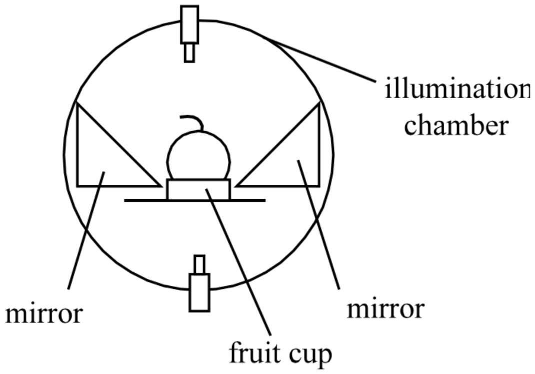


Figure 2-21: Setup of vision system [6]

After image acquisition, multiple methods were implemented for accurate defect detection. To not disrupt the segmentation process, image background removal algorithms were implemented to ensure that any backgrounds such as the mirror are removed. Segmentation is completed by using a reference fruit image and then subtracting it from the original fruit image. Then by the use of a simple thresholding method the defects could be easily extracted [6]. Because stem and calyx defects are very similar to each other the authors proposed the use of neural networks to distinguish the stem and calyx defects.

Forty samples of Fuji apples were used to test and validate the system. Figure 2-22 illustrates how the input apple image is segmented and the defected regions as well as the stem and calyx defects are detected. The neural network classifier proved to be very effective in detecting stem-calyx recognition with accuracies over 93% [6]. Overall this system proved to be successful in detecting defects on multiple sides of an apple simultaneously, while on a sorting line.

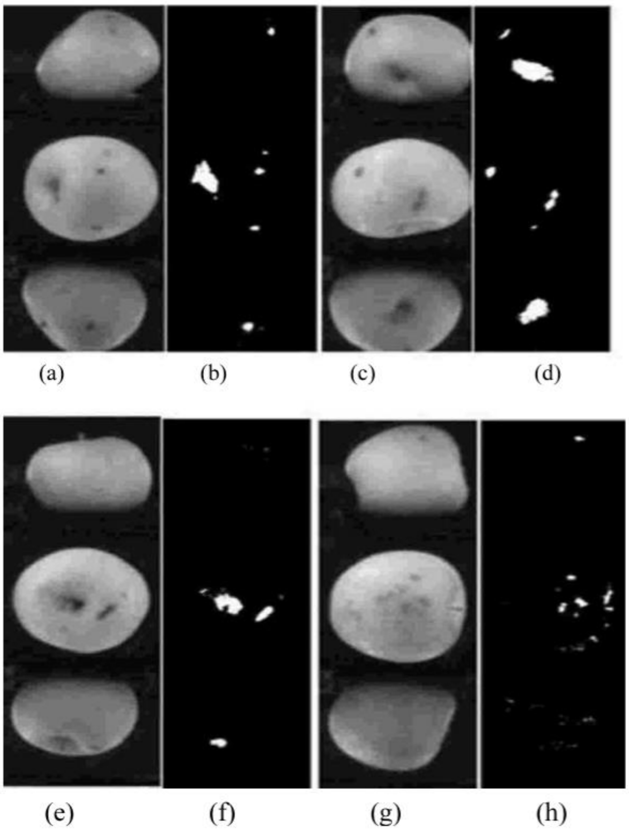


Figure 2-22: Defects segmentation results. (a), (c), (e), (g) original image; (b), (d), (f), (h) segmented defects [6]

[32] also implemented a non-destructive automated technique to detect bruises on apples using a multiple waveband technique. Hyperspectal imaging can provide enough information in several wavebands, but it is inappropriate in an on-line system due to its long acquisition and analysis time. Instead, [32] resided on using multi-spectral imaging where only a specific range of wavebands were used to detect apple bruises. A total of 128 ’Golden Delicious’ apples were purchased from two different locations and separated into non-bruised and bruised groups by visual inspection. Bruises are usually caused from human handling and vibration from transportation. Apples without visible bruises were impacted with a pendulum in the laboratory to achieve an average bruise size of 17mm in diameter [32]. The hyperspectral imaging system consisted of a conveyor belt, light source and camera for on-line bruise detection of apples as shown in Figure 2-23. The camera has high sensitivity from 400 to 1000 nm and used alongside a spectrograph to detect the separate wavebands of light. The system also consisted of light sources and operated under a controlled environment.

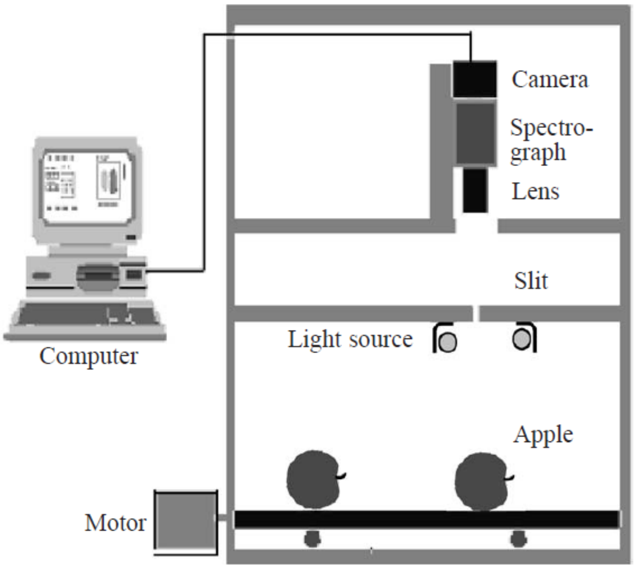


Figure 2-23: Schematic of hyperspectral imaging system [32]

A simple thresholding method was used to segment the image background and low intensity regions and further analyzed by principal component analysis (PCA). PCA is an effective tool in reducing data dimensionality and to enhance bruise features. Results showed that the wavebands centered at 558, 678, 728, and 892 nm were optimal in detecting bruises on ’Golden Delecious’ apples. A simple classification technique was introduced to determine whether apples are bruised or intact. This classification technique resulted in an accuracy of 93.5% for detecting intact apples and about 86% for detecting bruised apples.

To increase the speed of the sorting process, [33] suggested a computer vision based system to automatically grade apples. A monochrome digital camera with multiple band-pass filters was used to capture one-view images of ’Jonagold’ apples taken in a controlled illuminated environment. The data set consisted of 280 healthy apples and 246 apples included several skin defects such as bruises and rot. The 4 band pass filters used for image acquisition are centered at 450nm (Blue), 500nm (Green), 750nm (Red), 800nm (Infrared) as shown in Figure 2-24.

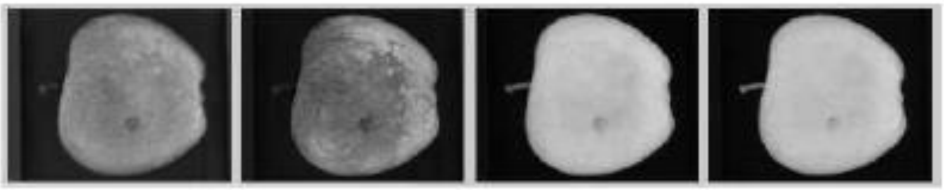


Figure 2-24: Filter images of apple. Left to right: Blue, Green, Red, Infrared filters [33]

Images of apples were taken in a uniform and low intensity background to ensure a controlled environment. Therefore, background segmentation can be easily achieved using a thresholding technique. Multiple global thresholding techniques such as Otsu, Entropy, and Iosdata were tested for defect segmentation. However, because of the similar appearance of the stem-end/calyx area and the apple defect a segmentation technique is required to distinguish them from one another. Stem-end and calyx are natural parts of the apple and usually appear as dark blobs which can be often mistaken as defects. Statistical, textural, and shape features are extracted and introduced to a support vector machine to distinguish the calyx from the defect. The result of this segmentation can be seen in Figure 2-25.

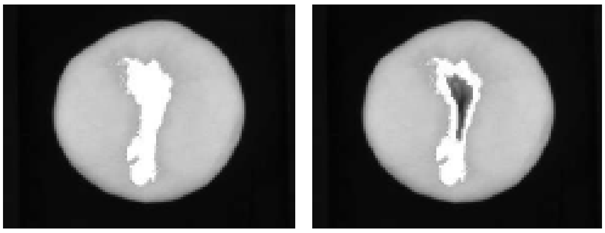


Figure 2-25: Example of stem-end/calyx removal. Before the removal on the left, and stem-end/calyx removal on the right. Defected area displayed in white in both images [33]

After calyx removal, a total of 13 features were extracted and introduced to multiple fruit classification techniques to test the accuracy of different classifiers. The 5 classifiers used were Linear Discriminant Classifier (LDC), Nearest Neighbor (k-NN), Fuzzy Nearest Neighbor (fuzzy k-NN), Adaptive Boosting (AdaBoost), and Support Vector Machine (SVM).

Figure 2-26 illustrates the different thresholding techniques with all 4 filters. It is quite apparent that unlike the blue and green filters, the red and infrared filter images provide a more accurate representation of defect segmentation. Blue and green filter images result in false segmentation because of the low contrast between healthy and defect skin in the wavelength range of 410 - 510nm. Figure 2-26 also shows that isodata thresholding accurately segments the defective area when compared to otsu and entropy thresholding.

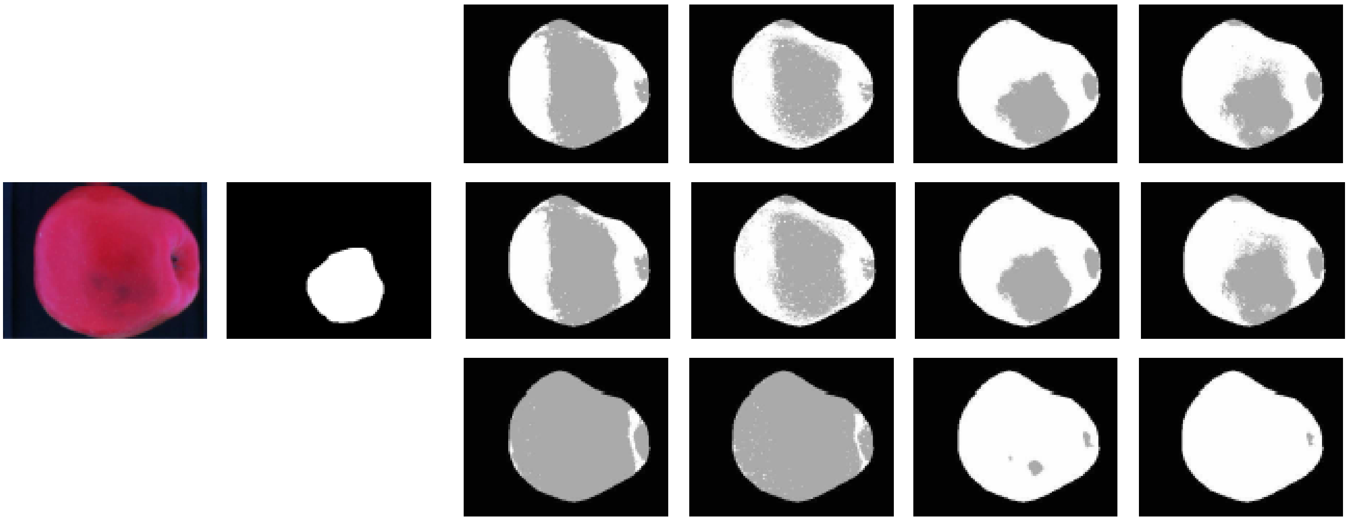


Figure 2-26: Band pass filters segmentation results [33]

To further validate the results more apple images were segmented using iosdata technique on Red and Infrared filters. Figure 2-27 shows that results of Red filter images give better segmentation results when compared to Infrared filter images. After calyx removal, defect segmentation, and feature extraction, apples are graded by different classifiers as mentioned before. SVM proved to be to most accurate classifier (89.2%) in this case when using the iosdata method with the Red filter images. The LDC and k-NN classifiers performed lower with accuracies of 79% and 83% respectively.

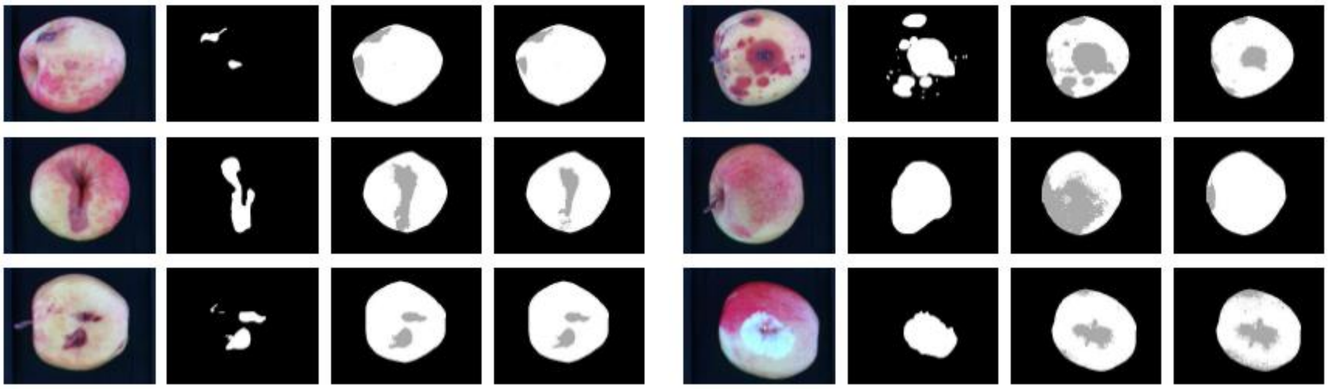


Figure 2-27: Band pass filters segmentation results 2 [33]

Table 2-6: Review of preharvest techniques

Paper	Acquisition Method	Test Plant	Target Disease	Environment	Automation/ Manual	Coverage Area	Segmentation Method	Classifier	Classification Accuracy
[30] (2009)	24-bit JPEG Image samples	Cotton Crop	Green stink bug, Bacteria angular, Ascochyta blight virus	Off Site	Images Taken Manually	Single leaf	NA	Support Vector Machine Classifier	90%
[28] (2011)	12 Mega pixel Digital Camera	Sugarcane Leaves	Fungi diseases	Off Site Controlled Background	Images Taken Manually	Single leaf	Triangle Thresholding	NA	98%
[7] (2011)	10 Mega pixel Digital Camera	Pomegranate leaves	General disease spots	Off Site Controlled Background	Images Taken Manually	Single leaf	K-means clustering	Fuzzy Logic Classification	NA
[29] (2011)	Digital Camera with 0.4 million CCD pixels	Tomato plants	Nutrient deficiency	Closed Chamber	Images Taken Manually	Single leaf	Percent Histogram	Fuzzy k-nearest neighbor classifier	82.5%
[31] (2014)	Multiple Gas Sensors (e-nose)	Soil fertility	VOCs	On Site	Semi-autonomous six-wheel robot	Lawn	NA	NA	NA
[8] (2016)	Three-CCD Multispectral Camera 1912x1076	Grapevines	Powdery mildew	On Site Controlled Lighting	Robot Rig Limited automation	Grapevine canopy	NA	NA	85%

Table 2-7: Review of postharvest techniques

Paper	Acquisition method	Test Fruit	Target Disease	Environment	Automation	Coverage Area	Segmentation Method	Classifier	Classification Accuracy
[6] (2002)	2 Monochrome cameras with band pass filter	Apples	Surface defects	Off Site Closed Chamber	Full automation	Multiple fruits	Simple subtraction thresholding	Neural Network	93%
[32] (2005)	Multispectral Camera (400 – 1000 nm)	Golden Delicious Apples	Apple bruises	Off Site Closed Chamber	Full automation	Multiple fruits	Simple thresholding	NA	86%
[33] (2005)	Monochrome Digital Camera with multiple band pass filters	Jonagold apples	Skin defects, bruises, and rot	Off Site Controlled Background and Lighting	Images Taken Manually	Single fruit	Isodata Thresholding	Support Vector Machine	89.2%
[2] (2012)	Apple Image dataset	Apples	Apple blotch, rot, and scab	Off Site Controlled Background and Lighting	Images Taken Manually	Single fruit	K-means Segmentation	Support Vector Machine	NA
[1] (2013)	Apple Image dataset	Apples	Apple scab and Apple rot	Off Site Controlled Background and Lighting	Images Taken Manually	Single fruit	Color and Texture Extraction	Neural Network	NA

2.3. Research Challenges

Both preharvest and postharvest disease detection techniques provide agriculturists and farmers with vital information on leaf and produce healthiness that can be used to monitor and revitalize plant and fruit health. However, both techniques face challenges that are discussed in this section.

2.3.1. Challenges with Preharvest Techniques

Preharvest imaging techniques is an early disease detection method, in which immediate action can be taken to revive plants and crops. This aspect is a major advantage, and cannot be achieved with postharvest techniques. However, there are limitations in the research when using preharvest techniques as listed below:

- Leaf images are taken off field and in controlled environments.
- Automation techniques are not fully developed.
- Limited applications for processing large numbers of plants in real time under field conditions.

2.3.2. Challenges with Postharvest Techniques

The use of imaging techniques for disease detection in postharvest fruits/vegetables proves to be very successful. Classification accuracies are relatively high when compared to preharvest techniques, and different diseases can be easily distinguished from one another. Many automation techniques such as fruit sorting have been developed with high efficiency and accuracy. However, there are some limitations listed below:

- Tests are not field based and conducted under controlled environments.
- Postharvest techniques are classified as late disease detection methods. It is difficult to cure the disease after the fruit has been fully developed.

Chapter 3
Research Methodology

Both preharvest and postharvest imaging as well as VOC profiling techniques have proven to be very effective in accurately classifying diverse types of plant diseases. Not only do these techniques give a good indication on overall plant health but also can accurately distinguish healthy produce from unhealthy produce. Preharvest techniques such as [29] and [30] achieved classification accuracy of plant diseases of 82.5% and 90% respectively. Postharvest techniques showed to be more promising as seen in [33] and [6] with classification accuracy of fruit defects of 89.2% and 93% respectively. Preharvest disease detection techniques are classified as an early disease detection method as seen in [29] and [8], in which immediate action can be taken to revive plants and crops. This aspect is a major advantage, and cannot be achieved with postharvest techniques. Postharvest techniques such as [6] and [32] have introduced automated systems in which defected postharvest produce can be distinguished and sorted automatically with an accuracy of 93% and 86% respectively. However, the integration of automation under real time and field based environments is still very limited. As seen in [8], in which a robot rig is used under field based conditions to detect diseases in grapevine canopy, but it is controlled manually, and full automation was not realized. Also, [31] implemented a semi-autonomous robot to test soil fertility under field based environments using multiple gas sensors, but results were ineffective in disease detection.

The methods presented in the literature differ greatly, from the use of simple digital cameras to the use of more advanced and sophisticated hyperspectral and multispectral imaging methods. Techniques such as [33], [6], and [32] used multispectral cameras and band pass filters, show higher classification accuracy as seen in Table 2-7 when compared to other techniques that use simple digital cameras for image acquisition. The use of multispectral cameras provides information of the image that can help to extract defected or diseased areas which may not be clear when using simple digital cameras with low sensitivity in the higher wavebands. [32] indicated that higher wavebands were helpful in detecting bruises in apples, and this cannot be achieved when using a simple digital camera. Preharvest disease techniques mostly use simple digital cameras for image capture of leaves and crops as seen with [28] and [29]. High accuracies of 98% and 82.5% are achieved respectively, because of the use of controlled backgrounds for precise segmentation of the diseased area.

It is clear, multispectral and hyperspectral imaging techniques proved to be the most reliable indirect method. However, hyperspectral imaging is a very costly technique, and there is still a limitation in the capability of designing systems for the detection of diseases in real time under field conditions. Most preharvest and postharvest techniques are completed under controlled environments and automation techniques are still not fully implemented. Because the use of preharvest techniques provide an early analysis of disease severity, it is more suitable to use over postharvest techniques.

3.1. Operational Scenario of the Proposed Preharvest Disease Detection Techniques

Because preharvest disease detection techniques provide an early analysis of plant health, two approaches are implemented to be used by agriculturists and scientists to monitor the overall health of plants. The first method is considered an Offline system in which plant leaf samples can be gathered and tested in a laboratory under controlled conditions, or leaf images captured on site. The second system is considered an Online system in which a robot will be designed to move along agriculture fields to test overall plant health. The two systems are developed and implemented to follow the scenario shown in Figure 3-1 in order to ensure both systems (offline and online systems) are functional and operational.

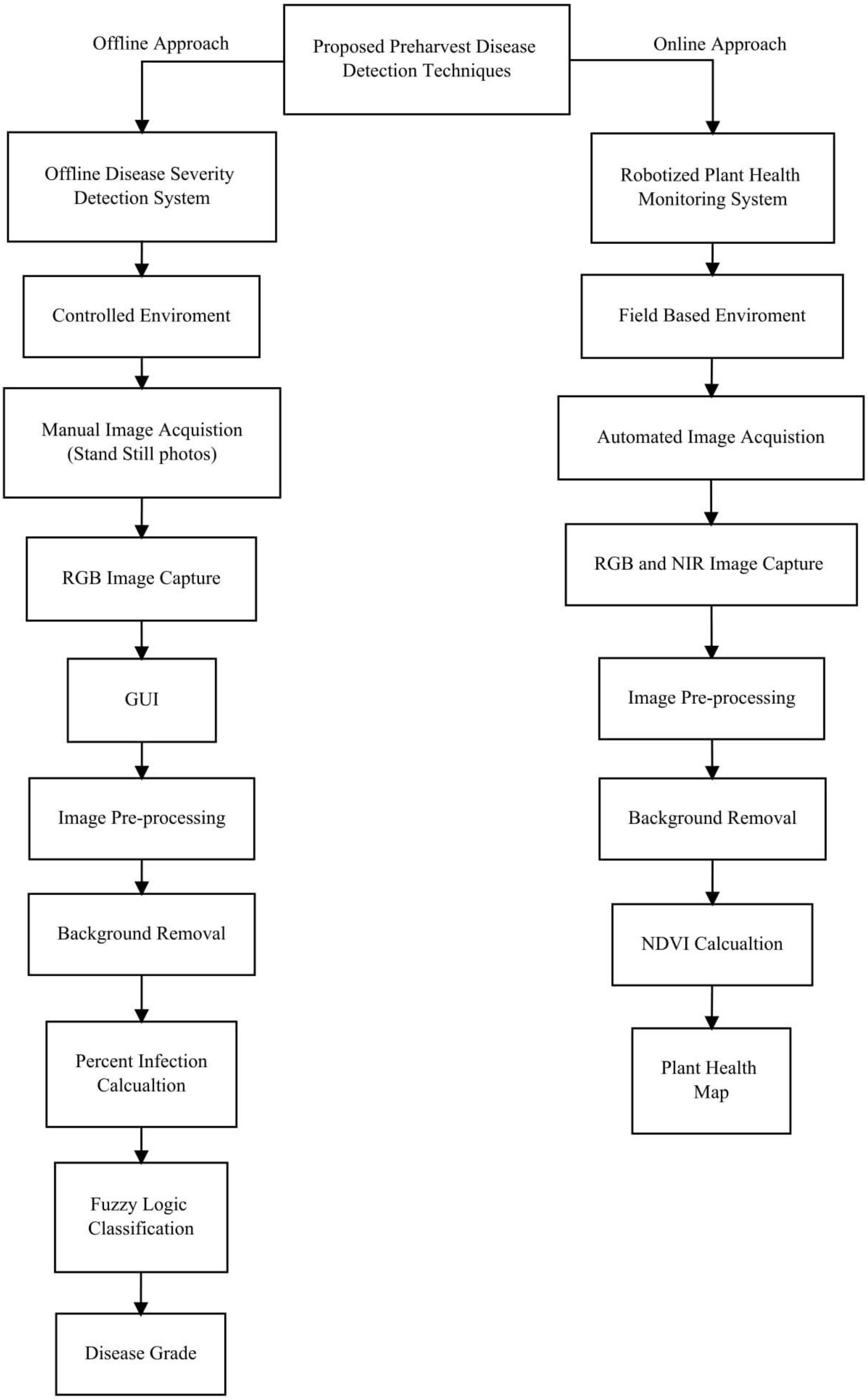

Proposed Preharvest Disease Detection Techniques
Offline Approach
Online Approach
Offline Disease Severity Detection System
Controlled Enviroment
Manual Image Acquistion (Stand Still photos)
RGB Image Capture
GUI
Image Pre-processing
Background Removal
Percent Infection Calcualtion
Fuzzy Logic Classification
Disease Grade
Robotized Plant Health Monitoring System
Field Based Enviroment
Automated Image Acquistion
RGB and NIR Image Capture
Image Pre-processing
Background Removal
NDVI Calcualtion
Plant Health Map

Figure 3-1: Research methodology flowchart

3.2. Scope

The scope of this thesis is to develop a pre-harvesting system with the capability to detect possible plant disease severity by understanding information about the overall plant health index information. The system under investigation will include a vision system that constitutes image acquisition, image processing and image analysis techniques to provide viable feedback on plant disease severity and overall plant health using an automated robotic solution by developing a mobile robot carrying the vision system to work within a greenhouse environment that has the suitable path segments and uniform terrain enabling the robot to move through it.

3.3. Research Methodology Outline

The outline for both systems are:

1. Offline Disease Severity Detection System
    a. Develop an offline leaf disease grading system using plant leaf images
    b. Use of Fuzzy Logic Classification to classify disease severity
    c. Implement a Graphical User Interface for ease of use
2. Online Plant Health Monitoring System
    a. Development and implementation of a robot integrated with RGB and NIR imaging cameras
    b. Implementation of two control schemes, where one is responsible for robot navigation and the other for image processing and analysis.
    c. Communication and synchronization of functions between control structures
    d. Calculation of NDVI for early information on overall plant health
    e. Provides real time and field based plant health monitoring
    f. Map of inspected locations indicating plant health

Chapter 4
Offline Plant Leaf Disease Severity System

A preharvest offline technique inspired by [28] and [7] is designed and implemented in which users can automatically calculate the disease severity of any leaf. Leaf samples are extracted from plants or taken on site in uncontrolled lighting, to be tested for infection. Fungi diseases are very common in leaves, and inhibit their growth immensely [28]. Because fungi-caused diseases in leaves appear as spots, it is applicable to use simple imaging techniques to detect the severity of the disease. Disease severity is expressed as the ratio between the affected area (A_D) and the total leaf area (A_T) [7]. If the lesion area ratio to leaf area ratio is high, then the leaf is said to have a high disease severity according to Table 2-4. Infection percentage (PI) is calculated as the area ratio between affected area and total leaf area as seen in Table 2-4. The classification of the disease severity is achieved by using a Fuzzy Logic technique. A graphical user interface is also implemented to enable agriculturists to monitor disease severity of leaves to have a clear indication on the overall health of the plant. The overall proposed methodology of the offline disease detection systems is represented in the flowchart shown in Figure 4-1. A few assumptions have been considered and listed below:

1. Leaf samples are picked and collected manually by a pathologist,
2. Images of leaf samples are taken with a digital color camera with a uniform black background,
3. Disease detection is non-specific (must have visible appearance on plant leaf), and
4. Only visible diseases can be detected and graded accordingly.

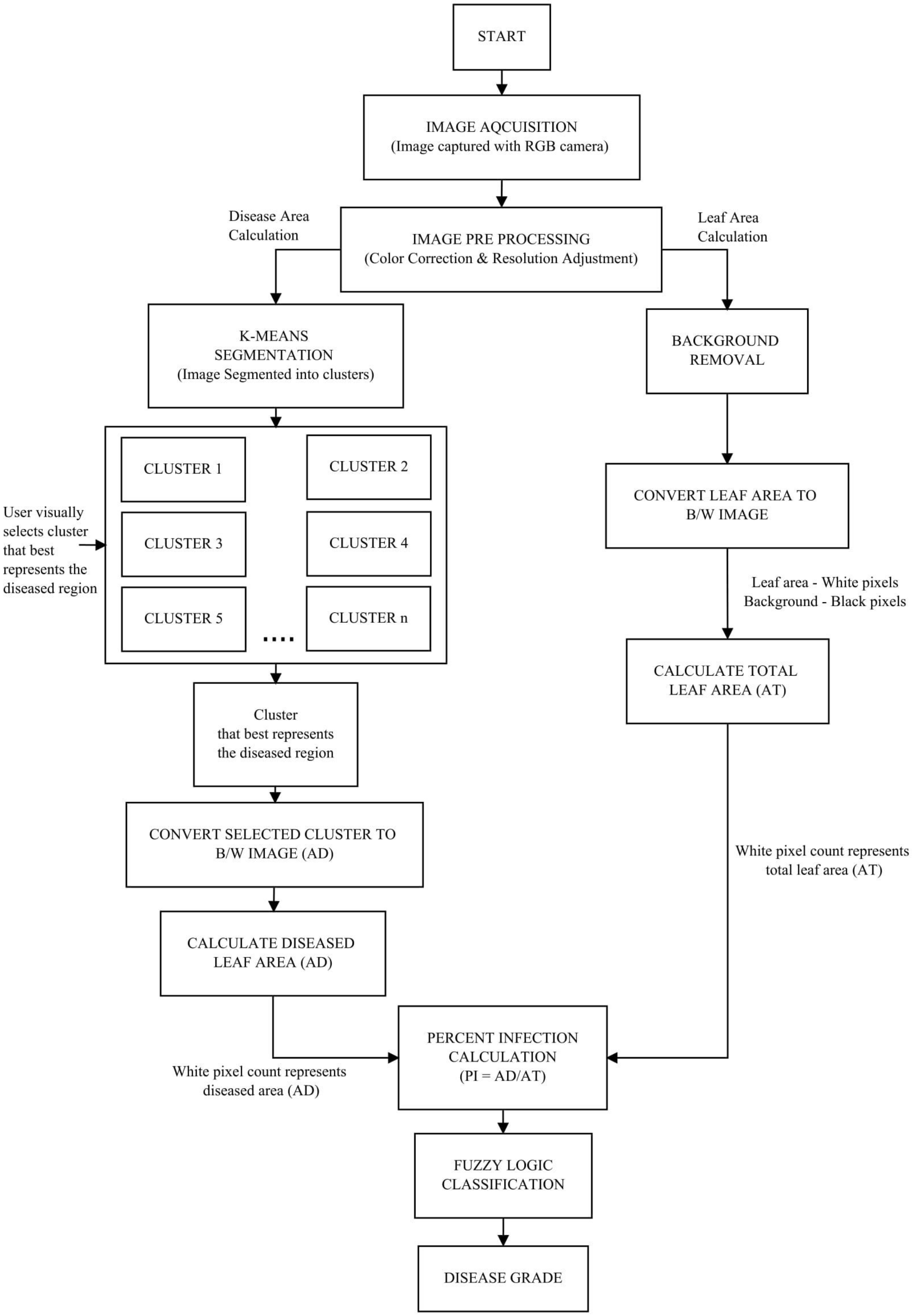

START
IMAGE AQCUISITION
(Image captured with RGB camera)
Disease Area Calculation
IMAGE PRE PROCESSING
(Color Correction & Resolution Adjustment)
Leaf Area Calculation
K-MEANS SEGMENTATION
(Image Segmented into clusters)
BACKGROUND REMOVAL
CLUSTER 1
CLUSTER 2
CLUSTER 3
CLUSTER 4
CLUSTER 5
....
CLUSTER n
User visually selects cluster that best represents the diseased region
CONVERT LEAF AREA TO B/W IMAGE
Leaf area - White pixels
Background - Black pixels
CALCULATE TOTAL LEAF AREA (AT)
Cluster that best represents the diseased region
CONVERT SELECTED CLUSTER TO B/W IMAGE (AD)
White pixel count represents total leaf area (AT)
CALCULATE DISEASED LEAF AREA (AD)
PERCENT INFECTION CALCULATION
(PI = AD/AT)
White pixel count represents diseased area (AD)
FUZZY LOGIC CLASSIFICATION
DISEASE GRADE

Figure 4-1: Flowchart for Offline Plant Leaf Disease Severity System

4.1. Image Acquisition

Image acquisition is the start of any image processing technique, in which images are digitalized and stored. For this system, leaf sample images with visible disease spots can be either captured with a digital color camera with a uniform background on site, or can be retrieved from any online database. It is recommended that images retrieved from the internet have a uniform background. This will allow for more accurate segmentation results as the background area can be easily distinguished from the leaf area. The resolution of images retrieved from an external database does not need to be set to a specific value, but this will be amended in the pre-processing stage. As for images taken on site, any digital RGB camera can be used. For this test, an iPhone camera was used to capture leaf images in the field and the specifications are shown in Table 4-1.

This camera provides high image capture detail with a resolution of 12 Megapixels. Another important feature of this camera is the low focal length of 4mm, which allows for a wider field of view. This aspect is vital for this system, as leaf images are taken at close range and it is important that the leaf area is completely represented in the image. A relatively low aperture of 1.8 allows for more light to enter the sensor, which can be beneficial in low light areas.

Table 4-1: iPhone 7 camera specifications

Resolution	3024 x 4032
Megapixel	12MP
Sensor	Sony Exmor RS (1.22 µm, 1/3")
Focal Length	4mm
Aperture	F/1.8

4.2. Image Pre-processing

In this stage images are resized to a fixed resolution to reduce the computational burden, and any other image adjustments can be achieved in this stage such as cropping, contrast enhancement, and angle correction.

4.2.1. Resolution Resize

To reduce computational complexity and time, it is recommended to decrease image resolution. However, this may diminish the accuracy of the disease severity calculation and the overall performance of the system. A good balance of computational time and accuracy should be achieved to ensure high accuracy of disease severity calculation in a reasonable amount of time. Therefore, to test the different resolution scales each image will be tested in multiple resolutions and the computational time and accuracy are compared to find an appropriate resolution scale to use shown in Sec. 4.8. Table 4-2 illustrates the resolutions that will be used for images retrieved from the online database and from iPhone camera. In order to preserve the original aspect ratio, the original width and height are both halved. This will result in an image with the same aspect ratio but with a quarter resolution of the original. The original images can be seen in Figure 4-2 and Figure 4-3.

Table 4-2: Image resolution scales

	Original Resolution (width x height)	Quarter Resolution (width x height)
Internet Database	590 x 443	295 x 222
iPhone camera	4032 x 3024	2016 x 1512

3 different interpolation techniques are considered and compared in terms of speed and its effect on edges or borders. Table 4-3 illustrates the different techniques, to find the most appropriate image resizing technique for this system.

Table 4-3: Interpolation methods

	Nearest-neighbor interpolation	Linear interpolation	Bilinear interpolation
Output pixel	Assigned the value of the pixel that the point falls within	Weighted average of pixels in the nearest 2-by-2 neighborhood	Weighted average of pixels in the nearest 4-by-4 neighborhood
Speed	Fast	Slower than Nearest-neighbor	
Edges	Sharp Edges	Smooth edges	

Nearest-neighbor is selected because the interpolation method provides edges that are sharp and can help in segmenting the leaf area from the background. The resized images using nearest-neighbor interpolation can be seen in Figure 4-4 and Figure 4-5.

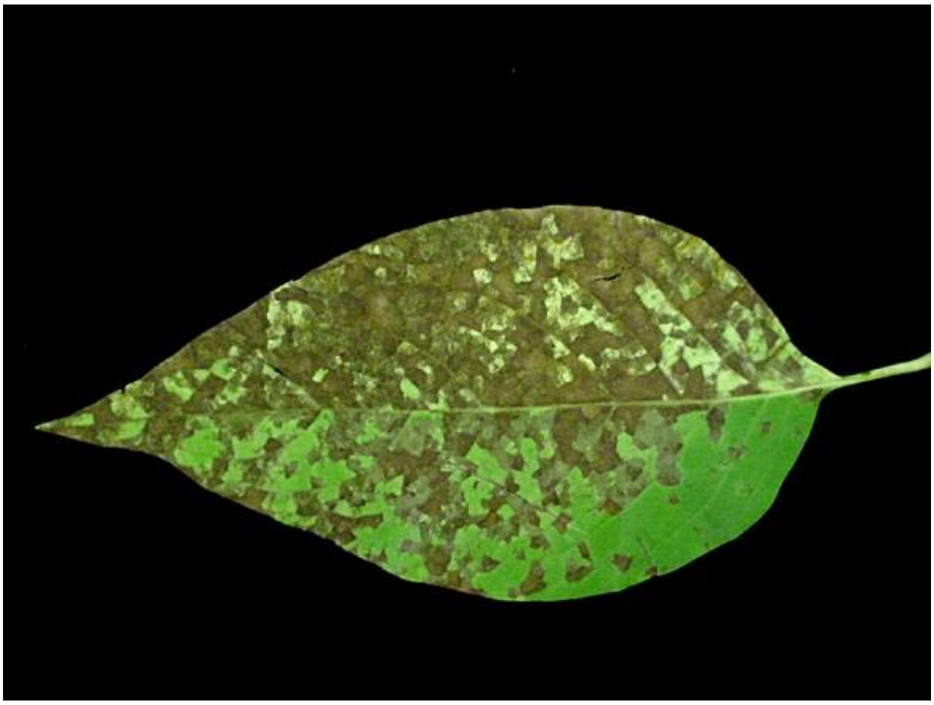


Figure 4-2: Input leaf image retrieved from internet in RGB color space (590 x 443 pixels)

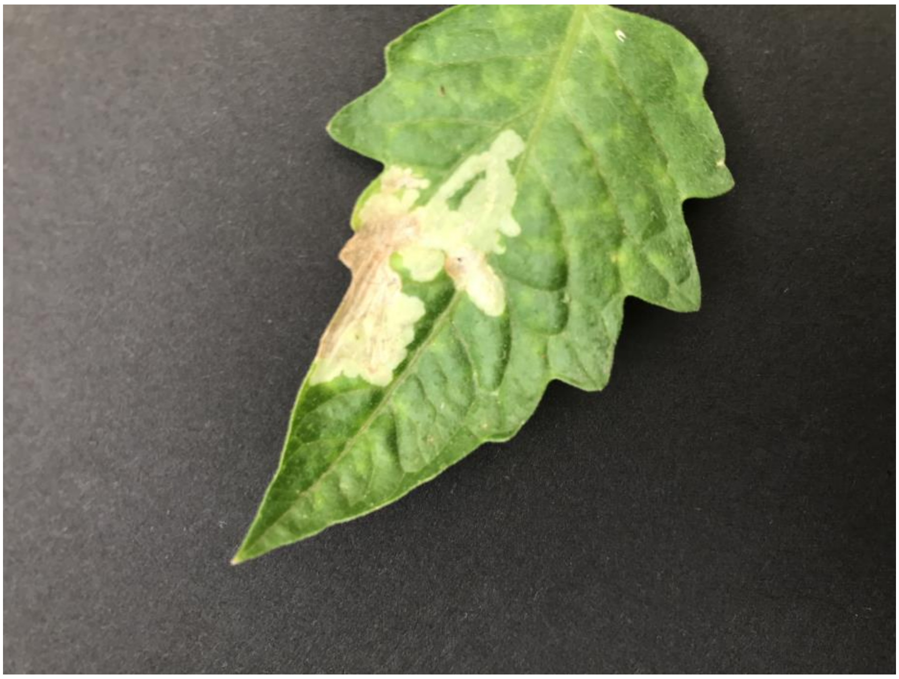


Figure 4-3: Input leaf image captured from iPhone in RGB color space (4032 x 3024 pixels)

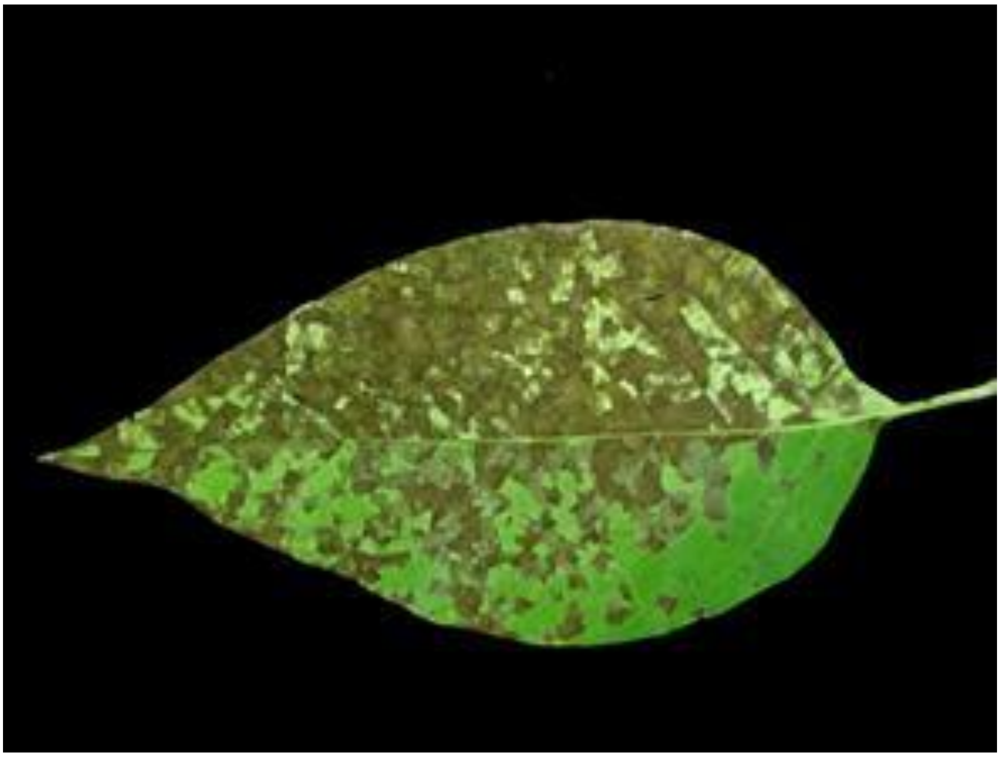


Figure 4-4: Internet leaf image quarter resolution (295 x 222 pixels)

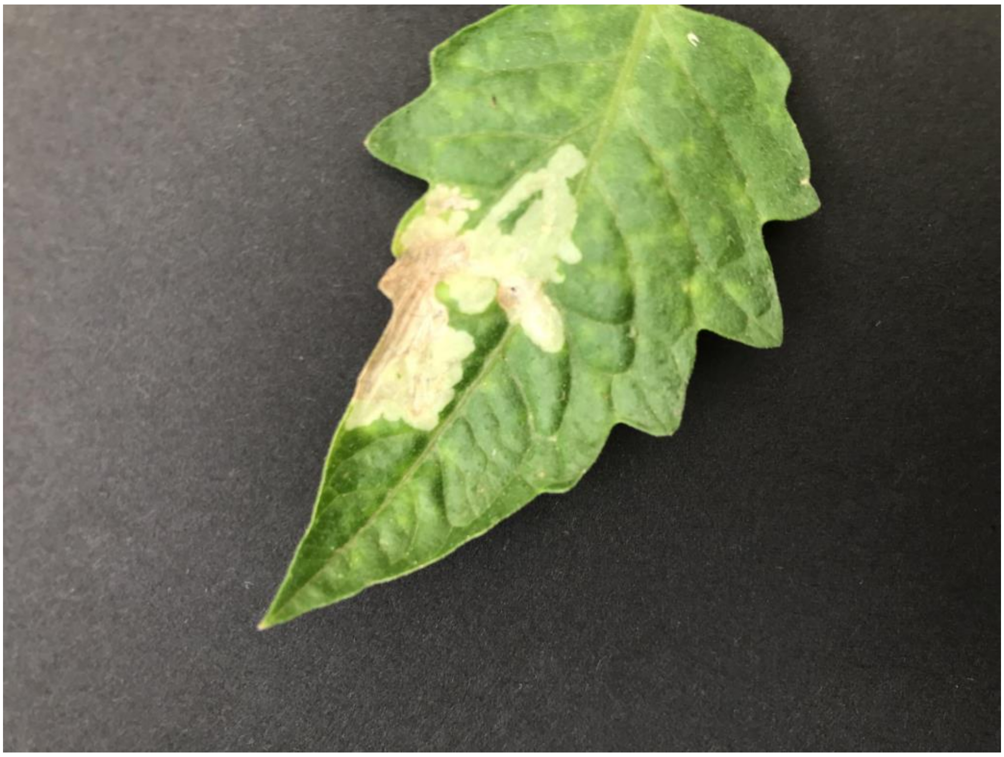


Figure 4-5: iPhone leaf image quarter resolution (2016 x 1512 pixels)

4.2.2. Color Space Conversion

Also, the selection and conversion of the color space can be completed in this stage. In this case the Lab color space is selected over RGB color model due to multiple reasons. The RGB color model can be seen in Figure 4-6. Not only does the Lab color space exceed the RGB color gamut, but it is also device independent [34]. This aspect is important because the colors are defined independent of their nature of creation, and is therefore less prone to changes due to light [1]. This is useful in cases where images are captured in areas with non-uniform lighting conditions. The space itself is a three-dimensional space, that contains an infinite number of possible representations of colors as shown in Figure 4-7. However, in practice, the space is usually mapped onto a three-dimensional integer space for digital representation. The L*a*b* space consists of a luminosity 'L*' or brightness layer, chromaticity layer 'a*' indicating where color falls along the red-green axis, and chromaticity layer 'b*' indicating where the color falls along the blue-yellow axis [35].

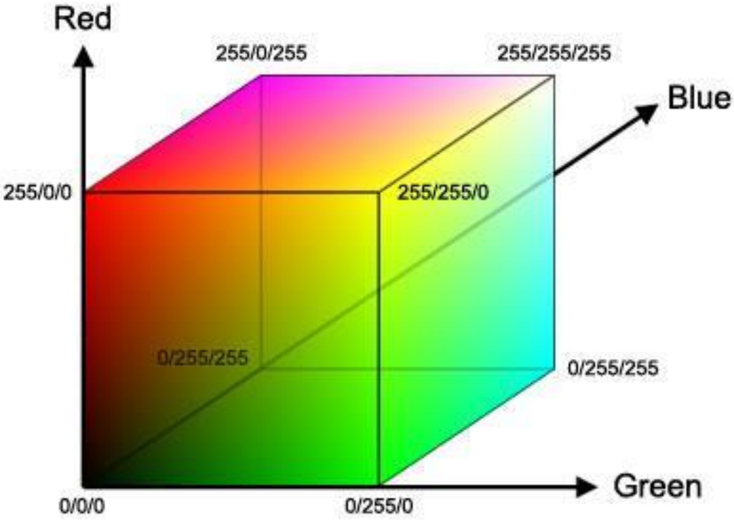


Figure 4-6: RGB color space [36]

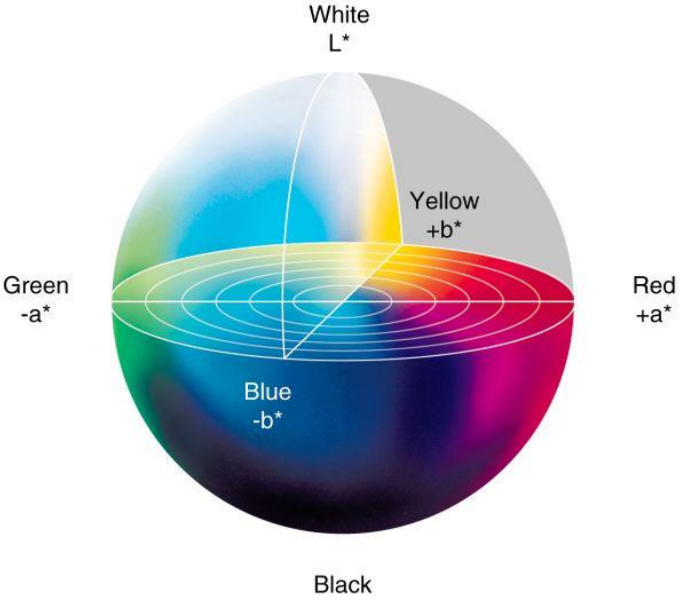


Figure 4-7: L*a*b* color space [37]

There are no simple formulas for conversion between sRGB and L*a*b*, because the sRGB color model is device-dependent. The sRGB values first must be transformed to the CIE 1931 color space and then transformed into L*a*b* [37]. The two leaf images shown previously in Figure 4-2 and Figure 4-3 converted to the L*a*b color space and can be seen in Figure 4-8 and Figure 4-9.

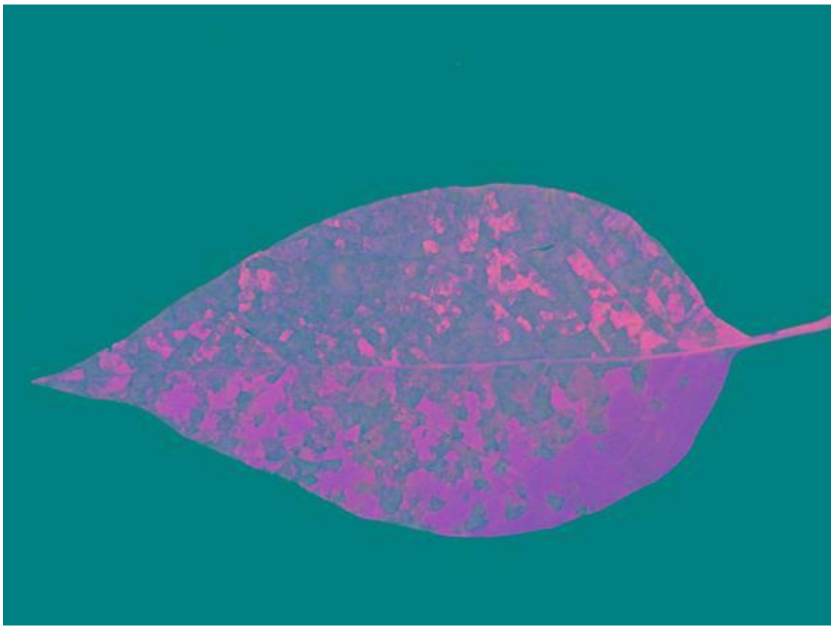


Figure 4-8: Internet leaf image in L*a*b color space

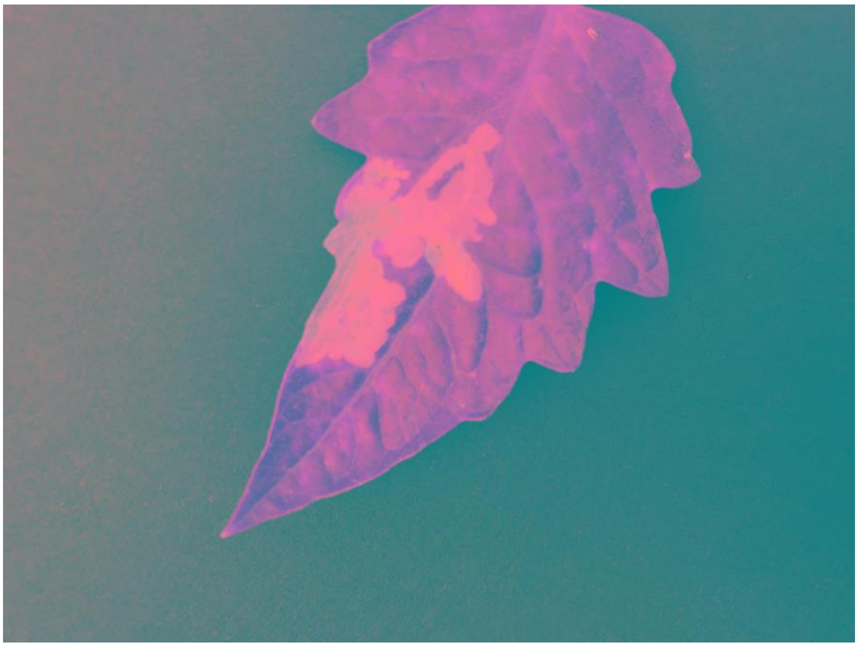


Figure 4-9: iPhone leaf image in L*a*b color space

4.3. Background Image Segmentation

Image segmentation is the process in which the digital image is partitioned into constituent regions, so that the different regions can be easily distinguished and analyzed. Segmentation can be achieved by various techniques such as clustering methods, compression-based methods, and histogram-based methods. The first step is to accurately segment the region of interest from the background. In this case the leaf area should be recognized and distinguished from the uniform background. Different thresholding techniques will be tested and compared to select the most appropriate for this system.

Before implementing thresholding techniques, the original RGB image is converted into a greyscale image using a simple conversion method. The luminosity method or weighted method considers for human perception by calculating the weighted average as illustrated in Equation 4.1. The red, green, and blue channels represent a 2D array of pixel values ranging from 0 to 255. The greyscale conversion of the Internet leaf image and iPhone leaf image are shown in Figure 4-10 and Figure 4-11.

$$Greyscale\ value_{i,j} = \left(0.2989\ \times R_{i,j}\right) + \left(0.5870\ \times G_{i,j}\right) + \left(0.1140\ \times B_{i,j}\right) \tag{4.1}$$

Where i and j represent the pixel location in the matrices

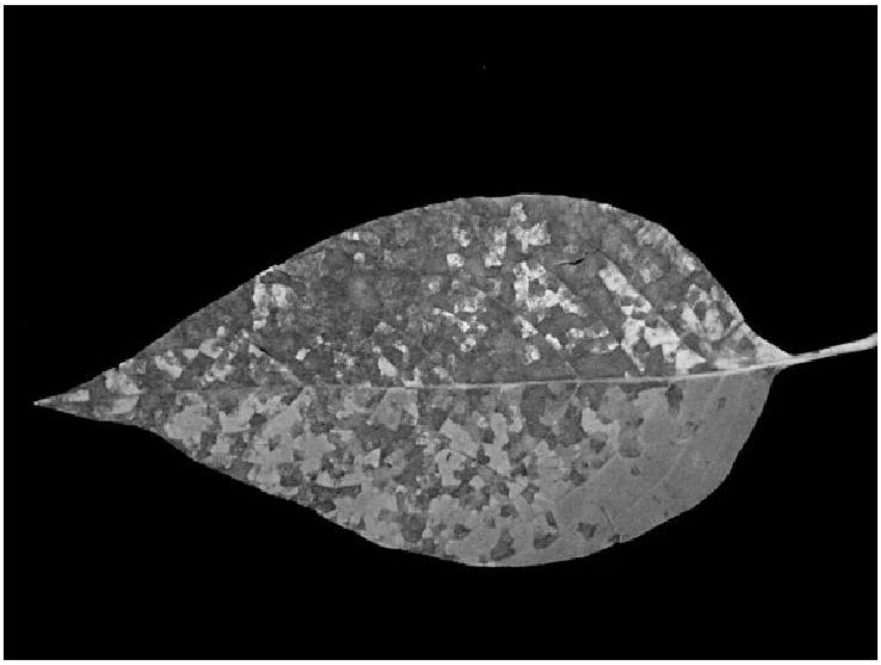


Figure 4-10: Internet leaf image after greyscale conversion

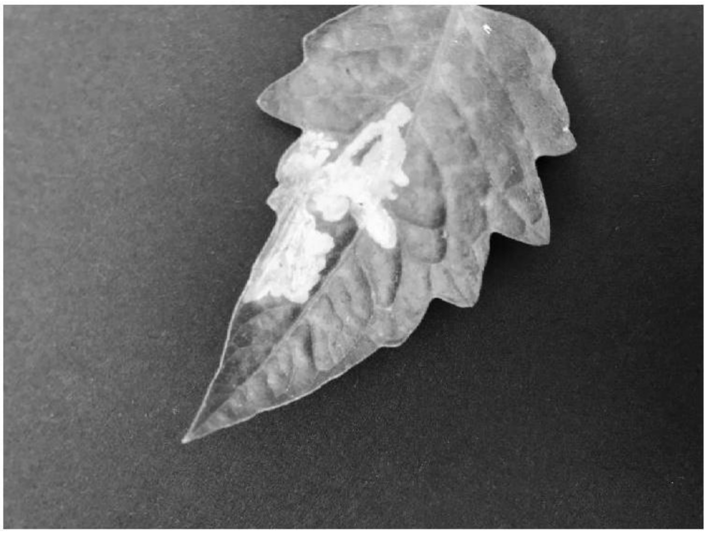


Figure 4-11: iPhone leaf image after greyscale conversion

4.3.1. Otsu Thresholding

Threshold segmentation is achieved by setting a proper threshold value to distinguish the foreground from the background. The greyscale image is then converted into a binary image according to the threshold set value. The output binary image replaces all pixels in the input image with luminance greater than the threshold value with the value 1 (white) and replaces all other pixels with the value 0 (black). The threshold is set as a value between 0 and 255 where values closer to 0 signify a threshold value closer to lower grayscale values (black) and vice versa. Otsu's thresholding method uses histogram information to iterate through all possible range of threshold values and selects the threshold that:

- Minimizes the within-class variance
- Maximizes the between-class variance

The Otsu method is demonstrated using a simple 6x6 image with a greyscale range from 0 to 5. The image and its histogram shown in Figure 4-12.

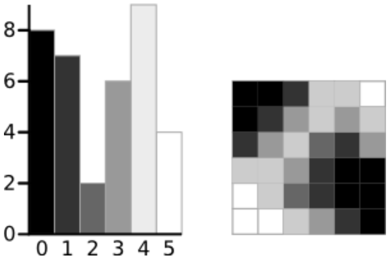


Figure 4-12: 6x6 Image (right), Histogram (left) [38]

Otsu method searches all possible threshold values from 0 to 5 to find the minimum within-class variance and maximum between class variance as illustrated in Equation 4.2 and 4.3 respectively [38].

$$\sigma_W^2(t) = w_b(t)\sigma_b^2(t) + w_f(t)\sigma_f^2(t) \tag{4.2}$$

$$\sigma_B^2(t) = w_b(t)w_f(t)\left[\mu_b(t) - \mu_f(t)\right]^2 \tag{4.3}$$

Where the, weights w_b and w_f are the probabilities of the background and foreground classes separated by a threshold t, and σ_b^2 and σ_f^2 are variances of these two classes respectively. The class probability $w_b(t)$ and $w_f(t)$ illustrated in Equation 4.4 is computed from number of threshold values (L), which is 6 in this example. The mean μ illustrated in Equation 4.5 and variance σ illustrated in Equation 4.6 are also calculated for the background and foreground [38] .

$$w_b(t) = \textstyle\sum_{i=0}^{t-1} p(i) \qquad\qquad w_f(t) = \textstyle\sum_{i=t}^{L-1} p(i) \tag{4.4}$$

Probability equation for background (left) and foreground (right)

$$\mu_b(t) = \textstyle\sum_{i=0}^{t-1} i\,\frac{p(i)}{w_b} \qquad\qquad \mu_f(t) = \textstyle\sum_{i=t}^{L-1} i\,\frac{p(i)}{w_f} \tag{4.5}$$

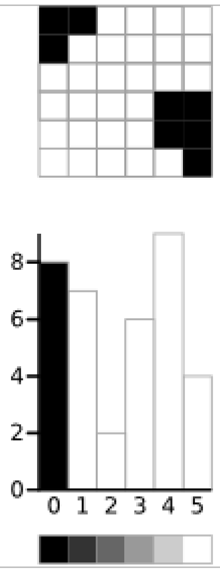
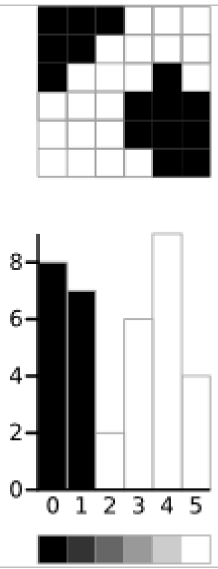
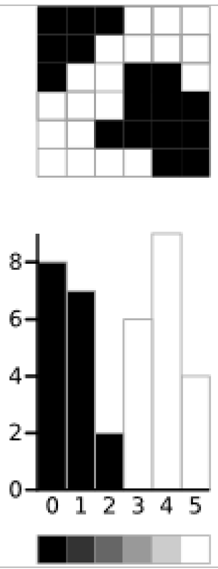
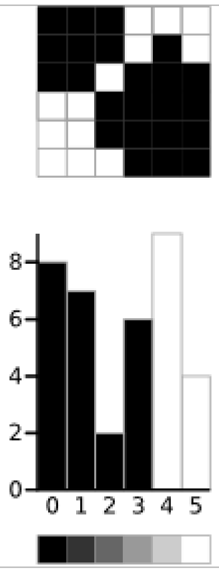
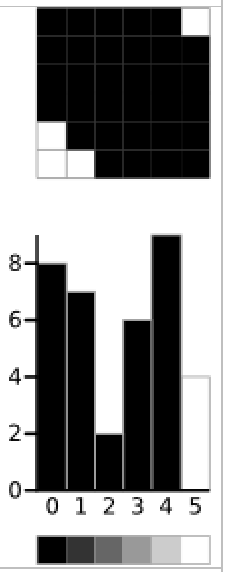
Mean equation for background (left) and foreground (right)

$$\sigma_b^2(t) = \textstyle\sum_{i=0}^{t-1} (i - \mu_b)^2\, p(i) \qquad\qquad \sigma_f^2(t) = \textstyle\sum_{i=t}^{L-1} \left(i - \mu_f\right)^2 p(i) \tag{4.6}$$

Variance equation for background (left) and foreground (right)

These equations are used to calculate within class variance $\sigma_W^2(t)$ for all threshold values (0 – 5) as seen in Table 4-4.

Table 4-4: Otsu Thresholding at different Threshold values [38]

Threshold	T=1	T=2	T=3	T=4	T=5
					
Weight Background	w_b = 0.222	w_b = 0.4167	w_b = 0.4722	w_b = 0.6389	w_b = 0.8889
Mean Background	μ_b = 0	μ_b = 0.4667	μ_b = 0.6471	μ_b = 1.2609	μ_b = 2.0313
Variance Background	σ_b^2 = 0	σ_b^2 = 0.2489	σ_b^2 = 0.4637	σ_b^2 = 1.4102	σ_b^2 = 2.5303
Weight Foreground	w_f = 0.7778	w_f = 0.5833	w_f = 0.5278	w_f = 0.3611	w_f = 0.1111
Mean Foreground	μ_f = 3.0357	μ_f = 3.7143	μ_f = 3.8947	μ_f = 4.3077	μ_f= 5.000
Variance Foreground	σ_f^2 = 1.9639	σ_f^2 = 0.7755	σ_f^2= 0.5152	σ_f^2 = 0.2130	σ_f^2 = 0
Within Class Variance	σ_W^2 = 1.5268	σ_W^2 = 0.5561	$\boldsymbol{\sigma_W^2}$ **= 0.4909**	σ_W^2 = 0.9779	σ_W^2 = 2.2491
Between Class Variance	σ_B^2 = 1.5928	σ_B^2 = 2.5635	$\boldsymbol{\sigma_B^2}$ **= 2.6287**	σ_B^2 **=** 2.1417	σ_B^2 = 0.8705

The lowest within class variance and highest between class variance can be seen for threshold value 3. Therefore, this is the final selected threshold. All pixels with a level less than 3 are background, all those with a level equal to or greater than 3 are foreground as shown in Figure 4-13.

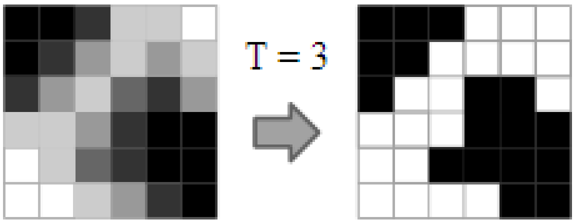


Figure 4-13: Result of Otsu Thresholding at threshold value 3 [38]

Figure 4-14 and Figure 4-15 represent the threshold background segmentation results for the internet leaf image and the image captured from the iPhone camera using the Otsu thresholding technique. Table 4-5 illustrates the optimal threshold value found using the Otsu thresholding technique.

Table 4-5: Otsu threshold values for leaf images

Image	Otsu threshold greyscale (0-255)
Leaf image retrieved from internet	58.0125
Leaf image captured from iPhone	102.9945

Internet Leaf Image

If pixel greyscale > 58.0125 then pixel set to 1 (white)

If pixel greyscale < 58.0125 then pixel set to 0 (black)

iPhone Leaf Image

If pixel greyscale > 102.9945 then pixel set to 1 (white)

If pixel greyscale < 102.9945 then pixel set to 0 (black)

The background segmentation result for the leaf image retrieved from the internet shown previously in Figure 4-10 seems to show good promise on accurately highlighting the leaf area from the background as seen in Figure 4-14. However, for the iPhone leaf image shown previously in Figure 4-11, this method proves ineffective in segmenting images with non-uniform lighting as seen in Figure 4-15. Because the intensity of black background is not uniform, different areas will incorrectly be defined as foreground. Therefore, this segmentation method will not be efficient to use, as leaf images will have different background intensities. The segmentation method should be able accommodate for various background intensities and partition the foreground from the background accordingly.

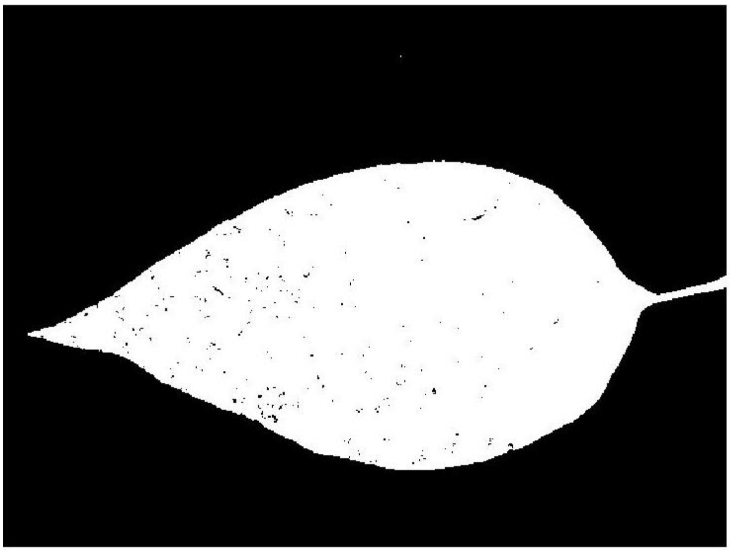


Figure 4-14: Internet leaf Otsu threshold background segmentation

Figure 4-15: iPhone leaf Otsu threshold background segmentation

4.3.2. Triangle Thresholding

The triangle thresholding method is a technique in which the threshold value is set according to the image histogram information. A line is constructed between the maximum peak of the histogram at brightness b_{max} and the lowest value b_{min} as shown in Figure 4-16. The distance d between the line and the histogram is computed for all brightness values from b_{max} to b_{min} . The objective of this technique is to find the maximum perpendicular distance (d_{b_i}) between the bin values and the line. Accordingly, the bin value with the maximum distance is concluded as the threshold value b_i. The threshold is used as the cut-off value in which pixels below are set as one group and pixels above as the other. [28]. b_i

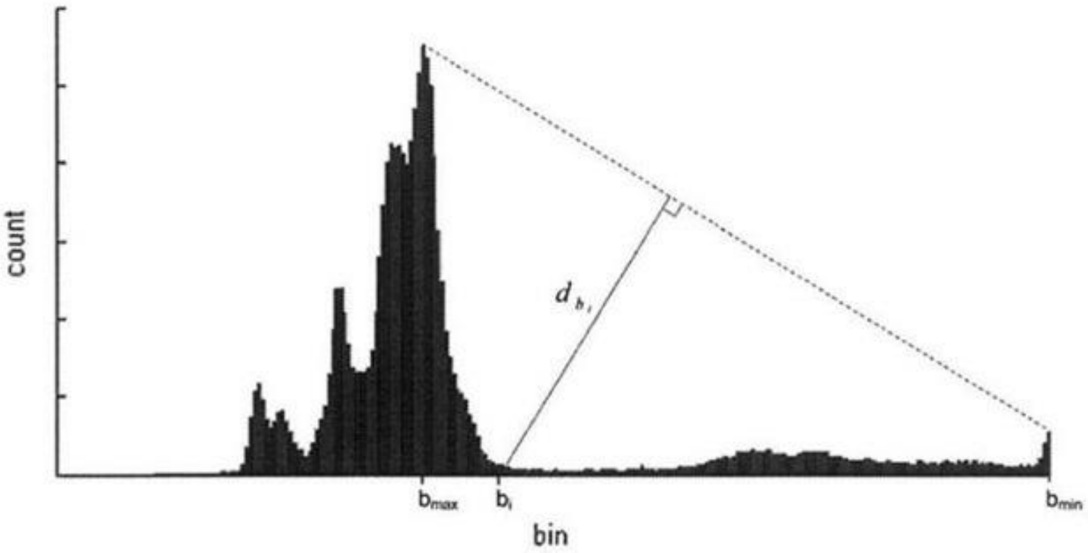


Figure 4-16: Intensity Histogram and Triangle Thresholding Technique [28]

Figure 4-17 and Figure 4-18 show the histogram results of the leaf image retrieved from the internet and the leaf captured from the iPhone camera from Figure 4-10 and Figure 4-11. The base line is also shown between the histogram peak and the farthest end of the histogram. It is clear the histogram for the leaf image retrieved from the internet that the grey levels of the pixels are set into two groups. The lower grey level range clearly indicate the background where the number of pixels is quite high, and the grey levels around the middle indicate the leaf area. On the other hand, the leaf image captured from the iPhone contains a wider range of grey level values because of the non-uniformity of the background intensity.

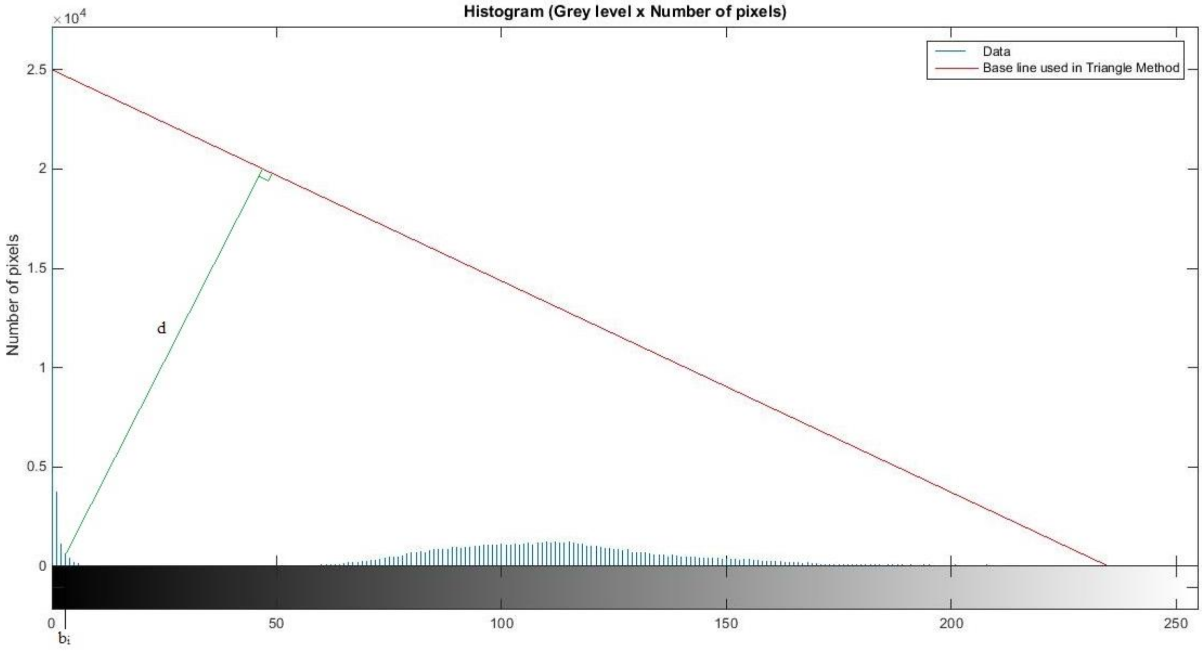


Figure 4-17: Internet leaf histogram

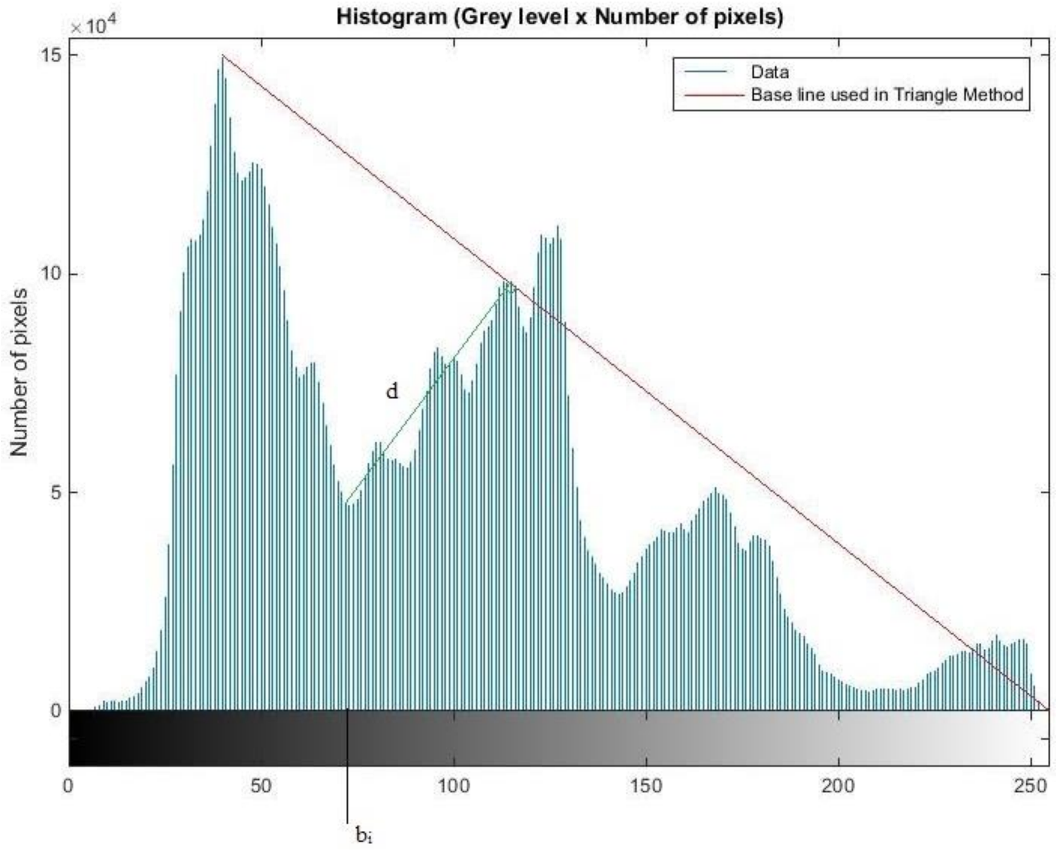


Figure 4-18: iPhone leaf histogram

After the histogram is constructed the threshold (b_i) is then found as the point of maximum distance between the line and the histogram (d) shown in Table 4-6.

Table 4-6: Triangle threshold values for images

Image	Triangle threshold greyscale b_i (0-255)
Leaf image retrieved from internet	1.989
Leaf image captured from iPhone	72.726

Figure 4-19 and Figure 4-20 show the result of the threshold value set on the binary conversion of both images. Figure 4-19 shows mixed results when compared to Figure 4-14 where Otsu thresholding was used. The internal area of the leaf is more properly represented, but the threshold selected is inaccurate in correctly segmenting the edges of the leaf with the background. Figure 4-20 shows a slightly improved result when compared to Figure 4-15, where the leaf area is characterized more accurately, but still proves to be ineffective in segmenting the total background from the leaf area because of the complexity of the histogram. There is no clear indication from the histogram of the foreground pixels and the background pixels, this is clearly due to the non-uniformity of the lighting.

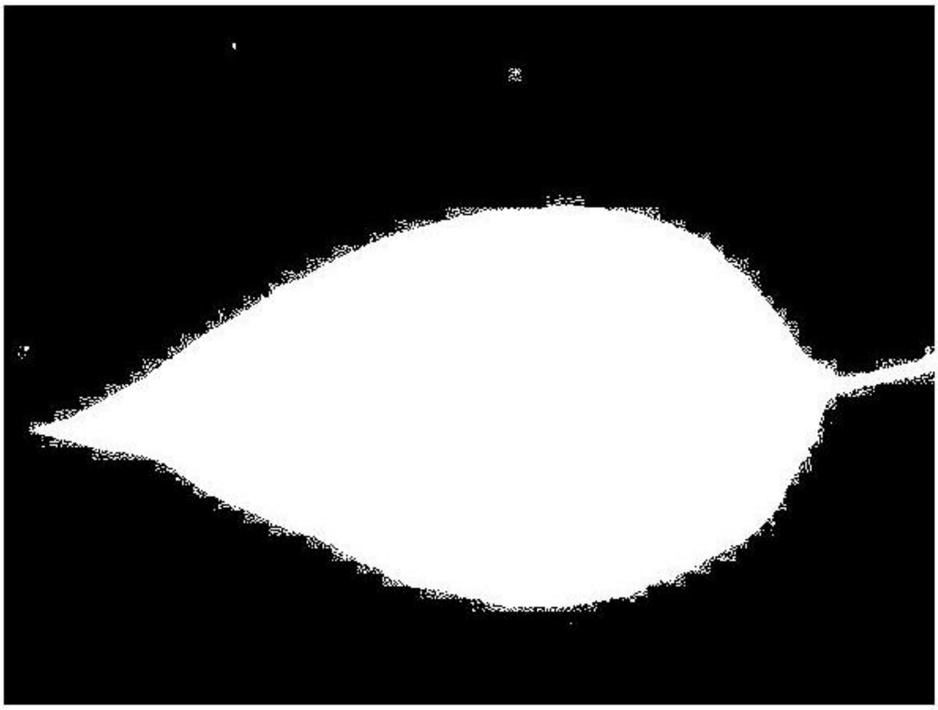


Figure 4-19: Internet leaf triangle threshold background segmentation

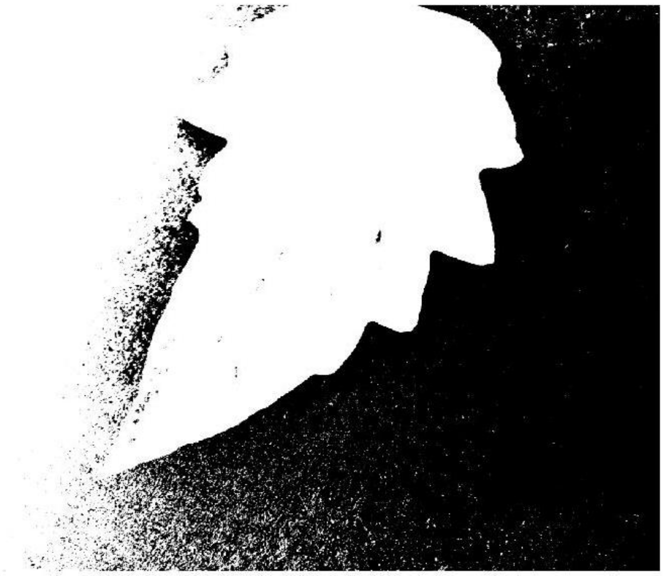


Figure 4-20: iPhone leaf triangle threshold background segmentation

4.3.3. K-means Segmentation with Triangle Thresholding

The goal of k-means clustering is to group similar objects together. Given a data matrix $X = [x_1, \dots, x_n]$ and k initial clustering centroids, k-means clustering targets to partition each x_i into k clusters, in which x_i belongs to its nearest cluster according to the squared Euclidian distance. K-means clustering has been adopted into many applications including market segmentation, astronomy and agriculture. K-means clustering example is shown in Figure 4-21. For dataset $X = [x_1, \dots, x_n]$ the main steps are as follows:

1. Initialize k cluster centroids $c_1, c_2, \dots, c_k$ randomly.
2. For all i, assign x_i to cluster g_i where
   $g_i = argmin_{j=1,2,..,k} \|x_i - c_j\|^2$
3. Recalculate the position of the centroids as the mean of its data points.
   $c_j = mean(\{x_i : x_i\} \in g_j)$
4. Repeat Steps 2 and 3 until the centroids no longer move.

$\|x_i - c_j\|^2$ is the squared Euclidian distance measurement between data point x_i and its corresponding cluster center c_j [39].

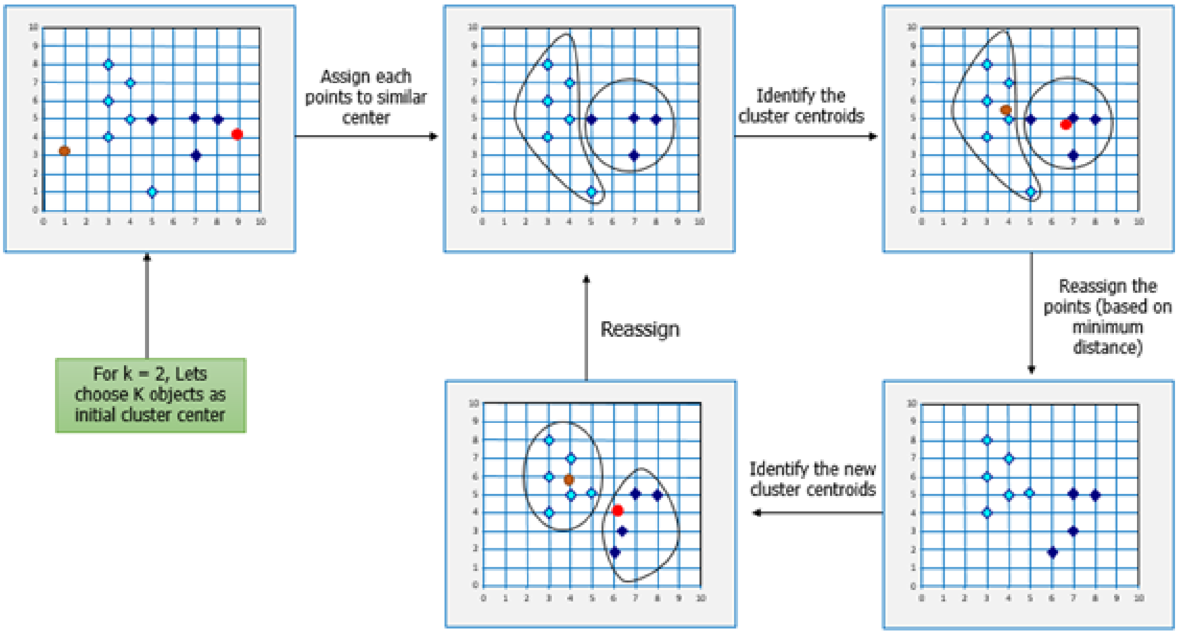

Assign each points to similar center
Identify the cluster centroids
Reassign the points (based on minimum distance)
Identify the new cluster centroids
Reassign
For k = 2, Lets choose K objects as initial cluster center
0 1 2 3 4 5 6 7 8 9 10

Figure 4-21: K- means clustering example [39]

The value k represents the number of clusters, in which each cluster has a defined centroid. The centroids are initially chosen at various locations and then each point in the data set is associated to the nearest centroid according to the squared Euclidian distance [39]. The centroids are re-positioned as the mean of its data points. This process is repeated until the centroids no longer change their position and the optimum position for the centroids has been found. K-means clustering can be used as a segmentation technique as seen in [2] and [7], in which the image is segmented into k number of clusters. Similar regions are clustered together and segmented from other regions of the image.

The major advantage of using k-means clustering is that the objective function is simple and is easy to implement. However, there are limitations that need to be considered:

1. The user must initially specify k (number of clusters)
2. The result of k-means heavily relies on the initialization of the cluster centroids
3. Computation time is dependent on k and number of data points

The selection of the value k is essential in accurately clustering multiple regions into their appropriate cluster without losing data. Higher values of k can result in some clusters that may need to be grouped with other clusters, and lower values of k may result in smaller number of clusters that may not accurately represent its data fully. The optimal choice of k should achieve a balance between maximum compression of data in a single cluster with maximum accuracy by assigning each data point to its own cluster. According to [2] and [7] values of k under 10 resulted in more accurate results for leaf images. Hence, after multiple trial runs the value of k was chosen as 10 in this case. To properly distinguish the background from the leaf area, the cluster that contains the background will be subtracted from the leaf image so that binary conversion using triangle thresholding can be completed accurately. This technique can be used to find the cluster that contains the background and properly segment it from the leaf area.

The 3-dimensional RGB image is converted into L*a*b color space as seen in Figure 4-22. This conversion is explained in Sec. 4.2.2. The color information is now stored in the 'a*b*' channel. Therefore, the 'a*' and 'b*' values are constructed into matrix form and plotted as points in a 2D space as shown in Figure 4-23.

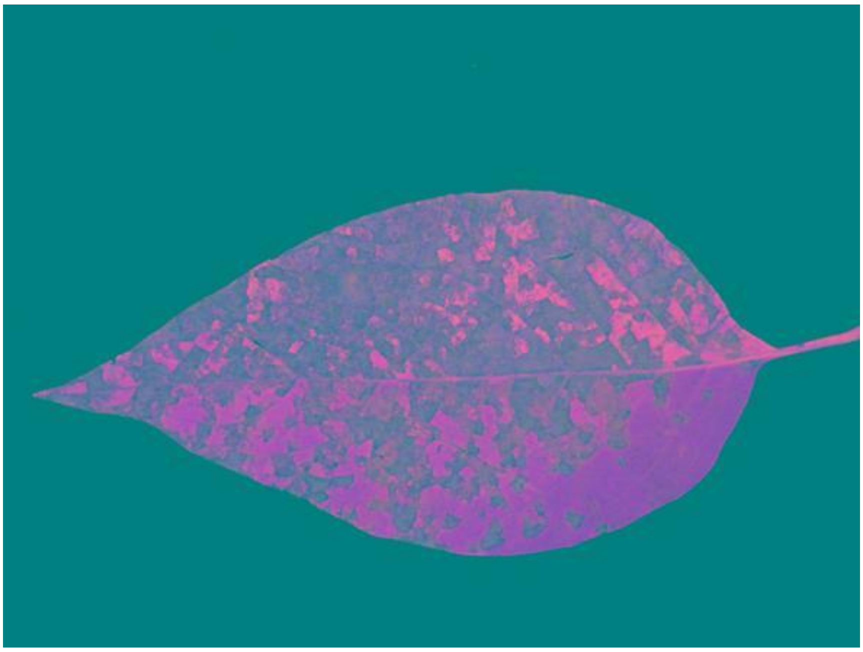


Figure 4-22: Internet leaf image in L*a*b color space

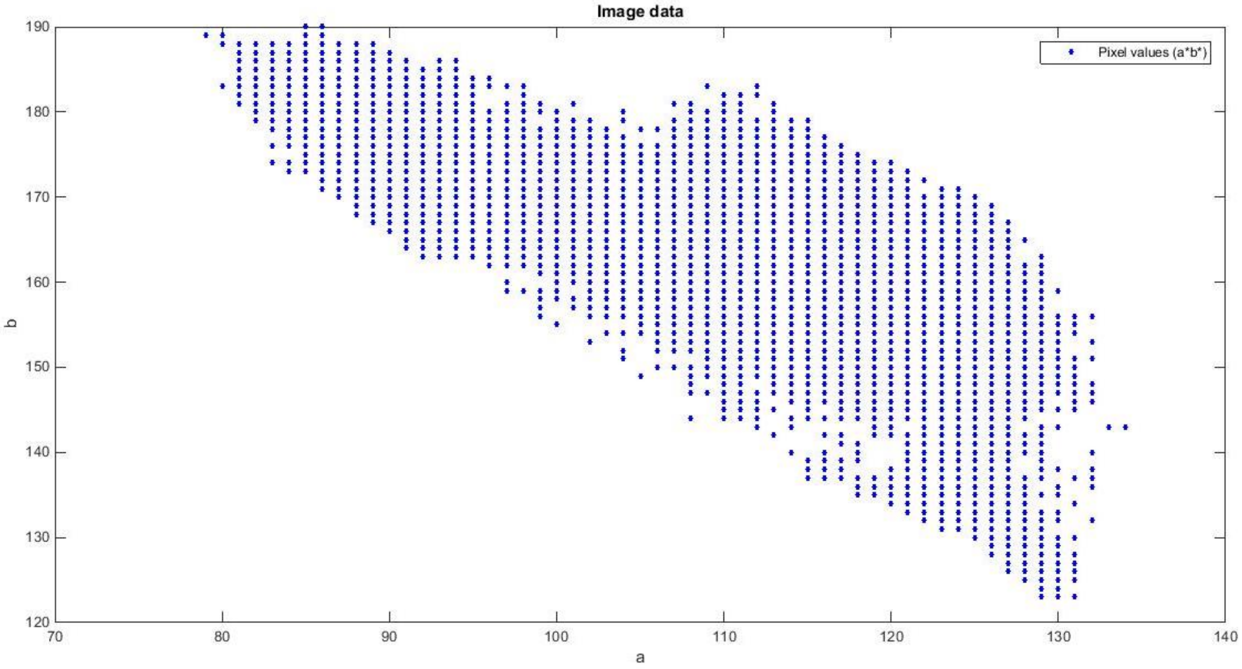


Figure 4-23: Image data for Internet leaf image

These points are then segmented into clusters using k-means clustering, and the 10 different clusters with centroids can be seen in Figure 4-24 for the internet leaf and Figure 4-25 for the iPhone captured leaf. The points are segmented into different clusters with each cluster representing a certain region of the leaf image. Each cluster is then extracted and the 'a*' and 'b*' values are used to draw the segmented image as shown through Figure 4-26 to Figure 4-29 .

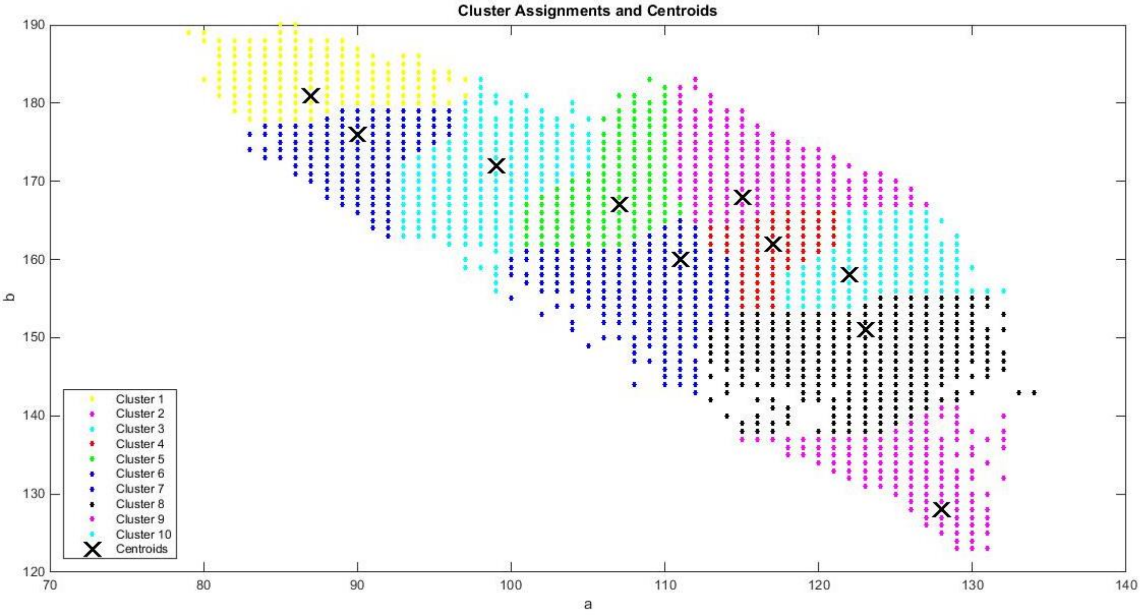


Figure 4-24: Internet leaf image cluster assignments and centroids

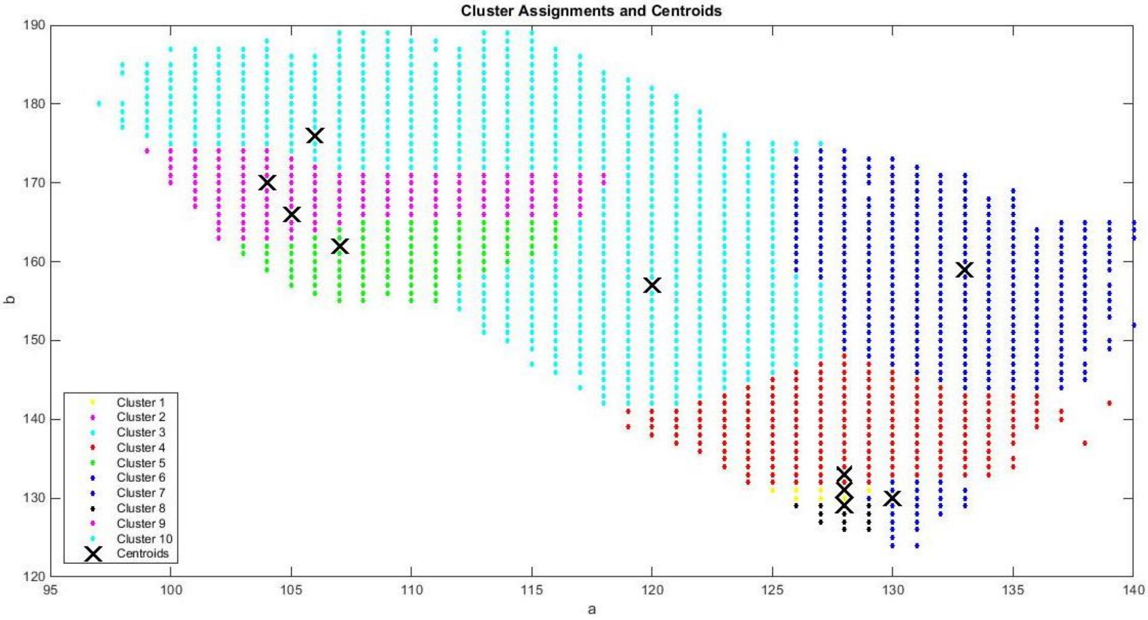


Figure 4-25: iPhone leaf image cluster assignments and centroids

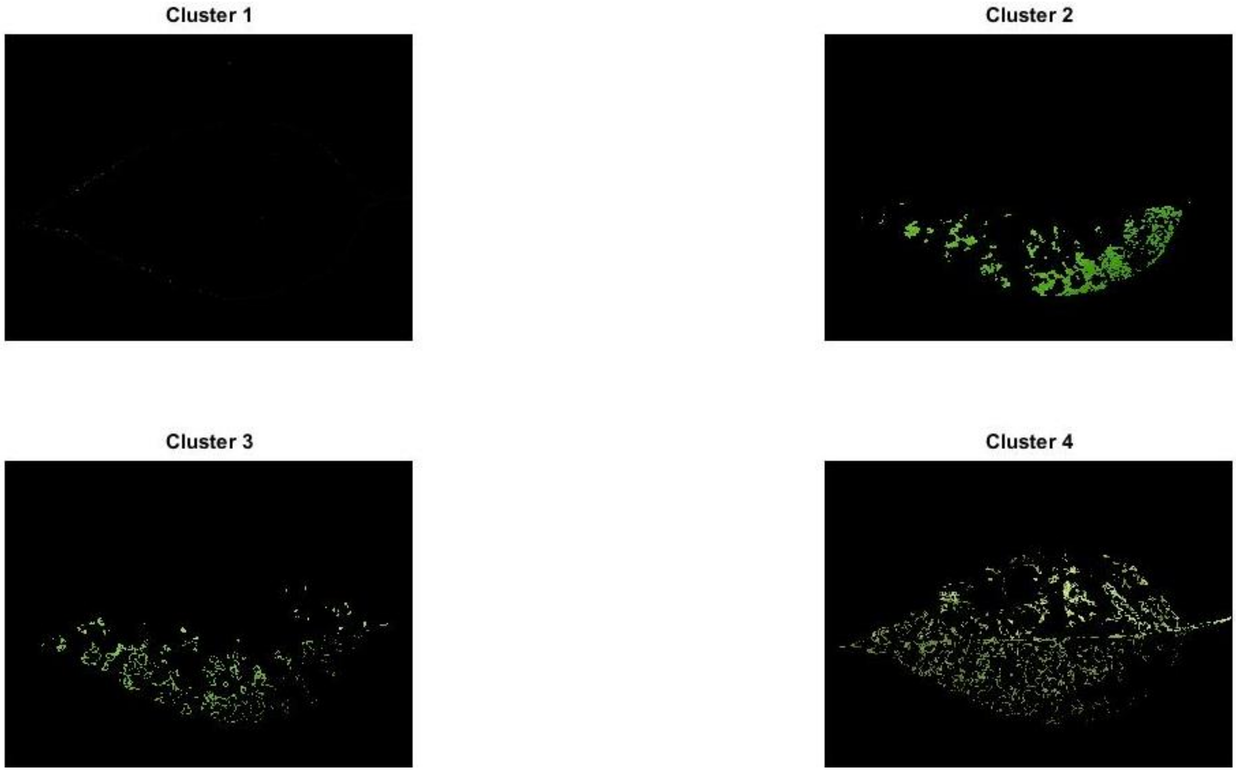


Figure 4-26: Internet leaf image Cluster 1-4

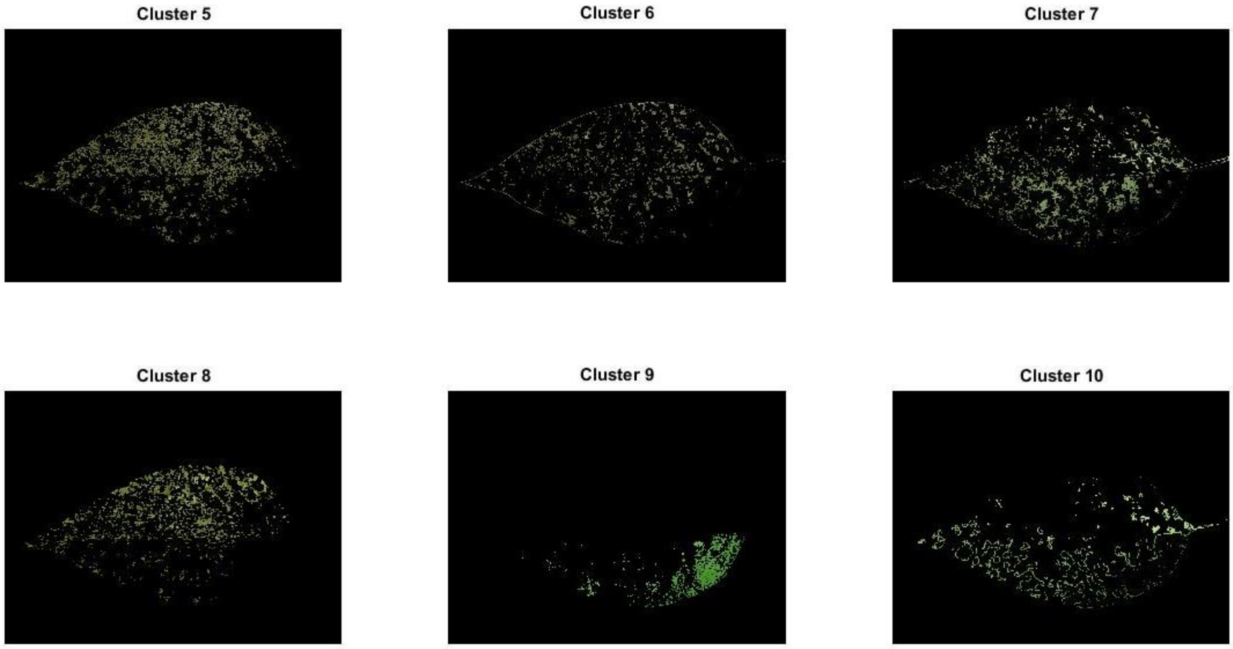


Figure 4-27: Internet leaf image Cluster 5-10

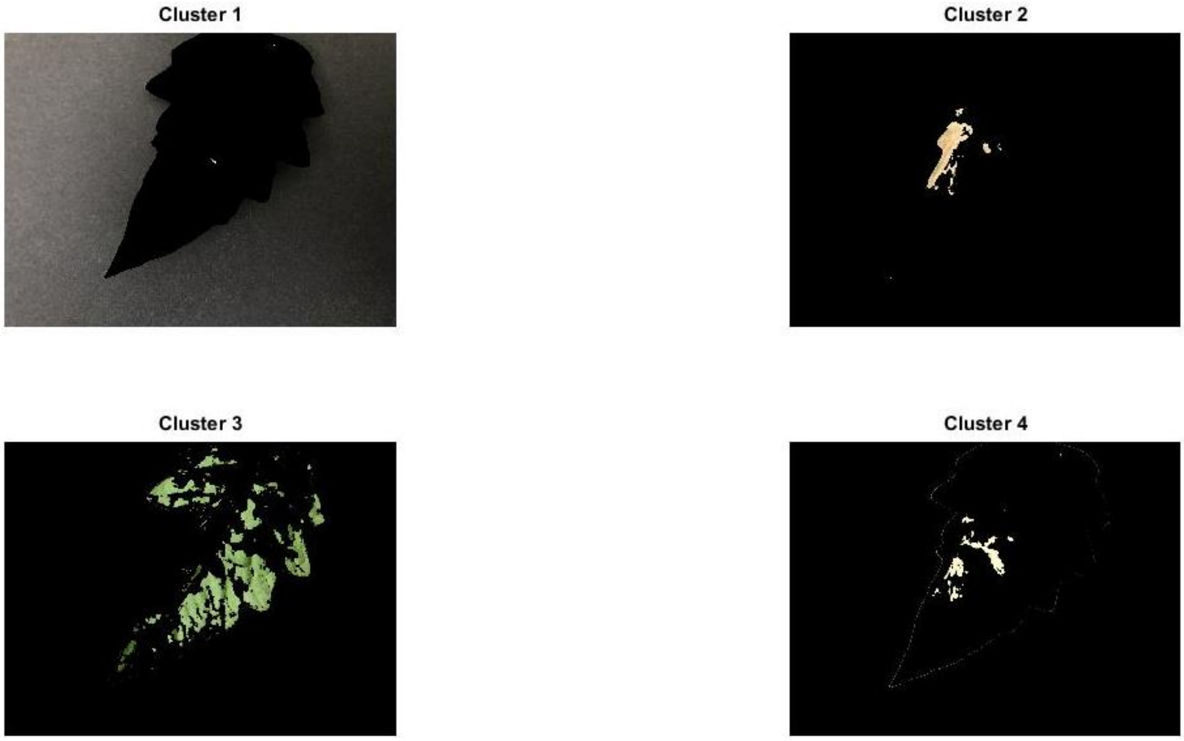


Figure 4-28: iPhone leaf image Cluster 1-4

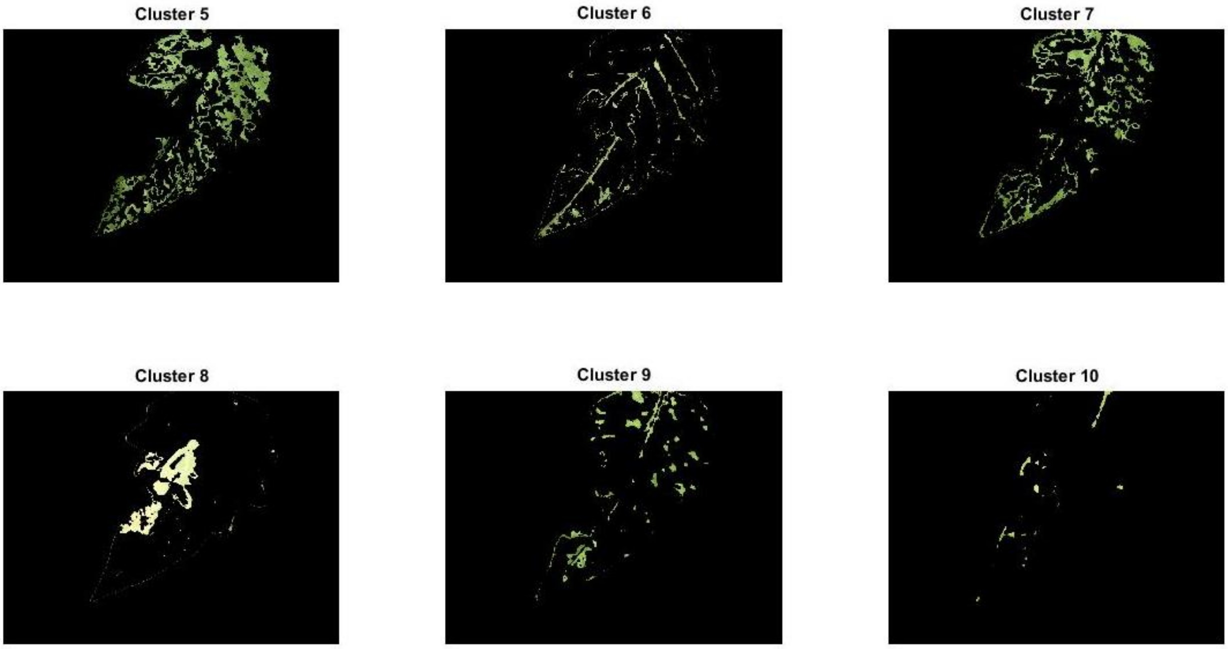


Figure 4-29: iPhone leaf image Cluster 5-10

In order to segment the leaf image from the background the cluster that contains the background must be subtracted from the original image to preserve the leaf area and remove the background. It is clear from Figure 4-26 and Figure 4-28 that cluster 1 includes the background region while the other clusters represent different areas of the leaf. Therefore, the image retrieved from cluster 1 will be deducted from the original leaf image so that the background is completely removed as shown in Figure 4-30 and Figure 4-31. Now that the background is completely removed and uniform, triangle thresholding is used to convert the image to binary as shown in Figure 4-32. The conversion shows much improvement over the previous techniques in which the leaf area was incorrectly represented from the background.

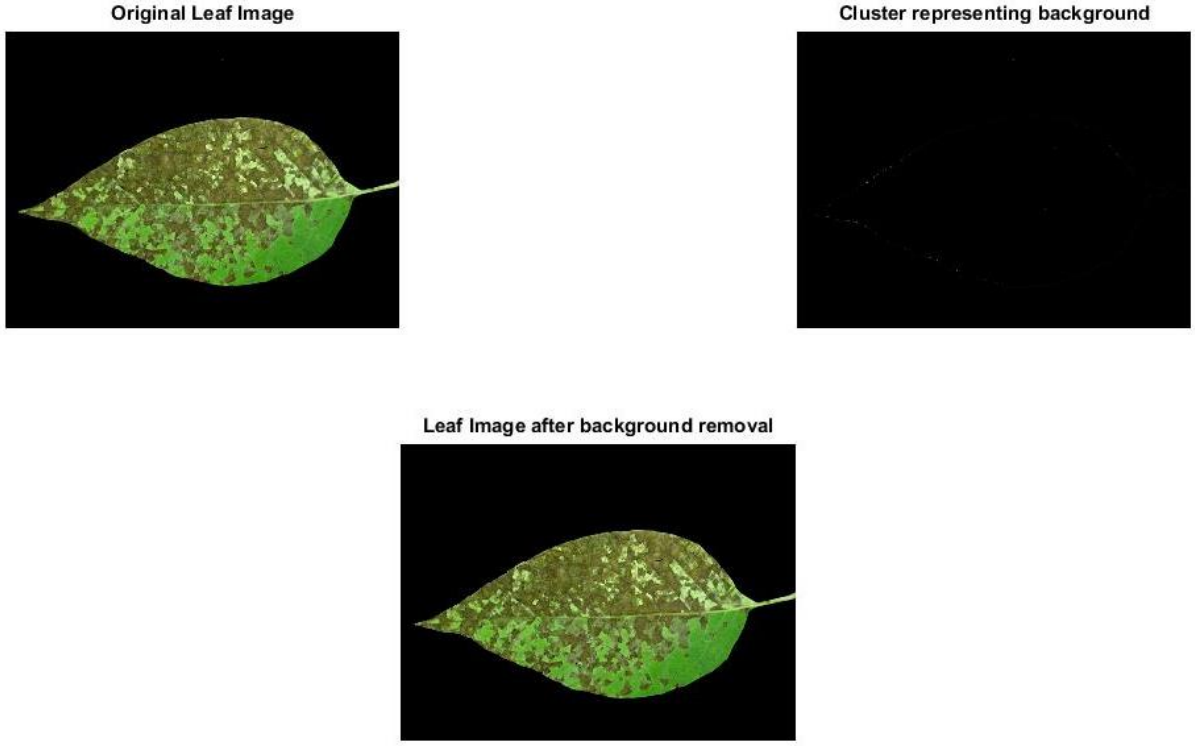

Original Leaf Image
Cluster representing background
Leaf Image after background removal

Figure 4-30: Internet leaf image after background removal

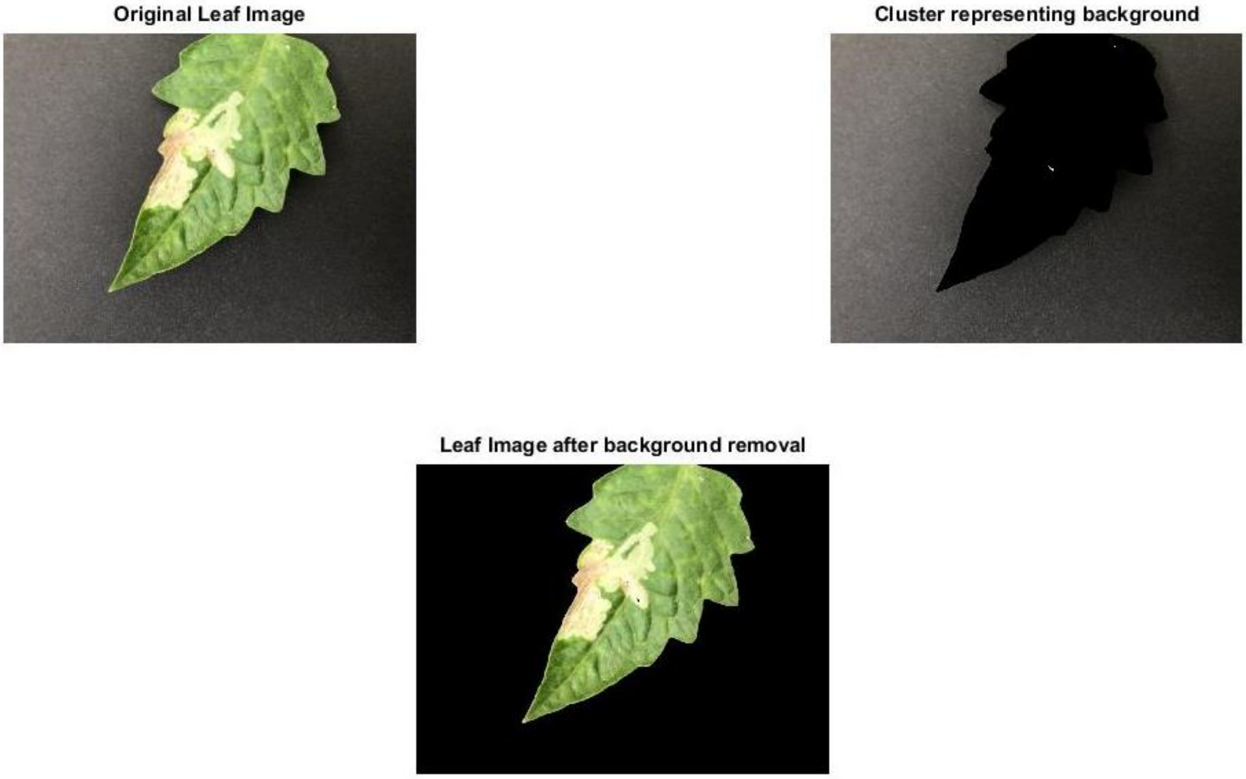

Original Leaf Image
Cluster representing background
Leaf Image after background removal

Figure 4-31: iPhone leaf image after background removal

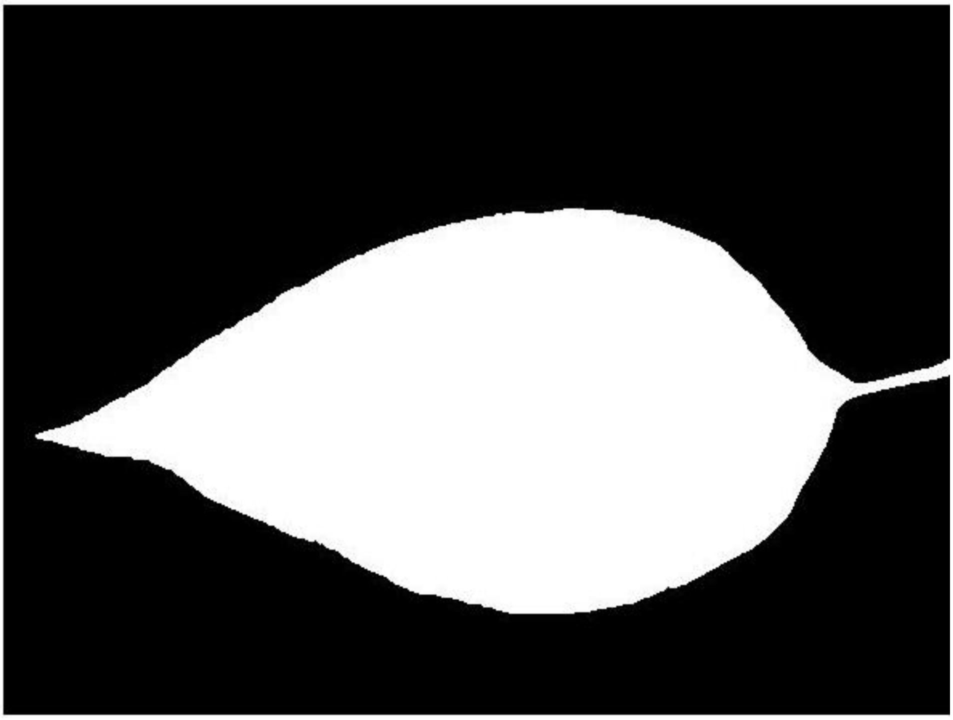


Figure 4-32: Internet leaf triangle threshold background segmentation with K-means clustering

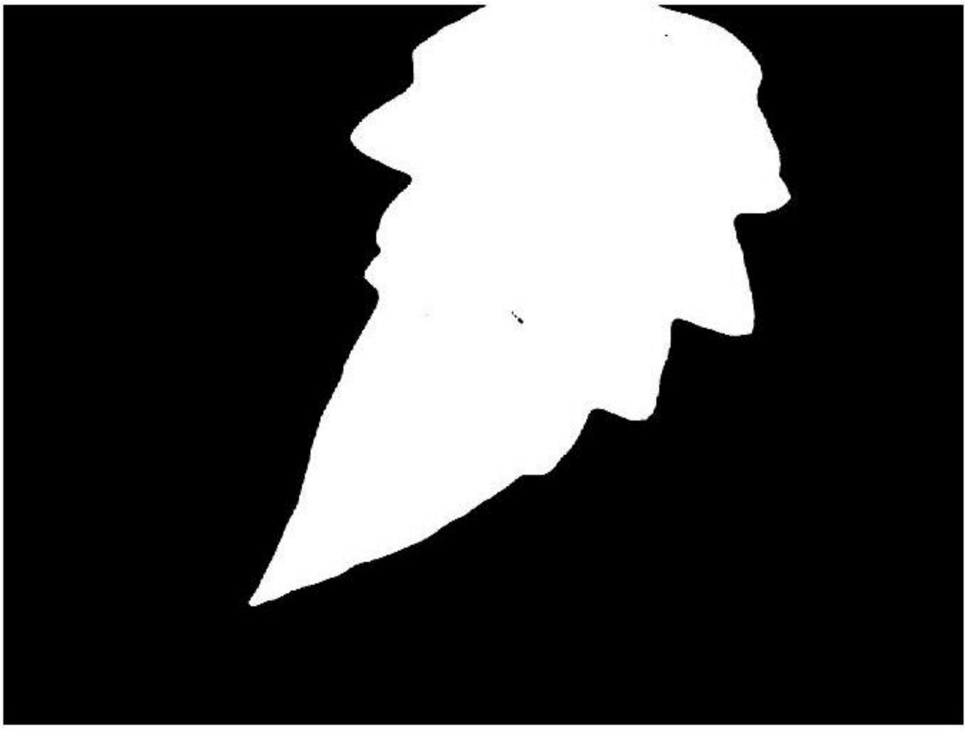


Figure 4-33: iPhone leaf triangle threshold background segmentation with K-means clustering

4.4. Disease Area Selection

In order to extract the diseased area, the cluster information from the k-means segmentation process is used. The cluster that visually best represents the diseased region will be selected as the diseased region. Internet leaf Cluster 5 shown in Figure 4-34 represents the diseased area most accurately in comparison to the other cluster segmentation results. As for the iPhone leaf image, cluster 8 represents the diseased as shown in Figure 4-35. As seen from Figure 4-35, the diseased region is clustered with other areas that do not represent the disease. The borders of the leaf image are segmented with the diseased area because of the similarity in color. Therefore, there is a need to remove unwanted lines and small miscellanies areas using morphological image processing methods.

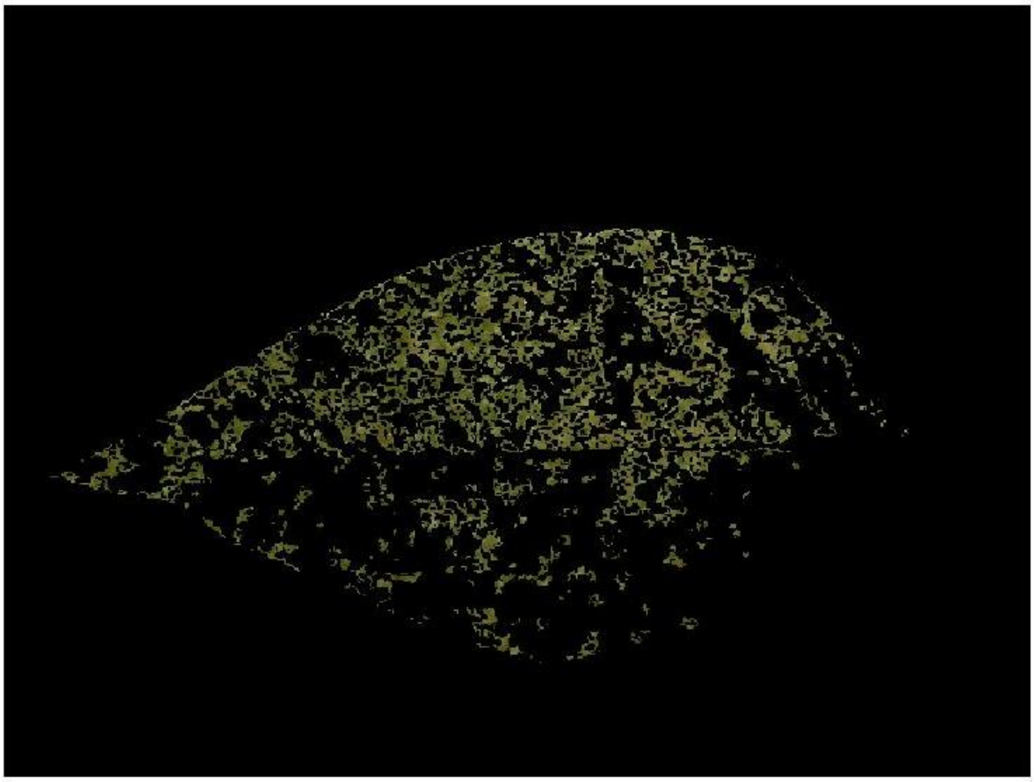


Figure 4-34: Internet leaf Cluster 5 containing disease

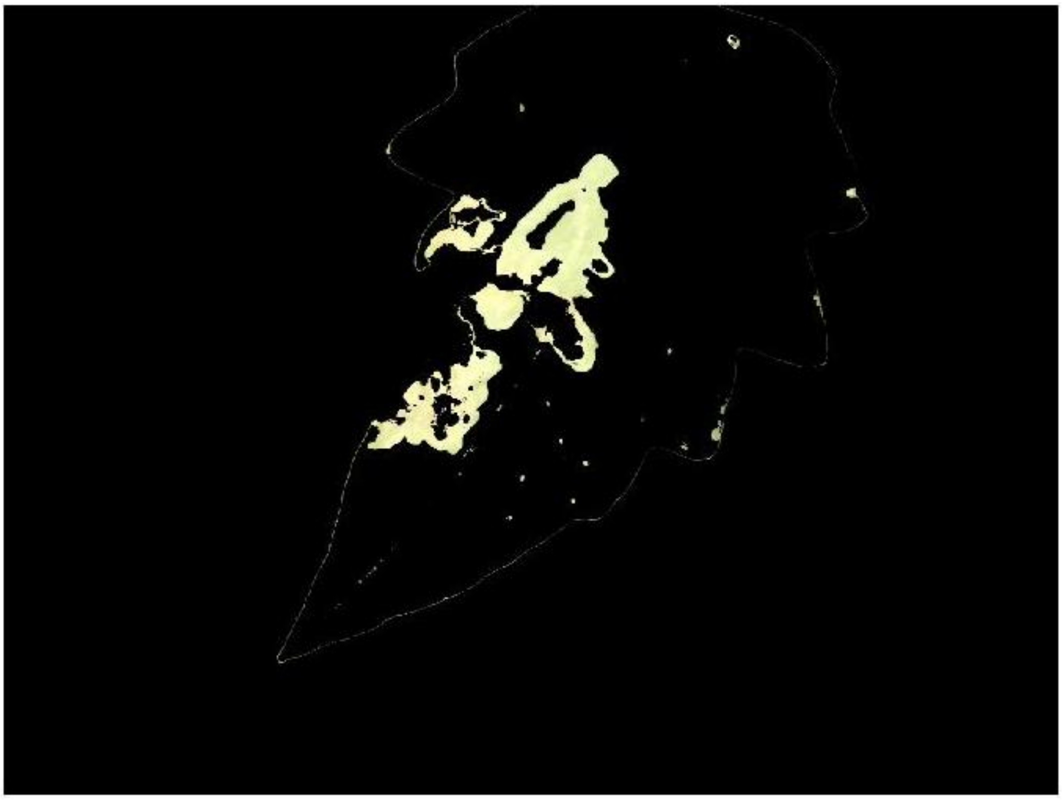


Figure 4-35: iPhone leaf Cluster 8 containing disease

4.4.1. Morphological Image Processing Techniques

To improve the accuracy of the binary conversion of the selected diseased cluster a few morphological image processing operations are applied on the diseased area binary image. Binary images may contain numerous imperfections, especially for regions produced by simple thresholding which can distorted by noise and texture. Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image. Figure 4-36 and Figure 4-37 represent the binary conversion of cluster 5 from the Internet leaf image and cluster 8 from the iPhone leaf image using triangle thresholding without any morphological image processing methods applied. It is clear especially in Figure 4-37, that the diseased area region is clustered with the border of the leaf image which leads to an inaccurate representation of the total diseased area.

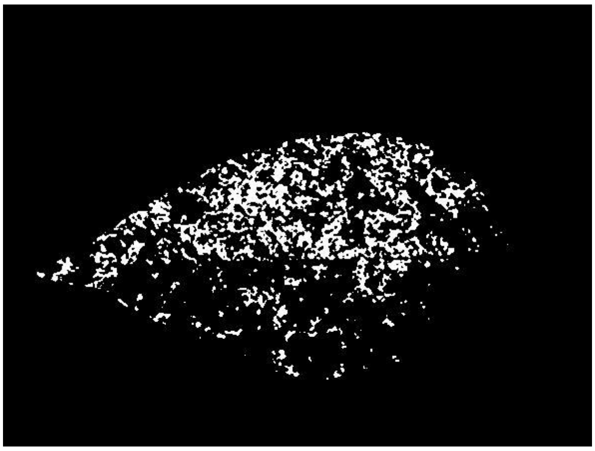


Figure 4-36: Binary conversion of internet leaf Cluster 5 using triangle thresholding

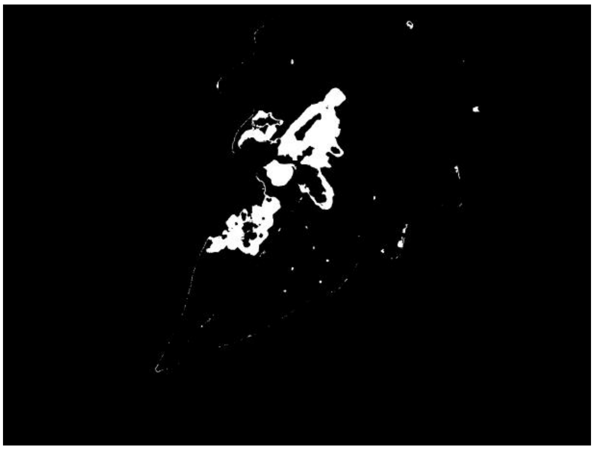


Figure 4-37: Binary conversion of iPhone leaf Cluster 8 using triangle thresholding

The first morphological method removes all connected components (objects) that have fewer than P pixels from the binary image. The value of P is chosen as the minimum area for an unwanted object in the image, anything below this area will be disregarded from the image. A structuring element of size P is passed over the image and objects smaller than the size of the structuring element is removed from the binary image. Figure 4-38 and Figure 4-39 show the output of the diseased area after removing smaller areas. Figure 4-39 represents an improved representation of the diseased area when compared to Figure 4-37.

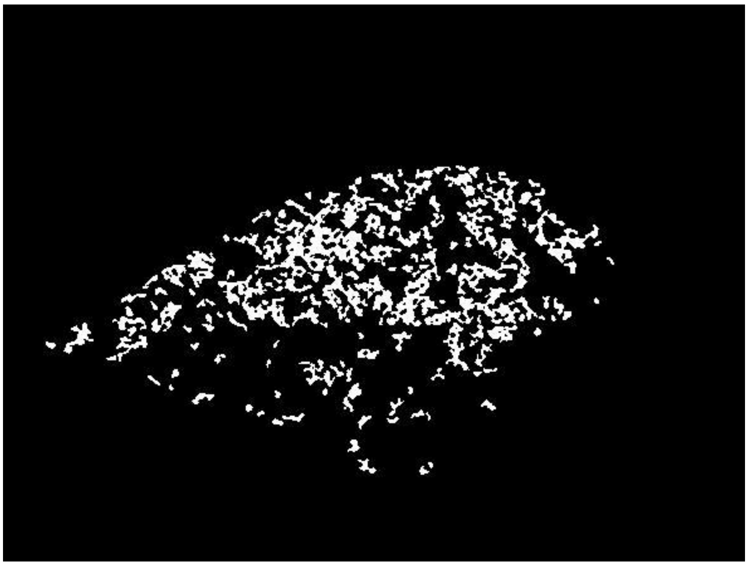


Figure 4-38: Binary conversion of internet leaf Cluster 5 after removing small areas

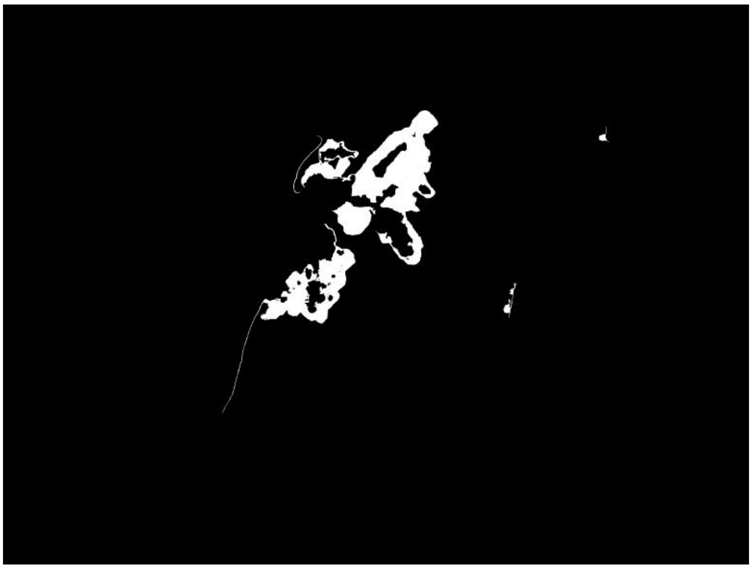


Figure 4-39: Binary conversion of iPhone leaf Cluster 8 after removing small areas

A circular flat morphological structuring element is then used to fill any gaps between small intersections. Figure 4-40 and Figure 4-41 illustrate how the diseased area is represented after removing small unwanted objects and after filling gaps to have a better representation of the overall leaf disease area.

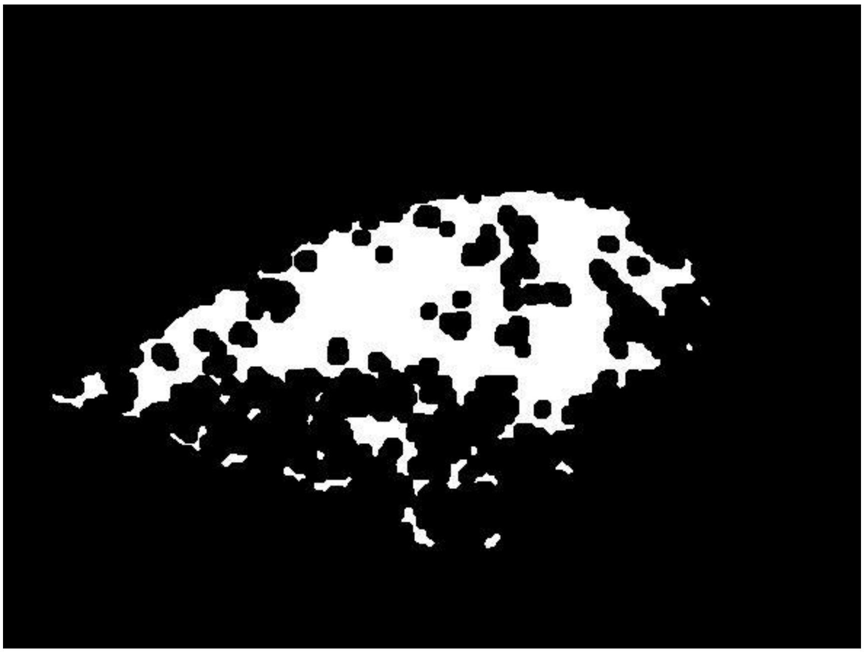


Figure 4-40: Binary conversion of internet leaf Cluster 5 after filling gaps

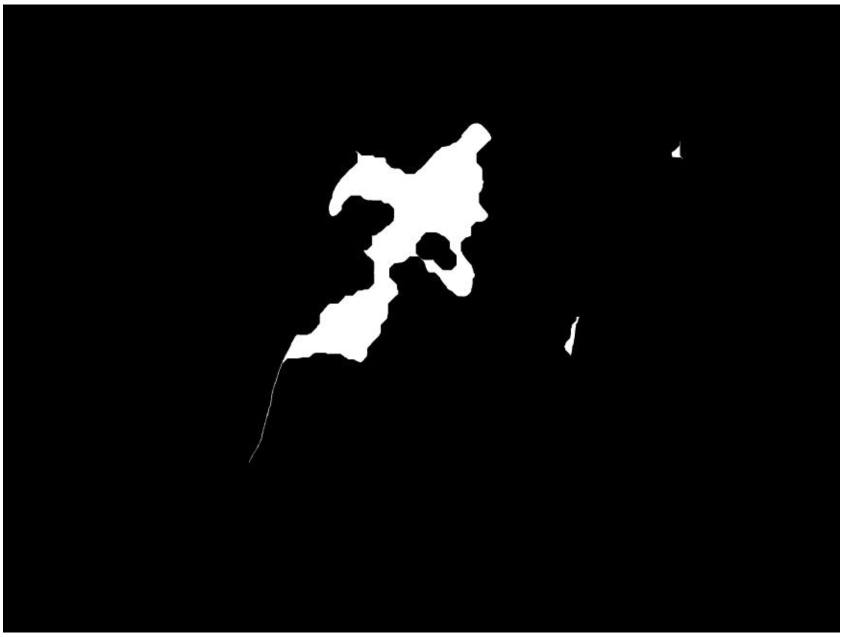


Figure 4-41: Binary conversion of iPhone leaf Cluster 8 after filling gaps

4.5. Percent Infection Calculation

To calculate the percent infection, the total leaf area (A_T) and diseased area (A_D) should be calculated appropriately to ensure high accuracy of disease severity calculation. The leaf area is clearly highlighted shown in Figure 4-32 and Figure 4-33 with white pixels, the area can be easily found as the summation of white pixels as the total leaf area A_T. The diseased region is shown in Figure 4-40 and Figure 4-41 and its area is area is also calculated as the summation of white pixels of the diseased area A_D.

After the image has been segmented into a healthy and diseased region the area of each can be easily calculated by counting the number of pixels in each region. The percent infection is calculated by applying Equation 4.7 in which A_D represents the diseased region and A_T represents the total leaf region [7].

$$Percent\ Infection = \left(\frac{A_D}{A_T}\right) * 100 \tag{4.7}$$

4.6. Fuzzy Logic Classification

A Fuzzy Logic system is implemented to characterize which disease grade the disease belongs to depending on the percent infection. It is difficult to assess which disease grade the disease belongs to because of the ambiguity and uncertainty of the table. Because the disease severity table represents a wide range of percent infection for every disease grade it may be problematic in accurately selecting the appropriate disease grade. Therefore, a Fuzzy Logic system can be effective in this case and a triangular membership function was implemented in this system to accurately depict the disease grade. For the disease grading Fuzzy Logic System the input variable is percent infection and the output variable is disease grade. The Triangular membership functions are used to define the variables and six fuzzy rules are set to grade the disease. The input variable was segmented into 6 triangular membership functions ranging from 0% to 100%, likewise the output variable was also segmented similarly from 0 to 5. Depending on the percentage of infection the disease grade will increase accordingly. The input and output triangular membership functions are shown in Figure 4-42 and Figure 4-43.

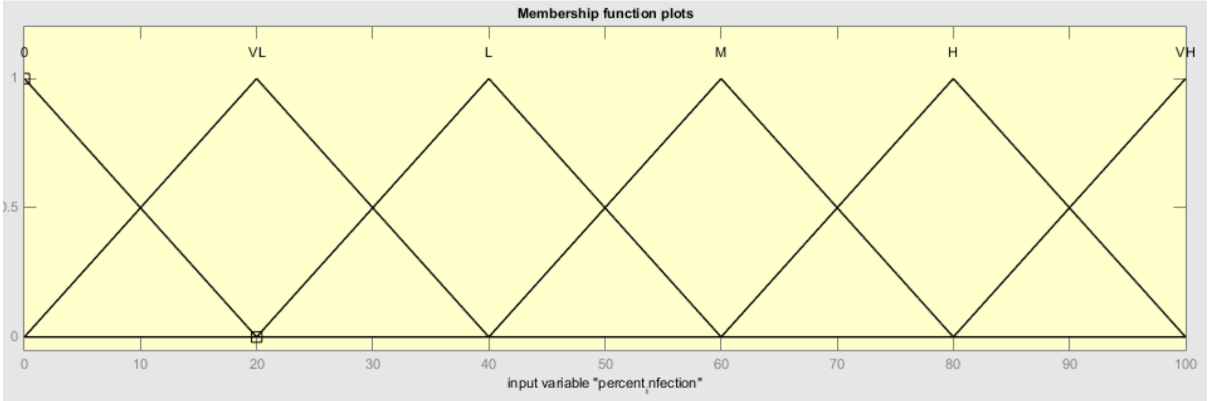


Figure 4-42: Input variable (Percent infection) triangular membership function

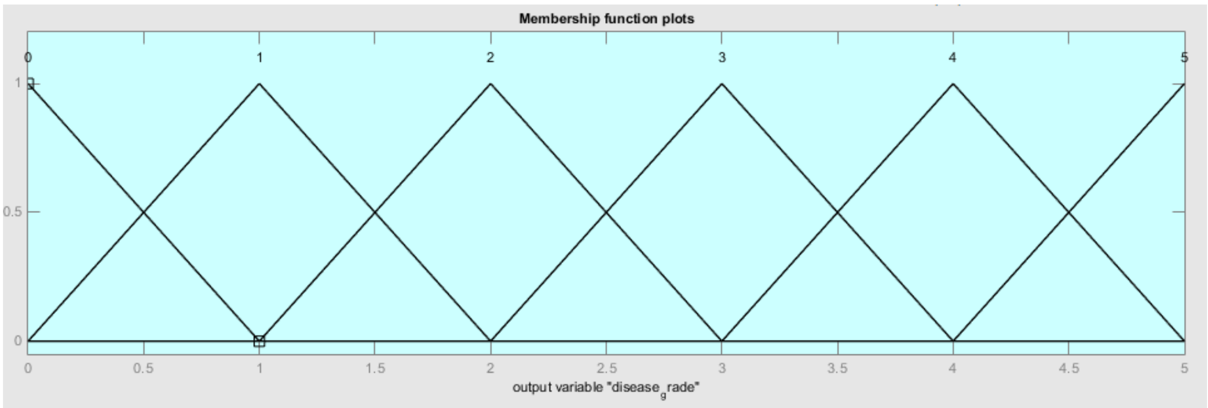


Figure 4-43: Output variable (Disease grade) triangular membership function

The percent infection input variable is defined as 0, Very Low (VL), Low (L), Moderate (M), High (H), and Very High (H), depending on the result of Equation 4.7. The logic if-then fuzzy rules that are set to select the appropriate disease grade are as Table 4-7.

Table 4-7: If-then Fuzzy rules

Percent infection	Disease Grade
0	0
VL	1
L	2
M	3
H	4
VH	5

4.7. Graphical User Interface (GUI)

A Graphical User Interface (GUI) is designed and implemented to help users monitor leaf disease severity and overall plant health. The core functions of the GUI displayed in Figure 4-44 are featured by:

a) Two large axes to clearly display the leaf image before (left) and after (right) processing.

b) Clear button placements to indicate the process flow, beginning with importing a leaf image to calculating the disease grade and saving all intermediate image outputs.

c) Drop down menus to help users choose between different presets for resolution and selecting among different clusters.

It has a simple layout with clear areas for images and large button placements. It can be used by users such as laboratory technicians to calculate the disease severity of any leaf sample. Each step of the process will be described in this section.

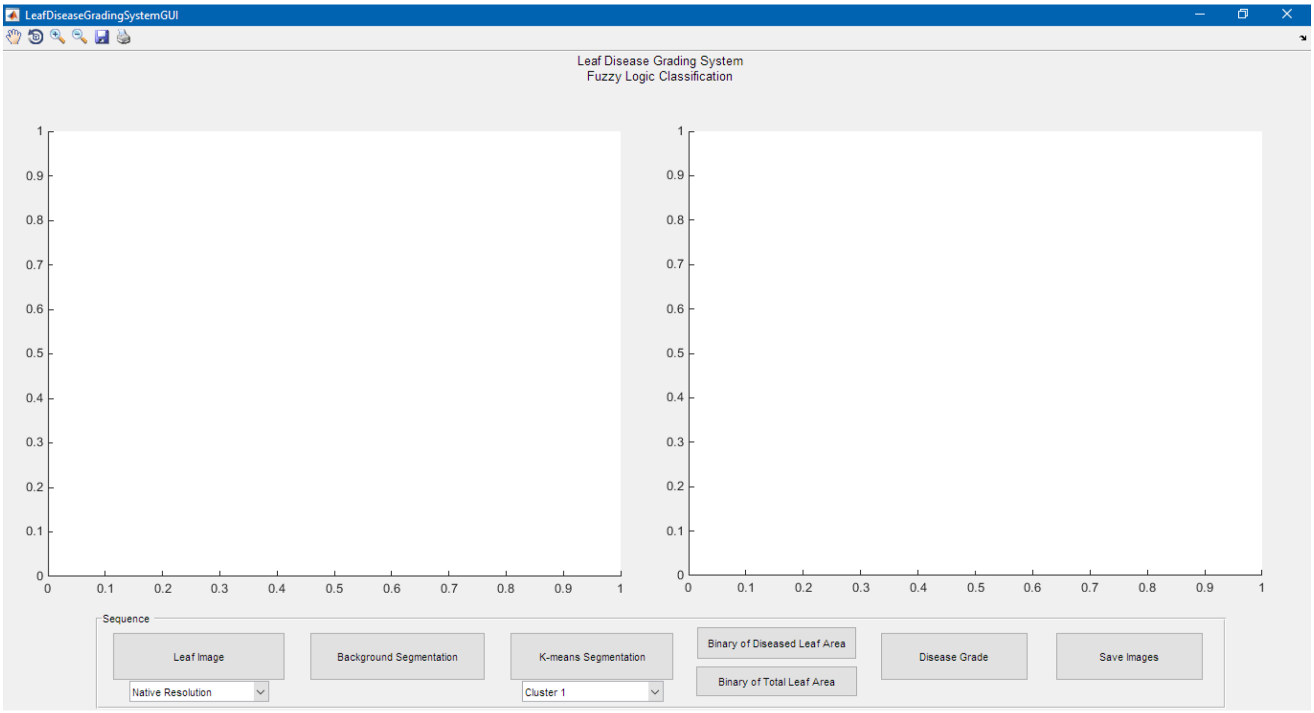


Figure 4-44: Graphical user interface

1. Selecting Leaf Image

The user has the option to insert a sample of a leaf image by pressing on the Leaf Image button. Once selected a new window appears as shown in Figure 4-45, in which the user can browse local folders to choose a leaf image. The folder can be updated by the user with new leaf images at any given time. After selecting the leaf image, it is displayed on the left axes as shown in Figure 4-46.

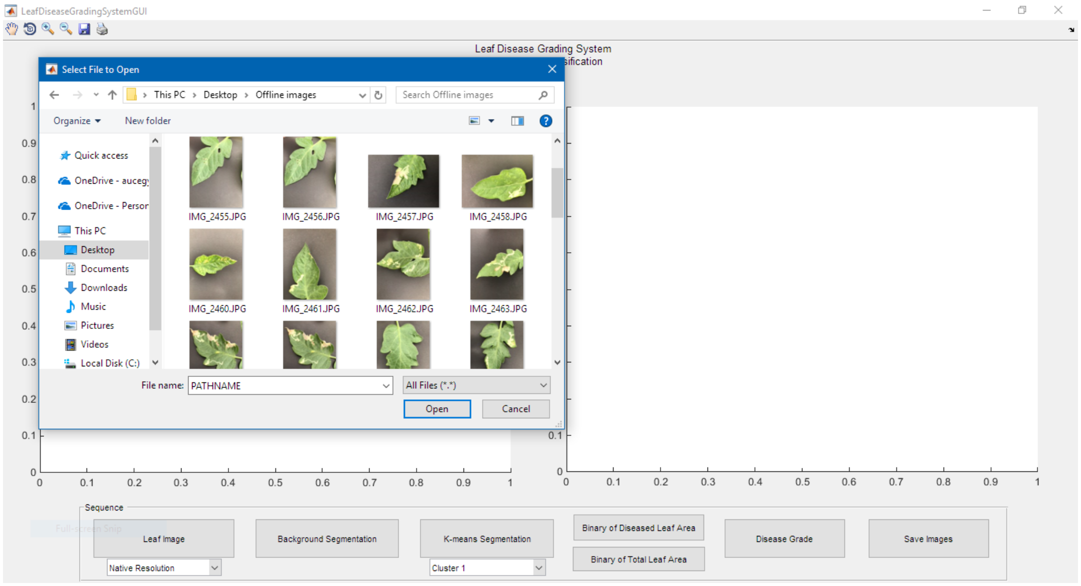


Figure 4-45: Selecting a leaf image

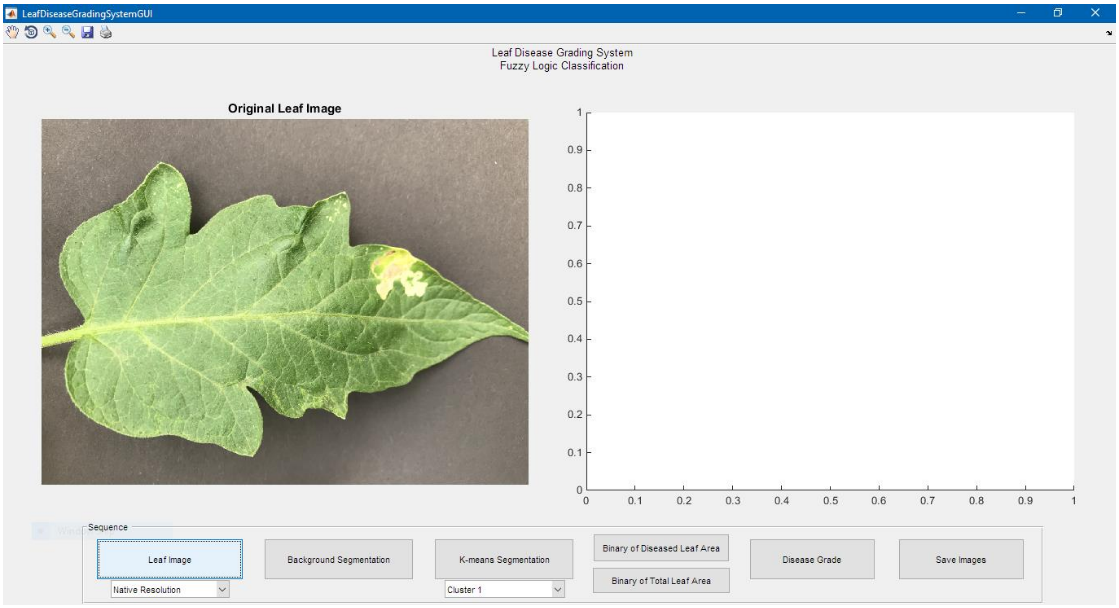


Figure 4-46: Leaf image selected

2. Resolution Adjustment

The resolution of the image can be changed by the drop-down menu available under the Leaf Image button as seen in Figure 4-47 and Figure 4-48. The two options are Native Resolution in which the resolution of the input image is kept as is, and the other is Quarter Resolution where the input image is resized to achieve a quarter of the original resolution.

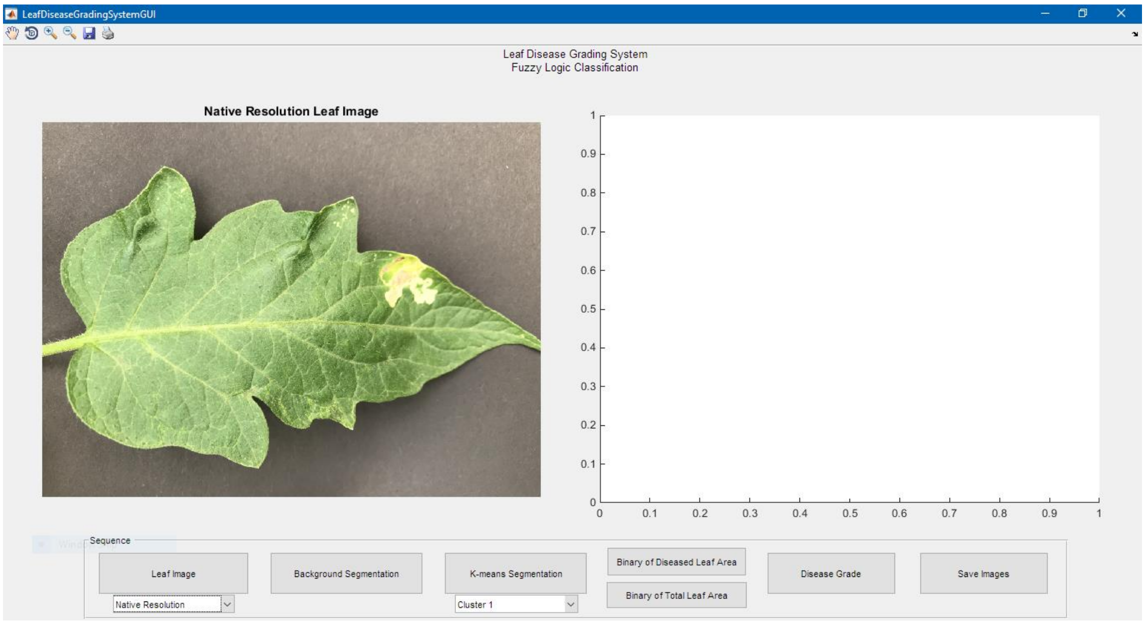


Figure 4-47: Leaf Image at Native Resolution

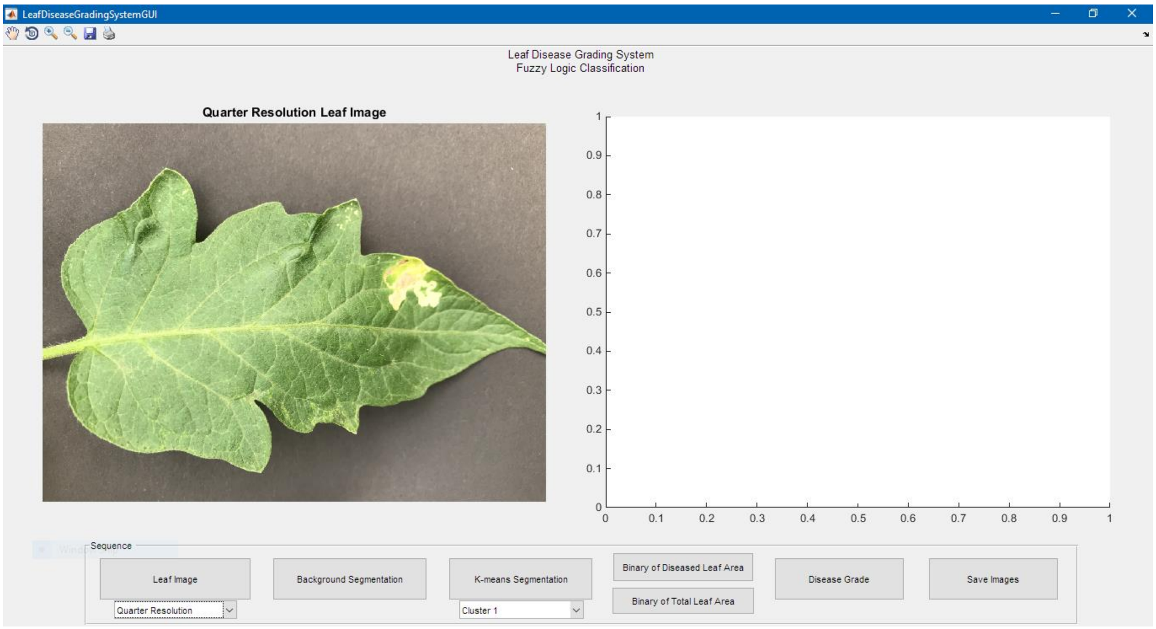


Figure 4-48: Leaf Image at Quarter Resolution

3. Background Removal

By pressing on the Background Segmentation button, the background of the leaf image is removed. Background removal is achieved by using K-means technique with Triangle thresholding. The output of this process is shown on the right axes as seen in Figure 4-49.

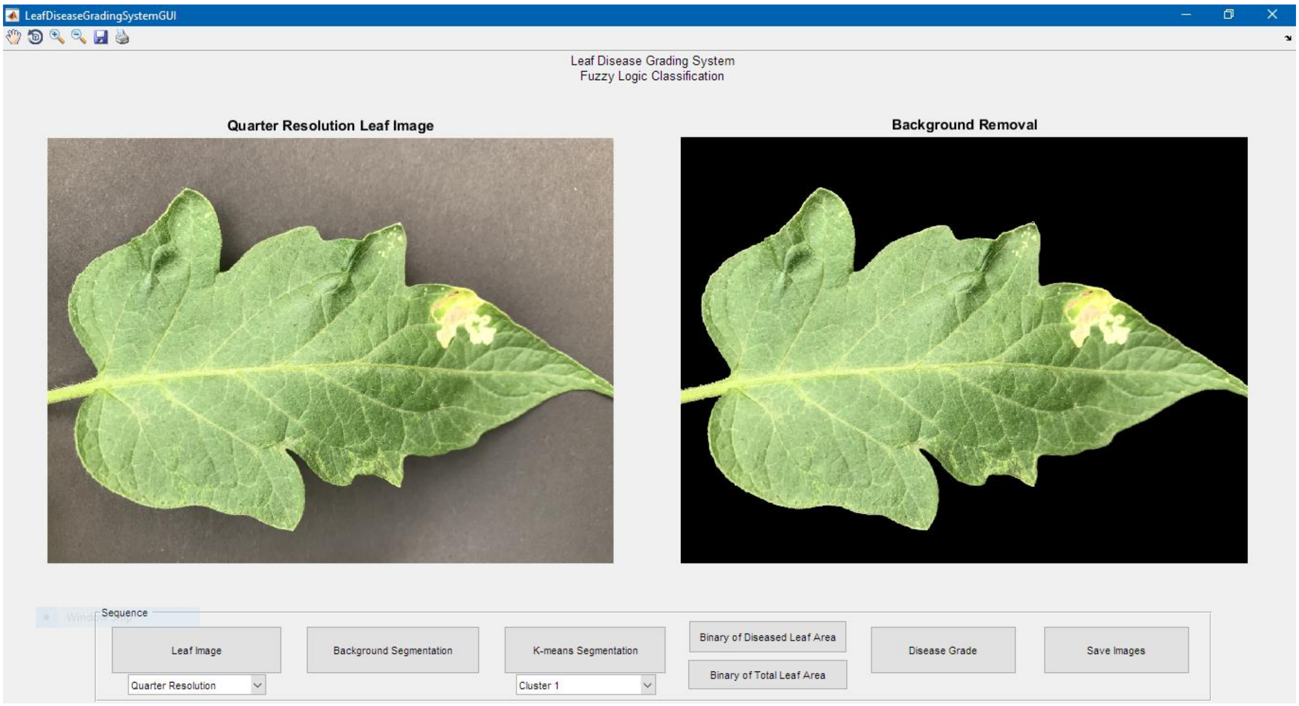

LeafDiseaseGradingSystemGUI
Leaf Disease Grading System
Fuzzy Logic Classification
Quarter Resolution Leaf Image
Background Removal
Sequence
Leaf Image
Quarter Resolution
Background Segmentation
K-means Segmentation
Cluster 1
Binary of Diseased Leaf Area
Binary of Total Leaf Area
Disease Grade
Save Images

Figure 4-49: Leaf image after background removal

4. Selecting Disease Cluster

By pressing on the K means Segmentation button the 1st cluster is shown on the right axes as see in Figure 4-50. The disease cluster can be selected by toggling through the Cluster drop down menu as shown in and Figure 4-51.

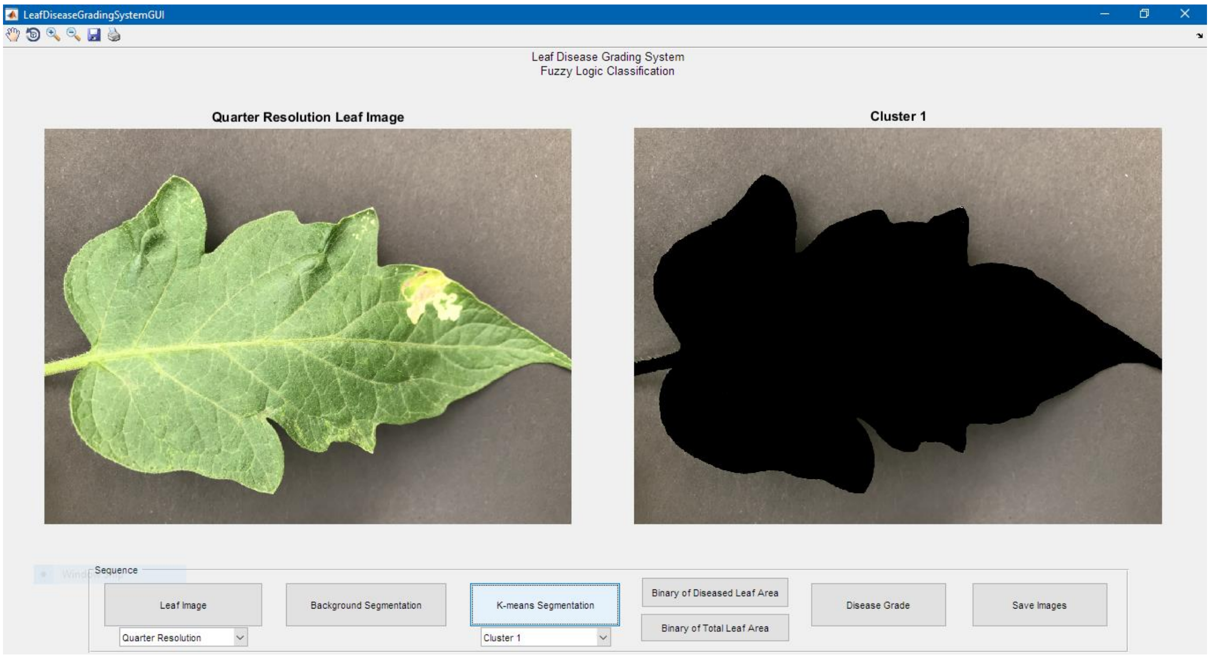

LeafDiseaseGradingSystemGUI
Leaf Disease Grading System
Fuzzy Logic Classification
Quarter Resolution Leaf Image
Cluster 1
Sequence
Leaf Image
Quarter Resolution
Background Segmentation
K-means Segmentation
Cluster 1
Binary of Diseased Leaf Area
Binary of Total Leaf Area
Disease Grade
Save Images

Figure 4-50: Cluster 1

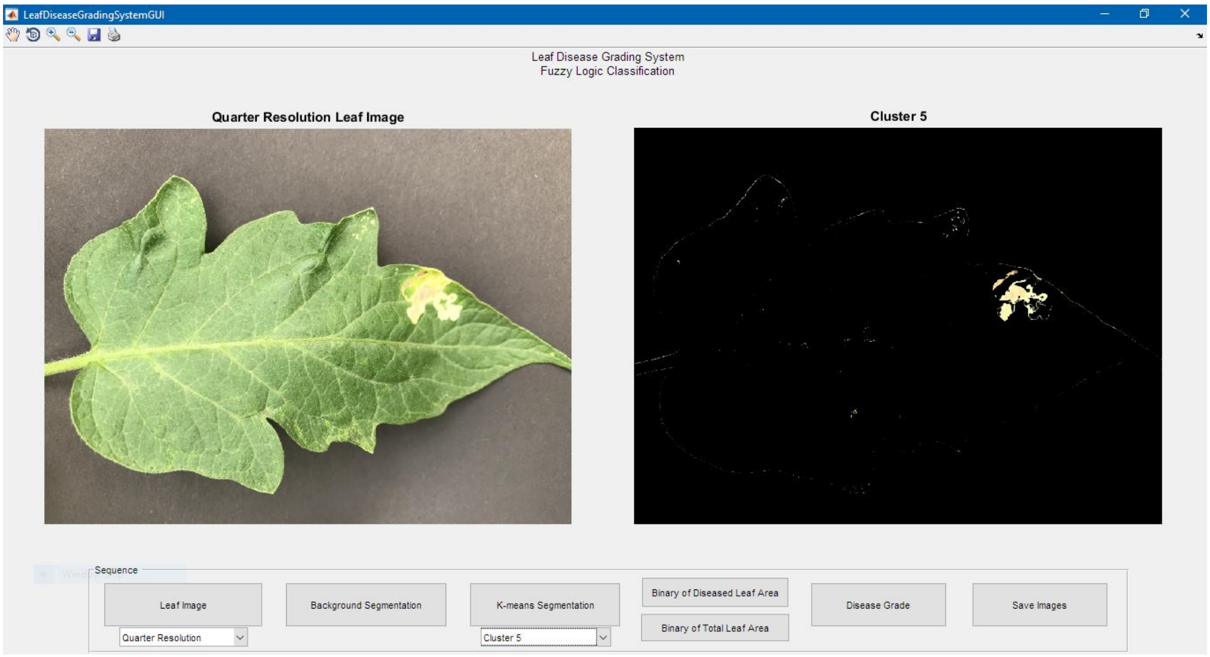

LeafDiseaseGradingSystemGUI
Leaf Disease Grading System
Fuzzy Logic Classification
Quarter Resolution Leaf Image
Cluster 5
Sequence
Leaf Image
Quarter Resolution
Background Segmentation
K-means Segmentation
Cluster 5
Binary of Diseased Leaf Area
Binary of Total Leaf Area
Disease Grade
Save Images

Figure 4-51: Selecting the cluster containing the disease

5. Leaf Area and Diseased Area

After the cluster with the disease is selected the binary of the total leaf area (A_T) and the diseased clustered area (A_D) is calculated by pressing on the Binary of Total Leaf Area and the Binary of Diseased Clustered Area button as shown in Figure 4-52 and Figure 4-53.

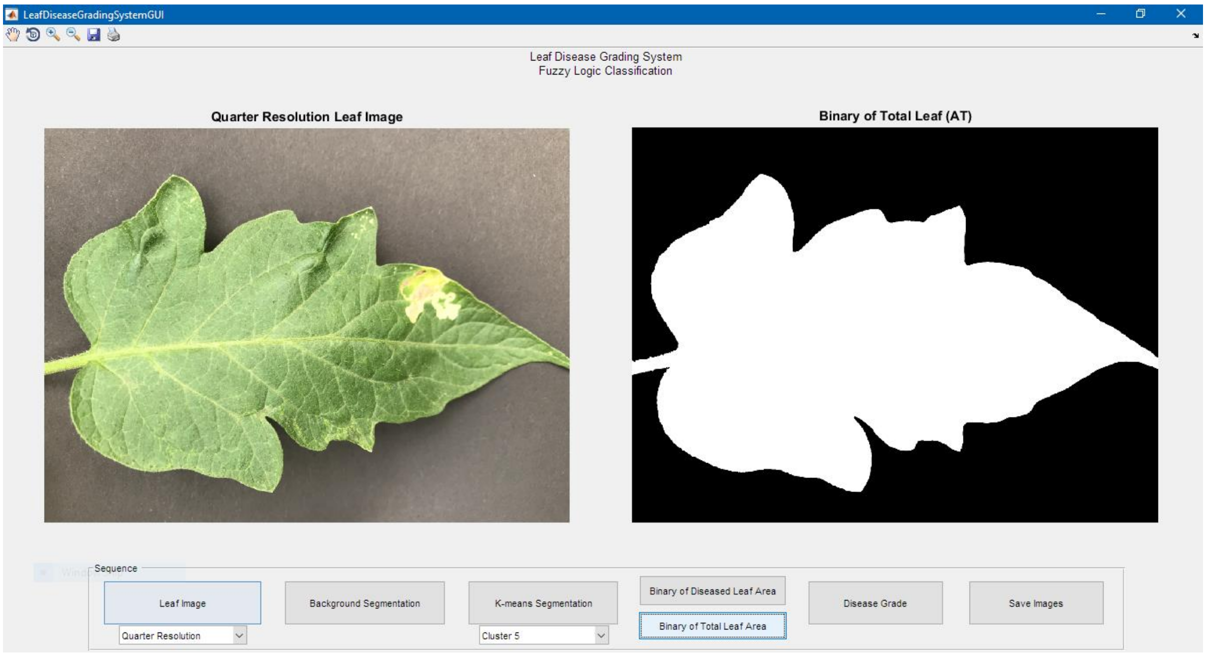

LeafDiseaseGradingSystemGUI
Leaf Disease Grading System
Fuzzy Logic Classification
Quarter Resolution Leaf Image
Binary of Total Leaf (AT)
Sequence
Leaf Image
Quarter Resolution
Background Segmentation
K-means Segmentation
Cluster 5
Binary of Diseased Leaf Area
Binary of Total Leaf Area
Disease Grade
Save Images

Figure 4-52: Binary of Total Leaf Area (A_T)

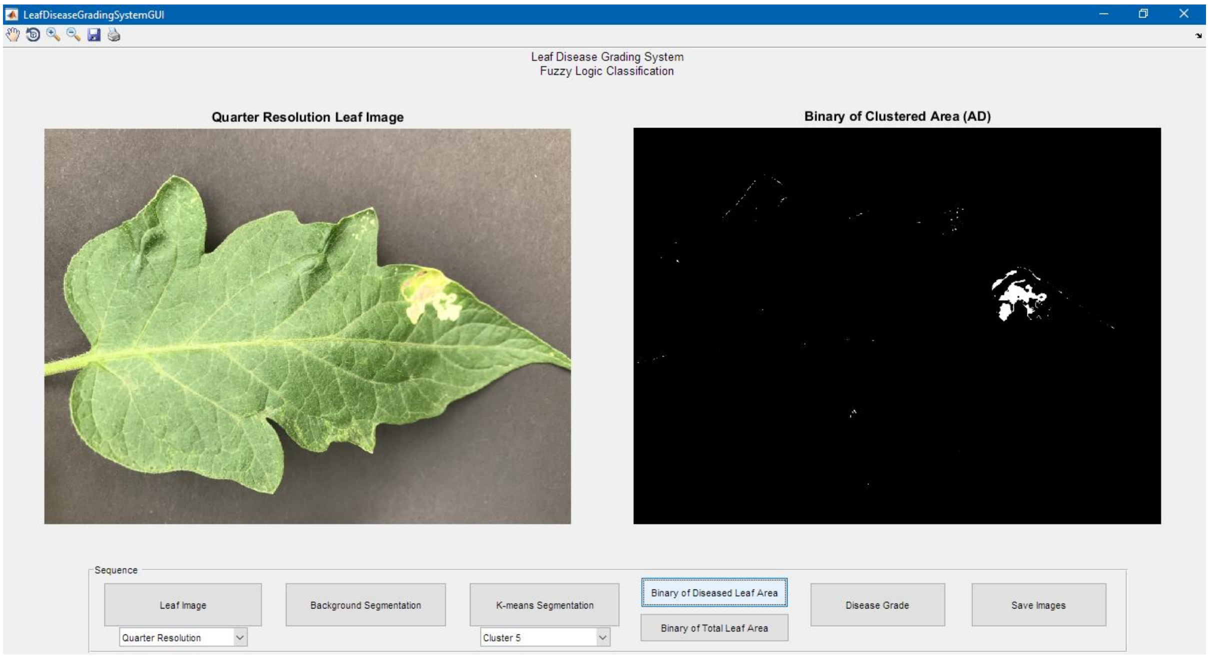
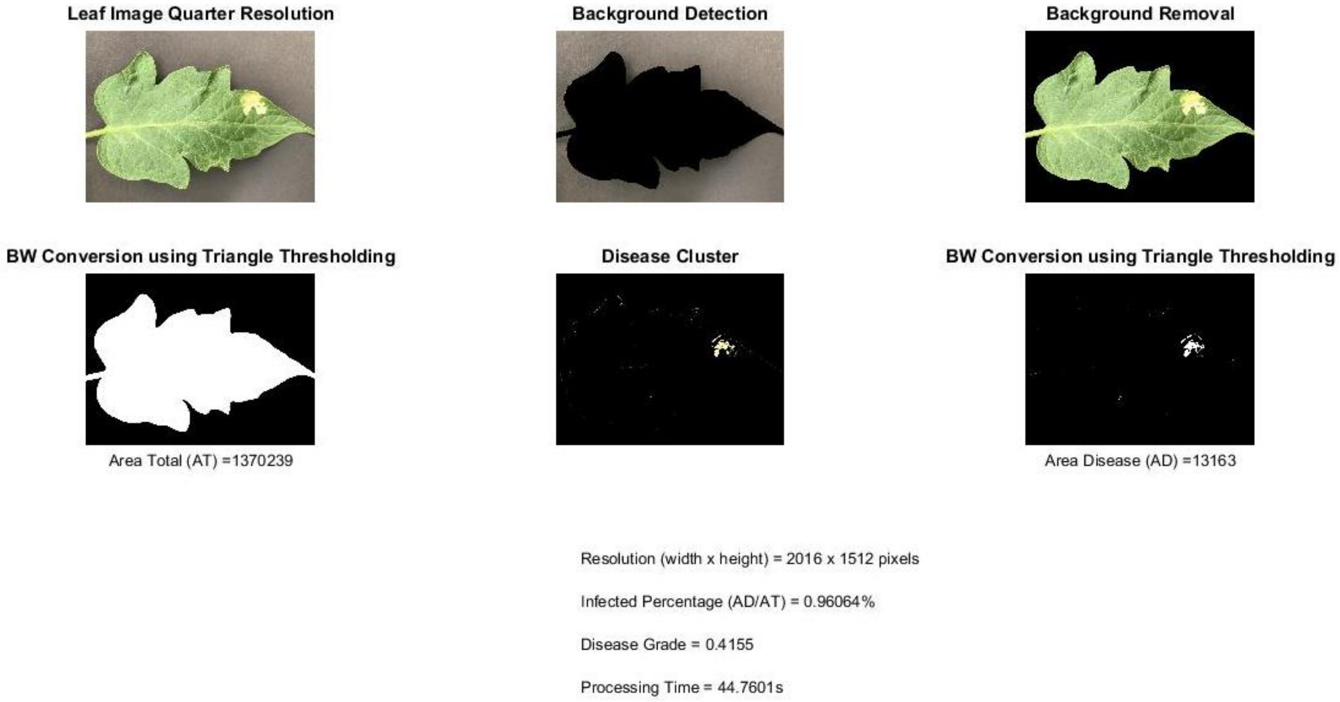

LeafDiseaseGradingSystemGUI
Leaf Disease Grading System
Fuzzy Logic Classification
Quarter Resolution Leaf Image
Binary of Clustered Area (AD)
Sequence
Leaf Image
Quarter Resolution
Background Segmentation
K-means Segmentation
Cluster 5
Binary of Diseased Leaf Area
Binary of Total Leaf Area
Disease Grade
Save Images

Figure 4-53: Binary of Diseased Area (A_D)

6. Save Results

By pressing on the Save Images button, a figure is displayed showing the original leaf image and the intermediate results such as the background removal and the disease cluster selection. Also, further information is shown under the images such as the disease grade and processing time.


Leaf Image Quarter Resolution
Background Detection
Background Removal
BW Conversion using Triangle Thresholding
Disease Cluster
BW Conversion using Triangle Thresholding

```
Area Total (AT) =1370239
Area Disease (AD) =13163

Resolution (width x height) = 2016 x 1512 pixels
Infected Percentage (AD/AT) = 0.96064%
Disease Grade = 0.4155
Processing Time = 44.7601s
```

Figure 4-54: Results of Leaf Image

4.8. Testing and Results using GUI

In order to test the system, multiple images of leaf samples were taken from a real field using an iPhone 7 camera and some were retrieved from the Internet as shown in Table 4-8. Each image was put through the system and processed to calculate the disease severity taking into consideration its different resolution sizes and its effect on the disease grade calculation as well as the processing time. This test is completed on a laptop with the specifications illustrated in Table 4-9. The processing times achieved for this system heavily rely on the computers processor speed and performance.

Table 4-8: Leaf Test Images

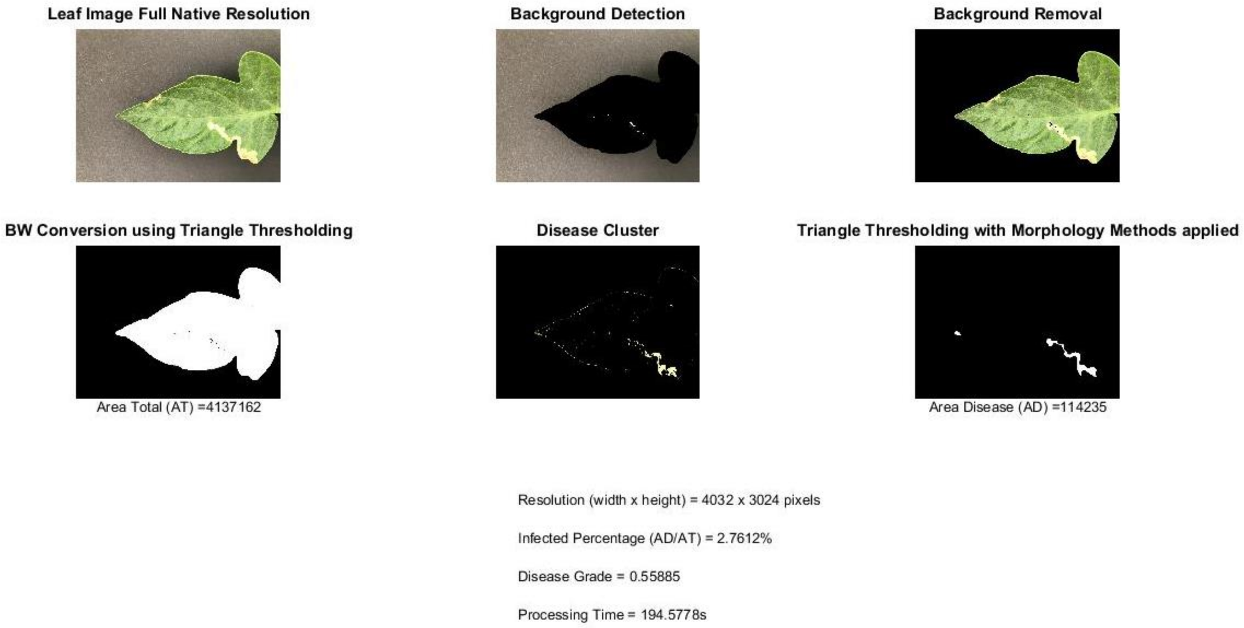
Leaf Image (JPEG)	Captured/Retrieved (Source)	Full Resolution (Pixels)	Reduced Resolution (Pixels)
1	iPhone	4032 x 3024	2016 x 1512
2	iPhone	4032 x 3024	2016 x 1512
3	iPhone	4032 x 3024	2016 x 1512
4	iPhone	4032 x 3024	2016 x 1512
5	iPhone	4032 x 3024	2016 x 1512
6	iPhone	4032 x 3024	2016 x 1512
7	iPhone	4032 x 3024	2016 x 1512
8	iPhone	4032 x 3024	2016 x 1512
9	iPhone	4032 x 3024	2016 x 1512
10	iPhone	4032 x 3024	2016 x 1512
11	iPhone	4032 x 3024	2016 x 1512
12	iPhone	4032 x 3024	2016 x 1512
13	iPhone	4032 x 3024	2016 x 1512
14	iPhone	4032 x 3024	2016 x 1512
15	iPhone	4032 x 3024	2016 x 1512
16	iPhone	4032 x 3024	2016 x 1512
17	iPhone	4032 x 3024	2016 x 1512
18	iPhone	4032 x 3024	2016 x 1512
19	Internet	590 x 443	295 x 222
20	Internet	850 x 565	425 x 283
21	Internet	1548 x 1031	774 x 516

Table 4-9: Laptop specifications

Processor	Intel Core i5 4200U Processor (2.6 GHz)
Operating System	Windows10
RAM	4 GB

4.8.1. iPhone Leaf Image Results (Images 1 – 18)

Each iPhone image was tested at its native full resolution and then again at quarter resolution as shown in Figure 4-55 and Figure 4-56. The overall processing time and disease grade of all tested images were recorded and compared as shown in Figure 4-57 to Figure 4-60.


Leaf Image Full Native Resolution
Background Detection
Background Removal
BW Conversion using Triangle Thresholding
Disease Cluster
Triangle Thresholding with Morphology Methods applied
Area Total (AT) =4137162
Area Disease (AD) =114235

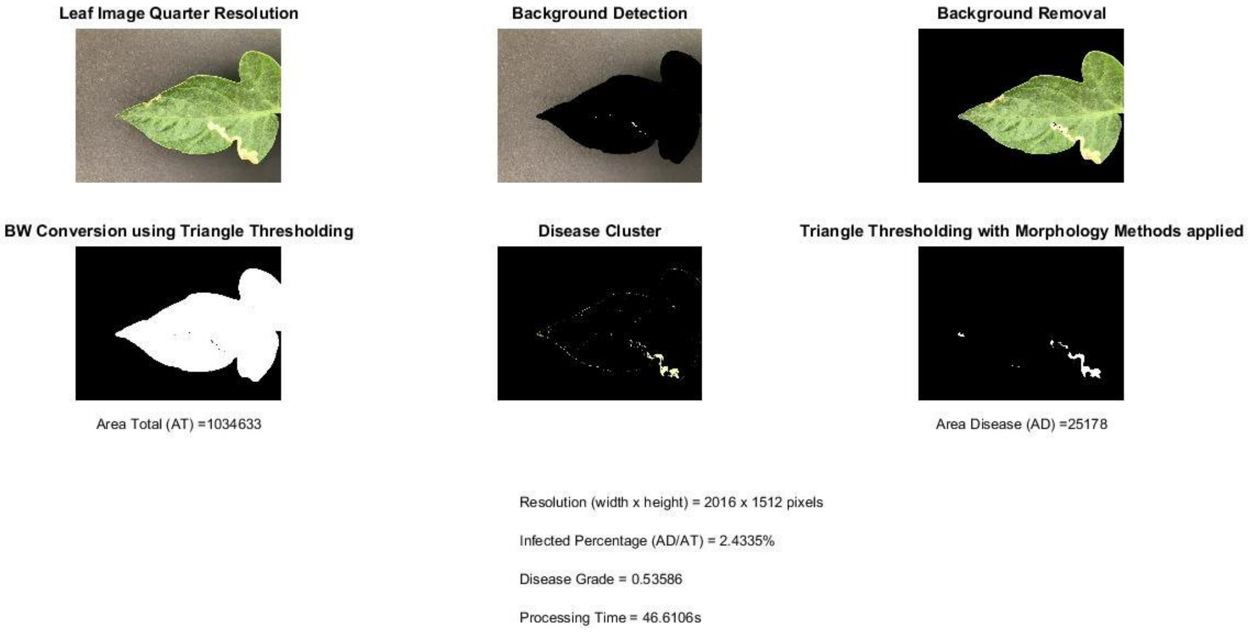
```
Resolution (width x height) = 4032 x 3024 pixels

Infected Percentage (AD/AT) = 2.7612%

Disease Grade = 0.55885

Processing Time = 194.5778s
```

Figure 4-55: Leaf image 4 results at full resolution


Leaf Image Quarter Resolution
Background Detection
Background Removal
BW Conversion using Triangle Thresholding
Disease Cluster
Triangle Thresholding with Morphology Methods applied
Area Total (AT) =1034633
Area Disease (AD) =25178

```
Resolution (width x height) = 2016 x 1512 pixels

Infected Percentage (AD/AT) = 2.4335%

Disease Grade = 0.53586

Processing Time = 46.6106s
```

Figure 4-56: Leaf image 4 results at quarter resolution

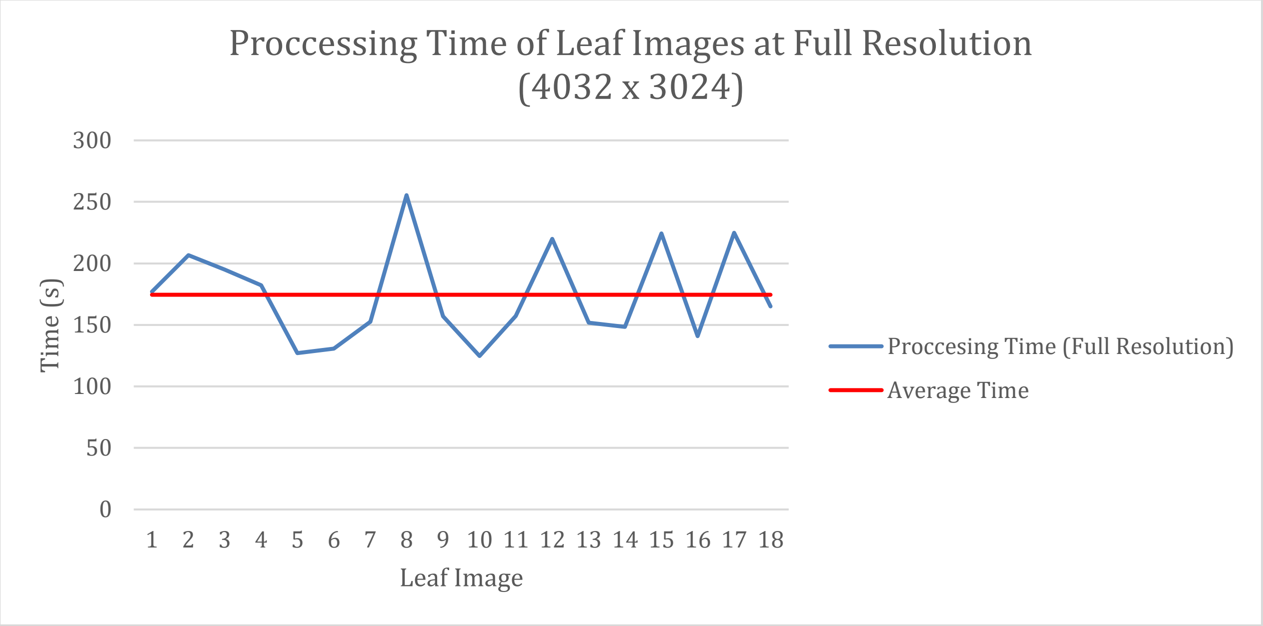


Figure 4-57: Processing Time of Leaf Images at Full Resolution (4032 x 3024)

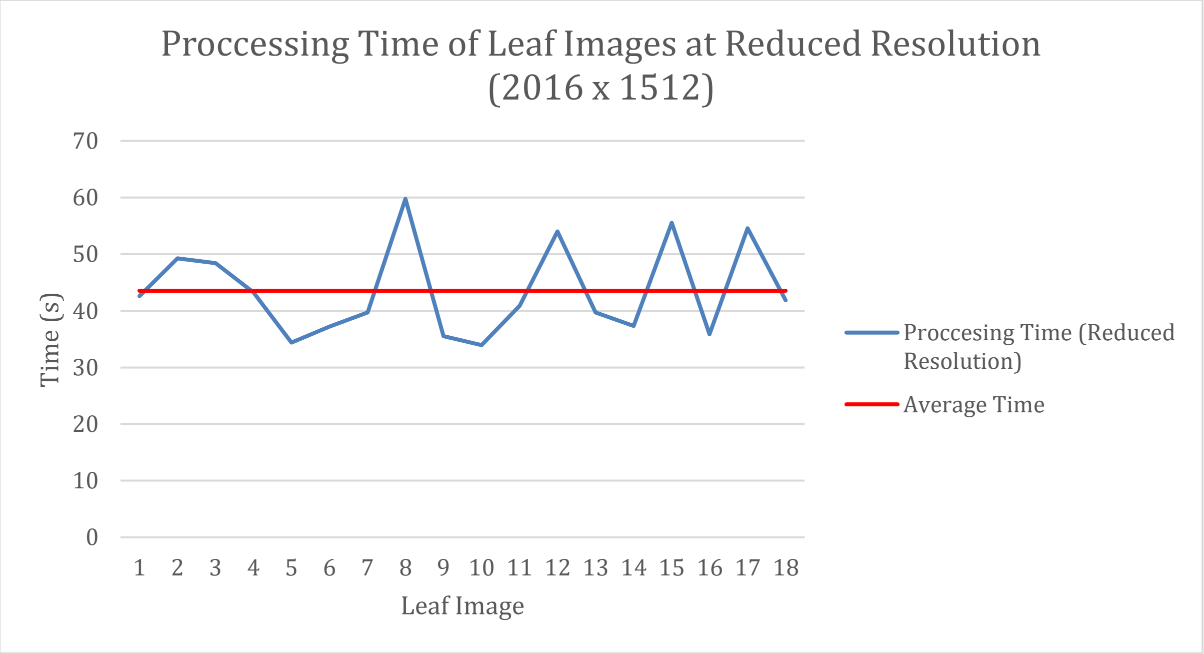


Figure 4-58: Processing Time of Leaf Images at Quarter Resolution (2016 x 1512)

It is clear from Figure 4-57 and Figure 4-58, that the average processing time is quite higher for images taken at full resolution when compared to images processed at quarter resolution. There is about a 4 time increase in processing time when using full resolution images when compared to quarter resolution images. This increase in processing time is expected as there are 4 more-time pixels to process, showing a direct relation to resolution and processing time.

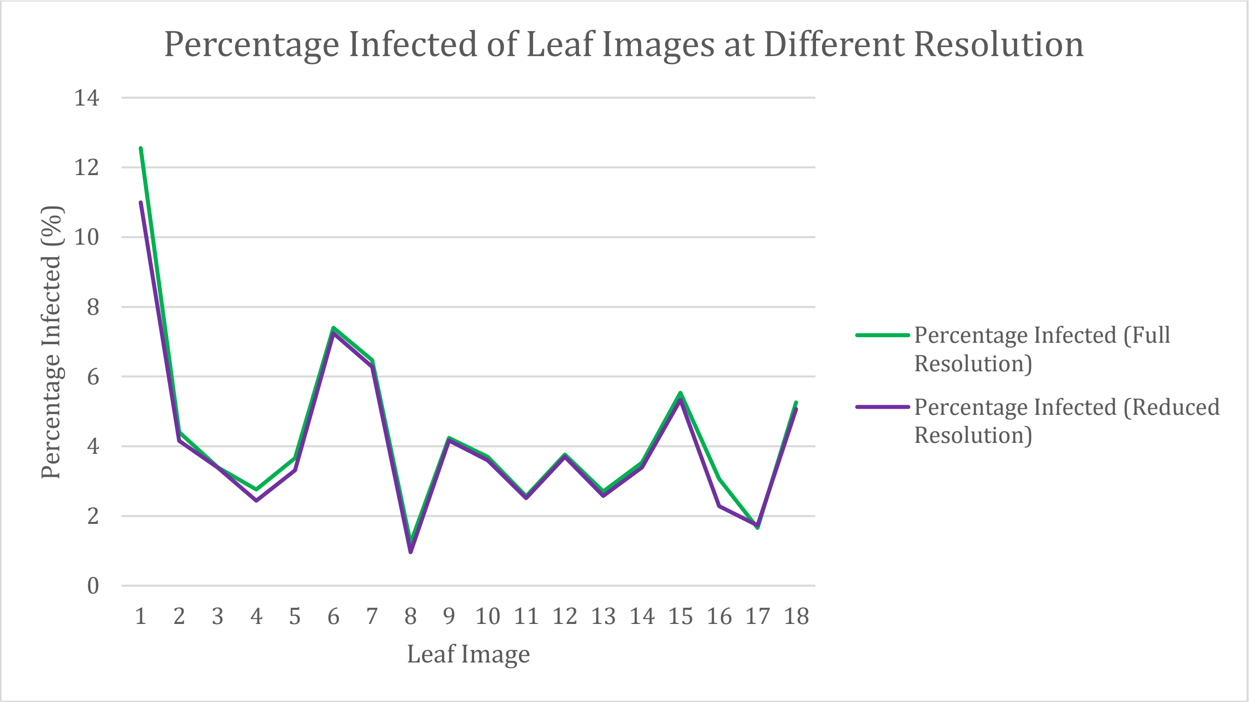


Figure 4-59: Percentage Infected at Full Resolution vs. Quarter Resolution

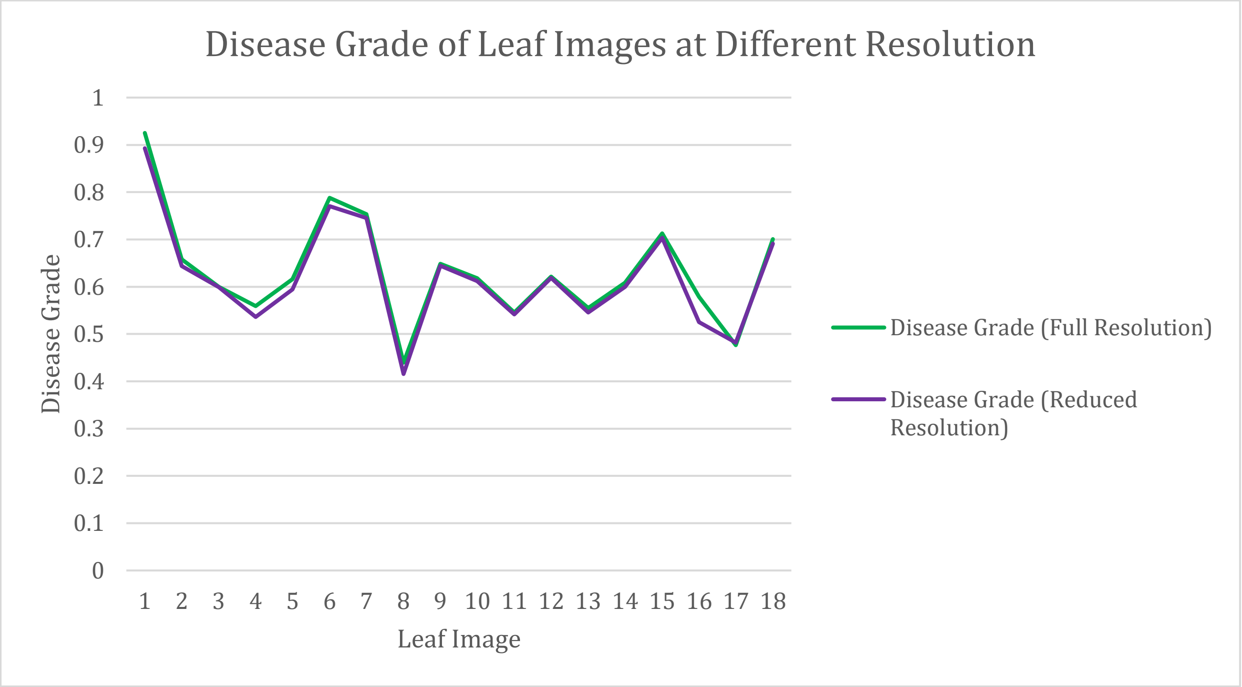


Figure 4-60: Disease Grade at Full Resolution vs. Quarter Resolution

Figure 4-60 illustrates how the disease grade classification is affected by the change in resolution. Clearly the affect the resolution has on disease grade is very minor if not negligible. The average difference in disease grade is 0.014 from images processed at full resolution and images at quarter resolution. This average difference is insignificant in affecting the overall accuracy of the disease grade classification. Because images at quarter resolution still have a high pixel count it does not affect the disease grade classification.

4.8.2. Internet Leaf Image Results (Images 19 – 21)

Each image retrieved from the Internet was tested at its native full resolution and then again at quarter resolution. Figure 4-61 illustrates the difference in processing time between images processed at full resolution and quarter resolution. Figure 4-63 illustrates the difference in disease grade classification between images processed at full resolution and quarter resolution.

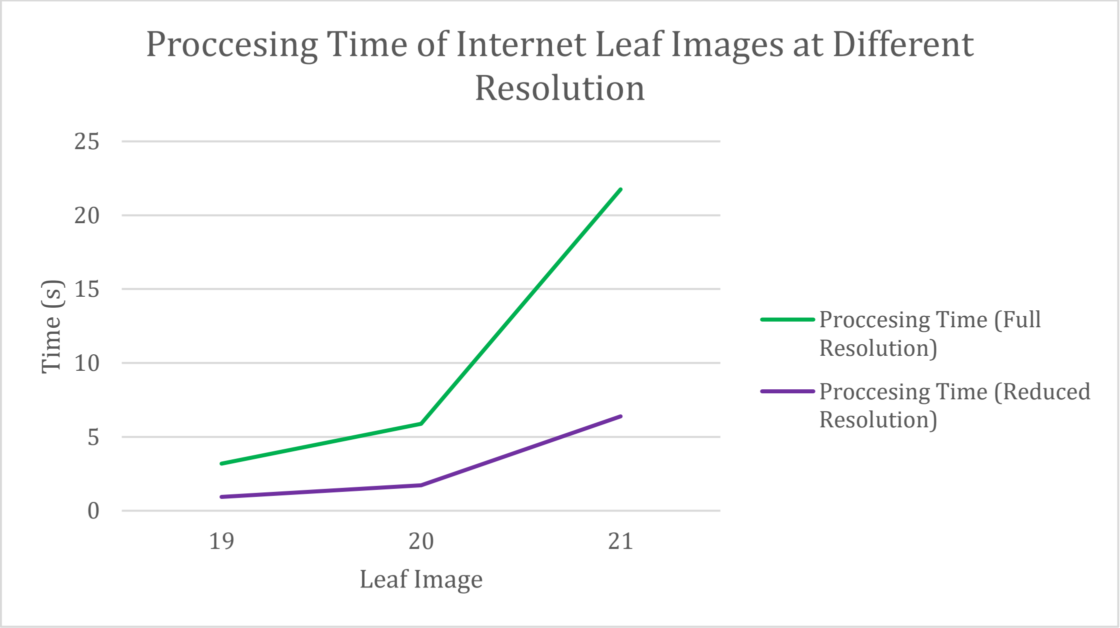


Figure 4-61: Internet Leaf Images Processing Time at Full Resolution vs. Quarter Resolution

The processing times as shown in Figure 4-61 are significantly lower than iPhone leaf images as the resolution is much lower for internet leaf images. In this case there is about a 70% decrease in processing time when using reduced resolution images when compared to full resolution images. Because leaf image 21 has a very high resolution compared to images 19 and 20, the processing time is significantly larger.

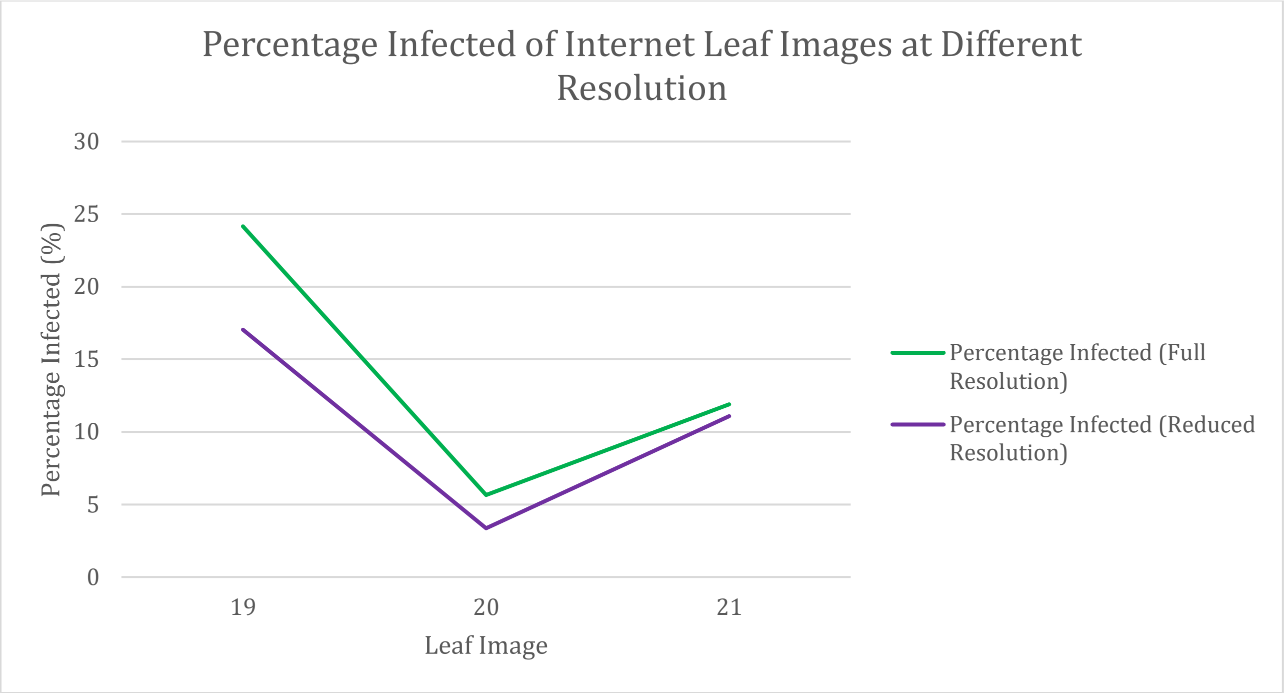


Figure 4-62: Internet Leaf Images Infected Percentage at Full Resolution vs. Quarter Resolution

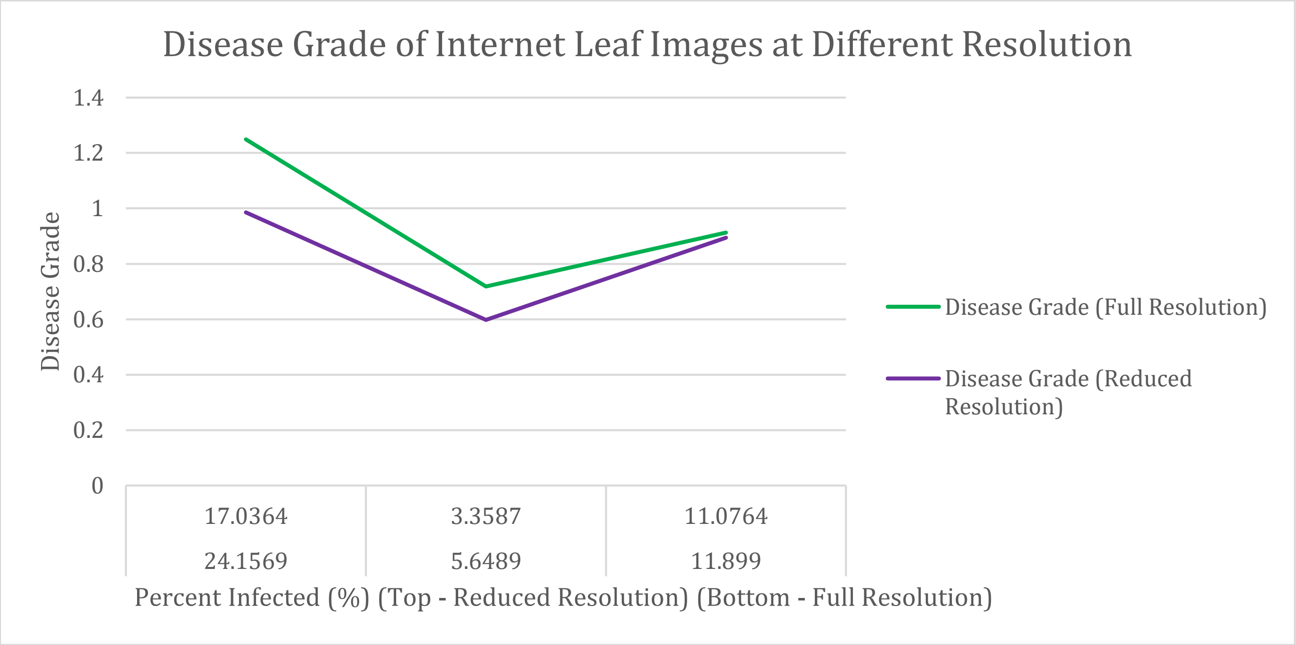


Figure 4-63: Internet Leaf Images Percentage Infected vs. Disease Grade

Figure 4-63 illustrates how the disease grade classification is affected by the change in resolution. In this case, the change in resolution has a more significant impact on disease grade classification when compared to iPhone leaf images as shown in Table 4-10. For instance, leaf 19 receives a disease grade of 1.2491 at full resolution and 0.98565 at quarter resolution resulting in a relatively high disease grade difference. Because the pixel count is already low at full resolution the decrease in resolution results in a high variance in disease grade classification. However, for leaf images that have high resolution such as image 21 the difference in disease grade is insignificant.

Table 4-10: Internet Leaf Images Disease Grade at Full Resolution vs. Quarter Resolution

Leaf Image	Full Resolution	Disease Grade	Reduced Resolution	Disease Grade	Disease Grade Difference
19	590 x 443	1.2491	295 x 222	0.98565	0.26345
20	850 x 565	0.71826	425 x 283	0.59761	0.12065
21	1548 x 1031	0.91237	774 x 516	0.89455	0.01782

From the results of both iPhone captured images and internet leaf images, the payoff between resolution and processing time can be concluded. Decreasing the resolution for cases where the resolution is high such as with iPhone leaf images, results in significant decrease in processing time. The ratio between processing time and resolution decrease is approximately 1:1, where a quarter decrease in resolution will result in a fourth of the processing time in almost all cases. Because the pixel count is already high even at quarter resolution the accuracy of disease grade classification is not impacted significantly. However, when the native resolution is already low in the case of internet leaf images, the decrease in resolution will have a slight impact on the disease grade classification.

4.8.3. Disease Grade Classification Accuracy

An important aspect that has a significant effect on the classification of the disease grade is the selected disease cluster. After the image is segmented into multiple regions it is vital for the user to select the region that most accurately depicts the defected area. This technique relies on the user's knowledge of leaf diseases to correctly identify the area which represents the disease most. Because the diseased region is most likely coherent it should be in one clear cluster. This system is not disease specific, so any disease or defected area can be identified from the rest of the leaf area. In some cases, the leaf may be infected with different diseases that are similar in color and could be clustered into the same cluster. However, if multiple diseases are visible and have a significant difference in color it can be distinguished from one another accurately.

To measure the accuracy of the disease area classification, the diseased area is highlighted manually as shown in Figure 4-64 and Figure 4-65. It is then compared with the diseased area that is selected using the Offline system as seen in Figure 4-66 and Figure 4-67.

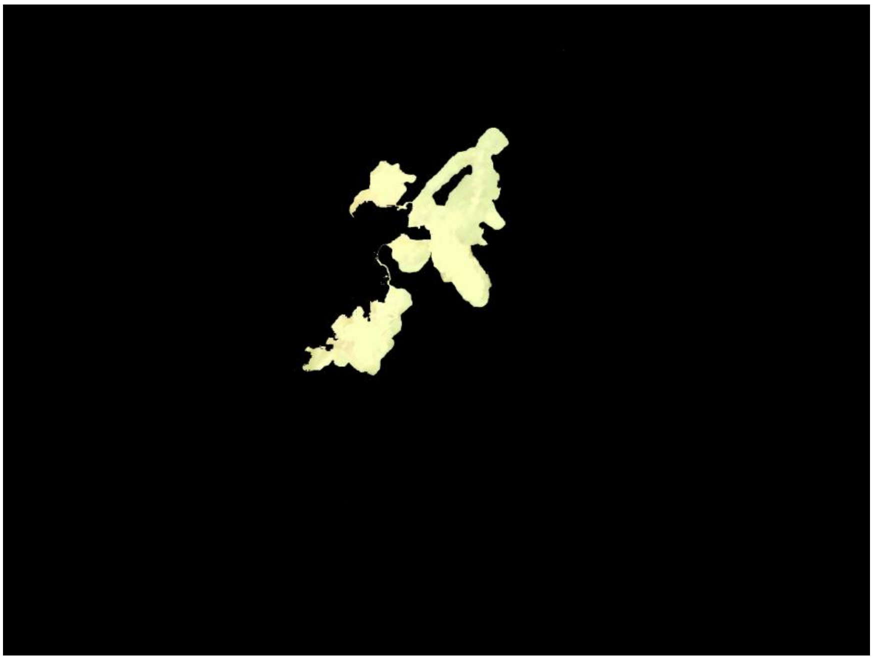


Figure 4-64: iPhone leaf image 1 diseased area highlighted manually

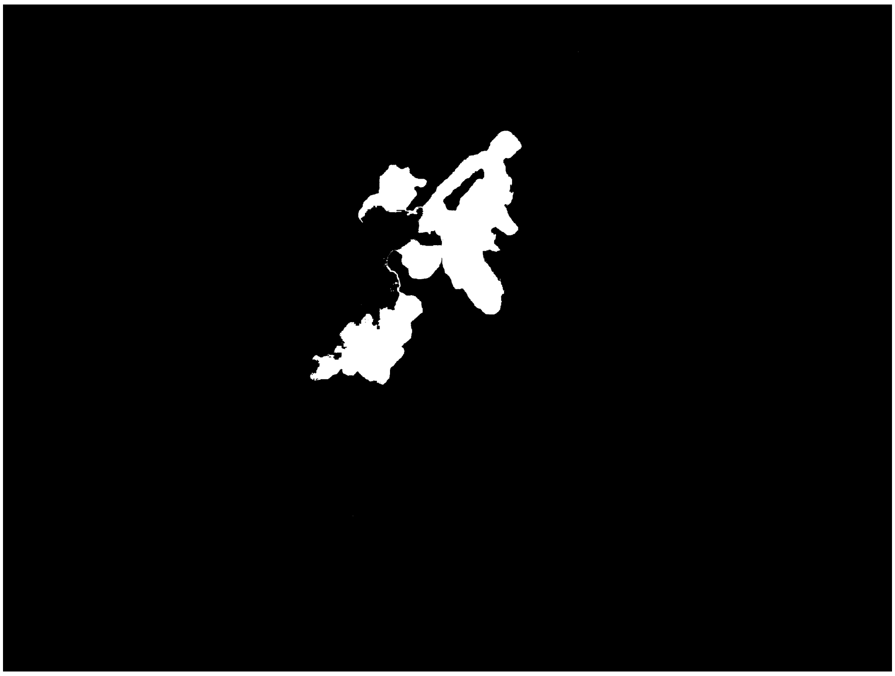


Figure 4-65: iPhone leaf image 1 diseased area highlighted manually in B/W

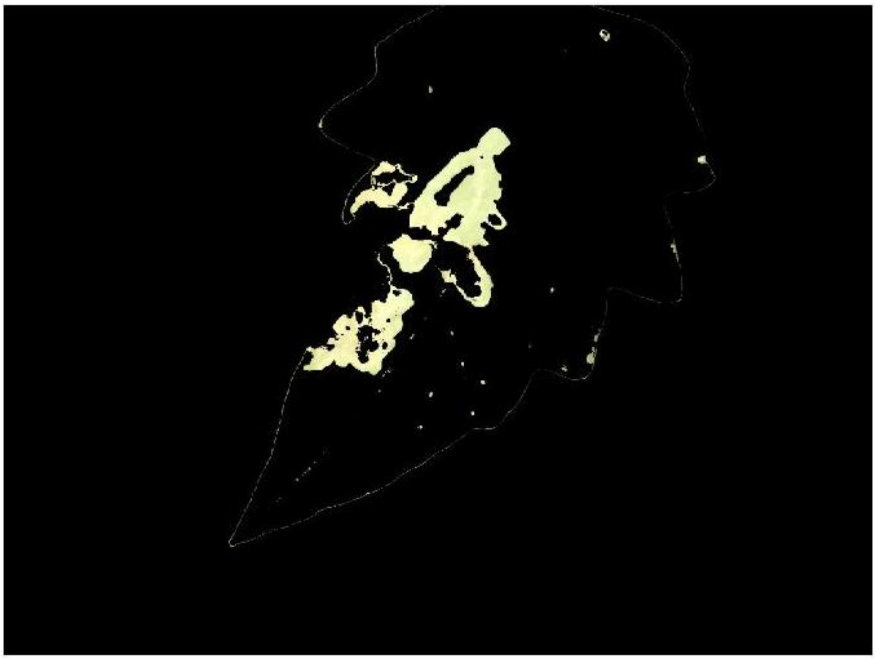


Figure 4-66: iPhone leaf image 1 diseased area from cluster 8

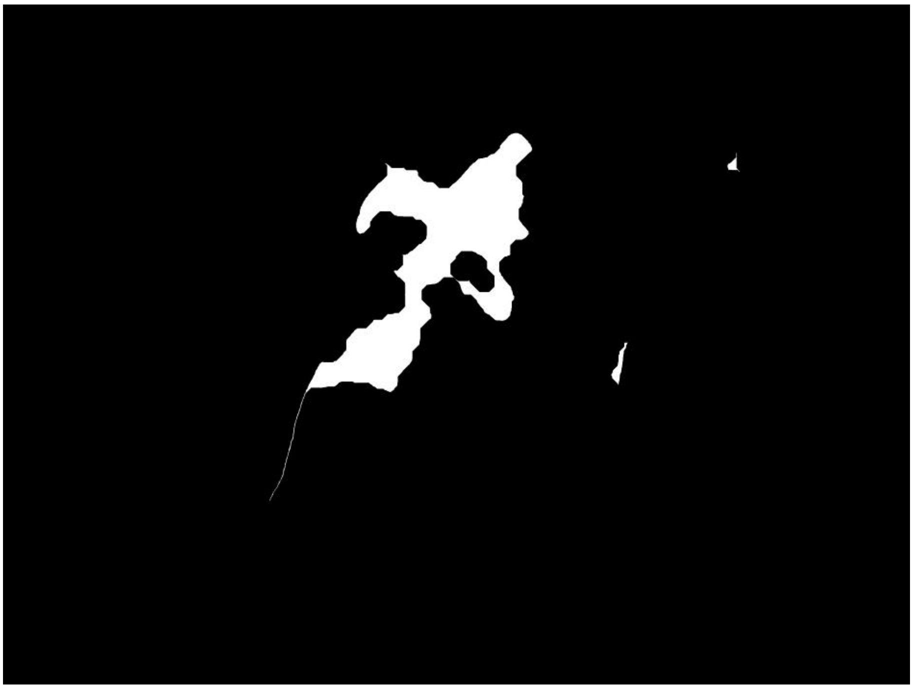


Figure 4-67: iPhone leaf image 1 diseased area after B/W conversion and morphological methods applied

Table 4-11 illustrates the difference in diseased area when selecting the infected area manually and when using the Offline system. This process is completed for the rest of the dataset to measure the accuracy of the Offline system. Figure 4-68 and Figure 4-69 illustrate the percentage infected and disease grade difference between the Offline system and selecting the disease area manually for iPhone leaf images.

For the iPhone leaf images, an average disease grade classification accuracy of 98% is achieved. Minor differences in percentage infection as seen in leaf images 6, 9, and 12 in Figure 4-68 have negligible impact on the disease grade calculation. When compared to other preharvest techniques such as [29] and [30], background segmentation in uncontrolled lighting conditions is achieved at a high accuracy, and the classification accuracy achieved for this Offline system is very promising.

Table 4-11: Disease Grade Classification Accuracy

Leaf Image	Manual Selection		Using Offline System		
	Percentage Infected (%)	Disease Grade	Percentage Infected (%)	Disease Grade	Disease Grade Accuracy
1	11.21245	0.8975	12.5545	0.92514	97
2	4.082	0.6402	4.4062	0.65767	97
3	3.32925	0.5958	3.3972	0.59998	99
4	2.89926	0.5682	2.7612	0.55885	98
5	3.44536	0.6029	3.6574	0.61571	97
6	6.95532	0.7722	7.3948	0.78819	97
7	6.27881	0.7453	6.473	0.75319	98
8	1.25511	0.442	1.2257	0.43946	99
9	3.8985	0.6299	4.2375	0.64866	97
10	3.80337	0.6243	3.6926	0.6178	98
11	2.45575	0.5375	2.5669	0.54534	98
12	4.173	0.6452	3.7575	0.62164	96
13	2.6691	0.5525	2.7081	0.55514	99
14	3.43221	0.6021	3.5329	0.60824	98
15	5.30473	0.7025	5.5321	0.71297	98
16	3.37869	0.5988	3.0591	0.57879	96
17	1.55267	0.4676	1.6576	0.47629	98
18	5.05104	0.6906	5.2615	0.70049	98

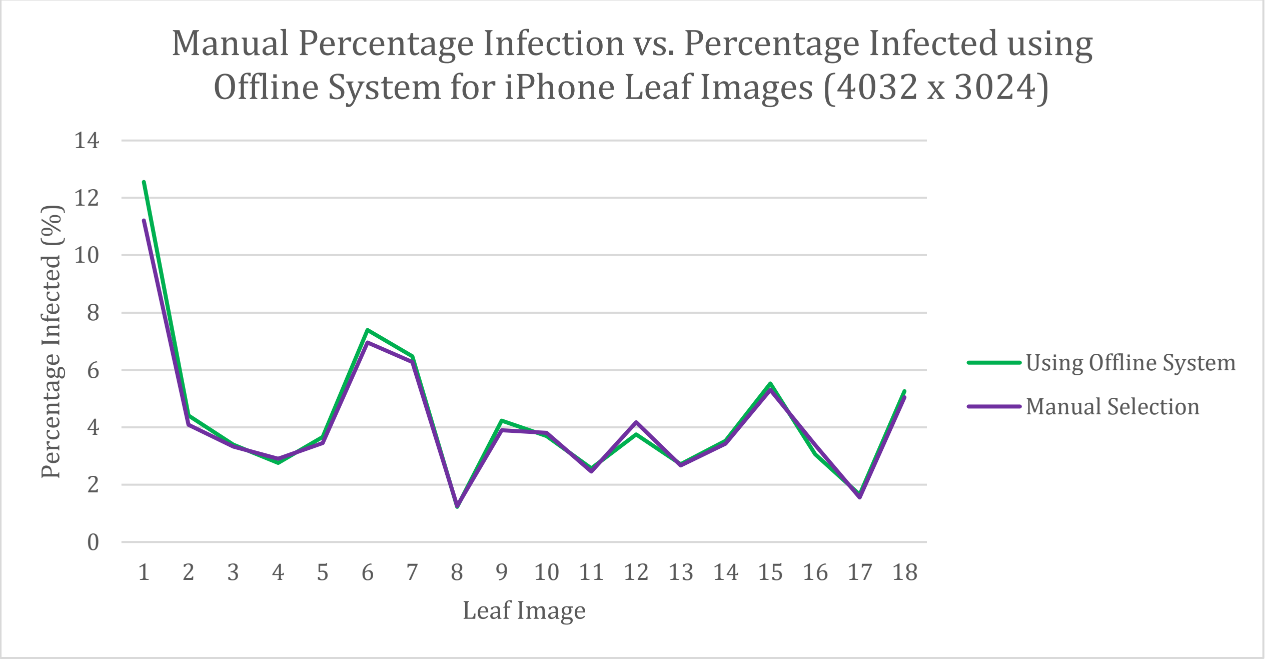


Figure 4-68: Manual Percentage Infection vs. Percentage Infected using Offline System for iPhone leaf images

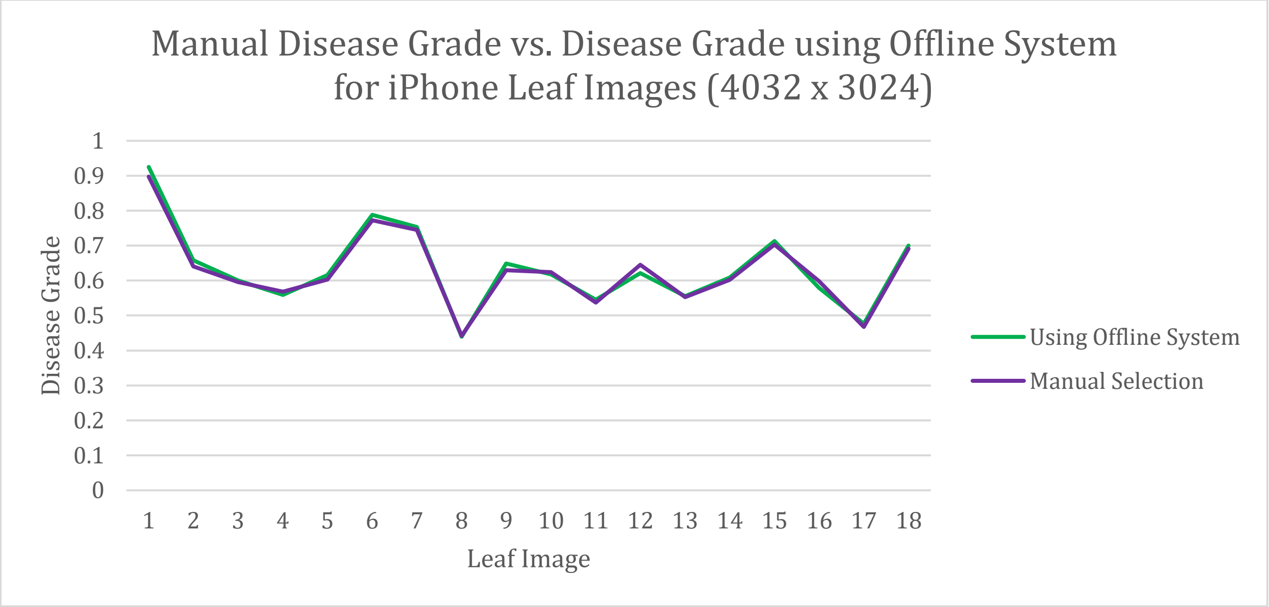


Figure 4-69: Manual Disease Grade vs. Disease Grade using Offline System for iPhone Leaf Images

Chapter 5
Robotized Plant Health Monitoring System

A major disadvantage of the Offline system is that it is required to be conducted off site and the system is not automated. Therefore, there is a need to design an Online automated system that has the capabilities of monitoring plant health and stresses under field conditions in real time. To fulfil such requirements, a robot is designed to move in a real agriculture field to measure plant health while providing position information about nutrient deficient plants. RGB and NIR imaging methods are integrated into a robot and used in synchronization to measure overall plant health using the Normalized Differential Vegetative Index (NDVI). Because of its low cost, NIR imaging techniques can be very effective when integrated with a robot. For this system, a few assumptions are considered and listed as follow,

1. Mobile robot moves along a specified path with a minimum width of 45cm
2. Images are taken under field conditions to distinguish healthy from unhealthy leaves
3. Monitoring plant health in a greenhouse environment with uniform terrain for movement

This section introduces how NDVI is calculated and how it is automated using a robotized system. This section also focuses on the design and the implementation of the robot with technical and functional requirements. The operational flow of the automated system is shown in Figure 5-1.

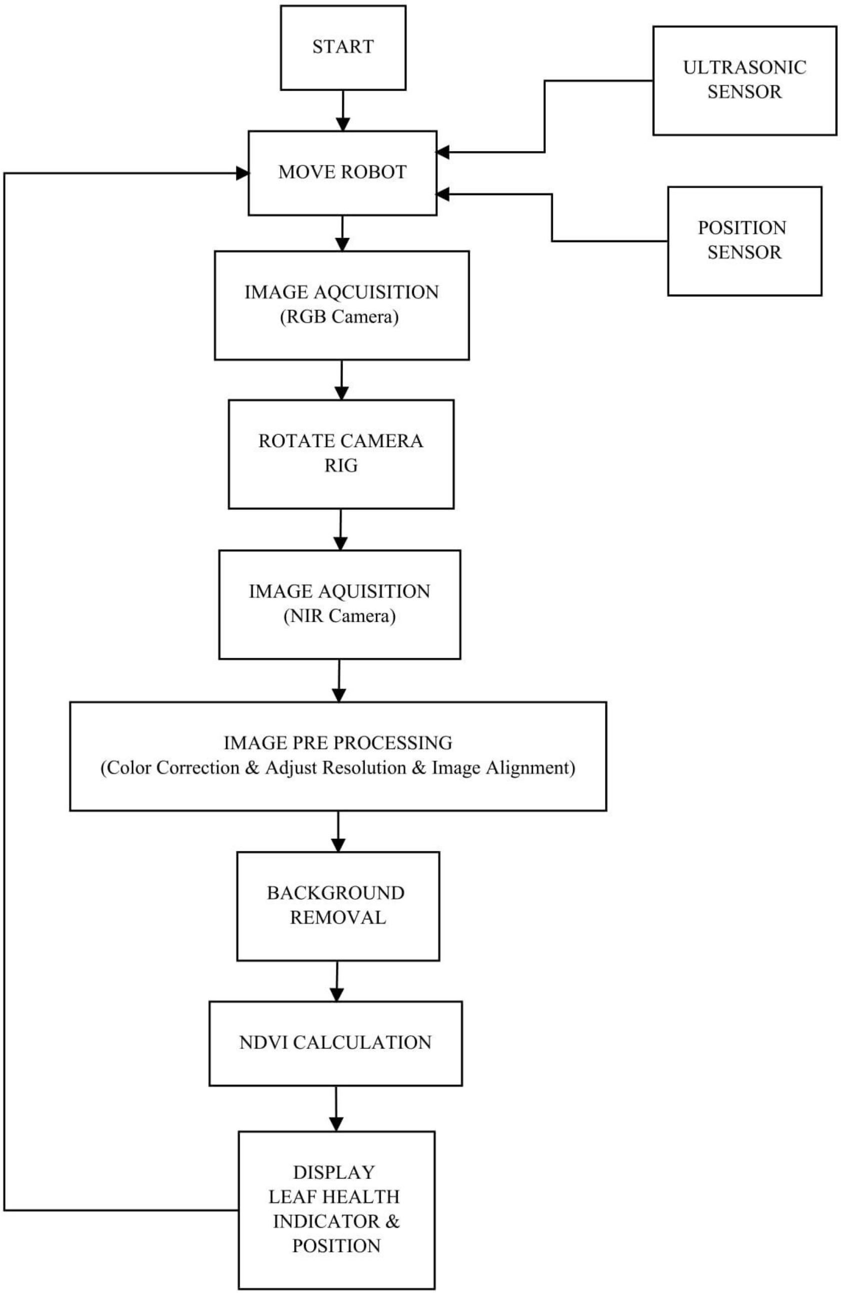

START
ULTRASONIC SENSOR
MOVE ROBOT
POSITION SENSOR
IMAGE AQCUISITION (RGB Camera)
ROTATE CAMERA RIG
IMAGE AQUISITION (NIR Camera)
IMAGE PRE PROCESSING (Color Correction & Adjust Resolution & Image Alignment)
BACKGROUND REMOVAL
NDVI CALCULATION
DISPLAY LEAF HEALTH INDICATOR & POSITION

Figure 5-1: Automated Robotic System Flowchart

5.1. Robot System Design

A robot was designed and implemented to fulfil the required functionality of moving along a specified path in a real field while capturing RGB and NIR images of leaves to measure NDVI of leaves which corresponds to an assessment of overall plant health. The overall design layout of the robot is shown in Figure 5-2. The robot after construction is shown in and Figure 5-3. The robot's functions are listed below:

1. Controlled movement from one location to the next,
2. Stop between locations to capture images,
3. Detect objects in its path (Obstacle Detection) while moving,
4. Provide map information associated with the NDVI calculation,
5. Calculate NDVI using two cameras fixed on a camera rig

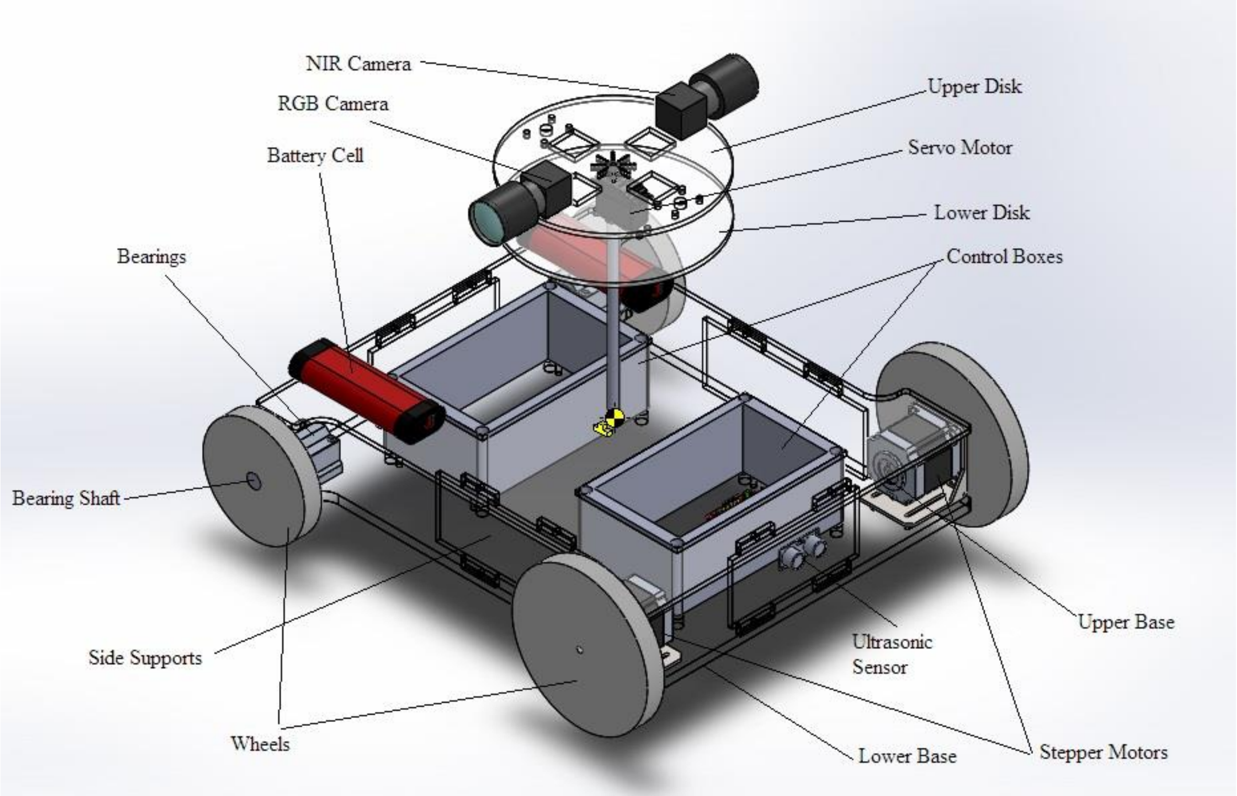


Figure 5-2: Robot mechanical design

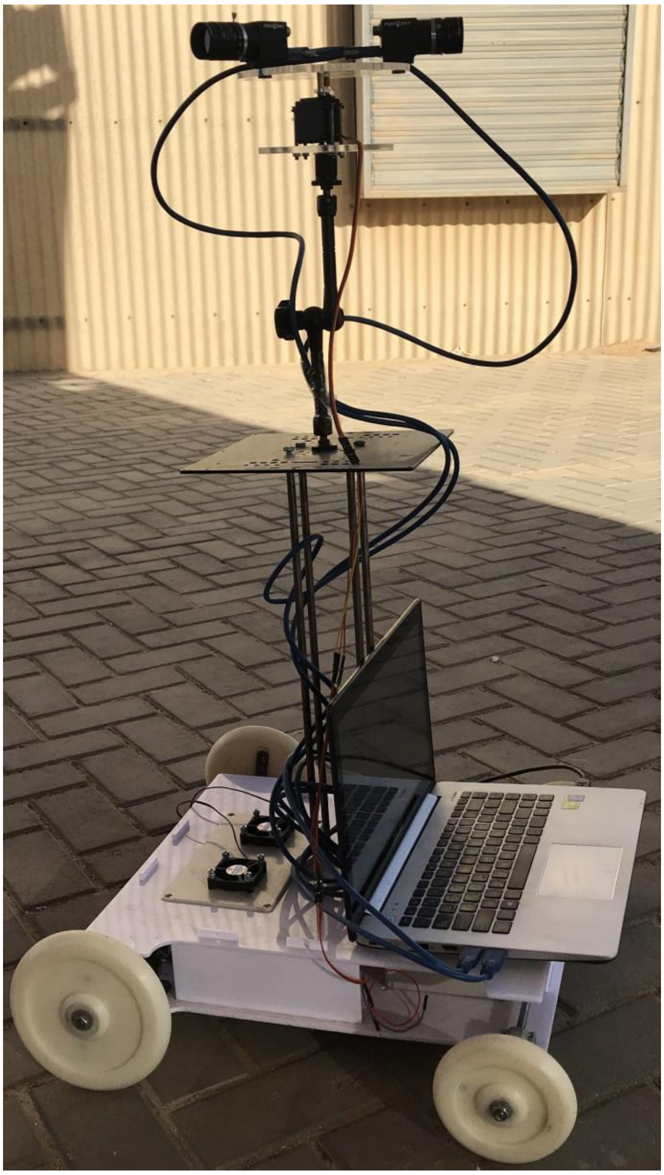


Figure 5-3: Implemented robot with all peripherals attached

5.1.1. Robot Frame

The body frame of the robot consists of multiple parts that can be easily assembled while maintaining structural strength and integrity. There are three main components that make up the robotic body; bottom base, top base, and side supports. The top and bottom base were coupled in a modular fashion using the side supports via the hole inserts. The initial structure is shown in Figure 5-4. To store the electrical components and protect them from outside interference such as sand and wind a control box is required to safety store all the components. The control box is of plastic material and stored inside the robot base between the upper and lower base as shown in Figure 5-5.

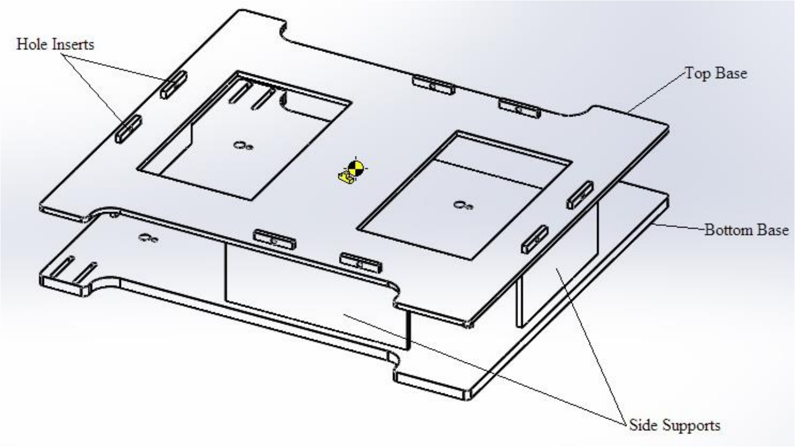


Figure 5-4: Body frame

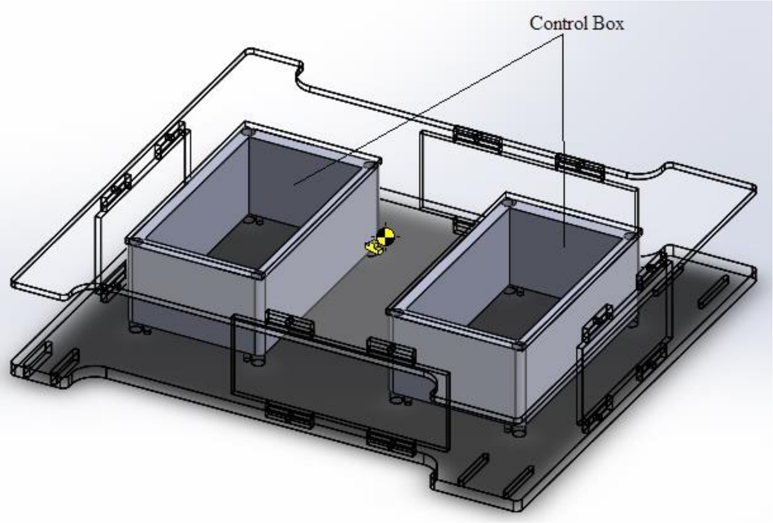


Figure 5-5: Control box

5.1.2. Motion Control of the Robot

For controlled movement, dual stepper motors were used as the main driving force for the robot. The motors are mounted on the robot using a motor bracket as shown in Figure 5-6. Two motor drivers were used and integrated with the robot controller to control the stepper motor. A dual full-bridge driver L298 chip is used for each motor to satisfy the motor requirements. The specifications of the stepper motor and driver are listed in Table 5-1

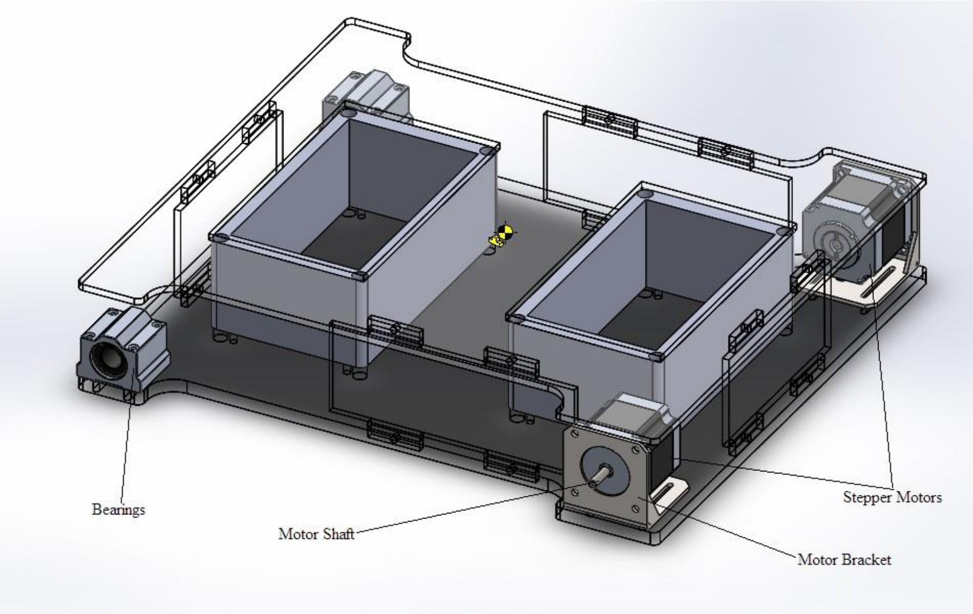


Figure 5-6: Stepper motor and Bearings

Table 5-1: Stepper motor and motor driver specifications

Step Angle (degrees)	1.8
Movement	Bi – Directional
Voltage supply	12V
Rated Current	2A/phase
Holding Torque	9 Kg.cm
Motor Weight	0.45 Kg
Maximum speed	12 rev/s
Motor driver supply voltage	24 V
Motor driver supply current	4A

5.1.3. Camera Rig

An adjustable camera monopod was used to create an upper level for the camera layout. As shown in Figure 5-7, a lower disk houses the servo motor that controls the of the two cameras, and the upper disk holds the camera layout.

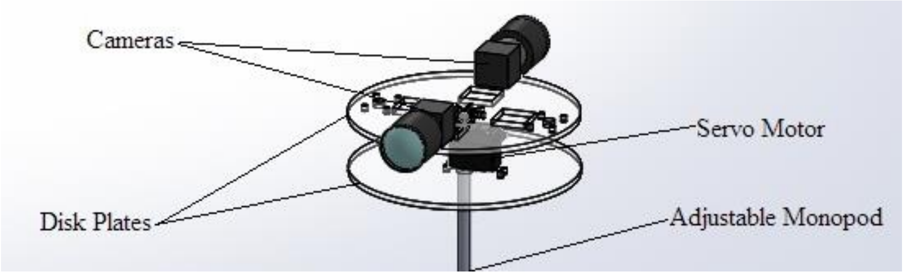


Figure 5-7: Servo motor and camera layout

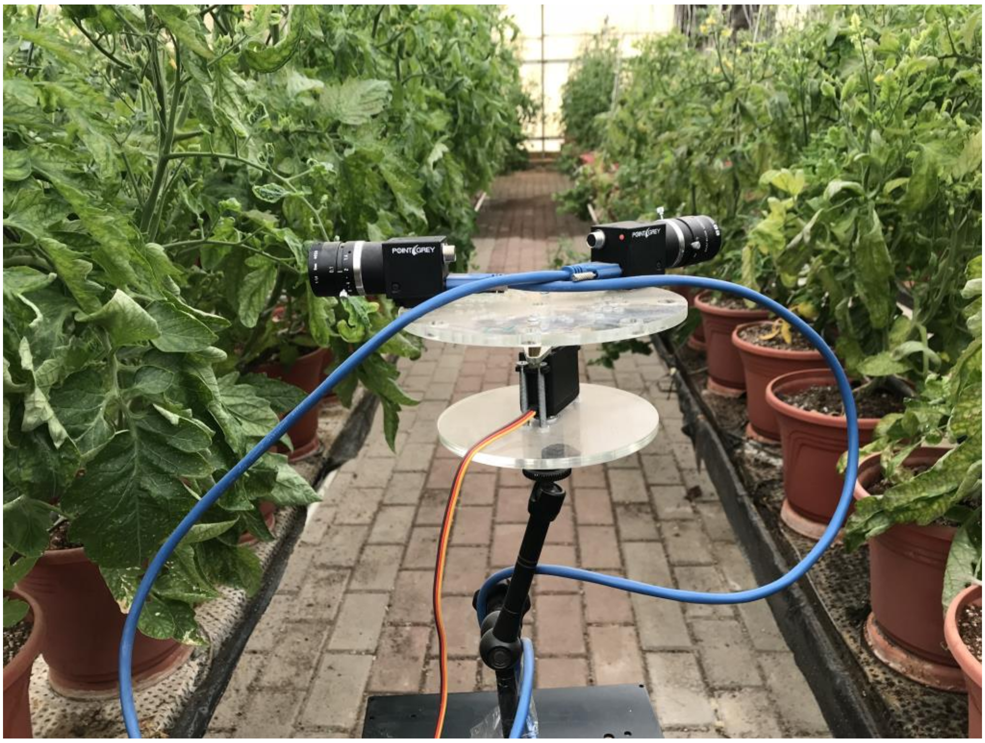


Figure 5-8: Camera rig

To control the position of the camera layout a servo motor is suitable because of its capability in rotating to specific angles or degrees. A metal gear high torque servo motor is selected because it allows increased servo reliability at high loads. The weight of the upper disk is about 160g, so the specifications of the servo motor listed in Table 5-2 is suitable for this case.

Table 5-2: Servo motor specifications

Voltage supply	7.2V
Speed	0.14 sec /60 degree
Torque	15 kg.cm
Dimension	40.8 x20.1x38 mm
Motor Weight	56g

5.2. System Controller

This section includes two control units integrated with the robotic system. These control units are:

1. Arduino Control Unit
2. Laptop Control Unit

5.2.1. Arduino Mega Control Unit

Arduino MEGA controller is selected because of the multiple serial ports available, for communication between the laptop and other peripherals that require serial communication, such as the GPS module. Arduino controller is responsible for the overall control and movement of the robot as well as send information over to the laptop. The Arduino controller is integrated with a range of hardware to facilitate multiple functions:

a. Stepper motor control
b. Servo motor control
c. Obstacle detection using Ultrasonic sensor

To detect for obstacles during the robots motion an ultrasonic sensor is used to measure distance between obstacles and the robot. The time of flight technique is used to measure distance between the robot and the object. Figure 5-9 illustrates how the ultrasonic sensor sends and receives the wave reflected from the object. Using Equation 5.1 the distance between the robot and the object can be found [47].

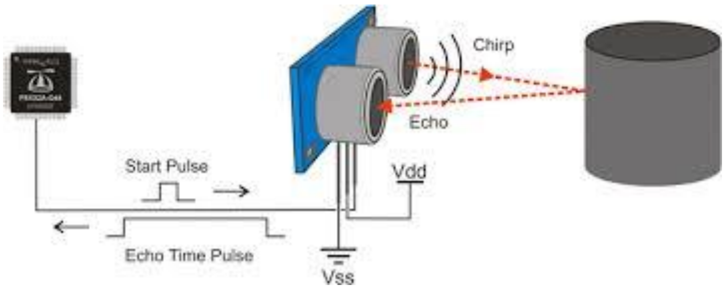


Figure 5-9: Ultrasonic sensor operation [47]

$$Distance = \frac{Speed\ of\ sound\ \times Time\ for\ echo}{2} \tag{5.1}$$

d. Position Sensor for location mapping

To provide position information of the stressful areas in a large agriculture field, a GPS module can be used. The module is the Skylab UART GPS Module. It is a small size and low weight GPS module as shown in Figure 5-10. It can be easily embedded into mobile robots because of its small size and low weight. The GPS module would be more suitable to use in larger agriculture areas where localization is more difficult. However, in a greenhouse environment the use of a GPS module is not as effective because of the smaller area.

Figure 5-10: GPS module

e. Serial communication with laptop

Because tasks are distributed by two control units, a communication protocol is required to ensure tasks are completed in a timely manner. Figure 5-11 illustrates how the laptop and the Arduino controller are connected for communication along with other peripherals.

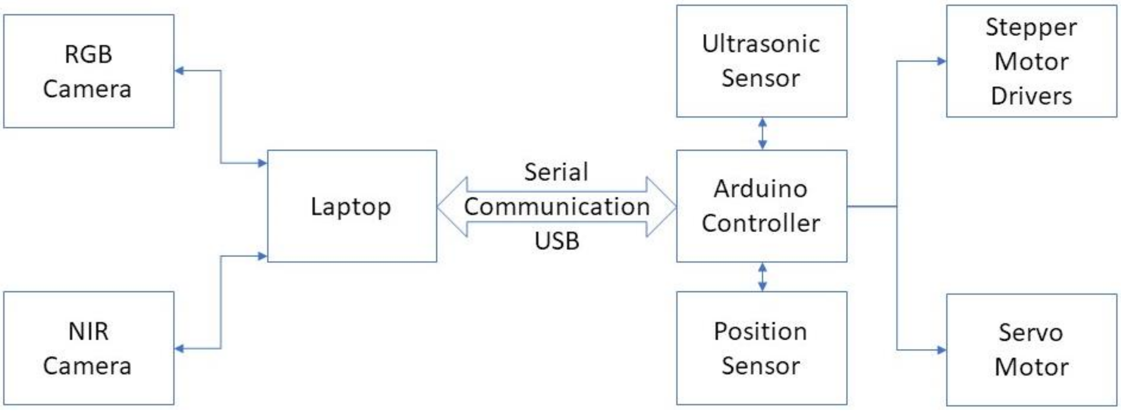


Figure 5-11: Arduino Controller and Laptop serial communication

5.2.2. Laptop Control Unit

The laptop is mainly responsible for image acquisition for the RGB and NIR cameras, image processing, and analysis. The tasks completed by the laptop are as follows:

a. Capture RGB Image

To capture images of the field and calculate the NDVI, two cameras are used. An RGB camera, to capture visible light and a NIR camera to capture near infrared light. These two cameras are attached to the top of the robot and controlled by a servo motor to take images of the same scene as the robot moves across the field line. The RGB camera used is BFLY-U3-13S2C-CS and its specifications are listed in Table 5-3.

Table 5-3: RGB camera BFLY-U3-13S2C-CS specifications [44]

Resolution	1288 x 964
Frame Rate	30 FPS
Megapixels	1.3 MP
Chroma	Color
Sensor Name	Sony ICX445
Sensor Format	1/3”
Quantum Efficiency (% at 525 nm)	61

b. Capture NIR Image

NIR camera used is BFLY-U3-13S2M-CS and the specifications are listed in Table 5-4.

.

Table 5-4: NIR camera BFLY-U3-13S2M-CS specifications [44]

Resolution	1288 x 964
Frame Rate	30 FPS
Megapixels	1.3 MP
Chroma	Mono
Sensor Name	Sony ICX445
Sensor Format	1/3”
Quantum Efficiency (% at 525 nm)	61

c. Image processing
d. NDVI calculation
e. Plant health map

To ensure fast image processing it is vital the PC have a powerful graphics processing unit with enough VRAM. Therefore, the laptop selected has an NVIDIA GeForce GT 740M graphics chip with 2GB DDR3 VRAM. The laptop is also small in size and relatively light, so it can be placed on top of the robot without adding too much weight. The laptop is physically connected to the Arduino via a serial cable and linked with the cameras using a USB cable as seen in Figure 5-11. The specifications of the laptop are listed in Table 5-5.

Table 5-5: Laptop PC specifications

Graphics	NVIDIA GeForce GT 740M with 2GB DDR3 VRAM
Dimensions	348 x 241.8 x 24.8 mm
Weight	2.20 kg (with 4 cell battery)

To explain the overall function of the robotic system a flowchart is shown in Figure 5-12.

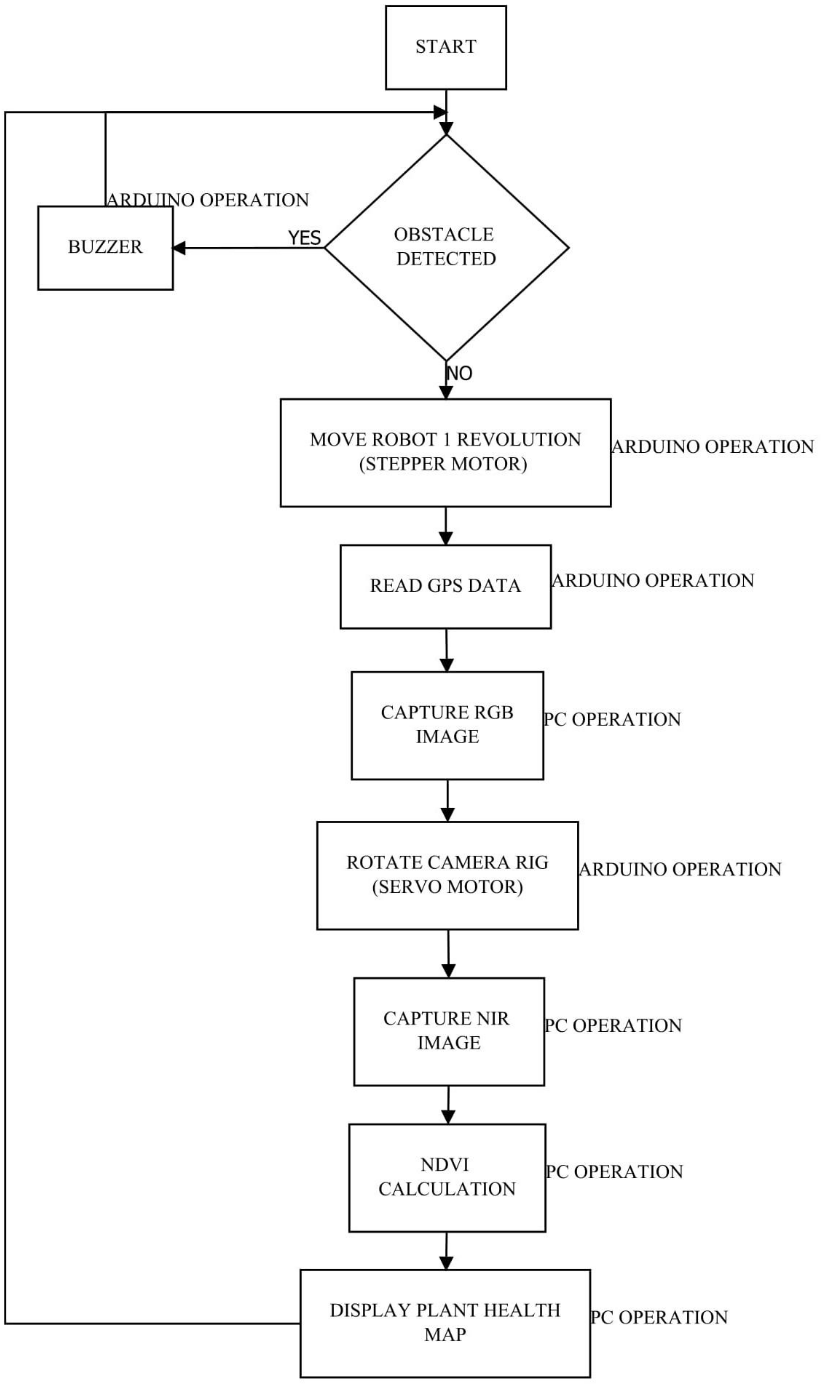


Figure 5-12: Automated Robotic System Process Flowchart

5.3. Normalized Difference Vegetation Index (NDVI)

NDVI is an indication used by many agriculturists and scientists to measure the overall health of plants and vegetative crops. This concept comes from the fact that healthy leaves with high amounts of chlorophyll will strongly absorb visible light and strongly reflect near-infrared light. While unhealthy or stressed plants with low chlorophyll content will strongly absorb near-infrared light and reflect most of the visible light [35]. Because chlorophyll is the most critical component in photosynthesis and plant growth it can be used as a good indicator of overall plant health and crop condition. As shown in Figure 5-13 healthy plants with high chlorophyll content will yield a higher NDVI when compared with unhealthy plants.

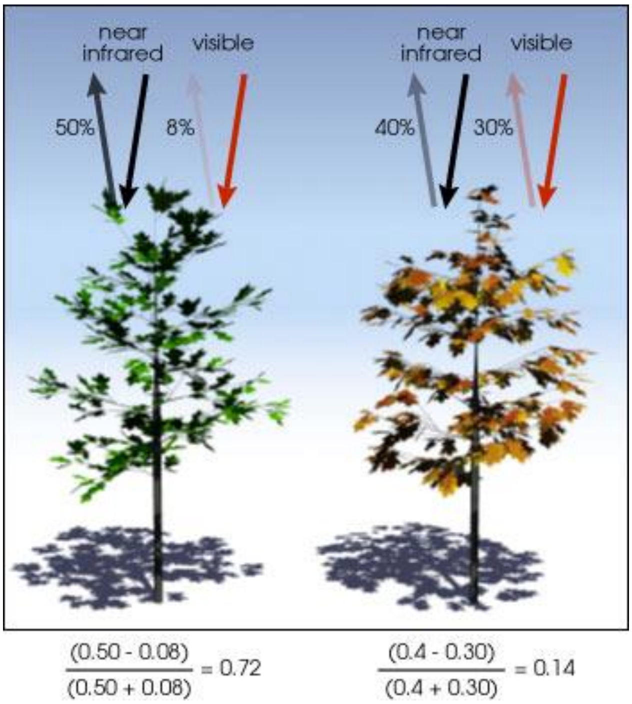


Figure 5-13: NDVI for healthy (left, NDVI = 0.72) and unhealthy (right, NDVI = 0.14) plants [45]

This concept has been extensively used by NASA and Earth observatory satellites to distinguish land areas from green areas such as forests and jungles [45]. Figure 5-14 shows an image taken with NASA satellites to map green areas of the entire Earth.

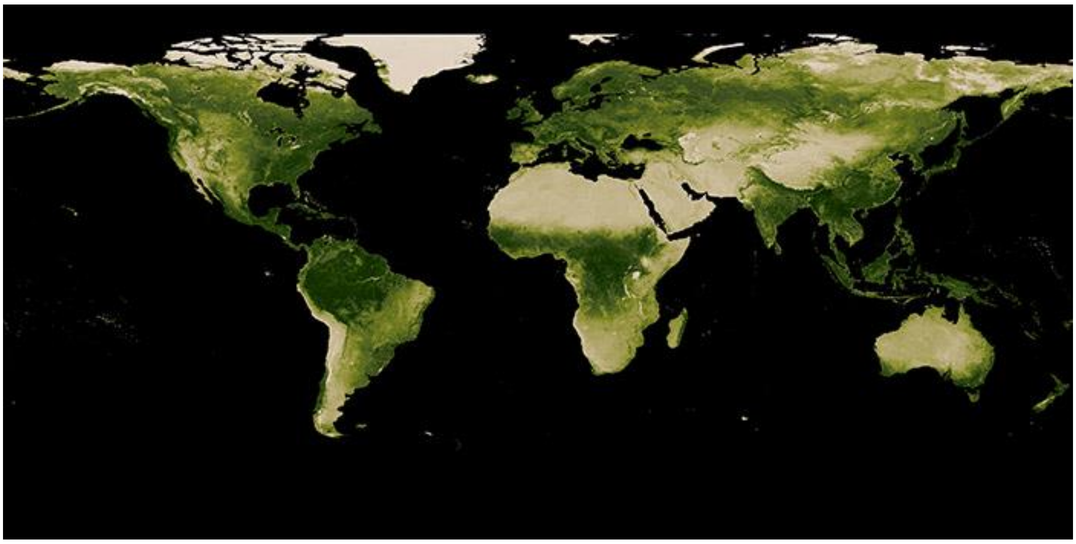


Figure 5-14: NDVI used to map green areas of earth [45]

The calculation of NDVI is illustrated in Equation 5.2, where NIR corresponds to the reflectance in the near infrared band and RED corresponds to the reflectance in the RED band [4].

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{5.2}$$

The RED band covers the wavelength range between 700nm and 760nm while NIR band covers the wavelength range from 760nm to about 1400nm as illustrated in Figure 5-15. The NDVI value is in the range of -1 to 1, where values greater than 0.4 will correspond to high foliar activity and values less than 0.2 correspond to low foliar activity. NDVI values between 0.2 and 0.4 indicate to sparse vegetation. This relation gives a direct indication of quality and development of vegetation [46].

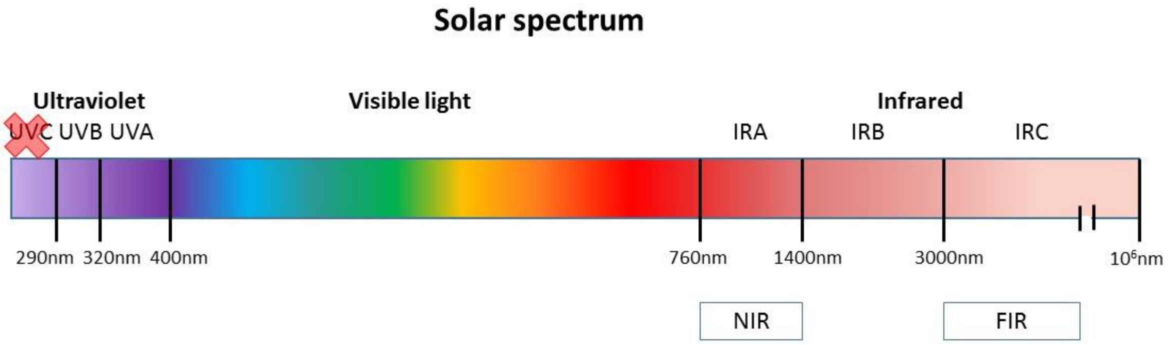


Figure 5-15: Ultraviolet, Visible and Infrared Spectrum [42]

5.3.1. Applications

NDVI can be used in many aspects of precision agriculture indicated below:

a. NDVI can be used to help agronomists and farmers to easily identify stressful areas, and to make timely decisions. NDVI maps can be produced at key dates to monitor plant growth and show variations in the field. This information can be used by farmers to determine the necessary actions for recovery [46].
b. Costly resources such as insecticides and mid-season fertilizers can be effectively used for better crop management while minimizing input costs [46].

5.3.2. NDVI Calculation

In order to extract the NDVI of a plant field, visible and NIR information is required. Therefore, two cameras are used in synchronization to capture both visible and NIR light reflected from vegetative plants. To test and validate the concept of using NDVI as a valid plant health indicator multiple test images were used and processed to create an NDVI map. Figure 5-16 and Figure 5-17 show the RGB and NIR images of tomato leaf plant 1.

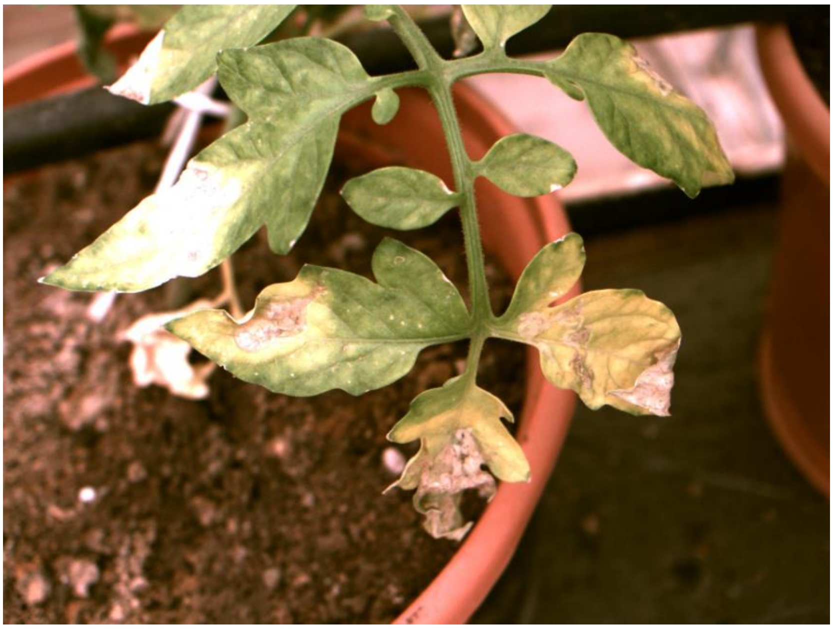


Figure 5-16: RGB image 1 of Tomato leaf

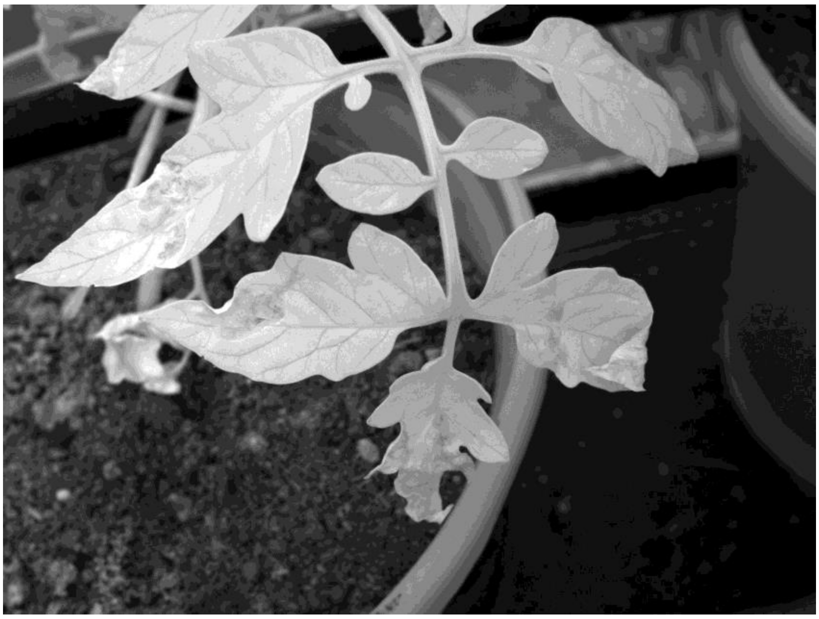


Figure 5-17: NIR image 1 of Tomato leaf

The RGB image is first segmented into R, G, and B channels. The RED channel shown in Figure 5-18, green channel shown in Figure 5-19, and blue channel Figure 5-20. Each channel represents vales of a 2D array of pixel values ranging from 0 – 255.

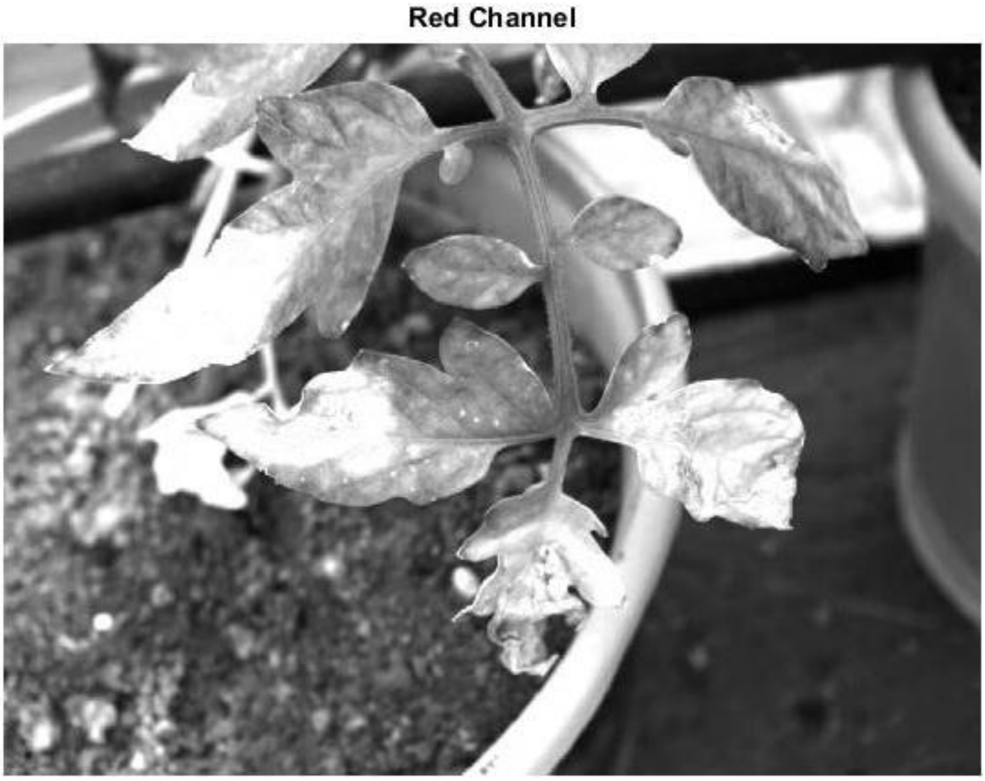


Figure 5-18: Red channel

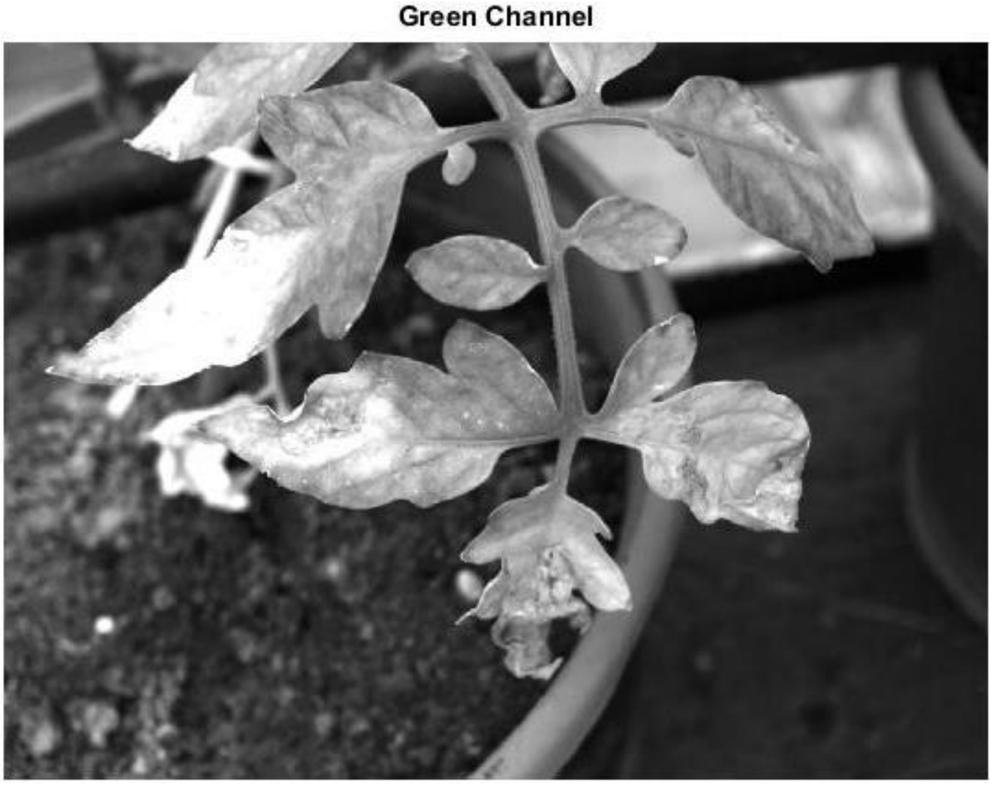


Figure 5-19: Green channel

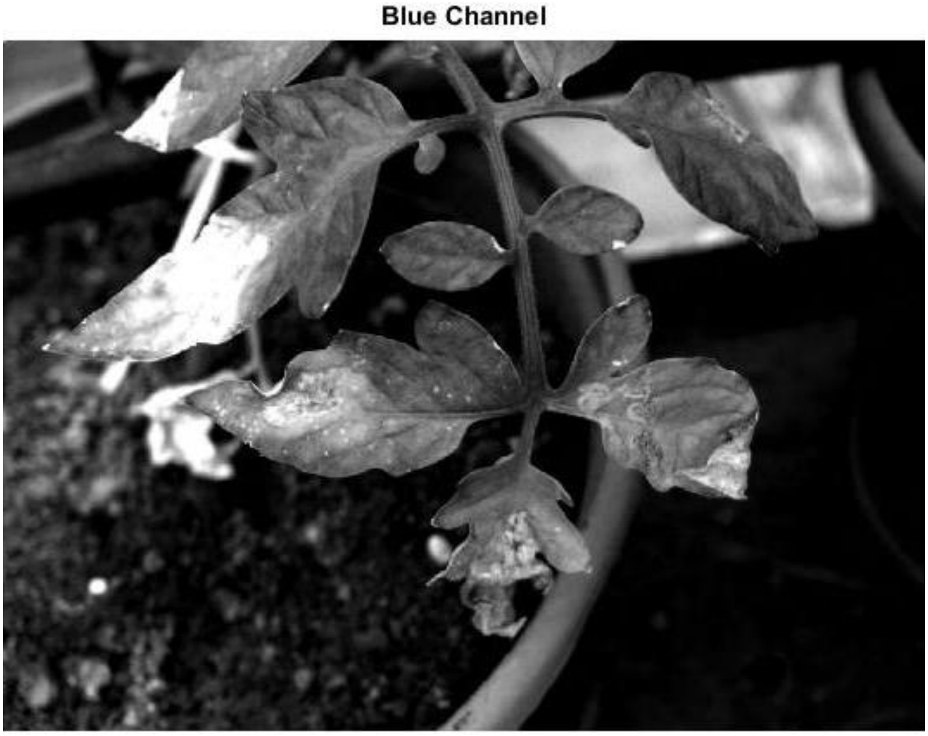


Figure 5-20: Blue channel

In order to calculate the NDVI, the RED channel from the RGB image and the NIR image pixel values must be normalized from [0 255] scale to a [0 1] scale. Equation 5.3 is computed for each pixel value in both the RED channel and NIR channel.

$$Pixel\ value_{[0\ 1]} = \frac{Pixel\ value_{[0\ 255]}}{255} \tag{5.3}$$

This is illustrated in the example matrices below.

$$\begin{bmatrix} 1 & 54 & 107 \\ 12 & 65 & 117 \\ 22 & 75 & 128 \end{bmatrix}$$

Pixel matrix before conversion

$$\begin{bmatrix} 0.0039 & 0.2118 & 0.4196 \\ 0.0471 & 0.2549 & 0.4588 \\ 0.0863 & 0.2941 & 0.5020 \end{bmatrix}$$

Pixel matrix after conversion

Once normalizing pixel values, each pixel from the RED band and its corresponding value from the NIR band are used to calculate the NDVI value using Equation 5.4. The NDVI values range from [-1 1] and are mapped onto a colormap of the same image size as shown in Figure 5-21.

$$NDVI_{i,j} = \frac{NIR_{i,j} - RED_{i,j}}{NIR_{i,j} + RED_{i,j}} \tag{5.4}$$

Where i and j represent the pixel location in the matrices

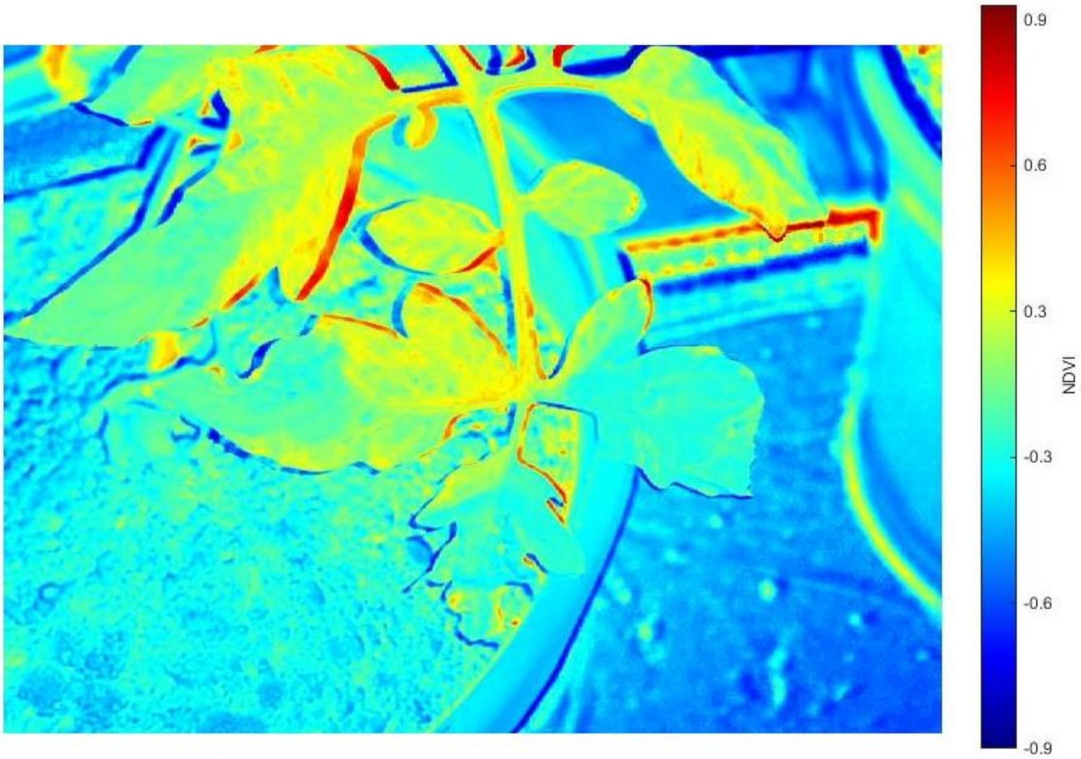


Figure 5-21: NDVI map 1 of Tomato leaf

Figure 5-21 presents the NDVI map, and it is clear that greener areas of the leaf give higher NDVI values when compared to other regions of the leaf. Healthy regions with high chlorophyll content will result in high NDVI values (>0.4) while undeveloped regions will display lower NDVI values (<0.2). Because diseases inhibit the overall content of chlorophyll this index can be used to detect plants under stress due nutrient deficiency problems. However, in order to give a clearer indication of plant health the background must be removed to avoid shadows and other objects that may affect the NDVI value.

5.3.3. Limitations

The calculation of the NDVI can be sensitive to a few factors:

a. NDVI calculations are sensitive to shadows from trees or nearby buildings, and can lead to misinterpreting NDVI values. These factors must be taken into consideration when taking images and calculating the NDVI [46]. This can be overcome by removing background from the images for more accurate representation of the scene.

b. Because soils tend to darken when wet, their reflectance may change accordingly, and this may lead to inaccurate NDVI values. The NDVI of an area can appear to change because of soil moisture changes from precipitation or evaporation and not because of vegetation changes [46].

c. The calculation and accuracy of NDVI values rely heavily on the sensor of the camera, and therefore may give different values depending on the sensor used. The quantum efficiency, characteristics, and performances of cameras in both the RGB and NIR range vary greatly. Therefore, a single formula like NDVI yields different results when applied to the measurements acquired by different instruments [46]. Quantum efficiency of the sensor in the NIR range is a crucial factor to be considered when selecting the cameras for NDVI calculation. Higher quantum efficiency in the NIR range, increases the sensitivity of the camera sensor in capturing NIR light.

5.4. Software Implementation

In this section the image analysis process for the RGB and NIR images is discussed starting from the image acquisition to the NDVI calculation and health analysis as seen in Figure 5-22.

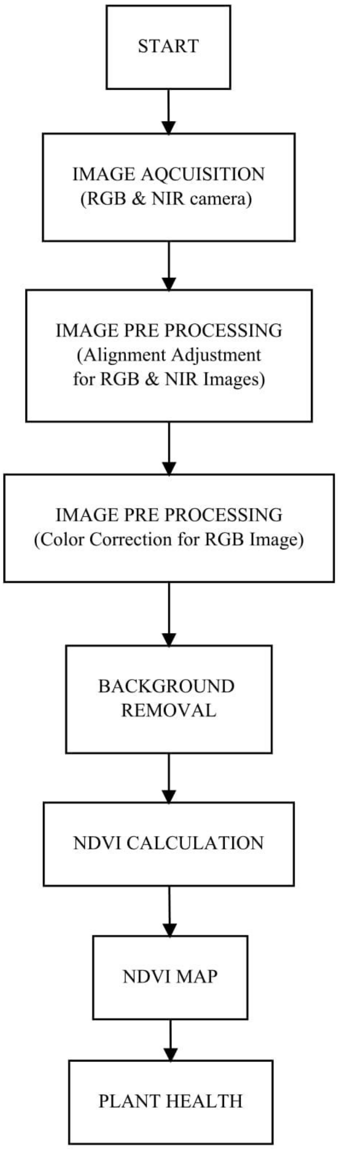


Figure 5-22: Online System Image Analysis flowchart for one cycle

5.4.1. Image Acquisition

It is vital to use cameras with high sensitivity in the NIR bands to ensure the effectiveness of the NDVI calculation. To ensure images captured by both cameras are identical, they both have the same specifications except for the chroma properties. The NIR camera should only allow NIR light to pass and to eliminate the RGB spectrum a monochrome camera was used instead of a color camera. Also, to eliminate visible light, a NIR optical pass filter was applied to the NIR monochrome camera. The plastic filter used is ideal for blocking visible light, while passing near infrared wavelengths only.

Another important property to consider is the quantum efficiency of the cameras in NIR wavelengths. As seen in Figure 5-23 the NIR camera chosen has relatively high quantum efficiency at NIR wavelengths (800 – 1000 nm). This gives the camera higher sensitivity in accurately detecting light in NIR wavelengths. Figure 5-24 and Figure 5-25 illustrate the RGB and NIR images of a pot tomato plant respectively.

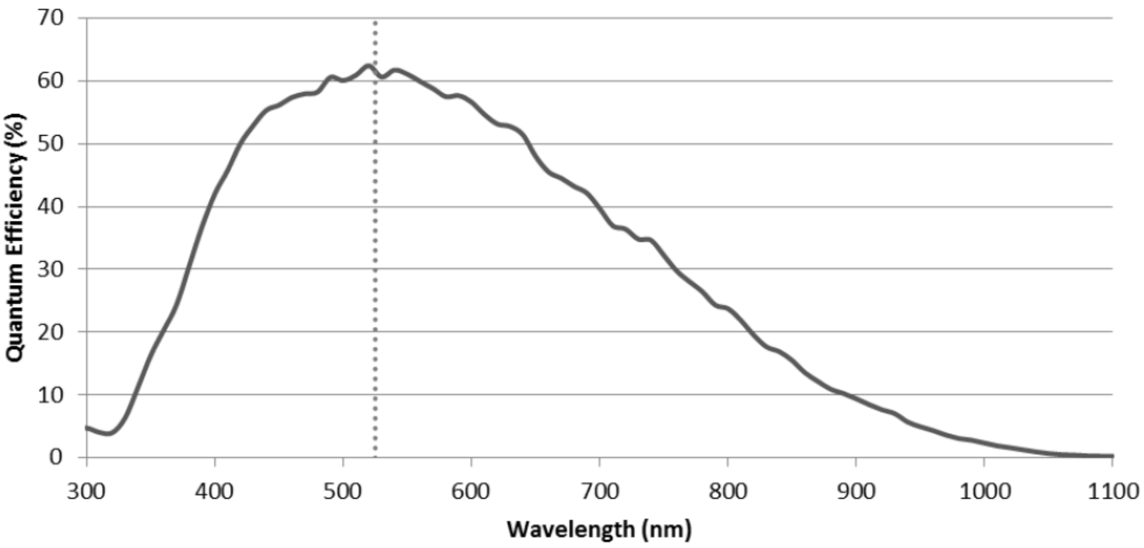


Figure 5-23: Quantum efficiency of cameras at different wavelengths [44]

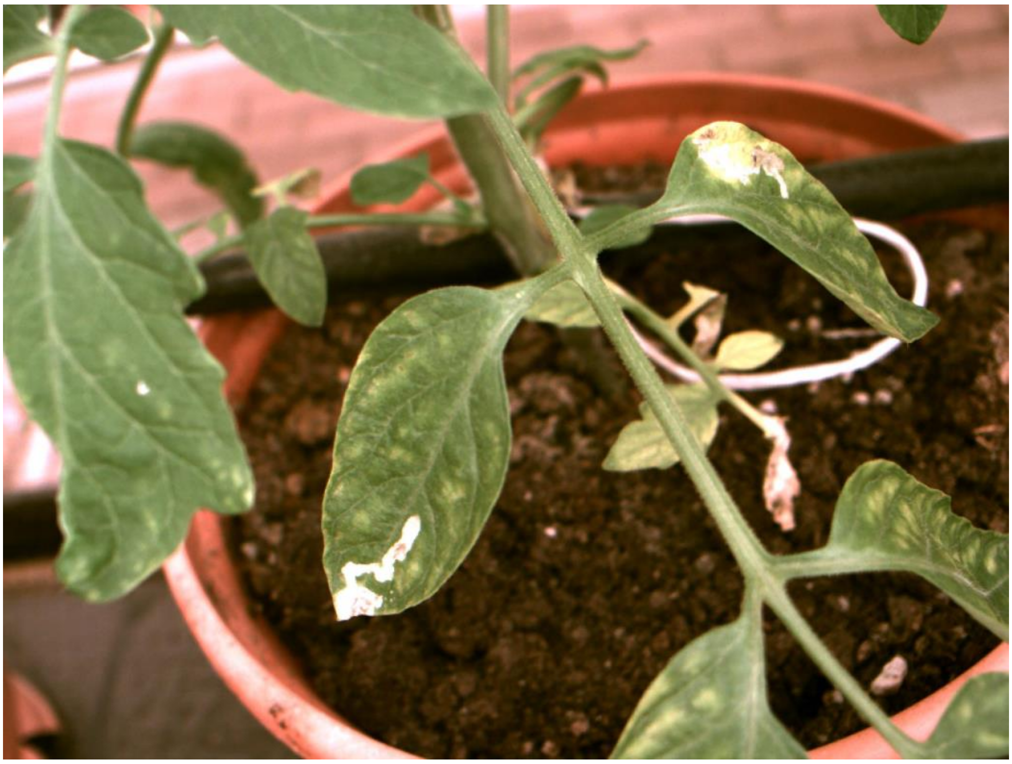


Figure 5-24: Leaf Image 1 RGB

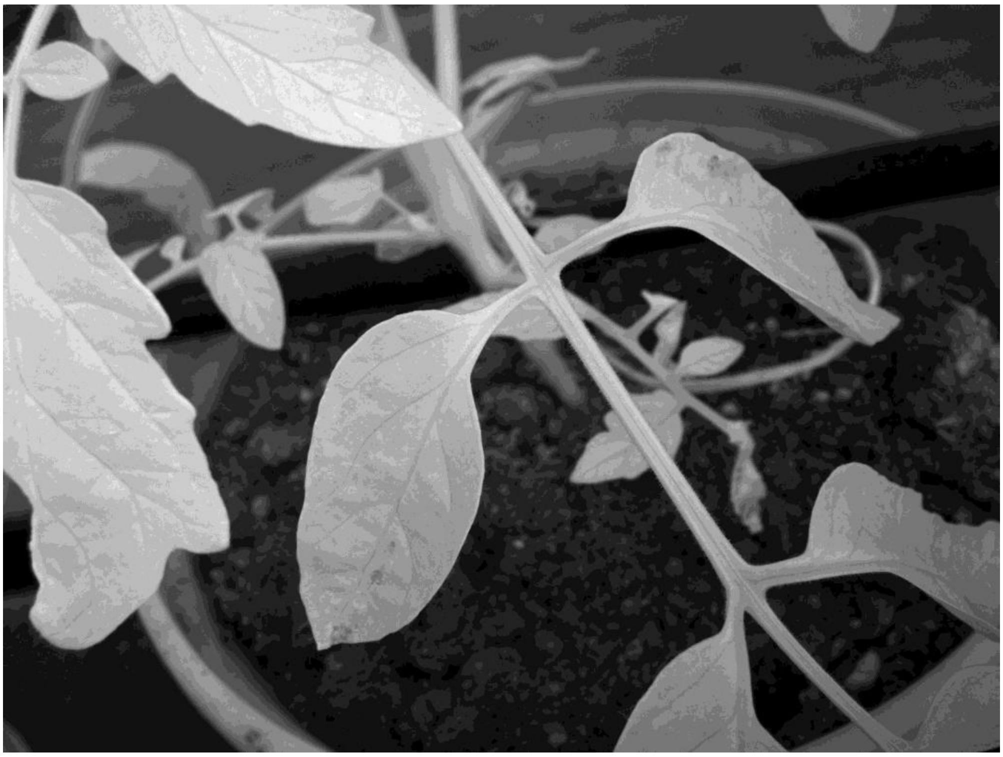


Figure 5-25: Leaf Image 1 NIR

5.4.1. Image Pre-processing

In this stage images are resized to a fixed resolution to reduce the computational burden, and any other image adjustments can be achieved in this stage such as cropping, contrast enhancement, and angle correction. Because the resolution of the images acquired is sufficient it is not necessary to resize the images to a lower resolution. However, image alignment is required as the RGB and NIR images are prone to offset during the servo motor rotation. Color correction is required for background segmentation using K-means clustering as illustrated in Sec. 4.2.2.

5.4.1.1. Image Alignment

To calculate the NDVI, the RGB and NIR leaf images must be aligned correctly. The RGB image is taken first and then via the servo motor the camera rig rotates and the NIR image is taken for the same scene. Because of the uncertainty of the servo motor rotation the NIR image may be misaligned from the RGB image. This angle must be taken into consideration and accounted for to ensure both images contain the same scene for NDVI calculation. This process is completed by matching features in both images and calculating the difference in position and angle of features. Speeded up robust features (SURF) detection is used to find prominent matching features in both images and use that information to rotate images so they would match. The SURF technique is similar to Scale-Invariant Feature Transform (SIFT) approach to feature detection. The key features of SIFT are listed below:

- Scale Invariant
- Rotation Invariant
- Invariant to changes in illumination
- Robust against noise

The SURF algorithm is based on the same principles and steps of SIFT, but it utilizes a different scheme and it provides better results at a reasonably faster time. SIFT is very accurate at matching local features, but is very computationally expensive [43]. Hence, it is more appropriate to use SURF as the system designed is an Online system where computation time should be kept at a minimum.

The SURF algorithm flowchart is illustrated in Figure 5-26, and the three main steps to SURF are listed below:

- Interest Point Detection: Identify prominent features, and should be detected regardless of viewpoint.
- Description: Each feature point has a unique description that is scale and rotation independent.
- Matching: Given two images, determine the transformation of the image, based on predetermined interest points.

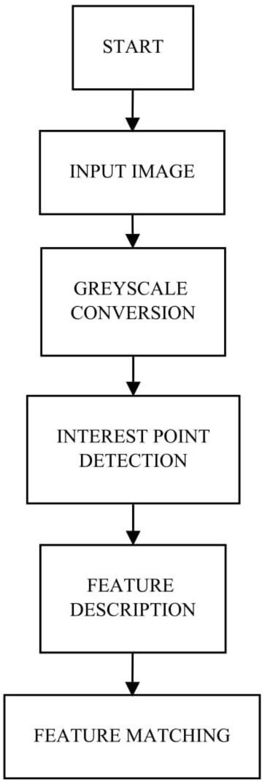


Figure 5-26: SURF algorithm flowchart

SURF function was implemented to detect matching features in both the RGB and NIR images. Figure 5-27 and Figure 5-28 illustrate all the key points detected in the RGB and NIR image after they are converted into greyscale. After removing key points with bad contrast and rejecting edges, the optimum features for both images are showed in Figure 5-29 and Figure 5-30. The matching features are shown in Figure 5-31. The transformation matrix is found between the matched features, and is used to transform the NIR image to match the RGB image as shown in Figure 5-32.

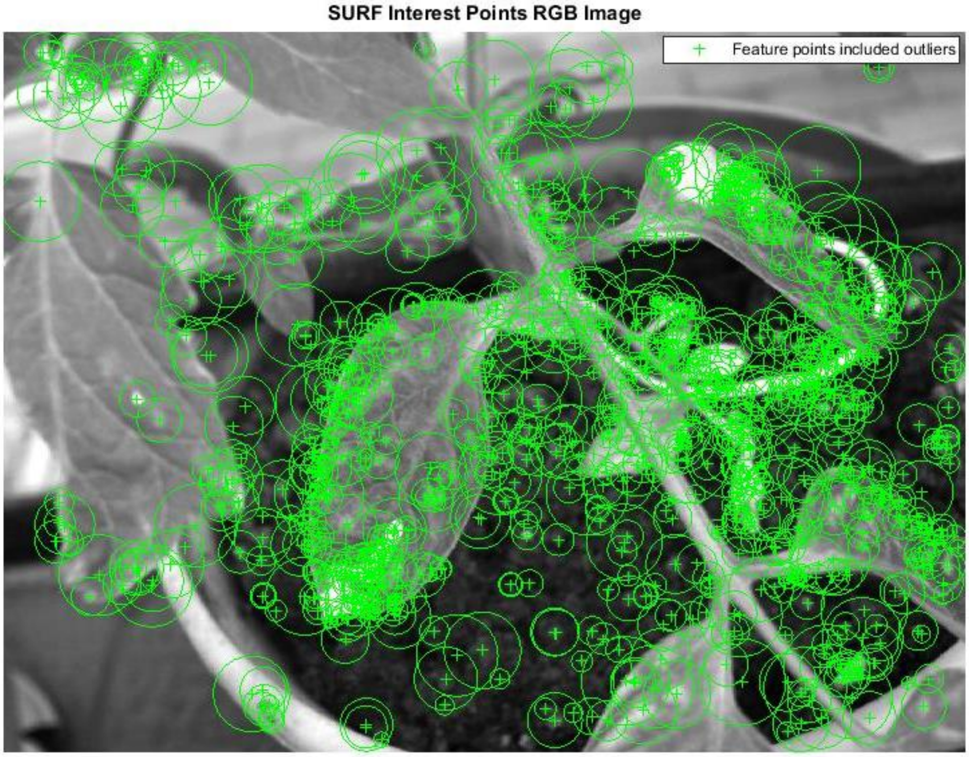

SURF Interest Points RGB Image
+ Feature points included outliers

Figure 5-27: Feature points in RGB image detected using SURF

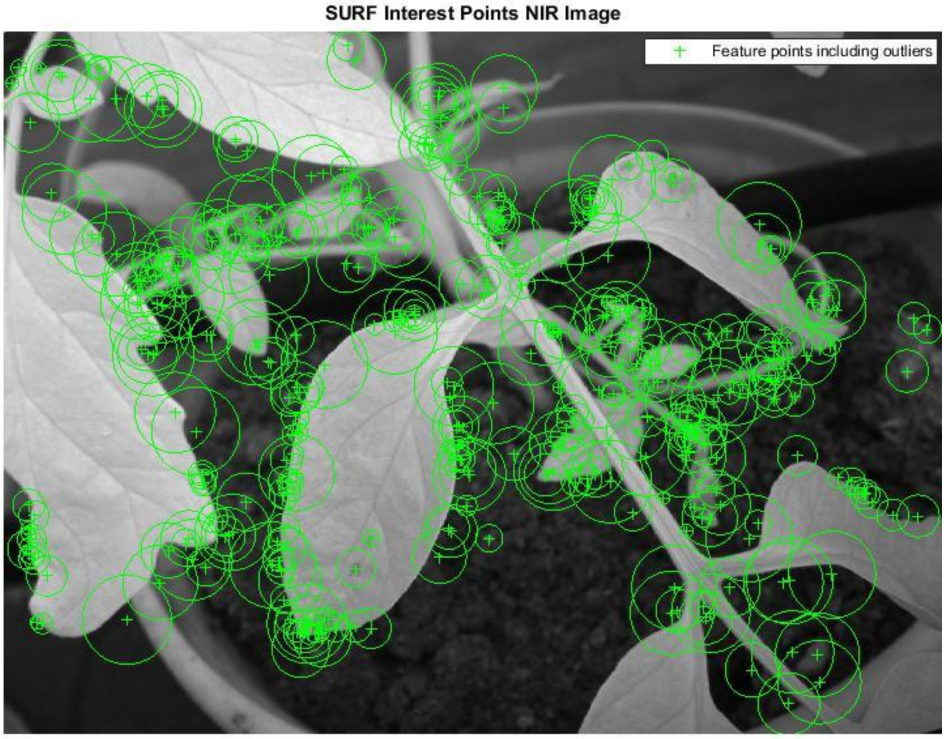

SURF Interest Points NIR Image
+ Feature points including outliers

Figure 5-28: Feature points in NIR image detected using SURF

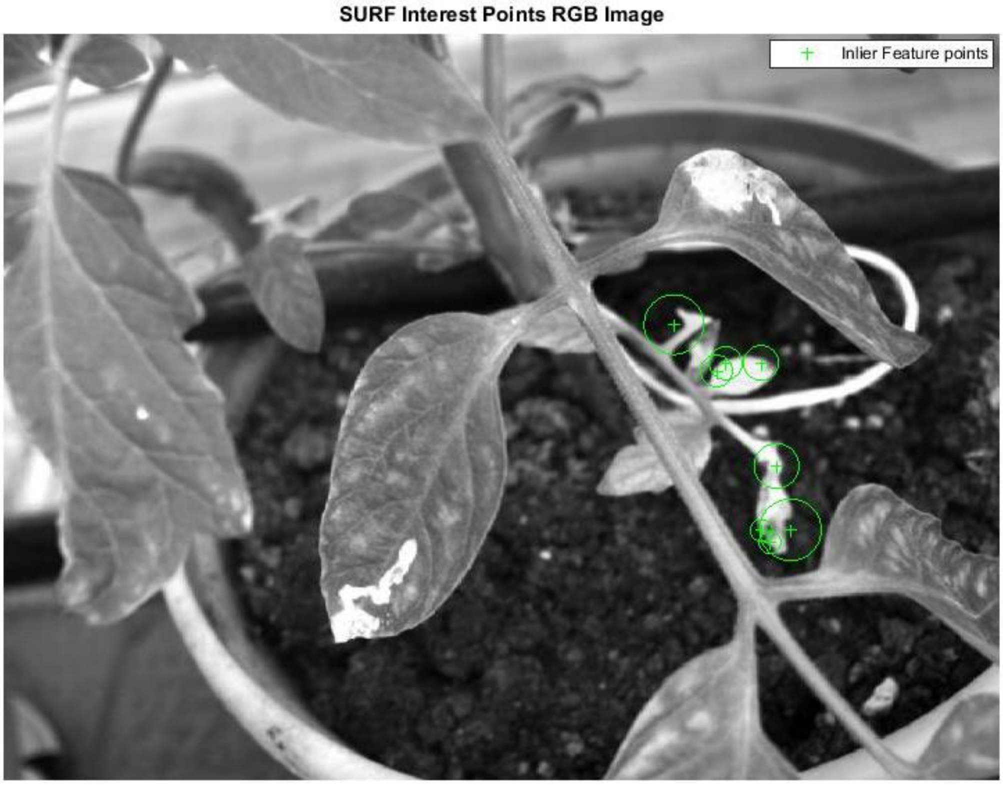


Figure 5-29: Inlier Feature points in RGB after removing outliers

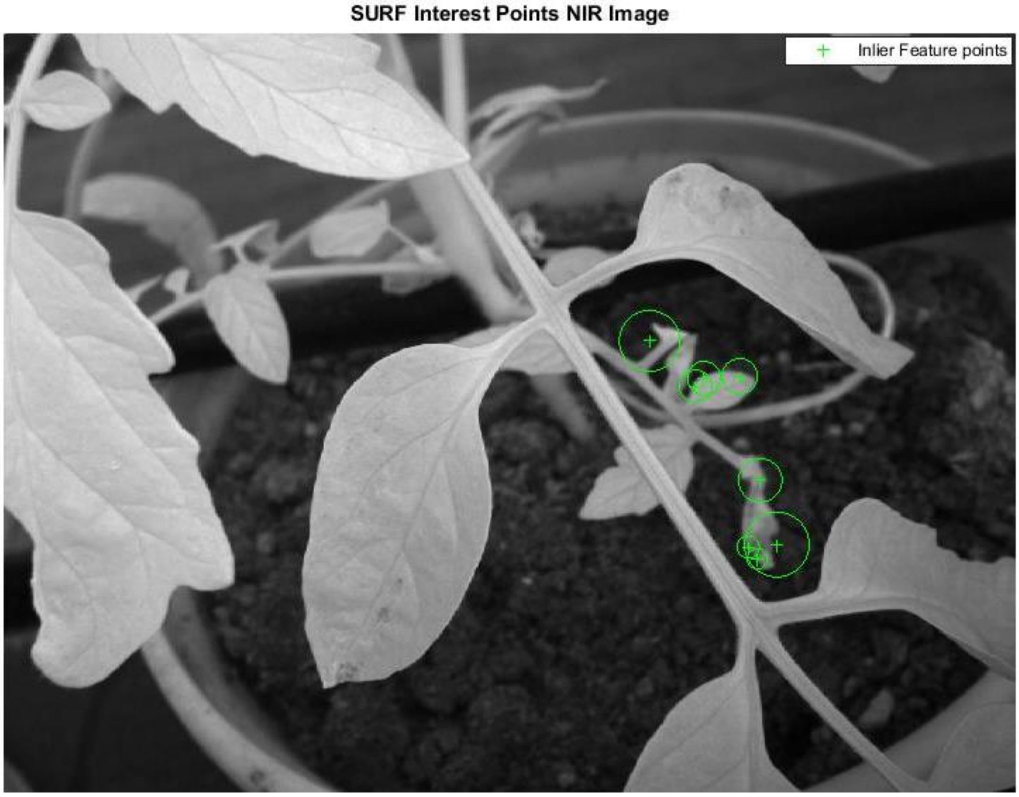


Figure 5-30: Inlier Feature points in NIR after removing outliers

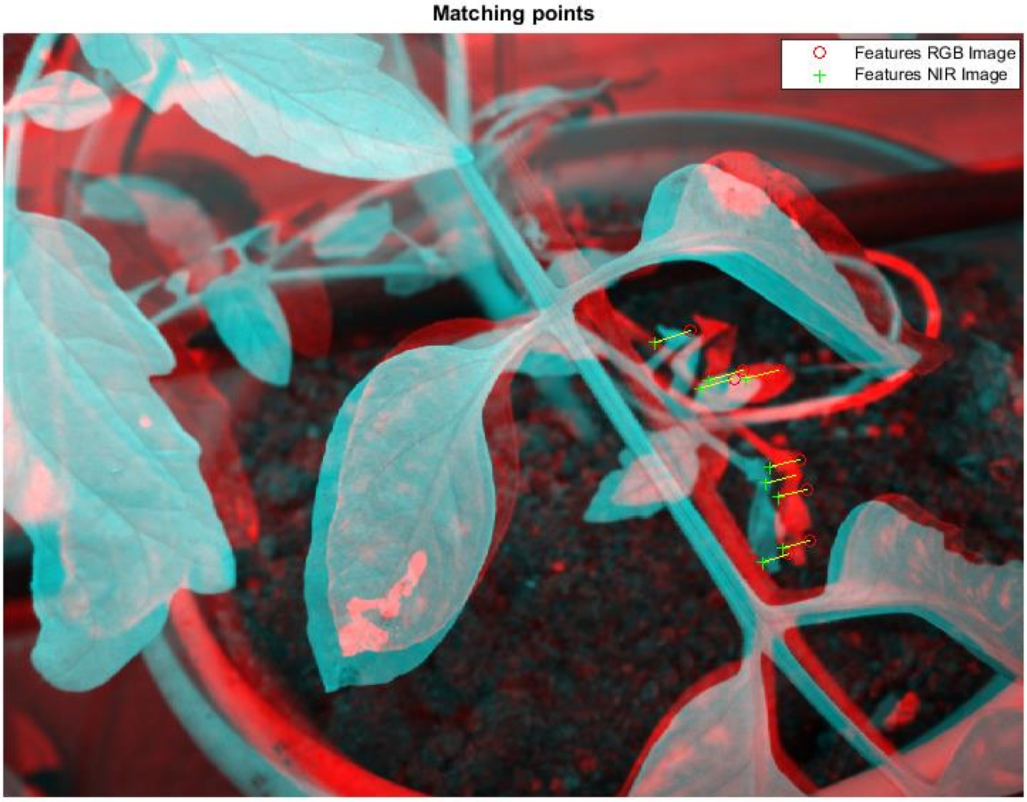


Figure 5-31: Matching Features in RGB and NIR images

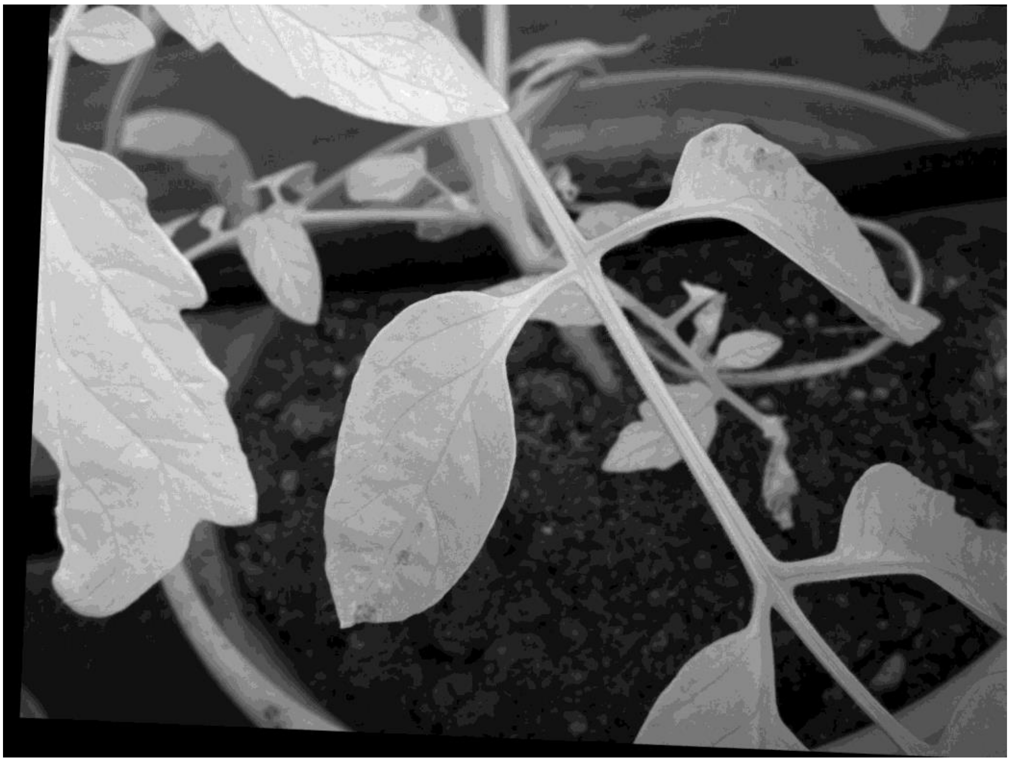


Figure 5-32: NIR image after translation

5.4.2. Background Removal

Background removal used here is completed using k-means segmentation as shown in Sec. 4.3.3. Because of the complicated background, traditional thresholding techniques is not suitable to be used in this system. Background removal is completed for both the RGB image and NIR image. After multiple trials, lower values of k showed to be more effective in this case as the scenery is more complex. Therefore, the value of k was chosen as 3. As seen cluster 1 illustrated in Figure 5-33 represents the foreground where the leaves are represented accurately because of the uniformity of the color. However, some of the information is lost due to the high variance in color and the complexity of the background. Cluster 2 shown in Figure 5-34 and cluster 3 shown in Figure 5-35 represent the background. These two clusters are used to remove the background from the NIR image.

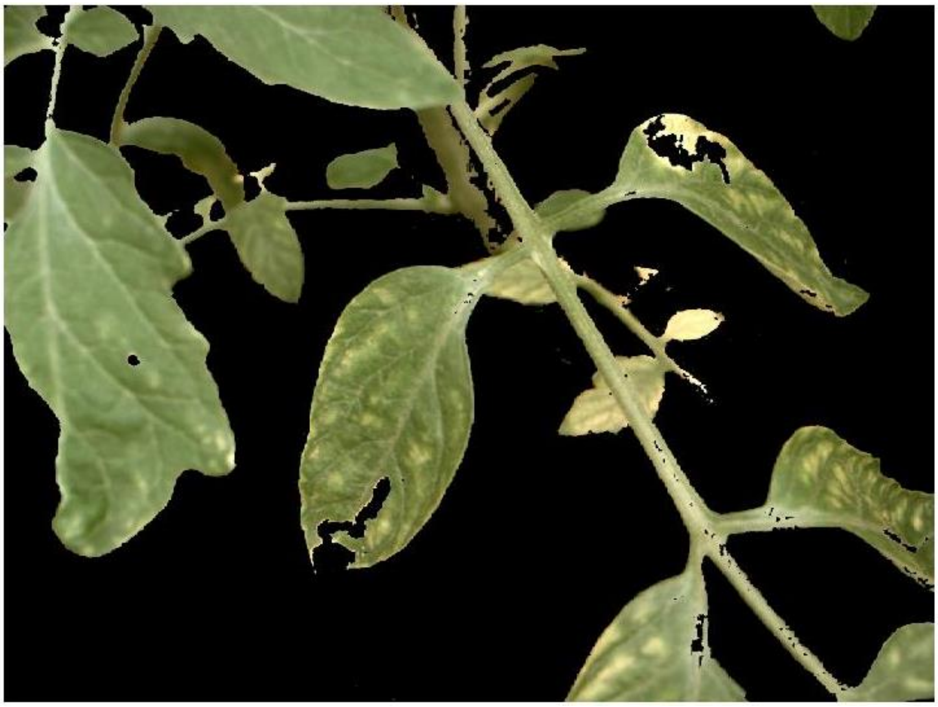


Figure 5-33: Cluster 1

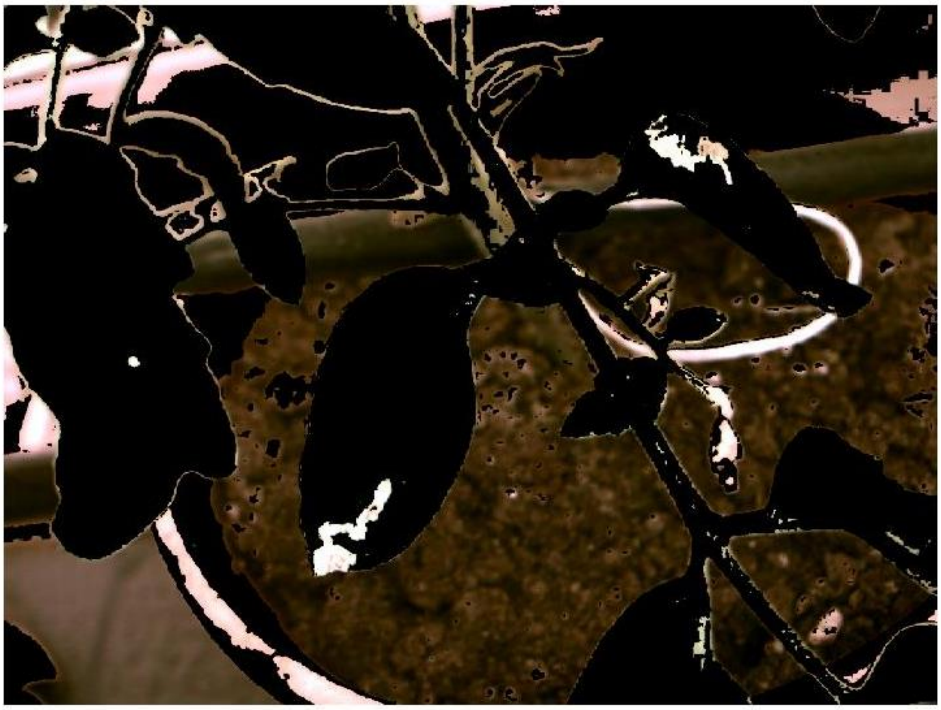


Figure 5-34: Cluster 2

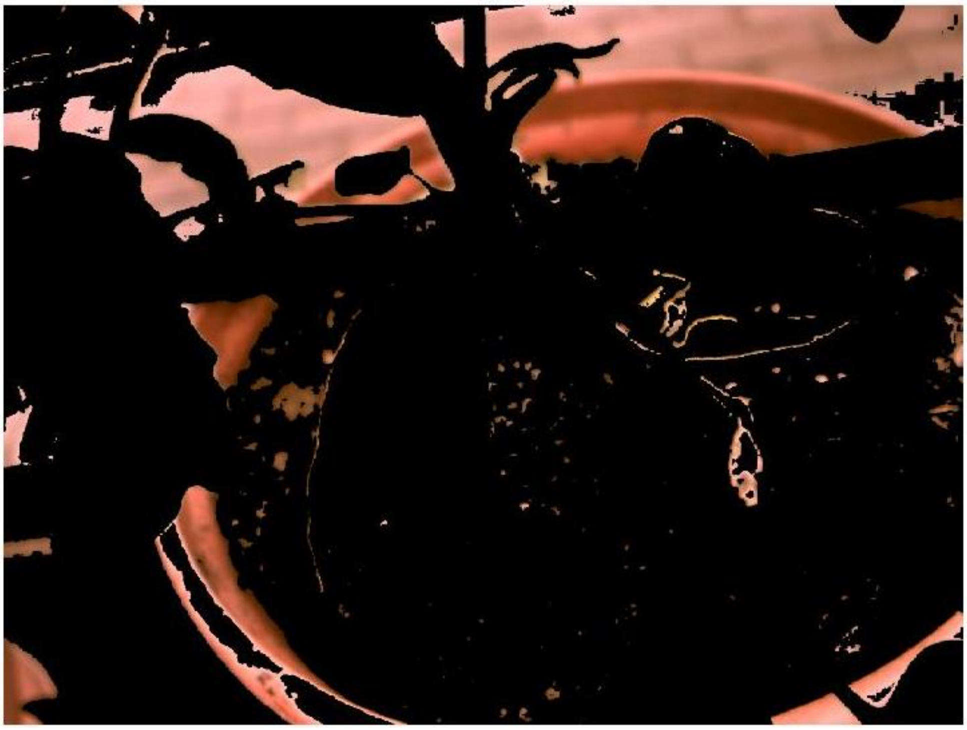


Figure 5-35: Cluster 3

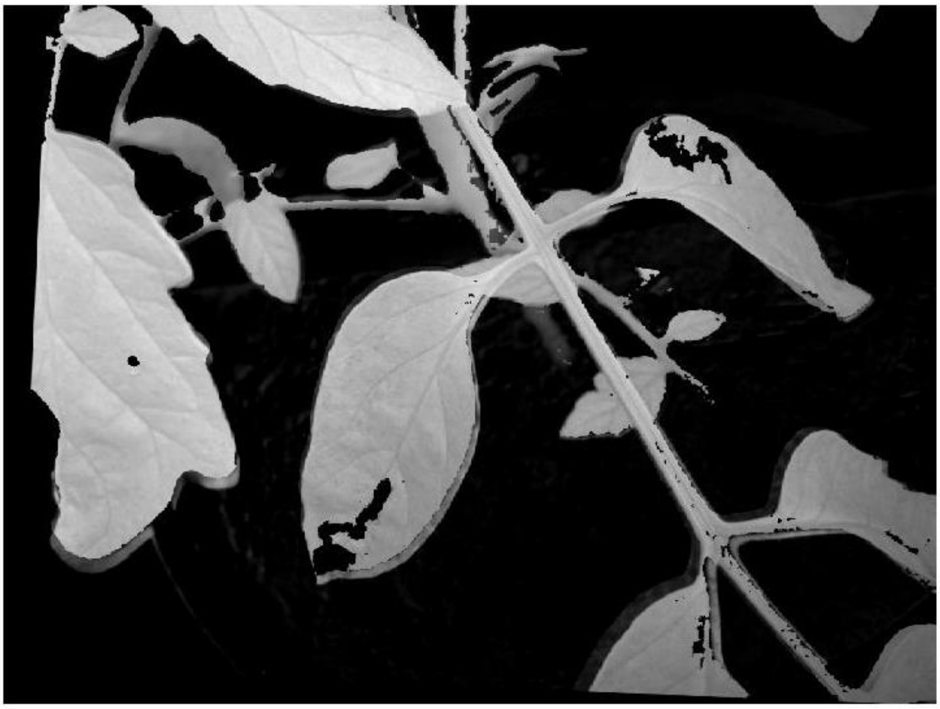


Figure 5-36: NIR image after background removal

Now that the background of both images has been removed the NDVI can be calculated accurately. The mean value of the NDVI values is found as 0.2977 which corresponds to a relatively high NDVI value and indicates to lower foliar activity.

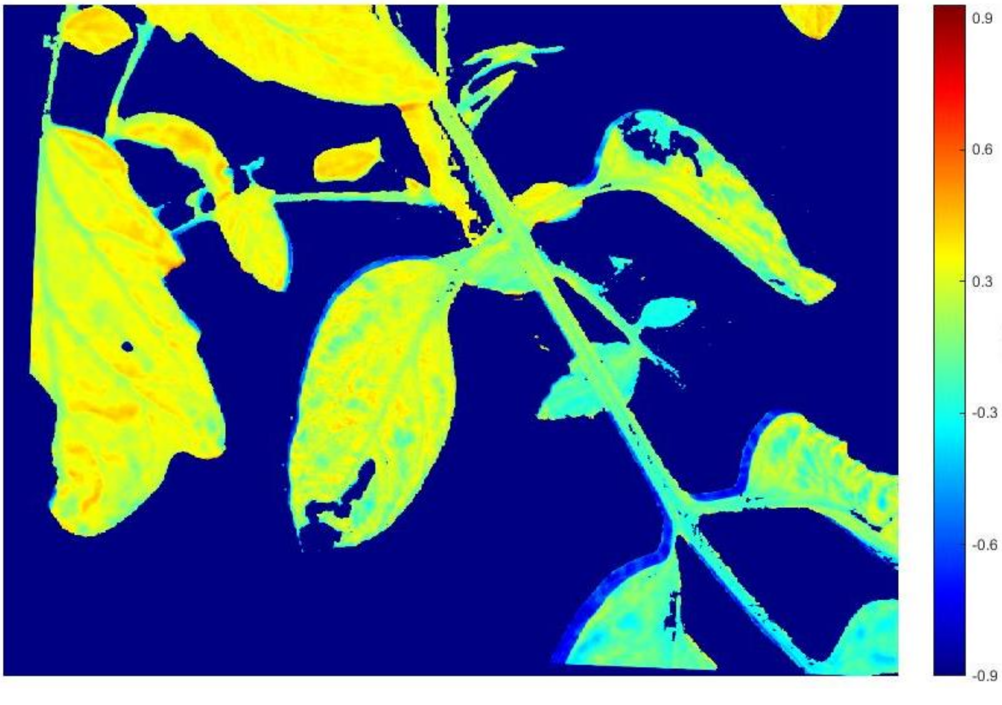


Figure 5-37: NDVI map

5.5. Testing and Results

The robot is tested in a greenhouse environment on tomato plants that are grown in pots. The test pots range from active and healthy plants to unhealthy plants and underdeveloped growth. Figure 5-38 and Figure 5-39 illustrates how the robotic system is used in the greenhouse measuring tomato plant health. The angle of the camera rig can be adjusted via the monopod to position the cameras at an angle the user desires. For testing, the camera rig was positioned at an angle as seen in Figure 5-40 and Figure 5-41. The robot moves across the field taking images of plant pots using both the RGB and NIR cameras for NDVI calculation and health mapping.

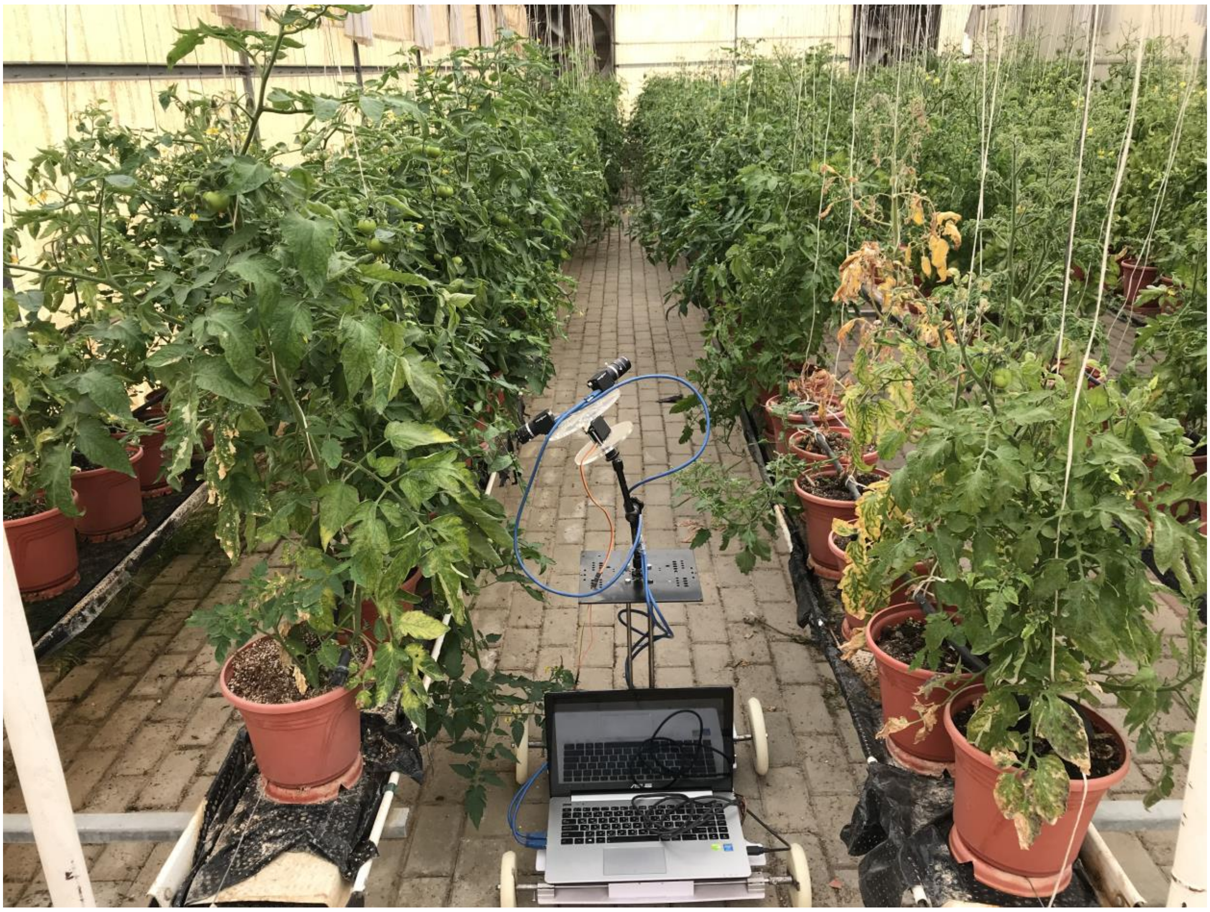


Figure 5-38: Robotic System in tomato plant greenhouse

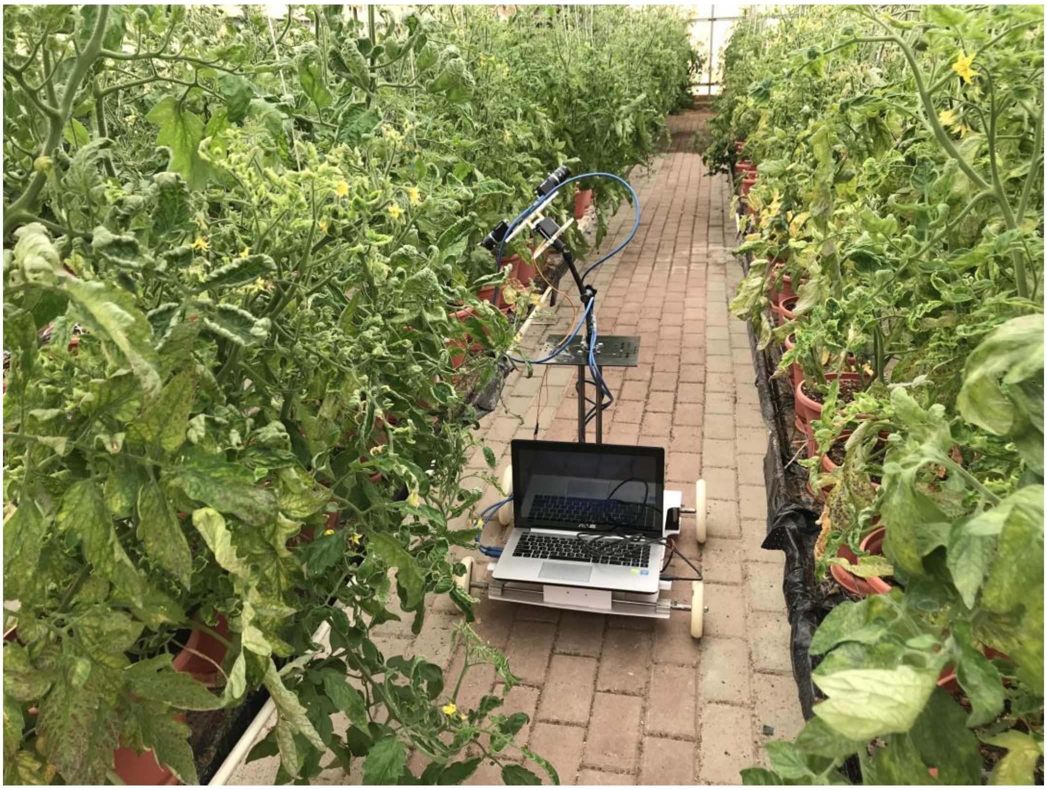


Figure 5-39: Robotic System in tomato plant greenhouse

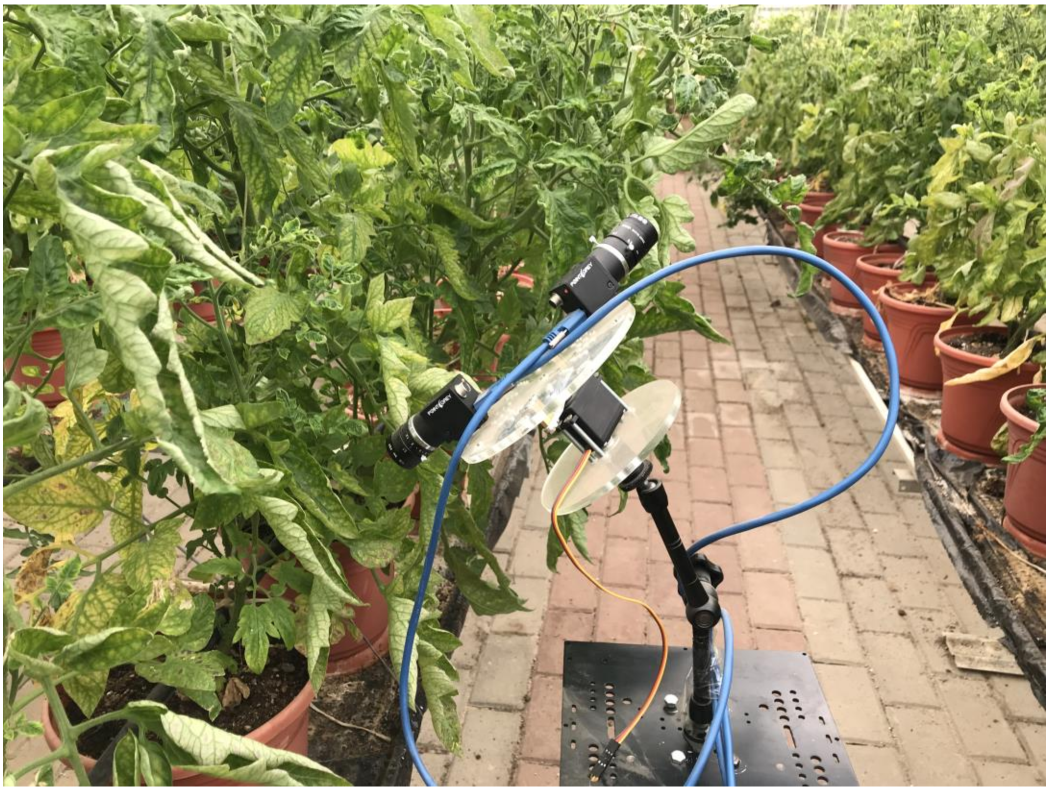


Figure 5-40: Camera rig positioned at an angle

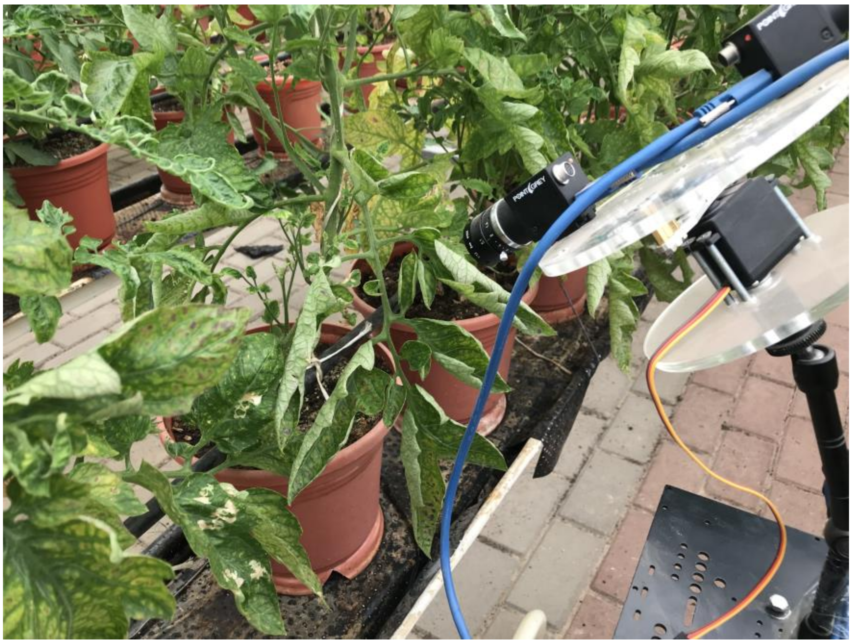


Figure 5-41: Capturing plant pot images

5.5.1. Underdeveloped plants

Multiple images are taken using the robot's cameras for cases with pots that have no sign of growth and the NDVI is used to validate this observation. Figure 5-42 and Figure 5-43 show the RGB image and NIR image after alignment of a pot with no plant growth. The NDVI map of this pot can be seen in Figure 5-44. It is clear from the map that the NDVI is very low with a mean value of 0.1826.

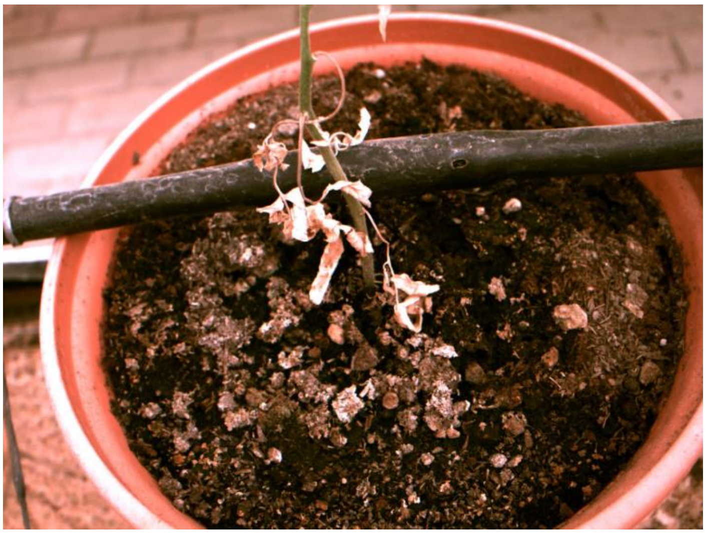


Figure 5-42: RGB Image of underdeveloped plants

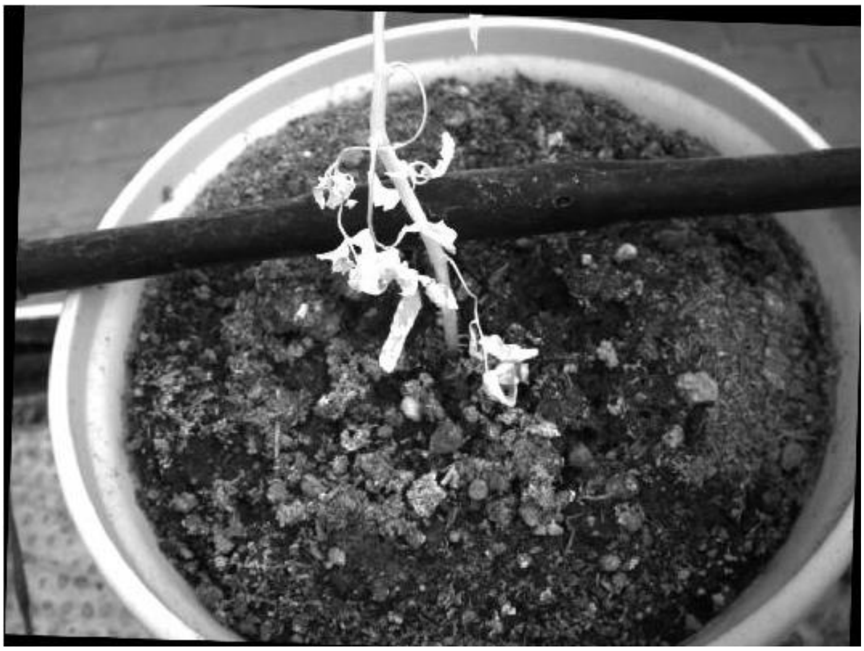


Figure 5-43: NIR Image of underdeveloped plants

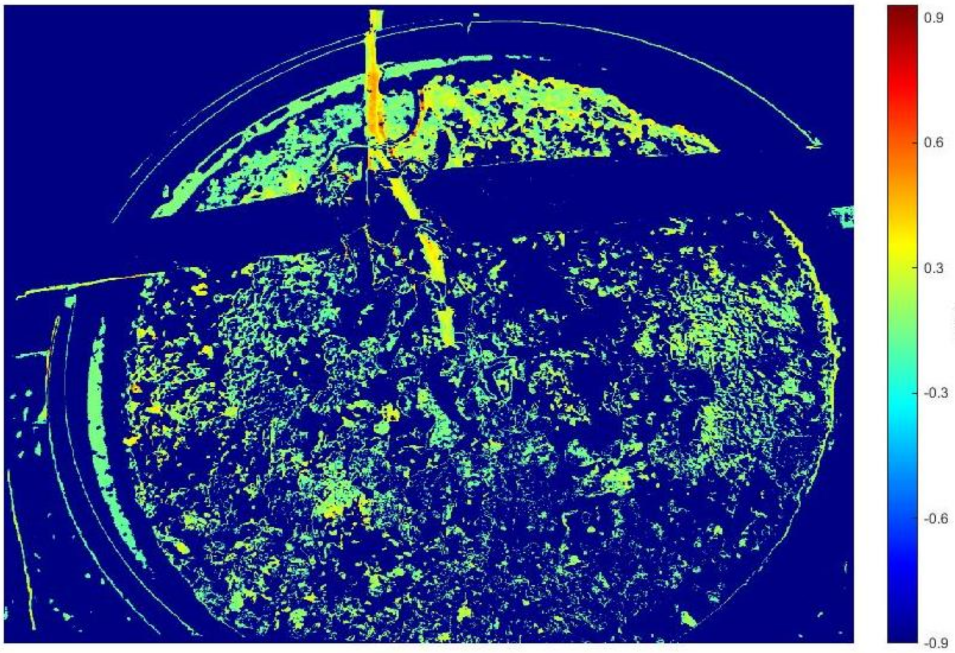


Figure 5-44: NDVI map of underdeveloped plants

5.5.2. Plants with Low Foliar Activity

Images are taken for cases with pots that have sign of unhealthy growth or low foliar activity. Figure 5-45 and Figure 5-46 show the RGB image and NIR image after alignment of a pot with unhealthy plant growth. The NDVI map of this pot can be seen in Figure 5-47. It is clear from the map that the NDVI is quite low with a mean value of 0.2072.

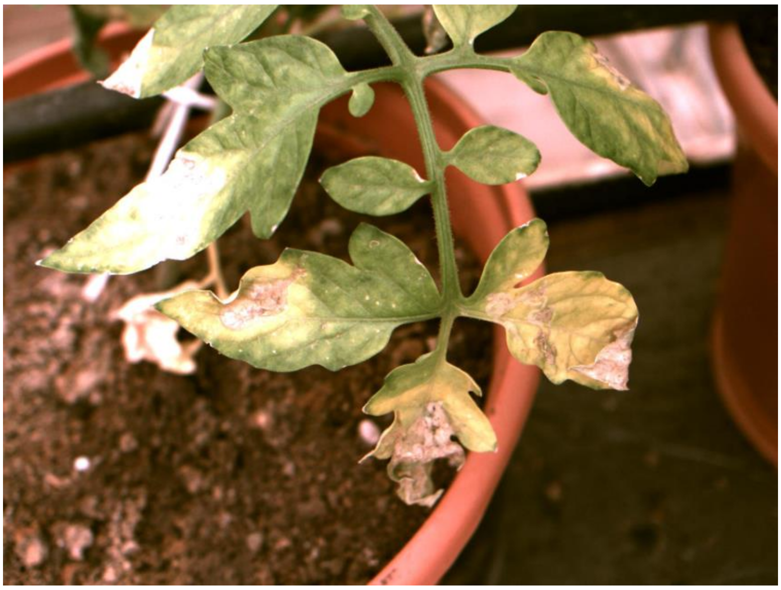


Figure 5-45: RGB image of unhealthy plants

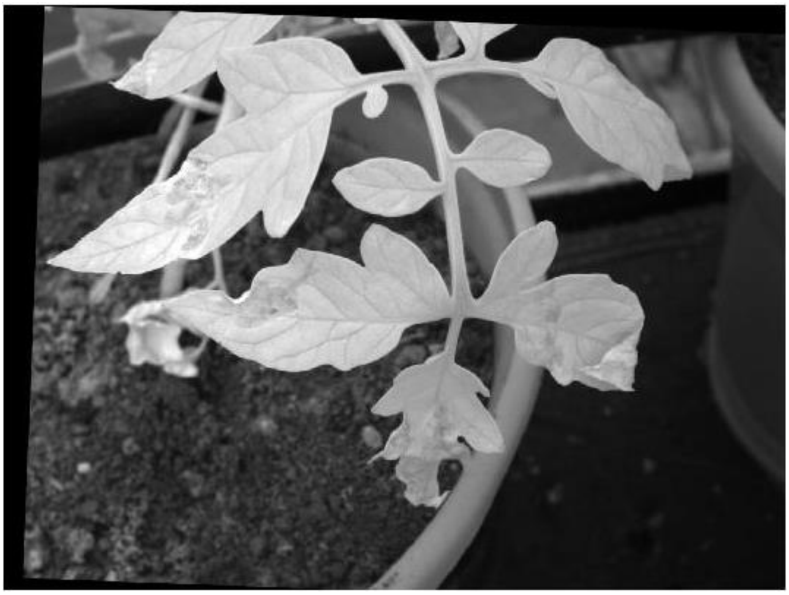


Figure 5-46: NIR image of unhealthy plants

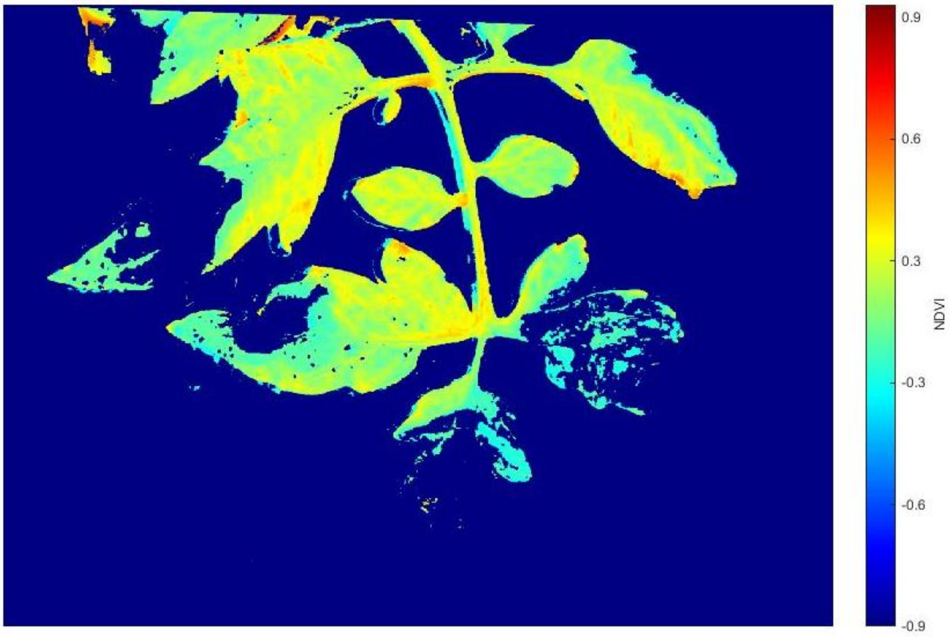


Figure 5-47: NDVI map of unhealthy plants

5.5.3. Plants with High Foliar Activity

Images are taken for cases with pots that have signs of healthy growth or high foliar activity. Figure 5-48 and Figure 5-49 show the RGB image and NIR image after alignment of a pot with high foliar activity. The NDVI map of this pot can be seen in Figure 5-50. It is clear from the map that the NDVI is relatively high when compared to underdeveloped plants or unhealthy plants with a mean value of 0.3679.

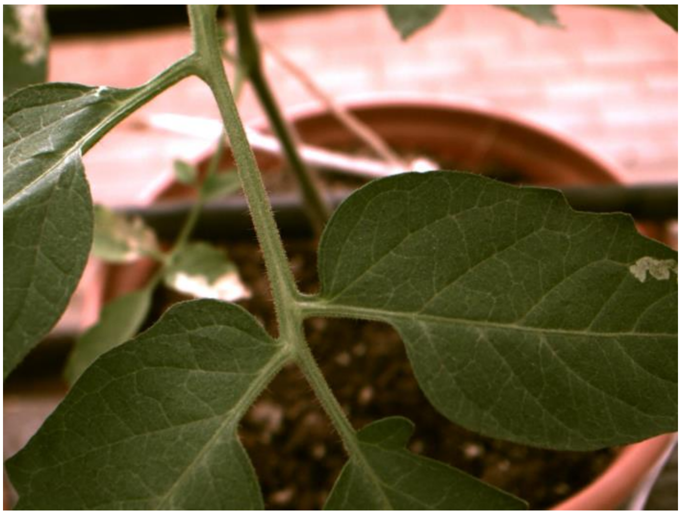


Figure 5-48: RGB image of healthy plants

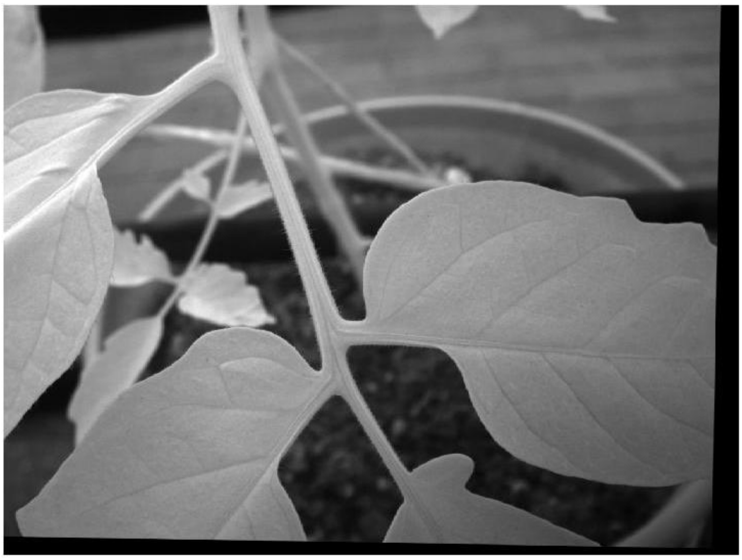


Figure 5-49: NIR image of healthy plants

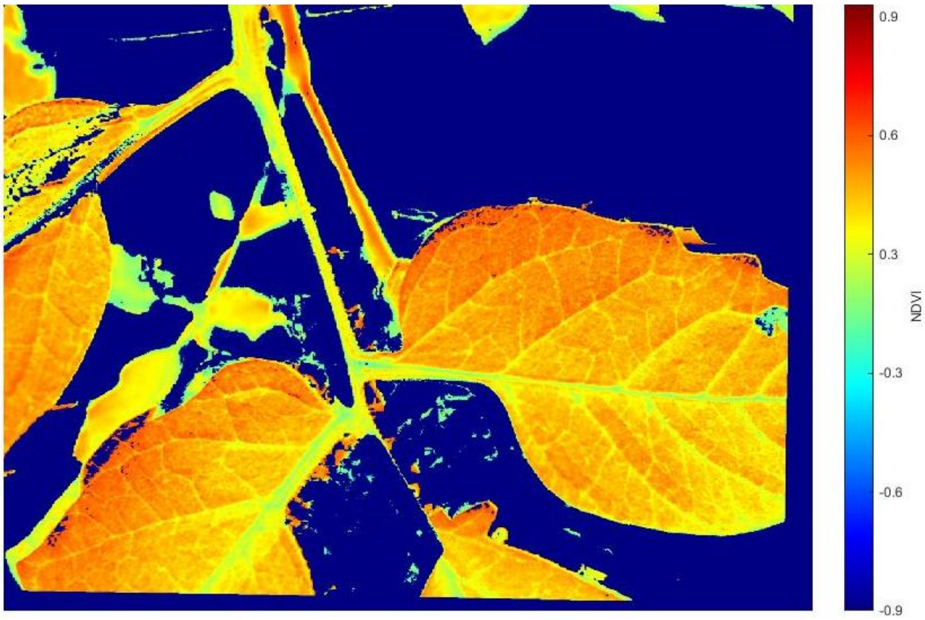


Figure 5-50: NDVI map of healthy plants

It can be concluded that plant regions or pots that have NDVI values lower than 0.2 are considered underdeveloped with very little sign of plant growth. While NDVI values between 0.2 to 0.3 show low foliar activity and values above 0.3 signify plants with high foliar activity.

5.5.4. Location and Mapping

The robot is tested across a tomato plant line that consists of 18 pots, in which each pot is considered a segment for location mapping. The results of each segment NDVI calculation is illustrated in Table 5-6. The NDVI values are measured and plotted so the locations of each pot and its health can be observed. A map is created indicating the status of different pot locations, whether the plants are underdeveloped, unhealthy or healthy. Figure 5-51 illustrates how each segment is categorized in one of 3 categories depending on its NDVI value. Plants with NDVI values lower than 0.2 are considered underdeveloped, while values between 0.2 and 0.3 are considered unhealthy. Healthy plants have NDVI values above 0.3.

Table 5-6: NDVI results for tomato plant line

Location segment	Mean NDVI	Health Classification
1	0.1911	Underdeveloped
2	0.1826	Underdeveloped
3	0.1737	Underdeveloped
4	0.1955	Underdeveloped
5	0.2133	Unhealthy
6	0.2756	Unhealthy
7	0.2072	Unhealthy
8	0.1747	Underdeveloped
9	0.2919	Unhealthy
10	0.2848	Unhealthy
11	0.3316	Healthy
12	0.2548	Unhealthy
13	0.2977	Unhealthy
14	0.3927	Healthy
15	0.2641	Unhealthy
16	0.3679	Healthy
17	0.2533	Unhealthy
18	0.2649	Unhealthy

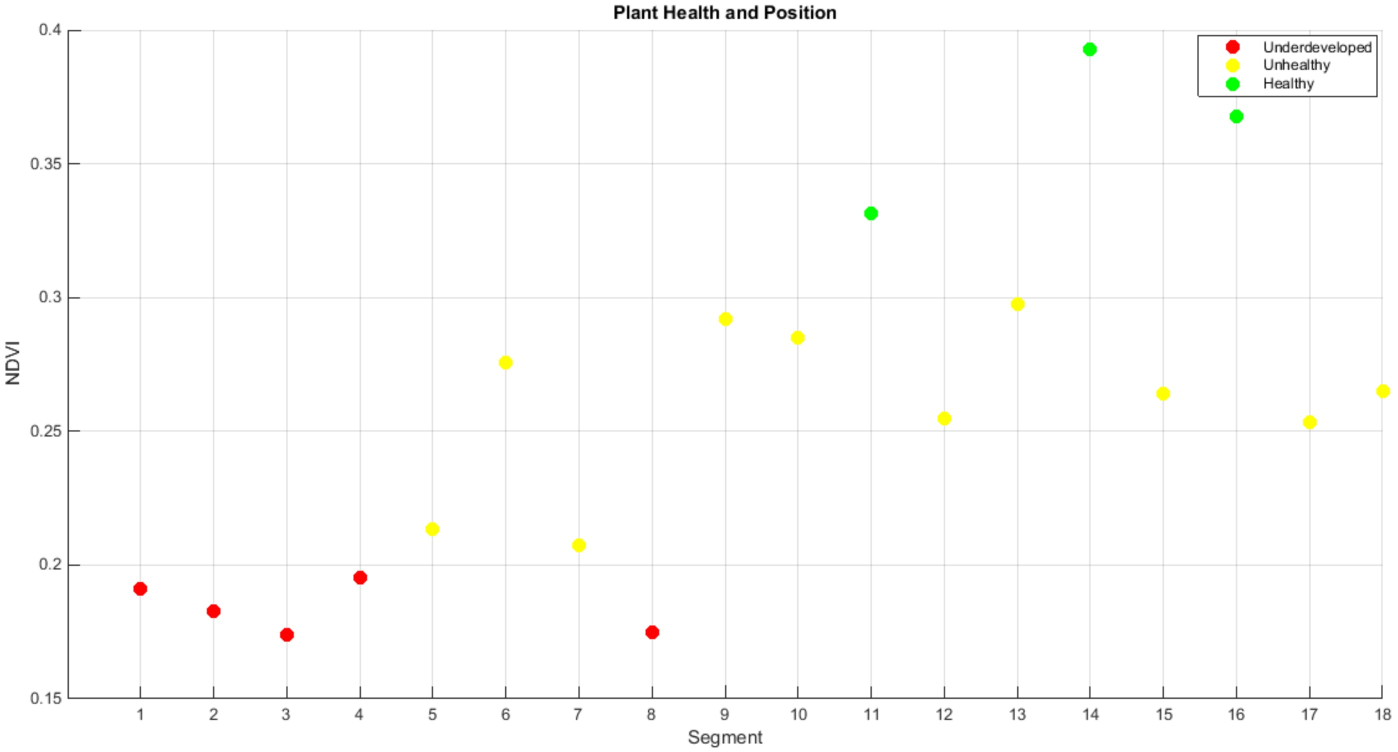


Figure 5-51: Plant health and segment location

5.5.5. Results Discussion

The robotized system results are compared to human observation to validate the results. Overall, the health map gives a good indication on the plant health of different pots, but in some cases the NDVI can give contradicting results. Figure 5-52 shows plant pot 5 with very high foliar activity, however the NDVI map shown in Figure 5-53 indicates to lower foliar activity with a mean NDVI of 0.2133.

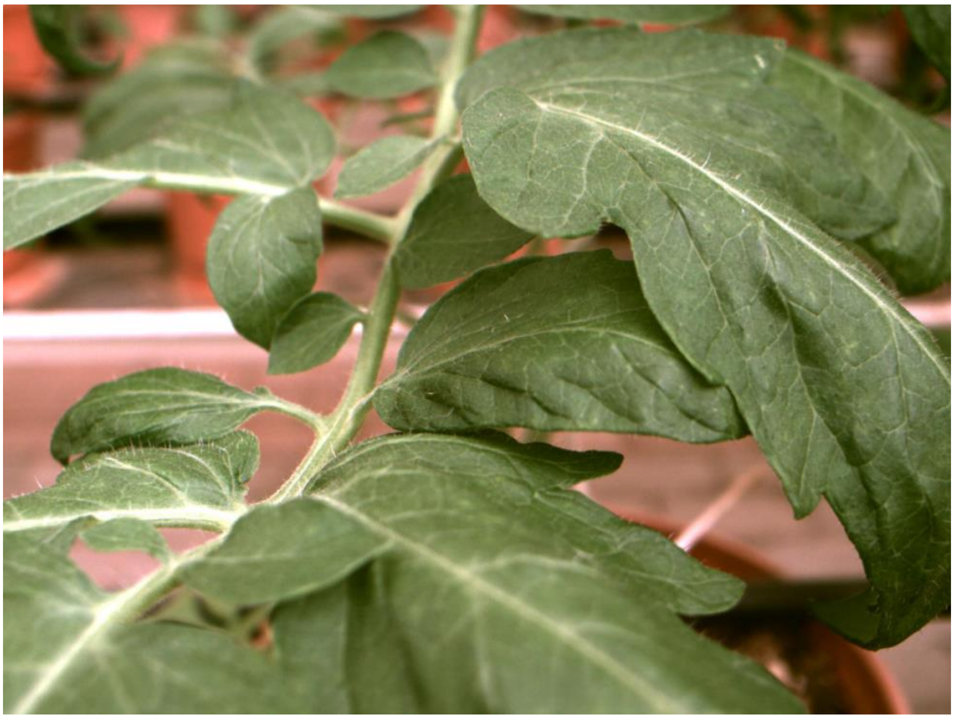


Figure 5-52: Plant pot 5

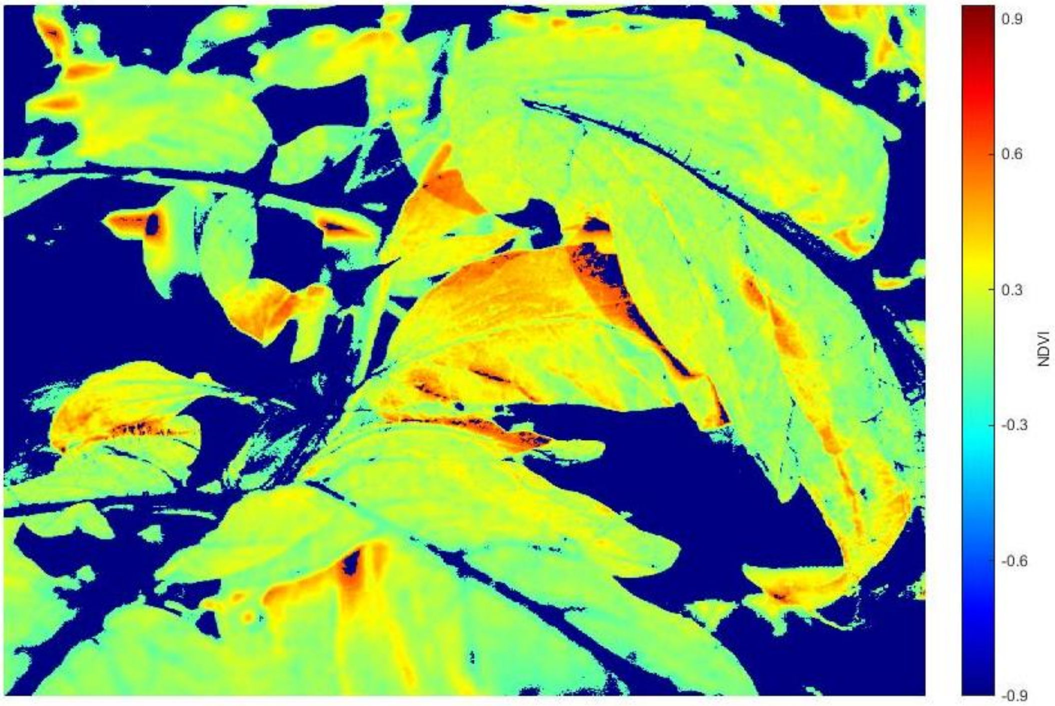


Figure 5-53: NDVI map of plant pot 5

Also using k-means clustering to segment the background showed diverse results. In some cases, segments of the background are clustered with the foreground. Figure 5-54 shows plant pot 9 and Figure 5-55 displays the output of the background segmentation. Improper segmentation is achieved here as areas of the soil are segmented with the leaf area. This leads to inaccurate calculation of the NDVI value as seen in Figure 5-56. Therefore, the health grade is prone to misclassification.

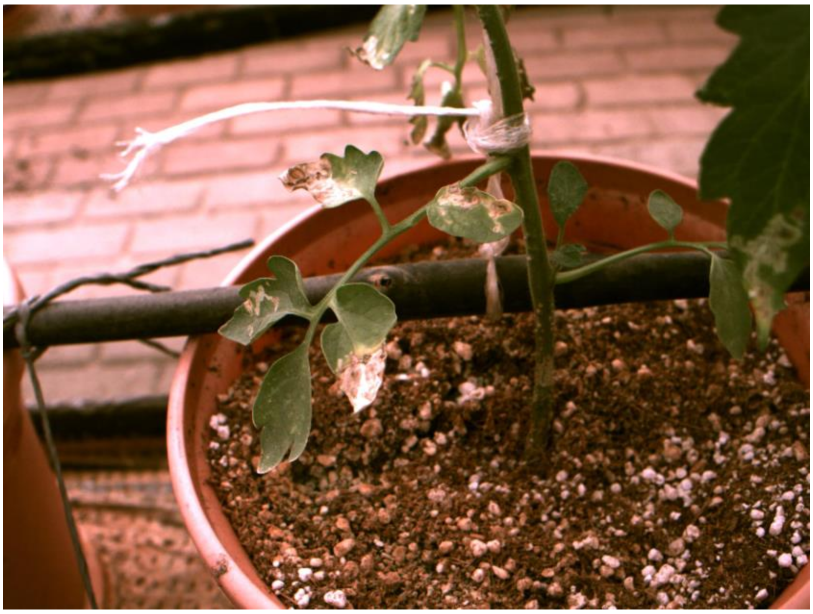


Figure 5-54: Plant pot 9

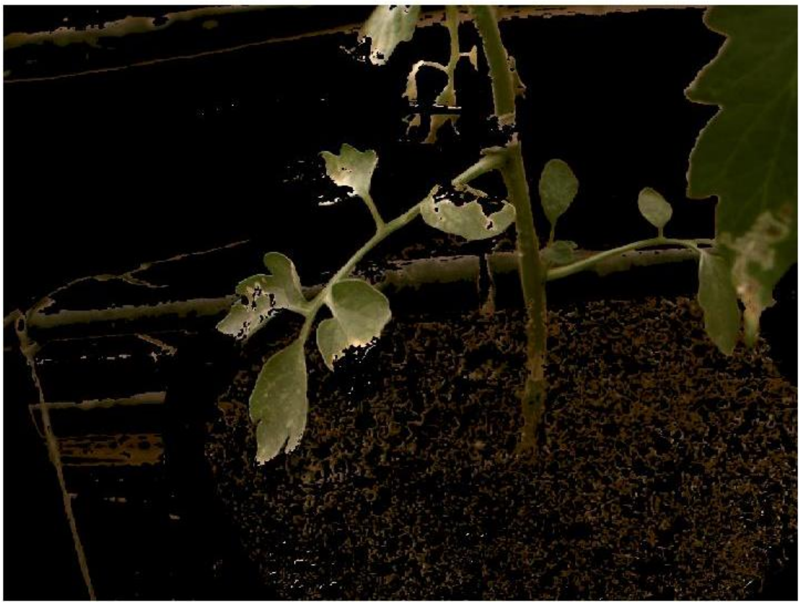


Figure 5-55: Plant pot 9 after background removal

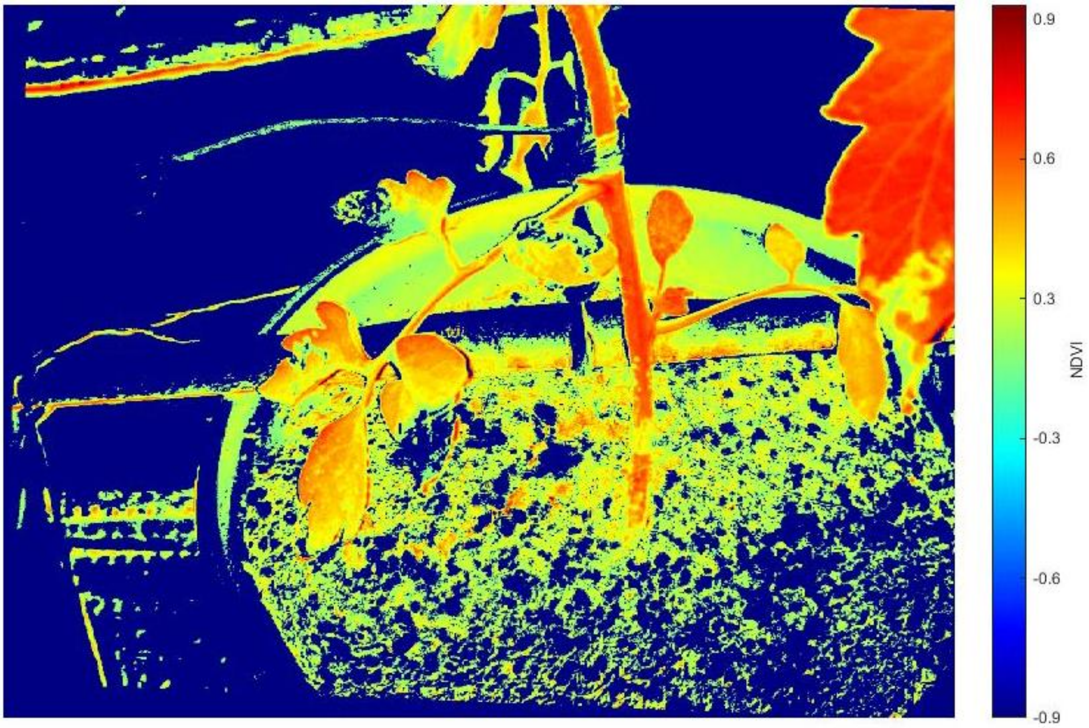


Figure 5-56: NDVI map of plant pot 9

Overall this system achieved a health classification accuracy of 83%. From the 18 plant pots, 3 were misclassified because of incorrect background segmentation and NDVI calculation. These results are comparable with [8] and [19], which include automated approaches for field based disease and health classification in preharvest applications. However, this system has the advantage of being fully automated and operating under uncontrolled environments which is not achieved in [8] and [19].

Chapter 6
Conclusions and Future Work

6.1. Conclusions

The developed preharvest disease detection systems with Offline and Online techniques showed to be successful in achieving the stated objectives to a high extent. The Offline plant leaf disease severity system proved to be very effective in segmenting and locating infected areas of leaf samples. Using a fuzzy logic classification technique to categorize disease severity proved to be very efficient and more reliable than relying on Table 2-1. The system was tested using plant leaf images of varying resolutions taken from a real site and achieved high background segmentation accuracies and disease classification. Because of the use of k-means clustering and triangle thresholding, background segmentation results were very robust against light variances. Therefore, this system can be used to test plant samples in uncontrolled lighting conditions and not only in laboratories. Leaf disease severity classification using Fuzzy Logic was very promising with a disease grade classification accuracy of 98%. Early identification of leaf disease severity can aid farmers into taking prompt actions to avoid disease spreading. Also, the system can be used to help monitor the changes in leaf severity over time, to ensure healthy growth. Overall the system devised is very user friendly and can successfully detect and measure disease severity of leaf plants, while saving time and effort. The system was integrated into a friendly GUI to help users such as pathologists to measure disease severity level in leaves easily.

The testing and results of the implemented robotized online system showed to be effective in providing vital health information for plants in real time under field conditions with an accuracy of 83%. Plants of varying health can be located and distinguished from one another according to health grade. Plants that have slow to underdeveloped growth have significantly lower NDVI values compared to unhealthy and healthy plants. The overall system has the capabilities of moving across a plant field and effective in categorizing plant health according to NDVI values. A map of plant locations and health is successfully implemented to help farmers distinguish healthy regions from unhealthy regions for early and immediate action to help revive plants. Nutrient deficient plants can be detected and categorized at preliminary stages of plant growth for controlled nutrient application. Plant stress and nutrient deficiencies can be measured according to NDVI map, and can be used for appropriate actions accordingly. The robotized system is a cost-effective solution to automated plant health monitoring. Selective pesticide spraying can also be achieved in this system as mapping of unhealthy plants can indicate to insect or viral infestations. Areas are distinguished from one another according to plant health so that farmers can focus on worrying areas, saving time and effort.

6.2. Future Work

As for future work, the following points can be considered:

- The Offline system can be used to train machine learning algorithms, such as Neural Networks and Support Vector Machines. Which can be implemented to automatically identify the diseased area.
- For the Online system, a more robust robot could be implemented to have the capabilities of maneuvering in uneven agriculture terrain such as sand and rock.
- Multispectral and hyperspectral cameras could be integrated into the robot for higher sensitivity in the infrared range for higher accuracy in vegetative index calculations.
- Use of a single hyperspectral camera to avoid alignment issues that may occur when using multiple cameras.
- More robust background segmentation technique to enhance accuracy of foreground representation of plant to increase NDVI accuracy.
- Integration of greenhouse nutrient management system for automatic water and nutrient irrigation.
- The robot can include a robotic arm for automatic and selective pesticide spraying.
- Robot integration with VOC profiling sensors for early indication on plant stress and possible diseases.

References

[1] M. Jhuria, A. Kumar and R. Borse, "Image processing for smart farming: Detection of disease and fruit grading," in *Image Information Processing (ICIIP), 2013 IEEE Second International Conference,* 2013.

[2] S. R. Dubey and A. S. Jalal, "Detection and classification of apple fruit diseases using complete local binary patterns," in *Computer and Communication Technology (ICCCT), 2012 Third International Conference,* 2012.

[3] D. Pimentel, R. Zuniga and D. Morrison, "Update on the environmental and economic costs associated with alien-invasive species in the United States," *Ecological Economics,* vol. 52, *(3),* pp. 273-288, 2005.

[4] S. Sankaran, "A review of advanced techniques for detecting plant diseases," *Computer Electronics in Agriculture,* vol. 72, *(1),* pp. 1-13, 2010.

[5] M. M. López, "Innovative tools for detection of plant pathogenic viruses and bacteria," *International Microbiology,* vol. 6, *(4),* pp. 233-243, 2003.

[6] Q. Li, M. Wang and W. Gu, "Computer vision based system for apple surface defect detection," *Computer Electronics in Agriculture,* vol. 36, *(2),* pp. 215-223, 2002.

[7] S. Sannakki, "Leaf Disease Grading by Machine Vision and Fuzzy Logic," *International Journal of Computer Technology and Applications,* vol. 2, *(5),* pp. 1709-1716, 2011.

[8] R. Oberti, "Selective spraying of grapevines for disease control using a modular agricultural robot," *Biosystems Engineering,* vol. 146, pp. 203-215, 2016.

[9] J. Belasque Jr, M. Gasparoto and L. G. Marcassa, "Detection of mechanical and disease stresses in citrus plants by fluorescence spectroscopy," *Applied Optics,* vol. 47, *(11),* pp. 1922-1926, 2008.

[10] L. G. Marcassa, "Fluorescence spectroscopy applied to orange trees," *Laser Physics,* vol. 16, *(5),* pp. 884-888, 2006.

[11] C. Bravo, "Early disease detection in wheat fields using spectral reflectance," *Biosystems Engineering,* vol. 84, *(2),* pp. 137-145, 2003.

[12] F. Spinelli, M. Noferini and G. Costa, "Near infrared spectroscopy (NIRs): Perspective of fire blight detection in asymptomatic plant material," *International Workshop on Fire Blight 704,* 2004.

[13] R. A. Naidu, "The potential of spectral reflectance technique for the detection of Grapevine leafroll-associated virus-3 in two red-berried wine grape cultivars," *Computer Electronics in Agriculture,* vol. 66, *(1),* pp. 38-45, 2009.

[14] B. Chen, "Spectrum Characteristics of Cotton Canopy Infected with Verticillium Wilt and Inversion of Severity Level," vol. 259, pp. 1169-1180, 2008.

[15] L. Chaerle, "Multicolor fluorescence imaging for early detection of the hypersensitive reaction to tobacco mosaic virus," *Journal of Plant Physiology,* vol. 164, *(3),* pp. 253-262, 2007.

[16] C. Bravo, "Foliar disease detection in the field using optical sensor fusion," *Agriculture Engineering International,* vol. 6, pp. 1-14, 2004.

[17] D. Moshou, "Automatic detection of 'yellow rust' in wheat using reflectance measurements and neural networks," *Computer Electronics in Agriculture,* vol. 44, *(3),* pp. 173-188, 2004.

[18] J. S. West, "The potential of optical canopy measurement for targeted control of field crop diseases," *Annual Review of Phytopathology,* vol. 41, *(1),* pp. 593-614, 2003.

[19] H. Z. Shafri and N. Hamdan, "Hyperspectral Imagery for Mapping Disease Infection in Oil Palm PlantationUsing Vegetation Indices and Red Edge Techniques," *American Journal of Applied Sciences,* vol. 6, *(6),* pp. 1031, 2009.

[20] A. Kumar, "Citrus greening disease detection using aerial hyperspectral and multispectral imaging techniques," *Journal of Applied Remote Sensing,* vol. 6, *(1),* pp. 063542-1-063542-22, 2012.

[21] J. Qin, "Detection of citrus canker using hyperspectral reflectance imaging with spectral information divergence," *Journal of Food Engineering,* vol. 93, *(2),* pp. 183-191, 2009.

[22] A. Vallat, H. Gu and S. Dorn, "How rainfall, relative humidity and temperature influence volatile emissions from apple trees in situ," *Phytochemistry,* vol. 66, *(13),* pp. 1540-1550, 2005.

[23] J. M. Cevallos-Cevallos, R. Rouseff and J. I. Reyes-De-Corcuera, "Untargeted metabolite analysis of healthy and Huanglongbing-infected orange leaves by CE-DAD," *Electrophoresis,* vol. 30, *(7),* pp. 1240-1247, 2009.

[24] H. Zhang and J. Wang, "Detection of age and insect damage incurred by wheat, with an electronic nose," *Journal of Stored Products Research,* vol. 43, *(4),* pp. 489-495, 2007.

[25] B. Prithiviraj, "Volatile metabolite profiling for the discrimination of onion bulbs infected by Erwinia carotovora ssp. carotovora, Fusarium oxysporum and Botrytis allii," *European Journal of Plant Pathology,* vol. 110, *(4),* pp. 371-377, 2004.

[26] A. Kushalappa, "Volatile fingerprinting (SPME-GC-FID) to detect and discriminate diseases of potato tubers," *Plant Disease,* vol. 86, *(2),* pp. 131-137, 2002.

[27] L. Lui, "Discrimination of three fungal diseases of potato tubers based on volatile metabolic profiles developed using GC/MS," *Potato Research,* vol. 48, *(1-2),* pp. 85-96, 2005.

[28] S. B. Patil and S. K. Bodhe, "Leaf disease severity measurement using image processing," *International Journal of Engineering and Technology,* vol. 3, *(5),* pp. 297-301, 2011.

[29] G. Xu, "Use of leaf color images to identify nitrogen and potassium deficient tomatoes," *Pattern Recognition Letters,* vol. 32, *(11),* pp. 1584-1590, 2011.

[30] A. Camargo and J. Smith, "Image pattern classification for the identification of disease causing agents in plants," *Computer Electronics in Agriculture,* vol. 66, *(2),* pp. 121-125, 2009.

[31] T. Pobkrut and T. Kerdcharoen, "Soil sensing survey robots based on electronic nose," in *Control, Automation and Systems (ICCAS), 2014 14th International Conference,* 2014.

[32] J. Xing, "Detecting bruises on 'Golden Delicious' apples using hyperspectral imaging with multiple wavebands," *Biosystems Engineering,* vol. 90, *(1),* pp. 27-36, 2005.

[33] D. Unay and B. Gosselin, "Thresholding-based segmentation and apple grading by machine vision," *Signal Processing Conference, 2005 13th European,* 2005.

[34] B. J. Lindbloom, "Accurate color reproduction for computer graphics applications," in *ACM SIGGRAPH Computer Graphics,* 1989.

[35] R. S. Hunter, "Photoelectric color difference meter," *Journal of the Optical Society of America,* vol. 48, *(12),* pp. 985-995, 1958.

[36] D. Maia and R. Trindade, *Face Detection and Recognition in Color Images Under Matlab,* 2016.

[37] J. Schanda, *Colorimetry: Understanding the CIE System,* 2007.

[38] A. Greensted, *Otsu Thresholding - The Lab Book Pages, 2002.*

[39] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,* 1967.

[40] N. Efford, *Digital Image Processing: A Practical Introduction using Java,* 2000.

[41] W. Wong, *Omnidirectional Thermal Imaging Surveillance System Featuring Trespasser and Faint Detection,* 2011.

[42] D. Barolet, F. Christiaens and M. Hamblin, *Infrared and Skin: Friend Or Foe,* 2015.

[43] J. T. Pedersen, "Study group SURF: Feature detection & description," *Department of Computer Science, Aarhus University,* 2011.

[44] Point Grey Machine Vision [Online]. Available: https://www.ptgrey.com/ [Accessed: 15- Oct -2016].

[45] Normalized Difference Vegetation Index (NDVI) [Online]. Available: https://earthobservatory.nasa.gov/Features/MeasuringVegetation/measuring_vegetation_2.php [Accessed: 2- Dec - 2016].

[46] NDVI - Vegetation Index [Online], 2015, Available: http://www.wikiagro.com/en/NDVI_-_Vegetation_Index [Accessed: 4- Dec- 2016].

[47] Ultrasonic Distance Sensor [Online], Available: http://arduino-info.wikispaces.com/Ultrasonic+Distance+Sensor [Accessed: 22- Nov- 2017].

Appendices

A. L*a*b Color Space Conversion

Equations A.1 to A.4 are used to convert from sRGB color space to CIE 1931 and then to L*a*b [37]. To compute the formula, the sRGB component values are first set in the range of 0 to 1. Which can be completed by dividing the R, G, and B values by 255.

$$C_{linear} = \begin{cases} \dfrac{C_{srgb}}{12.92} & , \quad C_{srgb} \leq 0.04045 \\ \left(\dfrac{C_{srgb} + a}{1 + a}\right)^{2.4} & , \quad C_{srgb} > 0.04045 \end{cases} \tag{A.1}$$

Where a = 0.055 and where C is R, G, B

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} \tag{A.2}$$

$$a = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \tag{A.3}$$

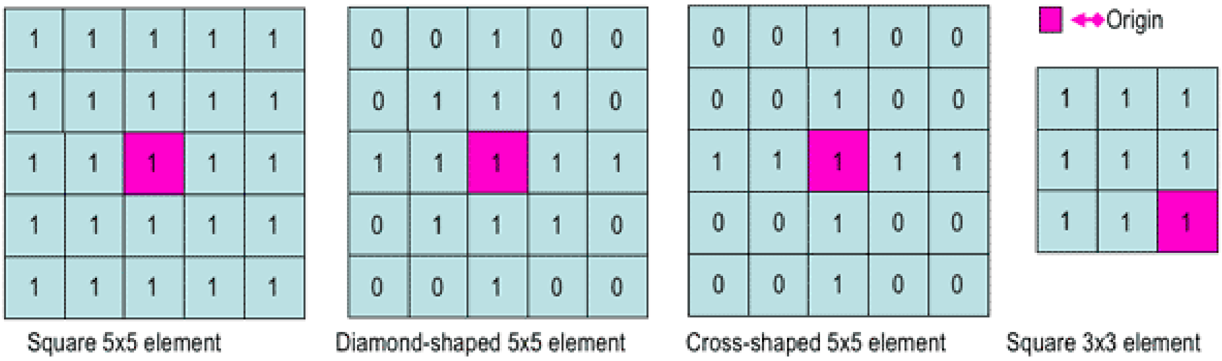
$$b = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \tag{A.4}$$

Where,

X_n, Y_n and Z_n are the CIE XYZ values of the reference white point

B. Image Morphology

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations rely on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological techniques probe an image with a small shape or template called a structuring element shown in Figure B.1. The structuring element is considered a small binary image, that is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. The value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image [40].



1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Square 5x5 element

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

Diamond-shaped 5x5 element

0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

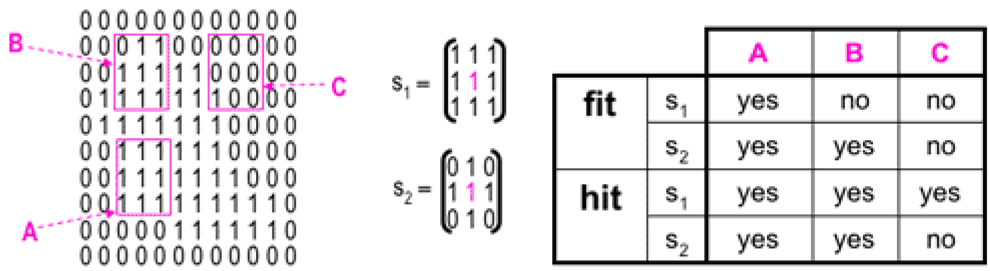
Cross-shaped 5x5 element

1	1	1
1	1	1
1	1	1

Square 3x3 element

Figure B.1: Examples of simple structuring elements [40]

When a structuring element is placed in a binary image it is said to fit the image if for each pixel set to 1, the corresponding image pixel is also set to 1. Similarly, a structuring element is said to hit an image if, at least for one of its pixels set to 1 the corresponding image pixel is also 1. This can be seen in Figure B.2 where for structuring element s_1 it does not fit section B in the image, but it does hit.



```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0 0 0 0 0
0 0 1 1 1 1 1 0 0 0 0 0
0 1 1 1 1 1 1 1 0 0 0 0
0 1 1 1 1 1 1 1 0 0 0 0
0 0 1 1 1 1 1 1 0 0 0 0
0 0 1 1 1 1 1 1 1 0 0 0
0 0 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0
```

$$s_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad s_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

		A	B	C
fit	s_1	yes	no	no
	s_2	yes	yes	no
hit	s_1	yes	yes	yes
	s_2	yes	yes	no

Figure B.2: Fitting and hitting of a binary image with structuring elements s_1 and s_2. [40]

Figure B.3 illustrates a structuring element of 2x2 pixels. In Figure B.4, this structuring element is passed over a binary image and anything smaller than the structuring element is considered undesirable such as section B and C.

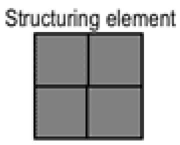


Figure B.3: Structuring element of size 2x2 [40]

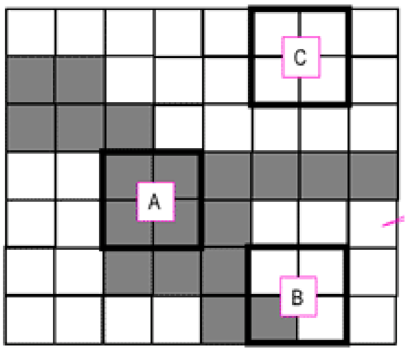
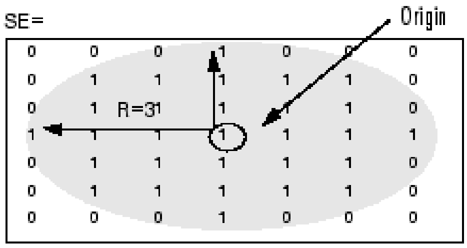


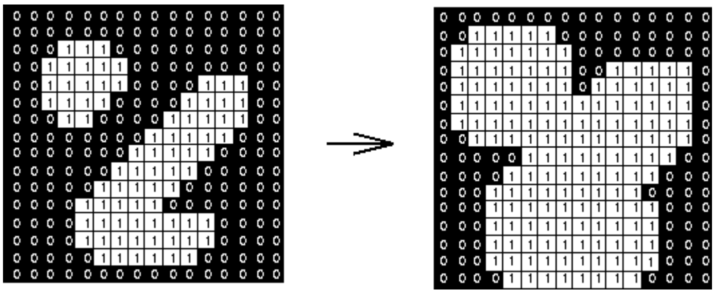
Figure B.4: Passing structuring element over a binary image [40]

The disk-shaped structuring element with a specified radius shown in Figure B.5 is passed over the image and elements that lie within the element will be filled. Figure B.6 illustrates how the binary image is affected by applying the circular structuring element.



$$SE= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Figure B.5: Circular structuring element [41]



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0

Figure B.6: Binary image before (left) and after (right) applying circular structuring element [40]