Faculty of Electrical Technology and Engineering

DEVELOPMENT OF GREENHOUSE LEAF HEALTH MONITORING SYSTEM USING MATLAB

CLEY ALEXSIUS JARIUS

Bachelor of Electrical Engineering Technology with Honours

2023

DEVELOPMENT OF GREENHOUSE LEAF HEALTH MONITORING SYSTEM USING MATLAB

CLEY ALEXSIUS JARIUS

**A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics)
with Honours**

Faculty of Electrical Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF GREENHOUSE LEAF HEALTH MONITORING SYSTEM USING MATLAB

Sesi Pengajian : 2023/2024

Saya CLEY ALEXSIUS JARIUS mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

✓	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
✓	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
	TIDAK TERHAD	

Disahkan oleh:

(TANDATANGAN PENULIS)

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap:
Kampung Namadan Baru Sook
89000, Keningau
Sabah

Tarikh: 14 Jan. 24

Tarikh: 14 Jan. 24

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Development of Greenhouse Leaf Health Monitoring System Using MATLAB" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Student Name	:	CLEY ALEXSIUS JARIUS
Date	:	14 Jan. 24

APPROVAL

I approve that this Bachelor Degree Project 1 (PSM1) report entitled "Development of Greenhouse Leaf Health Monitoring System Using MATLAB" is sufficient for submission.

Signature :

Supervisor Name : MOHAMAD HANIFF BIN HARUN

Date : 14 Jan. 24

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours.

Signature :

Supervisor Name : MOHAMAD HANIFF BIN HARUN

Date : 14 Jan. 24

Signature :

Co-Supervisor :

Name (if any)

Date :

DEDICATION

In the spirit of unwavering dedication to innovation and the relentless pursuit of sustainable agricultural practices, this report stands as a tribute to the visionary individuals who have dedicated their efforts to cultivating a healthier and more resilient future for our global agricultural landscape. At the heart of this dedication is a profound appreciation for those who recognize the critical role that technology plays in shaping the future of farming. It is with deep admiration that this work is offered to those who understand the intersection of cutting-edge technology and agriculture as a catalyst for positive change.

Specifically, this dedication is extended to the individuals and teams who have contributed to the realization of the project titled 'Development of Greenhouse Leaf Health Monitoring System Using MATLAB.' To my family, whose unwavering support has been my anchor throughout this journey, I express my deepest gratitude. Your encouragement, understanding, and belief in my endeavors have been the driving force behind each step of this project. To my supervisor, your guidance and expertise have been invaluable. Your mentorship has not only shaped the technical aspects of this work but has also instilled in me a deeper appreciation for the intricacies of agricultural technology.

To my lecturer, whose insightful teachings have laid the groundwork for the knowledge applied in this project, I extend my sincere appreciation. Your dedication to fostering intellectual growth has been a source of inspiration. This project is not merely a culmination of technical skills but a testament to the transformative power of education. To my friends, who have shared in the highs and lows of this academic journey, your camaraderie has added immeasurable richness to the experience. Your encouragement and shared passion for learning have made this endeavor all the more meaningful.

As we acknowledge the significance of this endeavor, it is essential to recognize that the fruits of our labor extend beyond the mere development of a monitoring system. This dedication is a tribute to the far-reaching impact of our collective work, reflecting the hope that our efforts will serve as a catalyst for transformative change in the agricultural sector. May the insights and advancements presented in this report resonate not only with the scientific and technological community but also with the farmers and stakeholders whose livelihoods are intricately tied to the health and prosperity of our global food systems.

In conclusion, this dedication expresses gratitude to my family, supervisor, lecturer, and friends who have played pivotal roles in the 'Development of Greenhouse Leaf Health Monitoring System Using MATLAB.' May our endeavors inspire future generations of innovators, fostering a legacy of sustainable and technology-driven agriculture that continues to flourish in the years to come.

ABSTRACT

The objective of this study was to design and implement a comprehensive monitoring system for a greenhouse environment. The system aimed to collect real-time data on various environmental parameters such as temperature, humidity, light intensity, and soil moisture. The data was captured using sensors strategically placed within the greenhouse and transmitted wirelessly to a central monitoring unit. The monitoring system utilized a microcontroller-based architecture, which allowed for efficient data collection and processing. A web-based interface was developed to visualize the collected data in real-time, enabling greenhouse operators to remotely monitor the environmental conditions and make informed decisions regarding the cultivation process. The implemented monitoring system demonstrated its effectiveness in providing accurate and timely information about the greenhouse environment. By continuously monitoring the key parameters, it facilitated proactive measures to optimize plant growth and yield. Moreover, the system incorporated alert mechanisms to notify users in the event of critical deviations from the desired environmental conditions, ensuring prompt action to prevent potential crop damage. The results obtained from the monitoring system indicated that it significantly contributed to improving the overall efficiency of greenhouse operations. It allowed for precise control of environmental factors, leading to enhanced crop quality and reduced resource wastage. Furthermore, the web-based interface provided a user-friendly platform for data visualization and analysis, empowering greenhouse operators with valuable insights into the cultivation process.

In conclusion, the developed monitoring system proved to be an invaluable tool in optimizing greenhouse management. Its ability to collect and analyze real-time data on environmental parameters facilitated informed decision-making and enabled timely interventions. The system's user-friendly interface and reliable performance make it a promising solution for greenhouse operators seeking to enhance their productivity and sustainability.

ABSTRAK

Objektif kajian ini adalah untuk mereka bentuk dan melaksanakan sistem pemantauan menyeluruh untuk persekitaran rumah hijau. Sistem ini bertujuan untuk mengumpul data masa nyata mengenai pelbagai parameter persekitaran seperti suhu, kelembapan, keamatan cahaya dan kelembapan tanah. Data telah ditangkap menggunakan penderia yang diletakkan secara strategik di dalam rumah hijau dan dihantar secara wayarles ke unit pemantauan pusat. Sistem pemantauan menggunakan seni bina berasaskan mikropengawal, yang membolehkan pengumpulan dan pemprosesan data yang cekap. Antara muka berasaskan web dibangunkan untuk menggambarkan data yang dikumpul dalam masa nyata, membolehkan pengendali rumah hijau memantau dari jauh keadaan persekitaran dan membuat keputusan termaklum mengenai proses penanaman. Sistem pemantauan yang dilaksanakan menunjukkan keberkesanannya dalam menyediakan maklumat yang tepat dan tepat pada masanya tentang persekitaran rumah hijau. Dengan memantau parameter utama secara berterusan, ia memudahkan langkah proaktif untuk mengoptimumkan pertumbuhan dan hasil tumbuhan. Selain itu, sistem ini menggabungkan mekanisme amaran untuk memberitahu pengguna sekiranya berlaku penyelewengan kritikal daripada keadaan persekitaran yang diingini, memastikan tindakan segera untuk mengelakkan kerosakan tanaman yang berpotensi. Keputusan yang diperoleh daripada sistem pemantauan menunjukkan bahawa ia menyumbang dengan ketara kepada peningkatan kecekapan keseluruhan operasi rumah hijau. Ia membenarkan kawalan tepat terhadap faktor persekitaran, yang membawa kepada peningkatan kualiti tanaman dan mengurangkan pembaziran sumber. Tambahan pula, antara muka berasaskan web menyediakan platform mesra pengguna untuk visualisasi dan analisis data, memperkasakan pengendali rumah hijau dengan pandangan yang berharga tentang proses penanaman.

Kesimpulannya, sistem pemantauan yang dibangunkan terbukti menjadi alat yang tidak ternilai dalam mengoptimumkan pengurusan rumah hijau. Keupayaannya untuk mengumpul dan menganalisis data masa nyata tentang parameter alam sekitar memudahkan membuat keputusan termaklum dan membolehkan campur tangan tepat pada masanya. Antara muka mesra pengguna sistem dan prestasi yang boleh dipercayai menjadikannya penyelesaian yang menjanjikan untuk pengendali rumah hijau yang ingin meningkatkan produktiviti dan kemampanan mereka.

ACKNOWLEDGEMENTS

I extend my sincere gratitude to all those who have played a pivotal role in the realization of the 'Development of Greenhouse Leaf Health Monitoring System Using MATLAB.' This project has been a journey filled with challenges and triumphs, and the support of numerous individuals has been instrumental in its success.

First and foremost, I would like to express my deepest appreciation to my family. Your unwavering support, encouragement, and understanding have been my pillars of strength throughout this academic pursuit. Your belief in my abilities and your constant encouragement propelled me forward, even in moments of uncertainty.

I am indebted to my supervisor for their invaluable guidance, mentorship, and expertise. Their insightful feedback and constructive criticism have significantly contributed to the refinement and success of this project. Their dedication to fostering a spirit of inquiry and innovation has been a guiding light.

A heartfelt acknowledgment goes to my lecturer, whose teachings laid the groundwork for the knowledge applied in this project. Their passion for the subject matter and commitment to excellence have inspired me to strive for academic and professional growth.

I extend my gratitude to my friends, whose camaraderie and shared enthusiasm for learning have added immeasurable value to this journey. Your support, encouragement, and shared experiences have made the challenges more surmountable and the victories more joyous.

To all the individuals who have contributed, directly or indirectly, to this project, your collective efforts have not gone unnoticed. Your expertise, insights, and encouragement have shaped the outcome of this endeavor.

Finally, I would like to express my thanks to the academic institution for providing a conducive environment for learning and research.

Thank you all for being a part of this journey. Your contributions have left an indelible mark on the success of this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Agriculture in the greenhouse system represents a significant advancement in modern farming practices. A greenhouse is a controlled environment structure that allows for the cultivation of crops in a protected setting. It provides an enclosed space where temperature, humidity, light, and other environmental factors can be manipulated to optimize plant growth and productivity. One of the key benefits of the greenhouse system is the ability to control the environmental conditions. Temperature, humidity, and ventilation can be adjusted to create the ideal growing conditions for specific crops. This level of control helps to protect plants from extreme weather conditions, pests, and diseases, reducing the reliance on pesticides and herbicides. It also minimizes water usage by allowing for precise irrigation management. Furthermore, the greenhouse system allows for the cultivation of crops that are not well-suited to the local climate. By creating a customized environment, farmers can grow a wide range of plants that may not thrive in the surrounding region. This opens up opportunities for diversification and the production of specialty crops, leading to economic benefits for farmers.

The controlled environment of a greenhouse also facilitates the implementation of advanced agricultural practices. For example, hydroponics, a soilless cultivation technique, is commonly employed in greenhouse systems. It involves growing plants in nutrient-rich water solutions, which allows for precise nutrient management and efficient water usage. Additionally, technologies such as artificial lighting can be used to supplement or replace natural sunlight, enabling year-round growth and enhancing productivity. The greenhouse system has gained popularity in both commercial and small-scale farming operations. Large-scale greenhouse complexes can produce high volumes of crops for commercial markets, while smaller-scale greenhouses are utilized by individual farmers and gardeners to grow fresh produce for local consumption.

In summary, the greenhouse system has revolutionized farming practices. It offers controlled environments that extend the growing season, provide crop protection, enable precise environmental control, and facilitate advanced cultivation techniques. By harnessing these benefits, greenhouse farming enhances productivity, promotes sustainability, and expands the range of crops that can be grown, contributing to food security and economic prosperity.

1.2 Problem Statement

Busy Farmer: Nowadays, people is getting busy with their works and not have sufficient time to be around their crops. If they outstation, then they will not be able to observe the condition of their plants.

Lack of Real-Time Data: Monitoring systems are often designed to collect and analyze data in real-time. However, a difficulties may arise when there are delays or gaps in data collection, making it difficult to have up-to-date and accurate information for decision-making.

Cost and Affordability: The cost of implementing a greenhouse monitoring system can be a significant barrier, especially for small-scale or resource-limited growers. The expenses associated with sensors, data loggers, control devices, software, and ongoing maintenance can be prohibitive. Finding cost-effective solutions without compromising on the quality and reliability of the system is essential to make greenhouse monitoring systems accessible to a wider range of growers.

1.3 Project Objective

The following are the project's objectives:

a) To develop greenhouse leaf health monitoring system using MATLAB.
b) Develop a robust leaf imaging system utilizing MATLAB to analyze high-resolution images of greenhouse leaves.
c) To develop MATLAB system to provide detailed reports on the identified leaf conditions, including their causes and recommended corrective measures.

.

1.4 Scope of Project

In this project, the main aim is to develop monitoring system for environment in greenhouse. To ensure this projects reach the goals is setting the scope of the project. The following are the scope of the project:

a) This project leverages the powerful capabilities of MATLAB to develop a comprehensive leaf health monitoring system, employing advanced image processing algorithms for analysis and classification of greenhouse leaf conditions.
b) Utilize machine learning techniques, integrated into MATLAB, for the classification and identification of various leaf conditions, such as diseases.
c) Conduct thorough validation and testing of the developed system to ensure the reliability of leaf condition analysis, with the flexibility to refine algorithms as needed for enhanced performance.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter covered the articles that were researched for this project. Greenhouses play a vital role in modern agriculture by providing controlled environments for optimal plant growth and production. To ensure efficient management and productivity, it is essential to monitor and control various environmental factors inside the greenhouse. Various strategies are available in precision agriculture (PA) to monitor and manage the necessary environmental parameters for the specific crop. Analysis of management methods for the appropriate setting is particularly crucial. This literature review aims to explore the advancements in monitoring systems for environmental control in greenhouses, focusing on the integration of sensors.

2.2 System Network

The system design is depicted in Fig.1 it is made up of two types of physical units: three remote sensor nodes and a central control station. An XBee radio and analogue sensors are used to build the remote sensor nodes(Hussain et al., 2013). These radios enable ZigBee topologies, which are designed to receive analogue signals directly from sensors and broadcast them as part of a data packet. Each node can detect temperature, humidity, and light levels. The measured data are relayed to the central computer on a regular basis. The central control unit is made up of an XBee radio kit that is connected to a personal computer via USB.

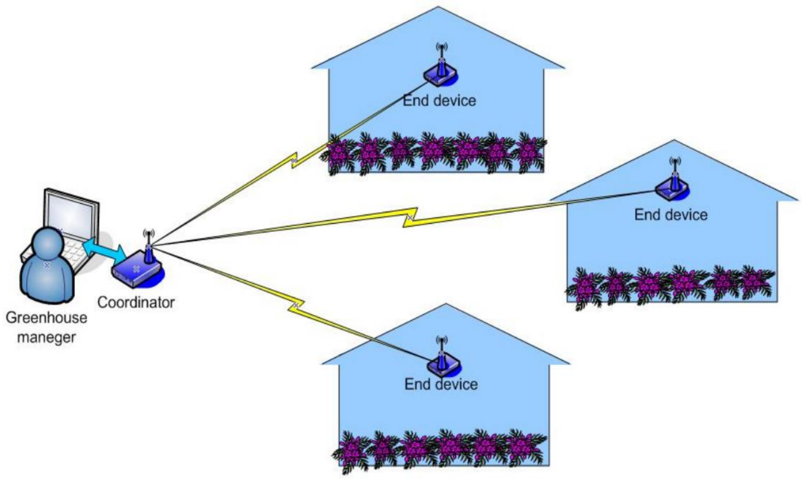


Figure 1 System design

Next, the Rgroup created a complete data acquisition system using a LabView (National Instruments) virtual instrument (Mancuso & Bustaffa, 2006) to allow simultaneous acquisition of three separate signals from each WSN sensor. The sensors network is a grid of wireless nodes, as shown in Fig. 2. Six nodes were used in a 20 by 50 metre tomato greenhouse, set in two rows 12.5 metres apart. The data acquired by the sensors is collected at the greenhouse's perimeter and relayed over LAN to a laptop computer (base station) for data logging and correlation. In the near future, collected data will be sent through WiFi from a base station to a server for data logging and correlation. The server will be linked to the Internet via LAN, and data will be uploaded to a Web server in XML format.

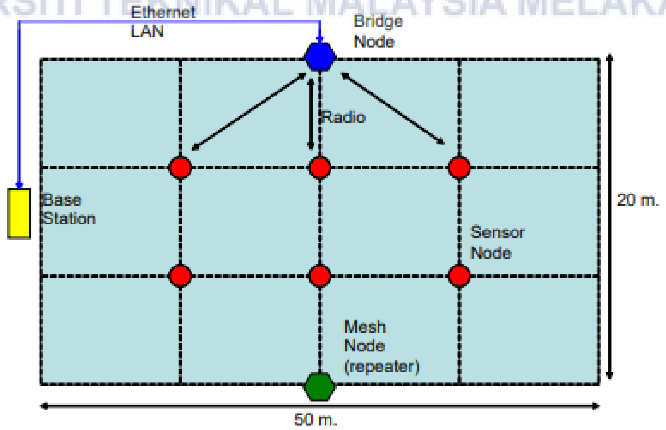


Figure 2 Sensors field structure for the experiment

In addition, between May 27 and June 25, 2008, the wireless sensor network was installed at the 640 sq.m R&D greenhouse in Labu, Negeri Sembilan (Tik, 2009). The greenhouse was coated with a 0.2 mm thick polyethylene sheet and outfitted with a 10-HP chiller unit that supplies chilled feed water to nine planting troughs, a 300-litre capacity nutrient tank for each trough, and four 1 HP ventilation fans. Each trough contains about 300 lettuce plants, which will grow in the trough for about a month until harvest.

Figure 3 Four ventilation fans and planting channels inside the greenhouse.

Sensor Node A, which comprises of two temperature sensors and one light quantum sensor, was placed in lieu of a lettuce plant in the trough. Sensor Nodes B and C, which were intended to be connected to the pipes supplying the solution to the nozzles, were later relocated to the feed tank to reduce the amount of piping modifications at the trough.

Figure 4 Node A, Node B and Node C planting

Lastly, a wireless sensor network and information control system were used in the hydroponic growing of cucumbers in a greenhouse facility at the University of Thessaly's experimental farm in Volos, Greece (Kalovrektis et al., 2013). Figure 4 depicts a plan view of the facility, which includes a chamber located roughly 70m away from the greenhouse and housing the computer for data collecting and control. A temperature sensor, an electric conductivity (EC) sensor, and an electronic balance were among the instruments used. The computer communicates with an embedded Full Function Device (FFD) based on ZigBee technology via the RS-232 serial port.

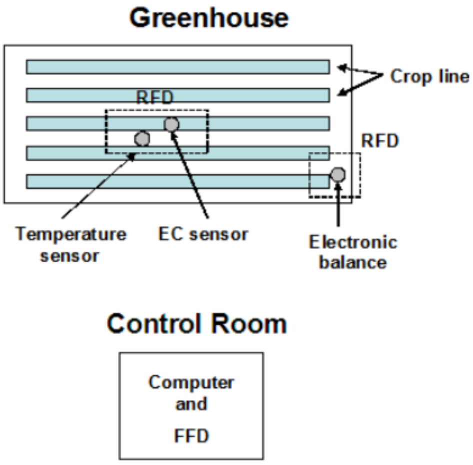


Figure 5 Installed sensors and data acquisition control room

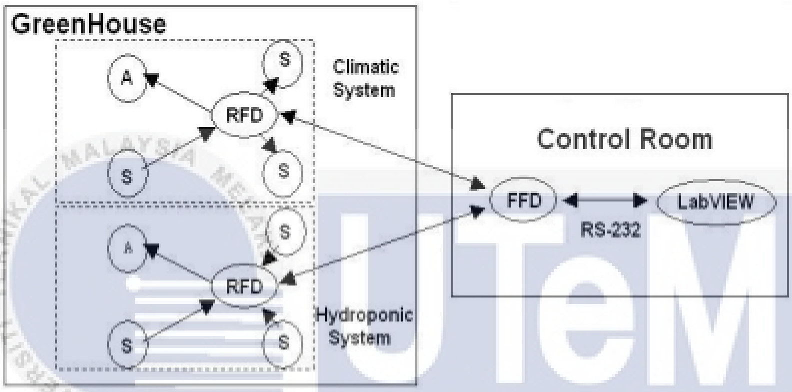


Figure 6 Wireless Model of the Star Network (Sensors (S), Actuators (A), Reduce Function Device (RFD), and Coordinator (FFD))

(Abdelhafidh et al., 2019) Architecture of WSN for dam monitoring. It provides a real time monitoring of dams in order to protect them from environmental anomalies and extern events. This work aimed to protect and worn people in case of emergency. To reach this goal, a set of parameters and environmental variables like temperature, water level and rainfall need to be sensed by a significant number of sensor nodes. For this, they deployed WSN cluster architecture with multi-hop communication. The proposed architecture for dam monitoring is illustrated by Figure 6.

A vast number of wireless sensor nodes are put in field regions as part of an environment monitoring system for agricultural applications to gather data on temperature, humidity, brightness, and air pressure. Using the MQTT (Message Queuing Telemetry Transport) protocol, the collected data are sent to the gateway over WiFi (Wr et al., n.d.). The gateway, which is a node centre, collects data from nodes, stores data, computes, and integrates data. A gateway can also construct a WiFi network and run a MQTT broker, which is used to send data from sensor nodes to the gateway and from the gateway to the cloud. Users can monitor environmental data in real-time thanks to the user interface, which is a web application based on a cloud platform. In this system, the author designed a loT gateway using a Raspberry Pi (running Raspbian OS), which is responsible for storing, analysing, and relaying sensor data to the cloud. The gateway was configured in access point mode in order to establish a WIFI connection to which the sensor nodes could connect. The Eclipse Mosquitto, an open source message broker that implements the MQTT protocol, was installed in the gateway. IBM Watson IoT in Bluemix is our chosen cloud platform for managing and storing sensor data. Node-RED flows include sending commands to sensor nodes and receiving data, with specific buttons serving different purposes. Environmental data can be monitored in two ways, the first option is for the user to connect to the gateway directly over WIFI using a web browser and then navigate to the host address (172.16.1.1:1880/ui). The real-time data will be displayed as soon as the gateway receives data. The second method is to monitor environmental information anywhere there is internet access by going to the appropriate web URL (http://agrinode.mybluemix.net/ui) to monitor the data. Figure show how environment monitoring system operates.

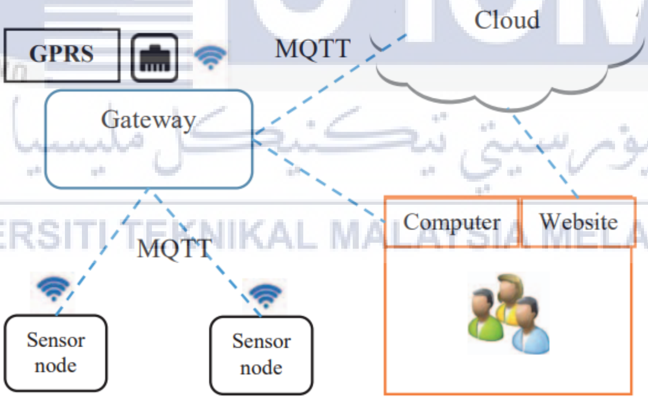


Figure 7 Environment monitoring system

SCADA (Supervisory Control and Data Acquisition) systems are increasingly being used in agriculture to enhance operational efficiency, automate processes, and improve overall productivity. The most important component of SCADA is the control section, which is based on closed loops and is in charge of system evolution(Mirabella et al., 2011). As shown in Figure 8, all devices are linked to the control system via an appropriate communication structure. The integration of wired and wireless networks enables the implementation of this structure. The wired network serves as the backbone, connecting greenhouses to the control room, while wireless networks connect clusters of sensors. A CAN/ZigBee bridge facilitates the integration of these protocols. Field devices are connected to either network based on their requirements. Wireless networks are preferred for easily movable sensors, avoiding disruption to cultivation caused by wired connections. Additionally, wireless networks help conserve battery life through power-saving strategies. Precise localization of mobile sensors allows for easy repositioning, automatically detected and displayed on the SCADA system monitor.

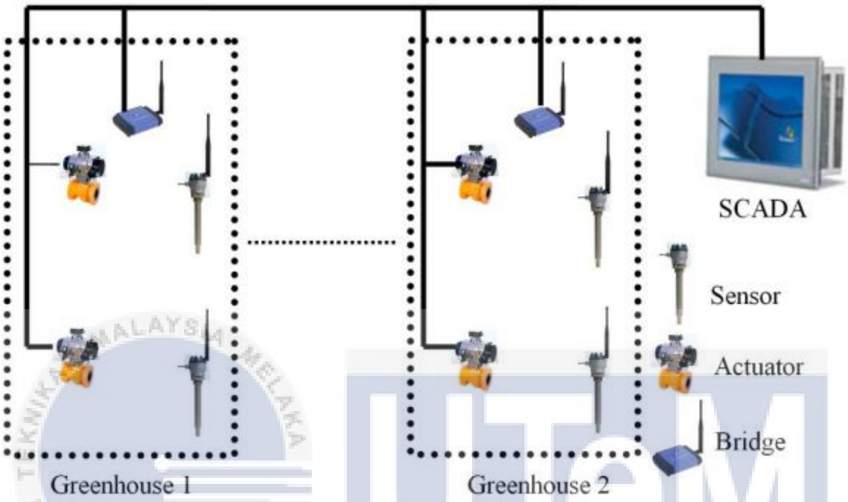


Figure 8 wired and wireless system network of greenhouse

In many wireless communication technologies GPRS, ZigBee, Bluetooth, WIFI networks, etc. important for agricultural surveillance systems Greenhouses have advantages such as high bandwidth, high transmission speed, strong compatibility, powerful anti-interference capability (He,2013), also important Direction of research on dissemination of greenhouse information Technology (Lin, 2014). Considering the above issues, this This paper proposes a WIFI-based greenhouse environment. Remote monitoring system. WIFI-based surveillance system converts TTL signals connect to wireless WIFI signal via sensor module. WIFI module, realize network connection and data communication with the server. WIFI module is expensive high-performance wireless LAN module USR-WIFI232-A2, manufactured by the manufacturer YouRen network company in Jinan. Physical devices can be sent via this module can connect to WIFI network to implement control management of the greenhouse environment. The WIFI module in this system can work in two modes. AP mode and STA mode. In this particular system, it is connected to the sensor board and operates in STA mode. To enable communication between the sensor and the server, the WIFI module needs to establish a socket network connection with the server. The communication process is shown in Figure 9. Figure 9 shows a block diagram of the interaction between the sensor and the server. To establish connectivity, connect both the server and the sensor to your WLAN router. The IP address obtained from the server computer is mapped to the router's public IP. At the same time, his WiFi module on the sensor is switched to his STA mode and added to the WiFi router. The WIFI module's network parameters are set to TCP client mode. Additionally, the WIFI module is added to the port opened by the server and associated with the public IP assigned by the server. When the server receives the request, a socket network connection is established between the sensor and server to allow data transfer.

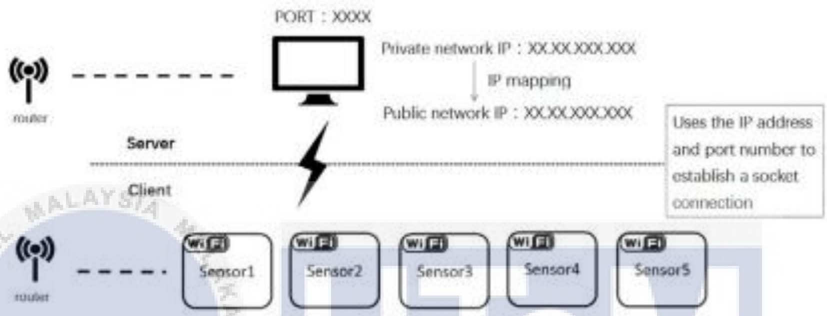


Figure 9 Block diagram of the communication between the sensor and the server

a)

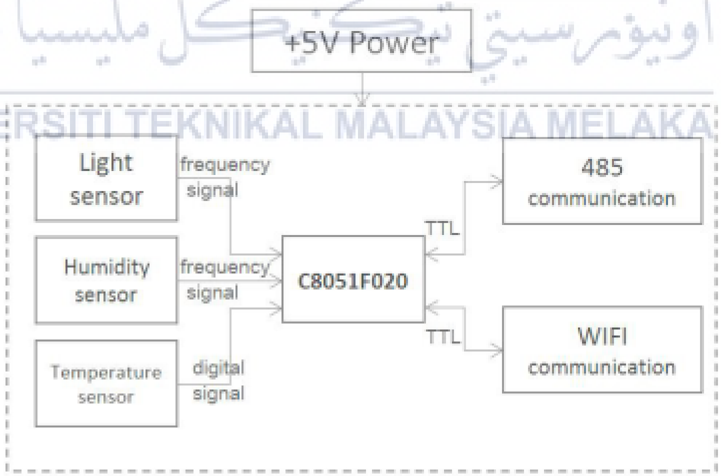


Figure 10 Working principle of the sensor module

The wireless sensor network system consists of a star topology. It has multiple integrated wireless sensor nodes as shown in Figure 11. A star topology allows each node to seamlessly communicate with each other (Rustia et al., 2020). Work with each other and send data directly over the Internet. Each wireless sensor node broadcasts environmental data 5 minutes, using the UDP protocol and sending the image via HTTP POST protocol. All data is also stored directly there to prevent data loss. Back up the SD memory card of each node. All recorded data is stored on a central server. main office the server runs on Windows 7 with an Intel Core i5 processor. Support for NVIDIA GTX630 GPUs. The server uses Apache as a server MySQL as web server software and database. Image editing and data analysis is done by the server to create processed data available online through a website accessible from your computer or your smart phone. Using processed images of sticky paper traps, websites can display the approximate spatial location of each node. Provided to monitor the situation on the ground. Temporary data are also provided to monitor changes in pest numbers. Environmental condition. Using the collected spatio-temporal information, data analysis such as the number of pest outbreaks is performed. Environmental conditions are posted on our website. Meanwhile, it base stations act as real-time onsite displays for farm owners shows data similar to the website.

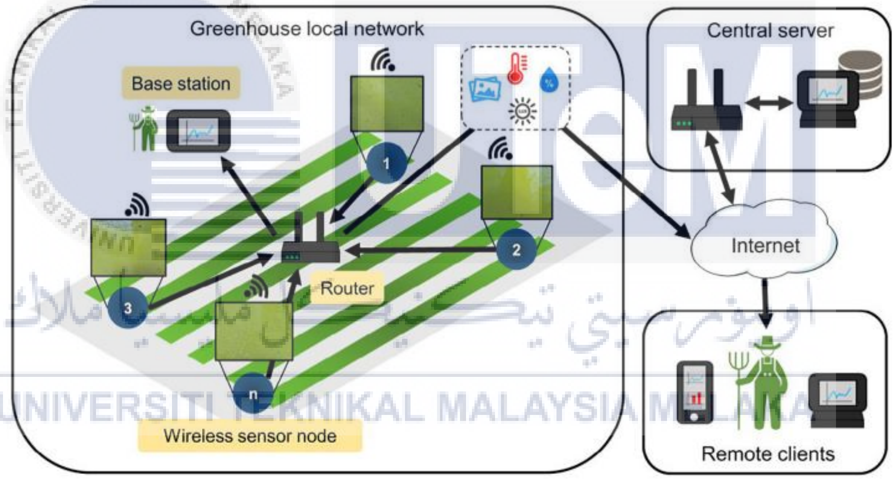


Figure 11 Integrated wireless sensor

2.3 Image processing for Monitoring System

An image-based algorithm for monitoring systems in agriculture refers to the use of computer vision techniques and image processing algorithms to analyze images or video data captured from agricultural fields. It involves extracting useful information from the images to monitor and assess various aspects of crop health, growth, and environmental conditions. The Normalised Different Vegetation Index (NDVI), which is often used in agricultural robots to monitor plant health or even identify diseases, is an example of an image-based algorithm. The Normalised Difference Vegetation Index, which is derived from a hyperspectral video imaging approach, was used to identify the response of plants to high Zn and Cu levels (Thai C.N., Evans M.D., Schuerger A.C, 1999).

To obtain the NDVI of crops, which may be used to estimate crop health in various sections of the field. A Raspberry Pi NoIR camera and a Raspberry Pi Zero were used by the author to build the camera setup. These were chosen because to the tiny amount of weight they add to the system. Furthermore, the Raspberry Pi camera is powerful enough to capture images of agricultural lands. An additional filter used in conjunction with this camera to obtain a visible and an NIR band for NDVI extraction. Following, the photos were transmitted via USB to a laptop or personal computer for stitching and NDVI extraction. These were accomplished by inventing and developing a software in Python that used OpenCV to perform all of the processing automatically, from stitching to NDVI extraction. The image was NDVI processed by dividing it into three bands: red, green, and blue. The red band of the image contained the visible values (R) of each pixel in the image, while the blue band of the image contained the NIR values of each pixel in the image, using a red filter with the camera system.

Figure 12 Scheme color used for the NDVI images

Sample Color	Color	Implication
	blue to violet	very healthy
	green	fairly healthy, doesn't have urgent needs
	yellow	unhealthy, needs some attention
	red to orange	soil, or very unhealthy vegetation, needs urgent attention/care

Figure 13 Different colour implication in the false coloured image

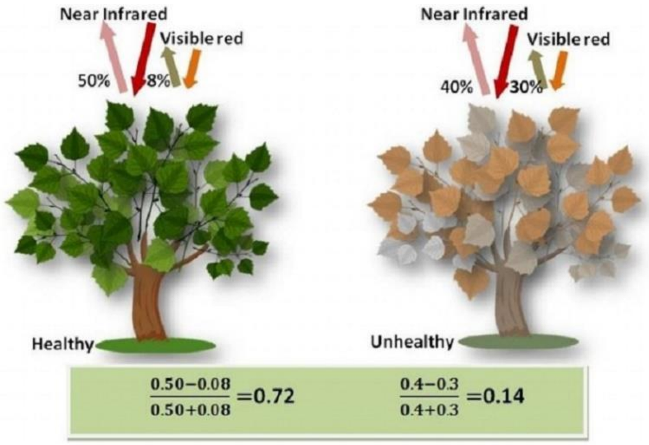


Figure 14 Leaf with NDVI value perform their condition

Next, to extract seedling traits from images, machine vision can be employed. Machine vision is the automated gathering and examination of video images for use in command or control. Employed a computer-controlled rotating stage's motion to capture images. (Lin T.T., Cheng S.F., Lin T.H., Tsai M.R.,, 1998) the logistic and Richard functions were adjusted to take into account seedling height, total leaf area, and top fresh and dry weights. Growth curves were then modelled using the logistic function-derived relative growth rate. These include self-driving tractors, unmanned aerial vehicles (UAVs) for surveying the state of the soil, and robots to help with milking, feeding, and harvesting. The quality and grade of seeds and produce are also inspected using machine vision. Under UV, VIS, or NIR lighting, machine. Vision systems capture images of agricultural commodities. Image processing is mainly done in computer vision detection and identification of the specific target. By analyzing images of leaves, stems, or fruits, algorithms can identify patterns, discoloration, lesions, or other signs of plant stress. Early detection allows farmers to take timely measures to prevent the spread of diseases and minimize crop damage. Machine vision can assess crop maturity by analyzing visual cues such as colour, texture, or size of fruits or grains. This information helps farmers determine the optimal harvest time, ensuring peak crop quality and reducing post-harvest losses.

Figure 15 Machine vision

Figure 16 Vision camera

Figure 17 Computer vision

Moreover, Red, green, blue (RGB) sensors are the least expensive and most common passive sensor type used on drones. These sensors capture visible light (400–700 nm wavelengths) in overlapping red, green, and blue channels, similar to human vision. Data from visible light sensors are frequently rather simple to understand qualitatively, even by relatively inexperienced persons. However, it is possible to calculate various vegetation indicators, like the green leaf index, to expand the usage of visible light sensors to more analytical applications.

Figure 18 Image captured by RGB sensor

Next, inter-plant weed detection is employed in agriculture. Inter-plant weed detection enables farmers to optimize the allocation of resources to crops. By accurately identifying and removing weeds, crops can receive adequate nutrients, water, and space without having to compete with weed plants. This promotes optimal growth and development of the cultivated plants, leading to improved crop yield and quality. Weeds not only compete with crops for resources but can also harbour pests and diseases. By detecting and removing weeds, farmers can reduce the risk of pest infestations and disease spread, leading to improved crop health and higher-quality produce. Figure 15 show steps in interplant weed detection.

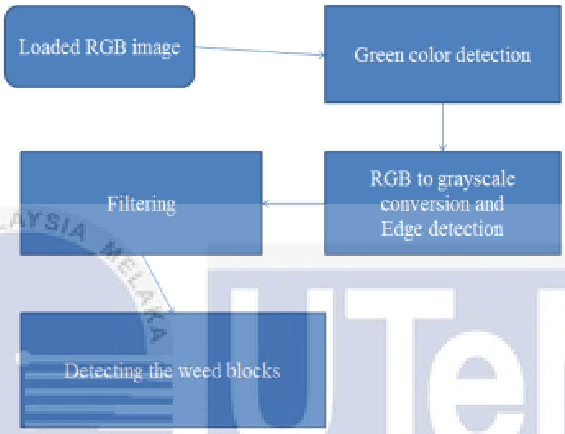


Figure 19 show steps in interplant weed detection.

This section of the algorithm loads the image from the source, performs colour segmentation, and performs edge detection to get an image ready for more sophisticated processing.

Figure 20 Input image for weed detection

One technique for separating the crop (which also includes weed) from the background in a photograph is colour segmentation (Poojith et al., 2000). Through Kmeans clustering, this is accomplished. The technique facilitates the separation of all visually discernible colours from one another. By assembling objects made up of pixels of the same colour, or "clustering," it becomes simpler to segment the data. Only two colours are present in the final image. After colour segmentation, the intended image is composed of green (the crop and the weed) and black (the remaining portion of the image), making edge detection possible.

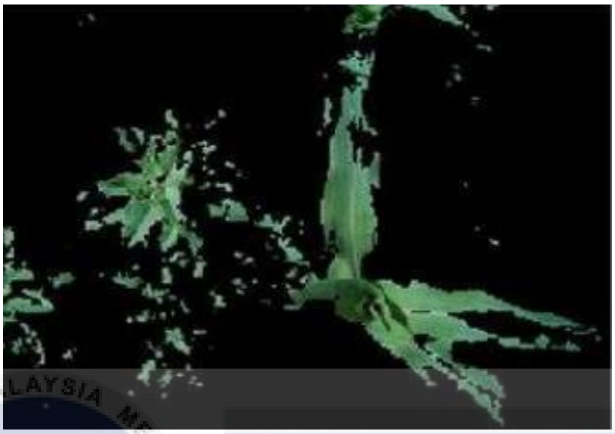


Figure 21 Output image (after colour segmentation)

In order to distinguish the crop from the weed, edge detection leverages the fact that the edge frequencies and veins of the crop and the weed have distinct density qualities (strong and weak edges).

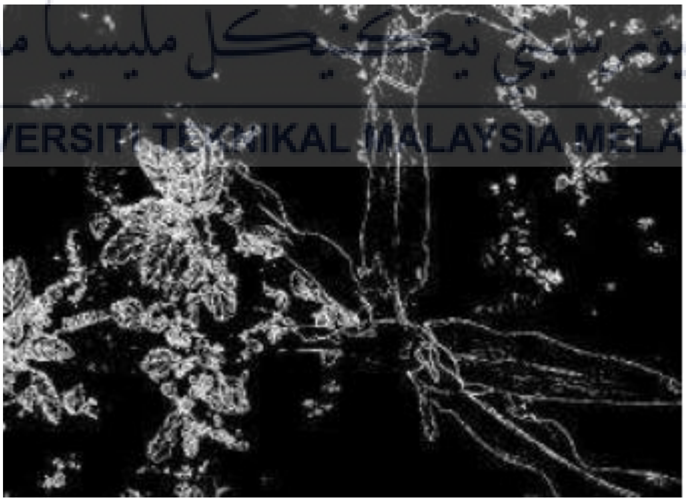


Figure 22 Output image (after edge detection)

Lastly, k-means clustering can also be applied to image processing in agriculture for tasks such as crop segmentation, disease detection, or weed identification. The system added new feature which is alerting. In India, this system help the farmer effectively minimize the decease spreading and improve the yield production of the crop and thereby suicides(Mugithe et al., 2020). Not only recognize diseases, but also notifies the farmer as soon as possible after an illness detection. Figure 19 show the system architecture.

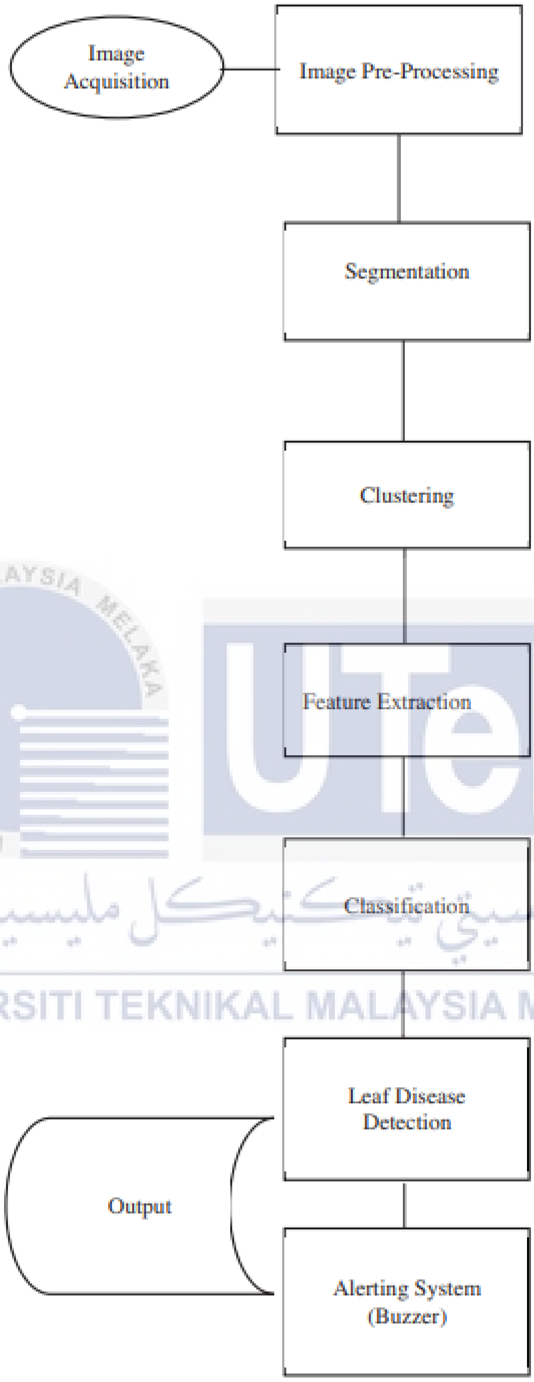


Figure 23 System architecture

MATLAB R2016 is use to develop this system to perform on leaf pictures and highlight separated sickness influenced bunch with the assistance of k-implies grouping calculation (Mugithe et al., 2020). The system did this in two ways:

- GUI (Graphical user interface)

  Detect decease in leaf and kept in Database, MATLAB software process in point to point.

Figure 24 Uploaded decease leaf

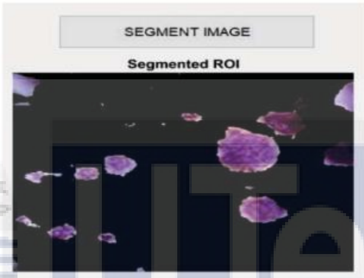


Figure 25 Segmented Image

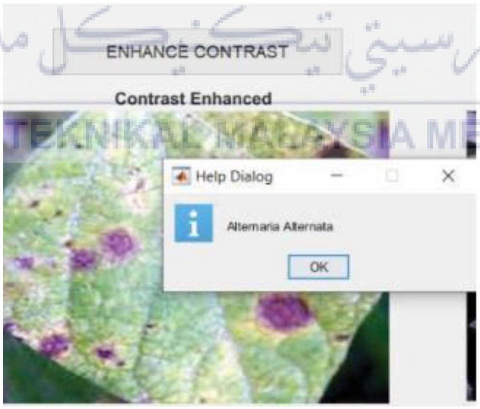


Figure 26 Decease detection

- Real time

  Collected real time leaves and processes under web-cam connected to Raspberry-pi. Buzzer will on when detect decease on leaf.

Figure 27 Before detect decease

Figure 28 After detect Decease

2.4 Sensor technology in agriculture

Current technology in agriculture has made significant advancements and continues to evolve rapidly. Precision agriculture involves the use of technologies such as GPS, sensors, drones, and satellite imagery to gather data and make informed decisions about crop health, soil conditions, irrigation, and fertilizer application. This allows farmers to optimize their practices, reduce waste, and increase productivity. Robots and automation are being used in various agricultural tasks, including planting, harvesting, and weeding. These robots can operate autonomously or be controlled remotely, reducing labour requirements and improving efficiency. Wireless sensor are used in agriculture to monitor and control various parameters such as temperature, humidity, soil moisture, and crop growth.

Sensors play an important part in agricultural development by gathering information about numerous aspects such as soil, water, climate, and so on. Numerous sensor technologies have been employed to monitor environmental parameters in greenhouses. Temperature, humidity, light intensity, carbon dioxide (CO2) levels, soil moisture, and nutrient concentrations are commonly measured variables. These sensor technologies have a significant impact on plant output and quality when integrated into a comprehensive monitoring system, provide valuable data for maintaining precise control over environmental conditions inside the greenhouse. A sensor is able to transform environmental physical or chemical readings into signals that a system can calculate.

2.4.1 Temperature sensor technology in agriculture

Temperature sensors in agriculture providing accurate and real-time temperature data for various applications. These sensors help farmers monitor and manage temperature conditions in different agricultural settings, ensuring optimal growth, productivity, and resource management. Temperature is a critical environmental factor that affects plant growth, crop development, and livestock health. Temperature sensors provide data that helps optimize growing conditions, prevent crop stress, mitigate heat or cold-related risks, and improve overall agricultural productivity. There are a few common types of temperature sensors and their basic working principles.

In agriculture, drone also play a significant part. Drones equipped with thermal sensors offer a unique and efficient way to monitor temperature variations in agriculture. The obtained thermal precision is suitable for most agricultural applications requiring comparative temperature analysis (van der Merwe et al., 2020). Drones are equipped with specialized thermal sensors, such as thermal cameras or infrared (IR) cameras. These sensors detect and measure the infrared radiation emitted by objects based on their temperature. They capture the temperature distribution across a scene and convert it into thermal images or data. The drone is flown over the agricultural area of interest, capturing aerial images or videos using the thermal sensor. The thermal sensor collects temperature data from the entire scene, capturing the thermal signatures of crops, livestock, infrastructure, and the surrounding environment. After the drone flight, the captured thermal images or data are processed and analyzed using specialized software or algorithms. Drone photos and ground sensor data are likely to play an important role in precision agriculture, allowing for extensive scientific research and development (Murugan D, Garg A, Ahmed T and Singh D

2016, 2016). The software interprets the temperature data and generates thermal maps or visualizations that represent the temperature distribution across the area. The thermal maps generated from the drone data provide valuable insights into temperature variations within the agricultural landscape. Farmers can identify areas of interest, such as variations in crop temperature, presence of heat or cold stress, water stress, or pest infestations. Uncooled thermal cameras, on the other hand, are typically mounted to UAVs (Figure 25 a, b) since they are smaller, lighter, and consume less energy (Gallo, Willits, Lubke, & Thiede, 1993).

Figure 29 a, b DJI Phantom 4 pro & DJI Inspire both equipped with a FLIR uncooled thermal camera

Next, the perception functions for robot skin by using several types of sensors to detect pressure (Li, et al., 2017), temperature, and sliding. The r-GO temperature sensor is appropriate for robot and electronic skins and can be widely employed in IoT applications. The r-GO temperature sensor performed consistently under varying degrees of deformation (Liu et al., 2018). The r-GO temperature sensors can be embedded in the soil to monitor soil temperature at various depths. The sensors can help identify temperature variations that may affect seed germination, root growth, and microbial activity in the soil. r-GO temperature sensors can be employed to detect near-freezing temperatures and help prevent frost damage in agricultural fields or orchards. The r-GO temperature sensors in agriculture enables farmers to make informed decisions, optimize resource management, and enhance overall agricultural productivity. By closely monitoring temperature variations, farmers can implement timely interventions, mitigate risks, and improve the efficiency of various agricultural processes. Figure 26 show the possible applications of the r-GO temperature sensor that was transform into fabric.

Figure 30 r-GO temperature sensor (Fabrication)

Moreover, plant canopy temperature was measured in the field using infrared thermometers. The infrared thermometer (IRT) was a research prototype that combined a temperature sensor and a data recorder(Sui et al., 2012). An LM35 analogue temperature sensor (National Semiconductor, Santa Clara, CA) was used to monitor air temperature, and an MLX90614 IRT module (Melexis SA, Ieper, Belgium) was used to assess plant canopy temperature. Measurements were taken at one-hour intervals and recorded to the data logger's memory chip, which was then downloaded on frequent trips to a handheld computer.. Thermometer with infrared technology. On the hot day, plant tends to regulate its leaf temperature around 28 degrees Celsius by releasing its heat load through the surface of its leaves through a process called transpiration. To do that, plants need access to sufficient soil moisture. By continuously monitoring the leaf temperature using canopy temperature sensors, authors able to take advantage of that relationship between canopy temperature and soil water. Author also use the continuous canopy temperature data to better forecast the timing of an irrigation. This is another tool that growers can use to improve their water use efficiency. Figure 27 depicts IRT sensors installed in the field inside thick-walled PVC plastic enclosures.

Figure 31 Cotton plant canopy temperature measured with an infrared thermometer

A thermal resistor is a temperature sensor consisting of a known resistor, which varies with temperature, such as the temperature measurement of the platinum resistor. Thermal resistance is the method of contact temperature measurement, which has many advantages including high precision, easy operation and low cost. Currently there isn't that much research on using thermal resistance to measure sheet metal temperature. The leaf temperature sensor LT-1 M (Yu et al., 2016) (see Fig. 28) has a subminiature touch probe that measures leaf temperature. The Lightweight stainless steel wire clip holds a high precision glass pack thermistor. and the probe is very small and specially designed, which has almost no impact the natural temperature of the leaves of the plant.

Figure 32 The leaf temperature sensor LT-1 M

2.4.2 Humidity sensor in agriculture

Humidity sensors in agriculture are devices used to measure and monitor the moisture content or relative humidity in various agricultural settings. They provide valuable data on atmospheric moisture levels, allowing farmers and growers to make informed decisions regarding irrigation, ventilation, disease management, and overall crop health. Humidity sensors are commonly used in controlled environments, such as greenhouses and indoor grow rooms, where maintaining optimal humidity levels is crucial for plant growth. They help regulate humidity by providing real-time data on moisture content in the air, allowing growers to adjust ventilation, irrigation, and heating systems accordingly. Maintaining proper humidity levels promotes photosynthesis, transpiration, and overall plant health, preventing issues like wilting, disease, and pest infestations. Humidity plays a significant role in the development and spread of many plant diseases and pests. By monitoring humidity levels, farmers can take preventive measures, such as adjusting ventilation, implementing fans, or applying appropriate fungicides or pesticides, to mitigate disease and pest pressures.

Firstly, agricultural UAV or drones become one of the most useful agricultural instruments utilized in smart farming, especially in the ground sensing applications. The majority of soil moisture applications today are based on IoT networking (Chen et al., 2019) or wireless sensor networking (WSN) (Viani et al., 2017). The goal is to get agricultural parameters automatically without requiring human-to-human or human-to-computer interaction. Figure 29 depicts the GS-UAV-SC model for smart agricultural applications. It should be emphasised that the IoT soil moisture GS kits were put in the field and subsequently connected to the UAVSC to stream data to the internet. The data is then stored and computed on the cloud platform. Furthermore, the farmer/user can direct the UAV throughout the field to collect data from all sensors. The data can be viewed by the user via the mobile application's application programme interface (API). The suggested approach seeks to compensate for the low energy resulting from multiple GSs connectivity. Figure 30 depicts the structure model of the soil moisture GS prototype. It should be mentioned that this platform would be used if there was sunshine. In this investigation, capacitive soil moisture was used. The capacitive soil moisture GS varies the capacitance based on the water content of the soil.

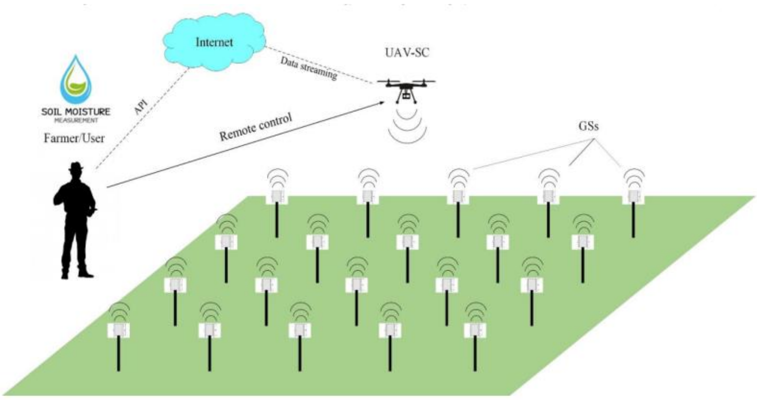


Figure 33 GS-UAV-SC model

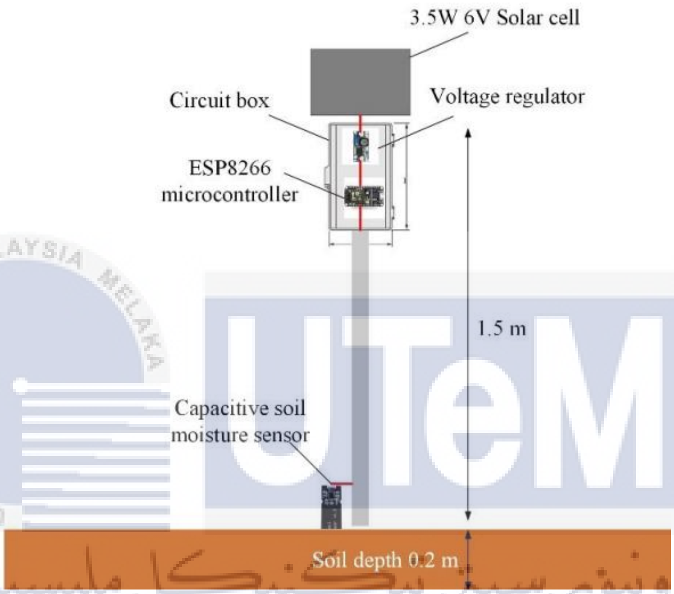


Figure 34 Soil moisture GS prototype

Figure 31 and 32 shows that real-time test:

Figure 35 Real-time test in Napier farm

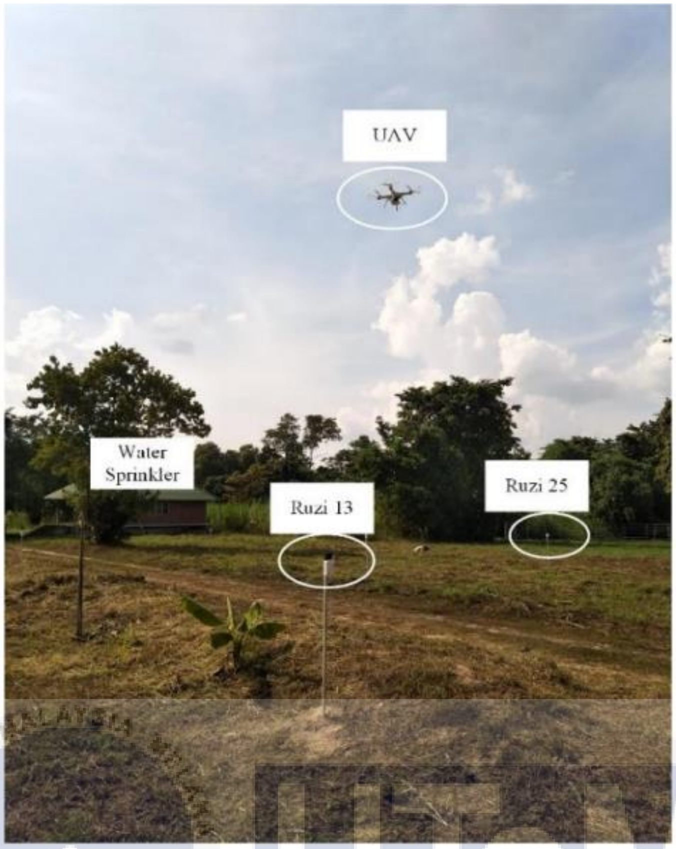


Figure 36 Real-time test in Ruzi farm.

Next, a transistor circuit as shown in Figure 33 can be used to implement the soil moisture sensor. The data trend of soil moisture is an advantage of the analogue output, this is critical for analysing soil moisture control in any condition. This is a basic automatic irrigation system. However, no irrigation settings, such as time, can be modified with this simple system. The system includes a soil moisture sensor placed in the soil at a desired depth. The sensor measures the moisture content in the soil and provides an electrical signal that represents the moisture level. As shown in Figure 34, the microcontroller-based irrigation system has been shown in practise.

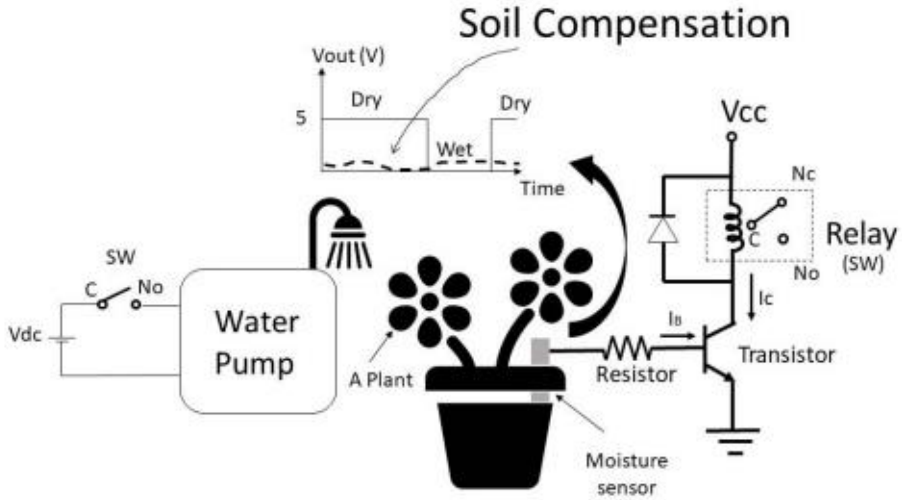


Figure 37 A soil moisture sensor and a transistor circuit are used in the traditional autonomous watering system.

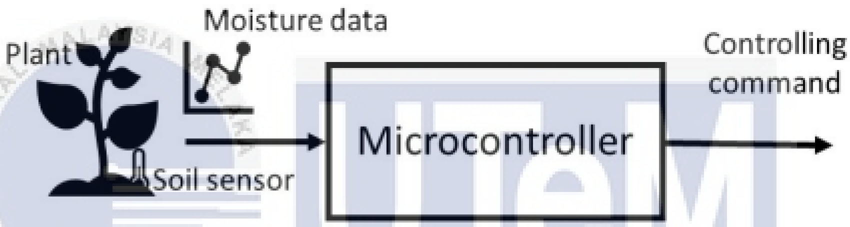


Figure 38 The traditional automatic watering system, which is based on a microprocessor and a soil moisture sensor.

Below shows the five components of the proposed IoT-based soil moisture sensor for smart farming:

- Soil moisture sensor
- ESP8266 NodeMCU (IoT controller)
- The user interface (Mobile Applications)
- Cloud platform: Thingspeak
- Data analytics: online MATLAB

The soil moisture sensor's duty is to measure soil moisture data and then communicate the electronic data to the analogue input of the IoT controller (ESP8266 NodeMCU). Figure 35 depicts a prototype IoT-based soil moisture sensor with a solar panel for field testing. Figures 37 and 38 depict real-time soil moisture data and soil moisture data analytics utilising Thingspeak/Matlab as the cloud platform.

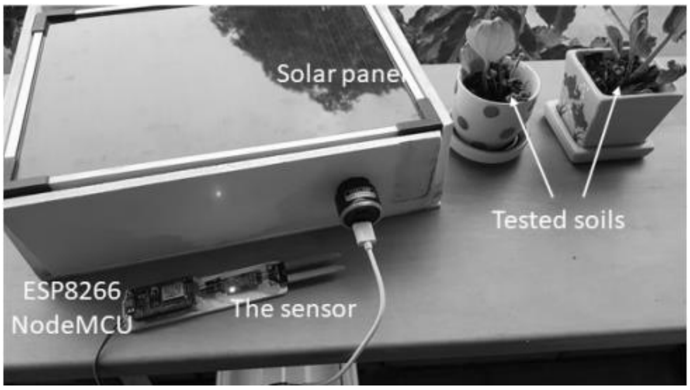
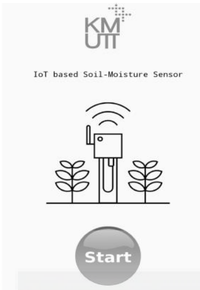


Figure 39 Soil moisture sensor Prototype



a) Figure 40 Mobile application monitoring and analyzing the soil moisture data

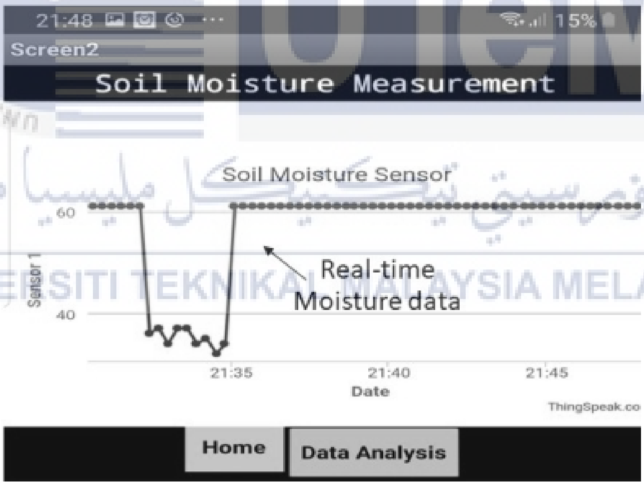


Figure 41 Mobile application monitoring and analyzing the soil moisture data

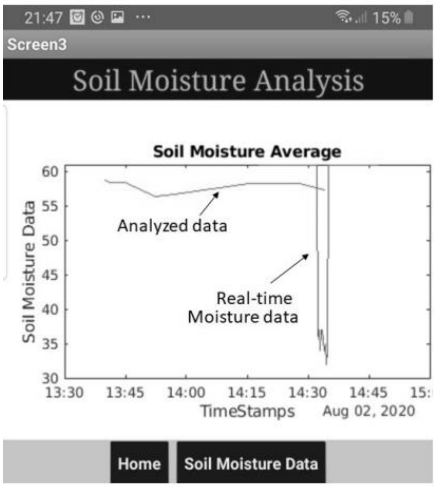


Figure 42 analyzed data and real-time moisture data.

Moreover, the HDC1010 digital humidity sensor is used and it provides accurate measurement of moisture level in environment at low power. It has excellent stability at high humidity (Prathibha et al., 2017). It has a typical accuracy of ±2% relative humidity and a resolution of 0.08% relative humidity. This level of accuracy is crucial for precise monitoring and control of humidity in agricultural environments. The HDC1010 sensor can operate in a wide range of humidity levels, typically from 0% to 100% relative humidity. By integrating the sensor with a microcontroller or IoT (Internet of Things) platform, farmers can receive timely alerts and take necessary actions to adjust ventilation, irrigation, or other environmental parameters. The HDC1010 sensor helps monitor and control humidity conditions that may favour the growth and spread of plant diseases or pests. The accurate and reliable data provided by the HDC1010 sensor can be used for research and analysis purposes in agriculture. Farmer can analyze long-term humidity trends, correlations with crop performance, or climate change impacts on agricultural systems. Figure 39 show HDC1010 digital humidity sensor.

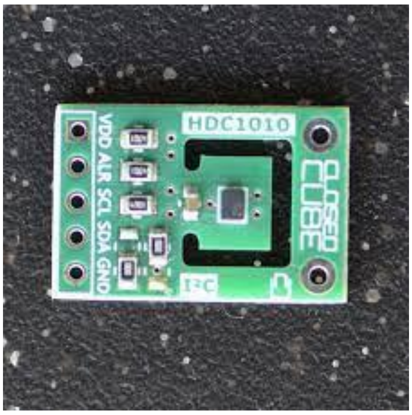


Figure 43 HDC1010 digital humidity sensor

In Turkey, agricultural irrigation consumes 75% of current fresh water (Dursun M, 2010). As a result, efficient water management is critical in irrigated agricultural cropping systems (Kim Y, 2009) (Sezen SM, 2010). In the designed system, 10 HS coded pre-calibrated Decagon Soil Moisture Sensors were employed to assess soil water content (Figure 40). The 10 HS requires very little power and has a very good resolution (Dursun & Ozden, 2011). Using a capacitance approach, the 10 HS determines the volumetric water content (VWC) of the soil by measuring its dielectric constant. It helps prevent overwatering or underwatering, leading to efficient water usage and improved crop health because water has a significantly larger dielectric constant than air or soil minerals, the dielectric constant of soil is a sensitive indicator of volumetric water content. The 10HS sensor indirectly assists in nutrient management by providing insights into soil moisture conditions. There are two types of sensor output: analogue and digital outputs (Puengsungwan, 2020). Output value produced by the sensor is like an analog data value and converted to digital data by the PIC and sent to the PC via serial ports. Two LEDs for notifications have been added in this system.

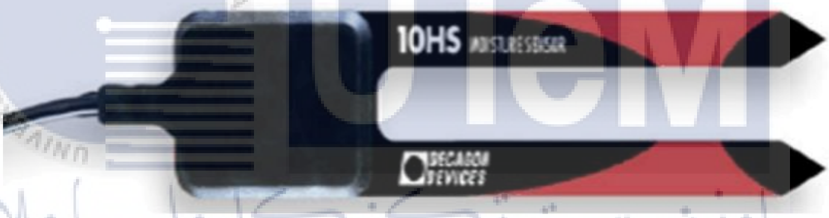


Figure 44 Soil moisture sensor

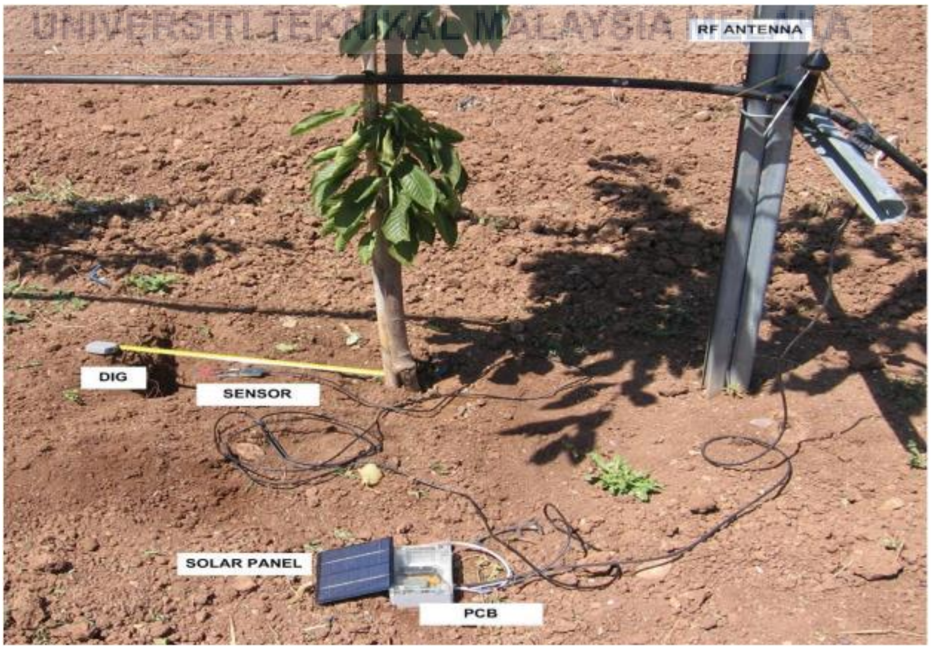


Figure 45 System’s application around dwarf cherry tree with sensor unit.

2.4.3 pH sensor in agriculture

pH sensors are widely used in agricultural research and education. pH sensors measure the acidity or alkalinity of the soil or nutrient solution. They help growers monitor and maintain the optimal pH range for plants' nutrient uptake. pH sensors are designed to measure a wide range of pH values typically found in agricultural settings. pH sensor also helping farmers assess soil conditions, select suitable crops, and make informed decisions about soil amendments and fertilization strategies. By monitoring pH levels, growers can adjust pH levels in the soil or nutrient solution to optimize nutrient availability and prevent nutrient deficiencies or toxicities. They can cover pH ranges from acidic (pH 0-6), neutral (pH 7), to alkaline (pH 8-14). Different crops have varying pH preferences for optimal growth. For the best absorption of magnesium and zinc salts and other micronutrients, the temperate crops in this study need a mildly acidic nutrition solution with a pH level range from 5.5 to 5.9 (Tik, 2009).

Soil pH ManagerTM by Veris Technologies is a real-time sensor for mapping soil pH. While in direct contact with the soil material, it automatically collects soil samples and assesses soil pH. 2011 (Schirrmann et al., 2011). It consists of three key components: a hydraulic soil sampling system, a pH electrode measurement system, and a water wash system (see Figure 21). A hydraulic cylinder on a parallel connection (2) lowers the soil sampler shoe (1) into the dirt while driving. The sample depth and time are also adjustable, but are usually set to 0.01 m and 2 seconds, respectively. The front of the shoe slices the soil material with a cone (3) while in the soil, resulting in dirt core flow down the shoe's trough. The shoe is then raised, allowing the soil sample to make direct contact with two antimony pH electrodes (5). At the same time, the shoe is being cleaned in the front using a scraper (4). The shoe's upward and downward movement is controlled by a proximity sensor. Because the measurement is performed on untreated, naturally damp soil material, no solution is put into the soil prior to its contact with the electrodes. The pH value is then computed by averaging the voltage outputs of the two electrodes. Voltage is converted to pH units using a calibration process that requires measuring two standard solutions with established pH values of 4 and 7. The pH measurement time ranges from 7 to 25 seconds. The soil sample is expelled after the pH measurement by additional soil material flowing through the cone. Two wash nozzles rinse the electrodes, and the rinse water is held in a 359

L tank with electric water pumps. Row cleaners are employed to clear crop debris in front of the sampler shoe, and furrow closers are present to fill the furrow left by the shoe. The sampling process and pH electrode signals are managed by an external controller, with data relayed to a user interaction device. The controller can be set to operate manually. When the sampler shoe is removed from the soil during field operations, differential GPS coordinates are acquired.

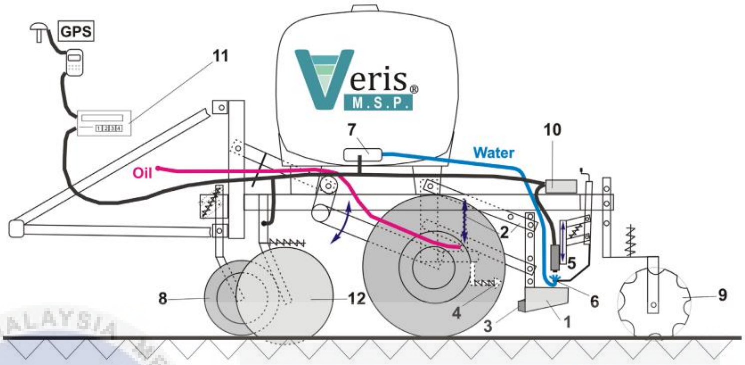


Figure 46 The Veris Multi Sensor Platform (MSP) schematic, which includes the Soil pH Manager and the Soil Electrical Conductivity Surveyor.

Calibration of the antimony electrode with many buffer solutions of known pH value is required for precise results (Bates, 1961). The sensor needs to be calibrated on a regular basis in order to provide precise readings. In order to accomplish decalibration and provide distilled water to clean the probe, three ready-to-use pH reference buffer solutions are needed. In order to execute both the calibration procedure and the measurement of the pH value of the nutrient solution in our hydroponic duct, Figure 23 depicts an auto-calibrated pH sensor with the micro-pumps that give the measured values of necessary liquids to the container.

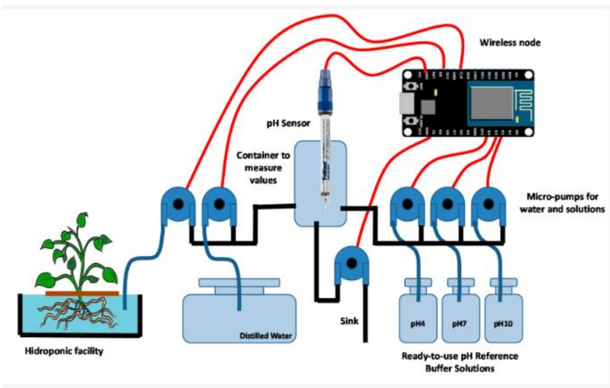


Figure 47 Auto-calibrated pH sensor

2.5 Relative sensor technology in agriculture

Sensor technology in agriculture covers a wide range of variables and measurements. The range values for different sensor technologies in agriculture can vary depending on the specific application and sensor type.

2.5.1 Relative Temperature

The temperature range in agriculture can vary depending on the specific crops, regions, and growing conditions. Different crops have different temperature preferences and requirements for optimal growth.

Germination Temperature Range:

- This refers to the temperature range required for seeds to germinate and initiate plant growth.
- The typical range is between 15°C to 30°C (59°F to 86°F) for many common crops, but specific crops may have different temperature requirements.

Vegetative Growth Temperature Range:

- This range covers the temperatures suitable for the vegetative growth stage of plants.

- For most crops, the optimal temperature range lies between 20°C to 30°C (68°F to 86°F).
- Cooler-season crops may tolerate lower temperatures, while warm-season crops may prefer higher temperatures.

Flowering and Fruit Set Temperature Range:

- During the flowering and fruit set stage, certain crops require specific temperature conditions for successful pollination and fruit development.
- The optimal range varies depending on the crop but is generally around 15°C to 35°C (59°F to 95°F).

- Extreme temperatures, especially high temperatures, can negatively impact pollination and fruit set.

Cold Tolerance and Frost Susceptibility:

- Some crops have varying degrees of cold tolerance and can withstand lower temperatures for short periods.
- The critical temperature threshold for frost susceptibility varies among crops, but it is generally around 0°C to -4°C (32°F to 24.8°F) for many common crops.
- Frost-sensitive crops may sustain damage or even crop loss when exposed to temperatures below their tolerance thresholds.

Heat Stress Threshold:

- Heat stress can occur when temperatures rise above the optimal range for crop growth.
- The specific heat stress threshold varies among crops, but prolonged exposure to temperatures above 30°C to 35°C (86°F to 95°F) can negatively impact crop growth, flowering, and yield.

For example, greenhouse cultivation of tomatoes five different growth stages including germination, seedling and growth nutrition, early fruiting, ripe fruiting identified by Decision Making System for growing high quality vegetables from the Ohio Agricultural Research and Center Development (Shamshiri, 2013). Scientific method, especially risk-benefit theory evaluation and decision support theory used. This program is designed to support recommendations for growing real plants that can be done from the optimal value. The program is accessed in two ways: as a graphical introduction and as an interactive decision support system. Definition of successful harvests with this program are of high yield and quality harvest. On the other hand, there is also the possibility that the harvest will fail low yield, high or inferior crop. With this program Ideal values of temperature and relative humidity for various tomatoes Dependent on climate and light conditions as well as certain growth stages Listed in Table 1.

Growth Stage	Temperature °C			RH (%)	Ideal VPD (kPa)		
	Sunny	Cloudy	Night		Sunny	Cloudy Day	Night
Germination	26	26	26	75	0.840	0.840	0.840
Seeding	27	24	20	[75,76]	[0.840,0.891]	0.746	0.584
Vegetable	[24,27]	23	19	[60,80]	[0.855,0.963]	[0.843,0.955]	[0.841,0.935]
Early fruiting	[24,27]	23	19	[60,80]	[0.855,0.963]	[0.843,0.955]	[0.841,0.935]
Mature Fruiting	[24,27]	23	19	[60,80]	[0.855,0.963]	[0.843,0.955]	[0.841,0.935]

Table 1 Ideal condition in the growth stages of tomato

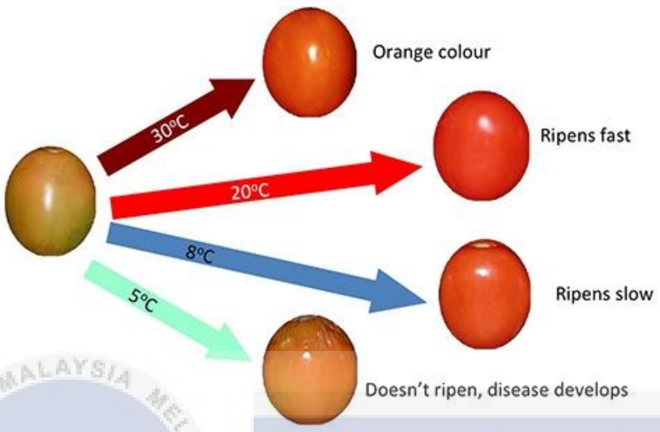


Figure 48 shows tomato condition in each temperature range

Common name	Scientific name	Maximum Cardinal	Optimal range	Maximum Cardinal
Alfalfa	Medicago sativa L.	8	24-26	36
Asparagus	Asparagus officnalis L.	4	18-22	28
Banana	Musa seppe. L.	12	25-30	40
Barley	Hordeum vulgare L.	2	18-28	34
Bean	Phaseolus vulgaris L.	10	24-30	36
Carrot	Daucus carota L.	3	16-22	28
Lemon	Citrus limon L.	13	23-30	35
Melon	Cucumis mela L.	15	25-35	40
Onion	Allium cepa L.	2	20-28	34
Sweet potato	Hipomea batata L.	15	25-33	38

Pineapple	Ananas comosus L.	15	22-30	35
Tomato	Solanum lycopersicum L.	12	22-26	35
Potato	Solanum tubrerosum L.	4	14-23	33
Rice	Oryza sativa L.	12	25-32	38
Safflower	Carthamus tinctorius L.	10	18-28	35
Soft Wheat	Triticum aestivum L.	2	18-26	31
Sorghum	Sorghum bicolor L.	12	24-30	36
Soybean	Glycine max L.	10	20-28	34
Strawberry	Fragaria X ananasia L.	4	15-20	28

Table 2 Cardinal temperatures for some crops with the sources of information used(Ferrante & Mariani, 2018)

2.5.2 Relative Humidity

In agriculture, the optimal relative humidity can vary depending on the specific crop, growth stage, and environmental conditions. However, here are some general guidelines for humidity ranges in agriculture:

Seed Germination:

- Range: 70% to 90% relative humidity.
- During the germination phase, higher humidity levels help promote seed moisture absorption and facilitate the germination process.

Vegetative Growth:

- Range: 50% to 70% relative humidity.
- Moderate humidity levels are generally suitable for vegetative growth, allowing for proper transpiration and nutrient uptake by the plants.

Flowering and Fruit Development:

- Range: 40% to 60% relative humidity.
- Lower humidity levels during flowering and fruit development help prevent excessive moisture and reduce the risk of fungal diseases.

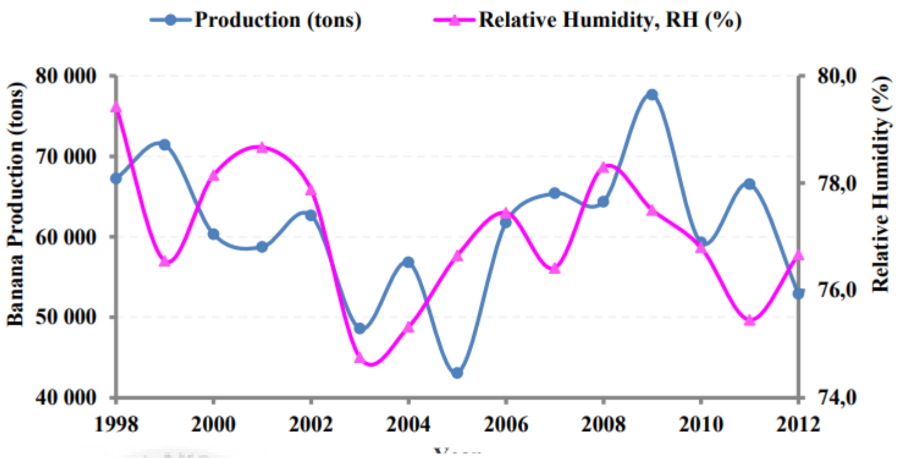


Figure 49 The graph of banana production (tons) against RH (%).

Figure 45 shows a general agreement between relative humidity and banana production. For example, a low peak in banana production in 2004 was followed by a sharp drop to a previous 48,085 tons peaked in 2009, after which there was a steady decline in production interrupted by a slight increase in 2011. In general, banana production is high to moderate high humidity, while production decreases as relative humidity decreases (Salau et al., 2016); The banana grows well in this study range within the HR of 74-79%.

For the investigation, the potato cultivars Norland, Russet Burbank, and Denali were used(Wheeler et al., 1989). 'Norland' matures early, while 'Russet Burbank' and 'Denali' mature late. 'Denali' has been demonstrated to produce well under continuous irradiation (Wheeler and Tibbitts,' unpublished data).

Growth characteristic	Relative humidity (%)	Cultivar		
		Russet Burbank	Norland	Denali
Leaf dry weight (g)	50	122 ± 18	119 ± 21	106 ± 16
	85	110 ± 9	98 ± 10	89 ± 10
Stem dry weight (g)	50	65 ± 13	51 ± 14	50 ± 14
	85	69 ± 6	40 ± 7	44 ± 7
Tuber dry weight (g)	50	28 ± 25	27 ± 26	66 ± 21
	85	42 ± 21	67 ± 11	105 ± 16
Total plant dry weight (g)	50	223 ± 16	204 ± 15	232 ± 18
	85	229 ± 9	210 ± 14	249 ± 14
Leaf area (m^2)	50	3.57 ± 0.35	3.22 ± 0.64	3.41 ± 0.99
	85	3.13 ± 0.40	2.57 ± 0.38	2.60 ± 0.46

Table 3 Effects of relative humidity on potato growth

2.5.3 Relative pH

In agriculture, the optimal pH range can vary depending on the specific crop and soil type. Based on the values of PH, its plant's growth will depend as follows:

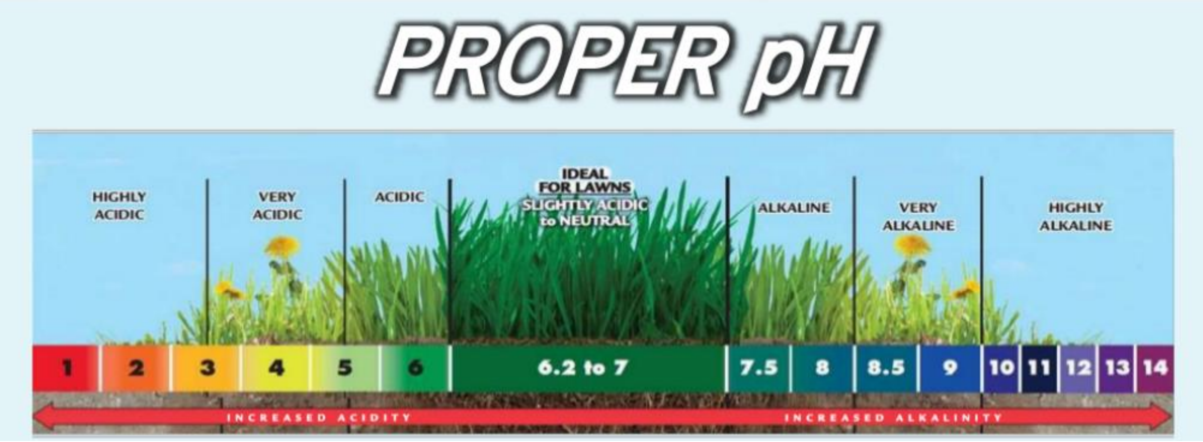


Figure 50 pH values and its plants growth

These ranges are general guidelines, and specific crops may have more specific pH preferences. Additionally, soil type and regional conditions can influence the natural pH of the soil. Conducting soil tests and consulting with local agricultural experts can help determine the optimal pH range for specific crops in your agricultural context. Adjusting soil pH, if necessary, can be done through various methods such as using soil amendments or pH-adjusting fertilizers.

2.6 Comparison between previous research paper

In terms of method and application, the table compares prior research journals or research papers.

Table 4 compares prior journals or studies on monitoring system in greenhouse.

No.	Author	Method	Application
1	(Mellit et al., 2021)	Type: Design (Smart Greenhouse) Sensor: CO2 and air quality sensors (CCS811), air temperature and	This research designs a novel prototype for remote greenhouse monitoring. The prototype enables

		humidity sensor (DHT11), light sensor (BH1750), Photovoltaic power sensor, Capacitive soil moisture sensor, Soncentration and ultrasonic sensor	the construction of an adequate artificial greenhouse environment, including water irrigation, ventilation, light intensity, and CO2 concentration.
2	(Rustia & Lin, 2017)	An IoT-based wireless imaging and sensor node system for remote greenhouse pest monitoring Sensor: Raspberry Pi 3, Raspberry Pi camera, and a multi-environmental	The device was created to concurrently measure environmental conditions and continually count the pest insects found on yellow sticky notes dispersed around various places.
3	(Ehret et al., 2001)	Automated monitoring of greenhouse crops David Sensor: monitor climate parameters, especially temperature, humidity and light	In order to increase quality and productivity and conserve resources, this review will cover potential applications for equipment that directly evaluate crop status in commercial greenhouses.
4	(Liang et al., 2018)	Greenhouse Environment dynamic Monitoring system based on WIFI	The wifi-based monitoring system converts the TTL

		Sensor: Temperature, Humidity, light intensity sensor	signal on the sensor module into a wireless WIFI signal via a WIFI module, after which it establishes a network connection and transmits data to the server.
5	(Satpute, 2018)	IOT Based Greenhouse Monitoring System Sensor: DHT 11 sensor, soil moisture sensor, LDR sensor module.	This study presents a system for monitoring and controlling the system in a greenhouse utilising a temperature sensor, a humidity sensor, a light intensity sensor, and a soil moisture sensor.
6	(Guo et al., 2010)	Greenhouse Monitoring System Based on a Wireless Sensor Network Sensor: temperature and humidity sensor	The purpose of this experiment was to investigate the degree of variety of the microclimate conditions in the greenhouse environment.
7	(Rustia et al., 2020)	Application of an image and environmental sensor network for automated greenhouse insect pest monitoring	The method provides an effective tool for observing long-term insect pest

		Sensor: Raspberry Pi camera, temperature and humidity sensor, light intensity sensor	behaviour as well as practical applications in integrated pest management (IPM).
8	(Mudaliar & Sivakumar, 2020)	IoT Based Real Time Greenhouse monitoring system using Raspberry Pi Component details: humidity and temperature sensor DHT 11, Soil Moisture sensor, Rain detector sensor, Smoke detector (MQ sensor), Buzzer	The raspberry pi will then transmit the data using IoT and the processed data will be displayed on IoT.

2.7 Summary

Technology for monitoring systems in agriculture enables real-time data collection, analysis, and decision-making to optimize farm management practices. Various types of sensors are deployed in agriculture to monitor parameters such as soil moisture, temperature, humidity, light intensity, pH levels, nutrient content, and weather conditions. These sensors provide continuous data, enabling farmers to make informed decisions regarding irrigation, fertilization, and pest control. IoT devices, including sensors and actuators, are interconnected to create a network that collects and transmits data. IoT enables remote monitoring of environmental conditions, crop health, and equipment performance. It allows farmers to access real-time data from their smartphones or computers and make timely interventions as needed. By utilizing monitoring technologies in agriculture, farmers can collect and analyze data, gain insights into crop conditions, optimize resource usage, detect issues early, and make informed decisions for improved productivity, sustainability, and profitability.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will go through the project research and methods in general. This chapter covers all of the information, explanations, and methodologies utilised to create this project. This chapter is also crucial in ensuring that the project runs successfully by following to the correct workflow. As a result, this chapter will go over the methods and phases of completing this project.

3.2 Methodology

In the initial phase, we will develop a sophisticated leaf imaging system designed to capture detailed images of greenhouse leaves. This involves the integration of cameras with the MATLAB environment to facilitate seamless data acquisition and processing. Advanced image processing algorithms will then be implemented in MATLAB, focusing on preprocessing to enhance image quality and subsequent feature extraction for key indicators like colour, texture, and shape. These algorithms will enable the system to classify leaves based on their health conditions.

Simultaneously, machine learning techniques integrated into MATLAB will play a crucial role in the project. A diverse dataset representing various leaf conditions, including diseases and stress factors, will be used to train machine learning models. These models will subsequently be integrated into the leaf health monitoring system to perform real-time classification, providing a dynamic and adaptive aspect to the monitoring process.

The project's robustness will be ensured through thorough validation and testing. Dataset splitting and cross-validation techniques within MATLAB will be employed to assess the accuracy and reliability of the developed algorithms. Additionally, a feedback loop will be established to continuously refine the algorithms based on testing outcomes, enhancing the overall performance of the system.

The generated insights from the leaf health monitoring system will be stored in a comprehensive database within MATLAB. This database will correlate identified leaf conditions with potential causes, drawing from existing agricultural knowledge. The system will then generate detailed reports on the identified leaf conditions, their causes, and recommended corrective measures. These reports will be seamlessly integrated into a user-friendly interface, providing accessible information for greenhouse operators and facilitating informed decision-making.

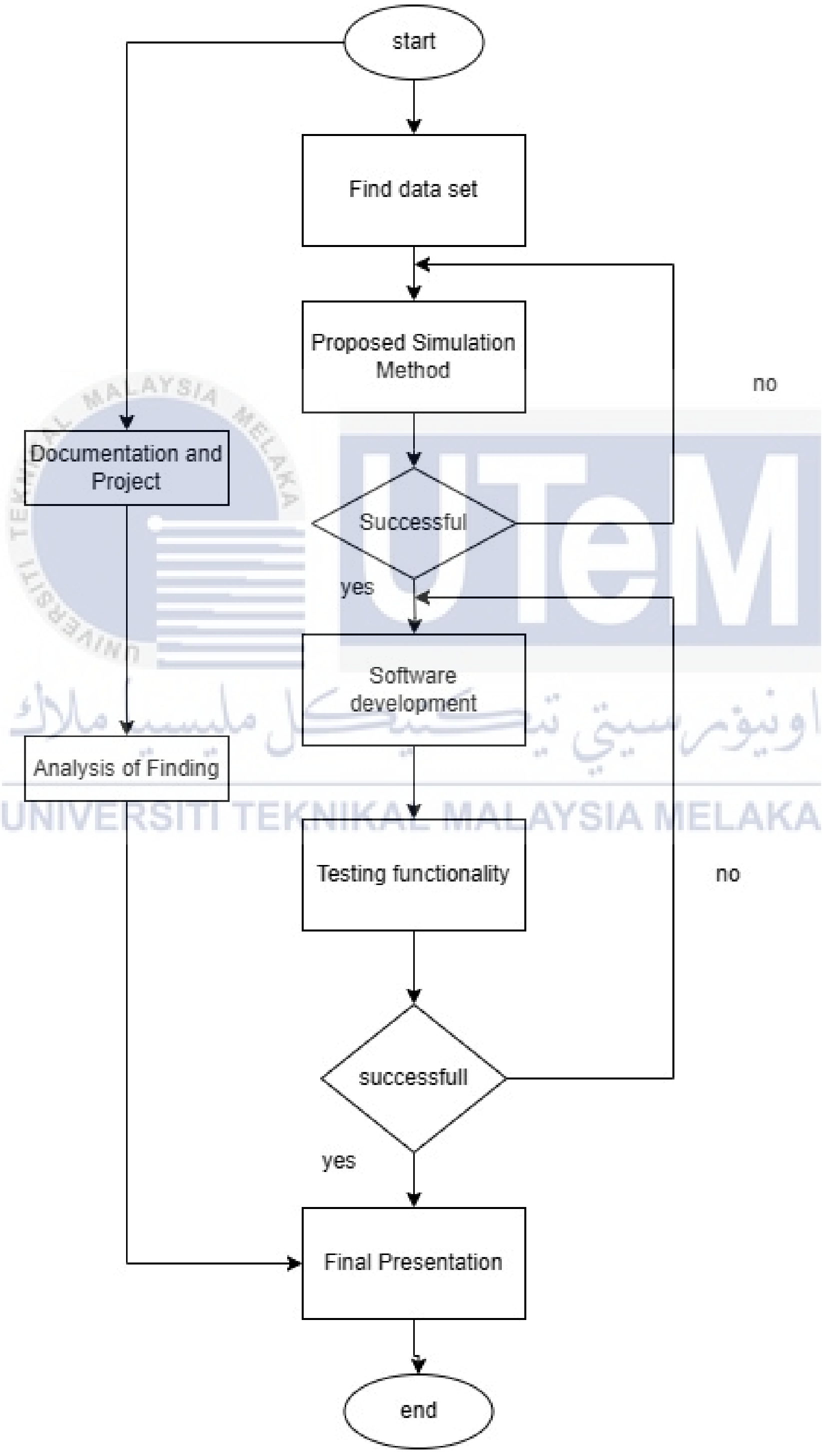
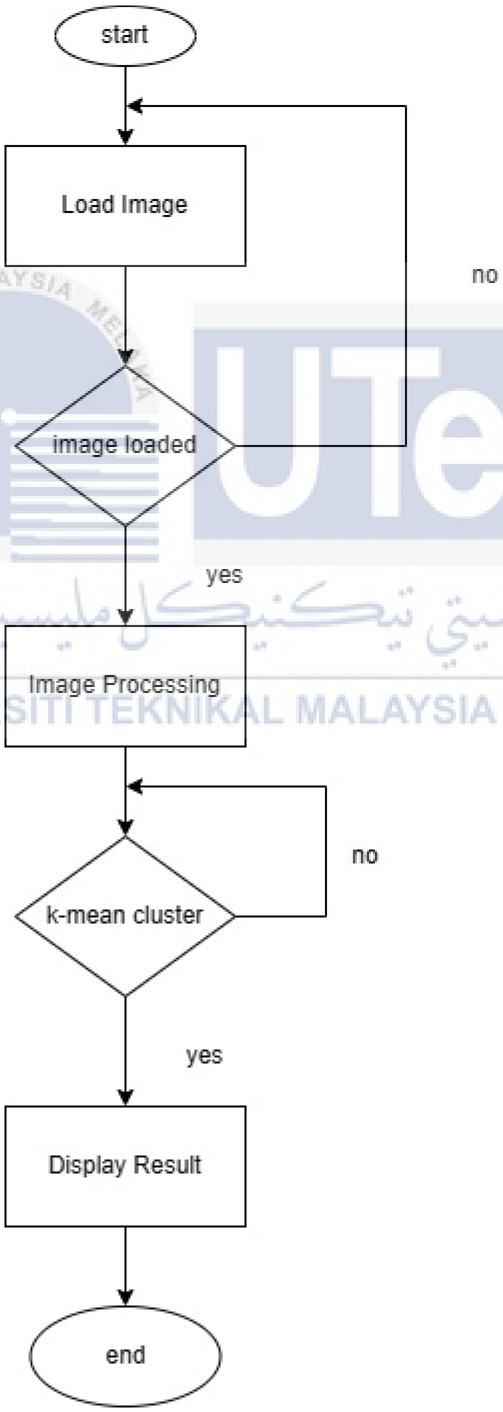


Figure 51 Overall project flowchart

3.3 Software Development

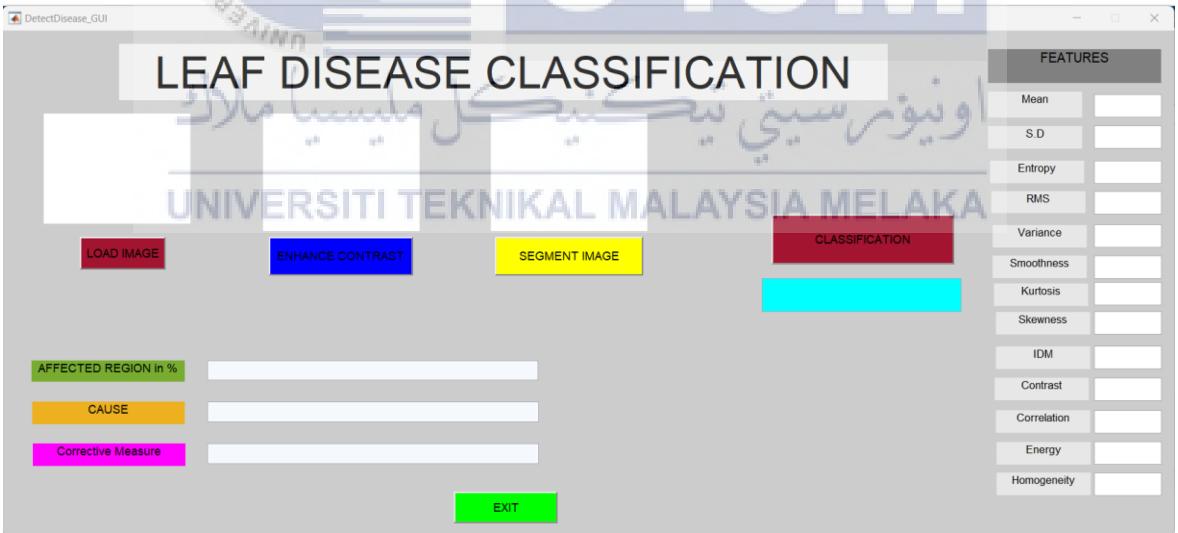
This software development initiative aims to revolutionize greenhouse management by leveraging the advanced capabilities of MATLAB, integrating sophisticated leaf imaging systems, powerful image processing algorithms, and machine learning models to create a comprehensive Greenhouse Leaf Health Monitoring System.

3.3.1 Overall Project Flowchart



The flowchart for the greenhouse leaf health monitoring system using MATLAB begins with the "Start" symbol, signifying the initiation of the monitoring process. Subsequently, the system progresses to the "Load Image" process, where an image of greenhouse leaves is loaded into the MATLAB environment. The "Image Loaded?" decision point follows to verify the successful loading of the image. In case of successful loading (Yes), the system advances to the "Image Processing" step, involving the implementation of MATLAB algorithms for leaf health analysis. Processed results are then displayed in the GUI through the "Display Results" step, which may include updating an axes component with the processed image or presenting health metrics in a text box. The process proceeds to the "End" symbol, marking the completion of the greenhouse leaf health monitoring system. This flowchart serves as a high-level representation of the primary steps involved in the project, and its structure can be further customized based on specific details and decision points in the MATLAB GUI implementation.

3.3.2 MATLAB GUI



The development of a MATLAB Graphical User Interface (GUI) for the greenhouse leaf health monitoring system involves the utilization of tools such as GUIDE or App Designer to architect a user-centric interface, incorporating essential components such as buttons, axes, and text boxes. Callback functions must be meticulously defined for interactive elements, such as the "Load Image" button, in order to effectively manage user interactions. The implementation of image processing algorithms using pertinent MATLAB functions like imread and the Image Processing Toolbox is imperative for comprehensive leaf health analysis. The presentation of processed results within the GUI, coupled with user guidance, error handling mechanisms, and rigorous testing, is essential to ensure optimal functionality. The documentation of the code is recommended for future reference, and upon satisfaction, deployment of the GUI can be accomplished through the creation of standalone applications or by sharing the MATLAB script.

3.3.3 MATLAB CODING

```
% Project Title: Plant Leaf Disease Detection & Classification
% Author: Cley Alexsius Jarius
% Contact: cleyalexsiusjarius01@gmail.com

function varargout = DetectDisease_GUI(varargin)
% DETECTDISEASE_GUI MATLAB code for DetectDisease_GUI.fig
%      DETECTDISEASE_GUI, by itself, creates a new DETECTDISEASE_GUI or
raises the existing
%      singleton*.
%
%      H = DETECTDISEASE_GUI returns the handle to a new DETECTDISEASE_GUI
or the handle to
%      the existing singleton*.
%
%      DETECTDISEASE_GUI('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in DETECTDISEASE_GUI.M with the given input
arguments.
%
%      DETECTDISEASE_GUI('Property','Value',...) creates a new
DETECTDISEASE_GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs
are
%      applied to the GUI before DetectDisease_GUI_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to DetectDisease_GUI_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help DetectDisease_GUI

% Last Modified by GUIDE v2.5 11-Jan-2024 10:56:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @DetectDisease_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @DetectDisease_GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before DetectDisease_GUI is made visible.
function DetectDisease_GUI_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% --- Executes just before DetectDisease_GUI is made visible.
function DetectDisease_GUI_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to DetectDisease_GUI (see VARARGIN)

% Choose default command line output for DetectDisease_GUI
handles.output = hObject;
ss = ones(300,400);
axes(handles.axes1);
imshow(ss);
axes(handles.axes2);
imshow(ss);
axes(handles.axes3);
imshow(ss);
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes DetectDisease_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = DetectDisease_GUI_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%clear all
%close all
clc
[filename, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'}, 'Pick
a Leaf Image File');
I = imread([pathname,filename]);
I = imresize(I,[256,256]);
I2 = imresize(I,[300,400]);
axes(handles.axes1);
imshow(I2);title('Query Image');
ss = ones(300,400);
axes(handles.axes2);
imshow(ss);
axes(handles.axes3);
imshow(ss);
handles.ImgData1 = I;
guidata(hObject,handles);

% --- Executes on button press in pushbutton3.
function pushbutton2_Callback(hObject, eventdata, handles)
I3 = handles.ImgData1;
I4 = imadjust(I3,stretchlim(I3));
I5 = imresize(I4,[300,400]);
axes(handles.axes2);
imshow(I5);title(' Contrast Enhanced ');
handles.ImgData2 = I4;
guidata(hObject,handles);
```

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
I6 = handles.ImgData2;
I = I6;
%% Extract Features

% Step 1: Allow the user to select a folder
selectedFolder = uigetdir();

% Check if the user clicked Cancel
if isequal(selectedFolder, 0)
    disp('User cancelled the operation.');
    return; % Exit the script
else
    [~, folderName] = fileparts(selectedFolder); % Extract folder name
end

% Step 2: List of items and their representations
items = {
    'Alternaaria Alternata' 'Fungal Pathogen: Infects plants, causes leaf
spots and blights.';
    'Anthracnose'           'Caused by Colletotrichum fungi; infects many
plants.';
    'Bacterial Blight'      'Caused by various bacteria; affects different
plants.';
    'Cercospora Leaf Spot'  'Fungal infection by Cercospora species.';
    'Healthy leaves'        'Healthy leaves';
    % Add other items as needed
};

% Display the available items
disp('List of Items:');
for i = 1:size(items, 1)
    fprintf('%d. %s\n', i, items{i, 1});
end

% Step 3: Select an item
choice = input('x: ');

if choice >= 1 && choice <= size(items, 1)
    selectedItem = items{choice, 2};
    fprintf('Disease cause: %s\n', selectedItem);
    associatedItem = items{choice, 1};
else
    disp('Invalid choice.');
    return; % Exit the script
end

% Display the selected folder and associated disease classification
disp(['Selected Folder: ', folderName]);
disp(['Associated Disease Classification: ', associatedItem]);

% Continue with the remaining steps (selecting an image and displaying it)
% ...


% Step 2: Allow the user to select an image file from the selected folder
[fileName, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'}, 'Pick a
Leaf Image File');

% Check if the user clicked Cancel
if isequal(fileName, 0) || isequal(pathname, 0)
    disp('User cancelled the operation.');
    return; % Exit the script
end
```

```
% Step 2: Allow the user to select an image file from the selected folder
[fileName, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'}, 'Pick a Leaf
Image File');

% Check if the user clicked Cancel
if isequal(fileName, 0) || isequal(pathname, 0)
    disp('User cancelled the operation.');
    return; % Exit the script
end

% Step 3: Display the name of the selected image file and the folder name
disp(['Selected image: ', fileName]);
disp(['Disease Classificaiton: ', folderName]);

% Step 4: Display the selected image
fullImagePath = fullfile(pathname, fileName);
imageData = imread(fullImagePath); % Read the image data
imshow(imageData); % Display the image
title(['Selected Image: ', fileName], 'Interpreter', 'none'); % Display the
image title

% ... (rest of your code for image segmentation, feature extraction, and GUI
updating)

% Function call to evaluate features
%[feat_disease seg_img] =  EvaluateFeatures(I)

% Color Image Segmentation
% Use of K Means clustering for segmentation
% Convert Image from RGB Color Space to L*a*b* Color Space
% The L*a*b* space consists of a luminosity layer 'L*', chromaticity-layer 'a*'
and 'b*'.
% All of the color information is in the 'a*' and 'b*' layers.
cform = makecform('srgb2lab');
% Apply the colorform
lab_he = applycform(I,cform);

% Classify the colors in a*b* colorspace using K means clustering.
% Since the image has 3 colors create 3 clusters.
% Measure the distance using Euclidean Distance Metric.
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);
nColors = 3;
[cluster_idx cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
                                      'Replicates',3);
%[cluster_idx cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3);
% Label every pixel in tha image using results from K means
pixel_labels = reshape(cluster_idx,nrows,ncols);
%figure,imshow(pixel_labels,[]), title('Image Labeled by Cluster Index');

% Create a blank cell array to store the results of clustering
segmented_images = cell(1,3);
% Create RGB label using pixel_labels
rgb_label = repmat(pixel_labels,[1,1,3]);

for k = 1:nColors
    colors = I;
    colors(rgb_label ~= k) = 0;
    segmented_images{k} = colors;
end



figure,subplot(2,3,2);imshow(I);title('Original Image');
subplot(2,3,4);imshow(segmented_images{1});title('Cluster 1');
subplot(2,3,5);imshow(segmented_images{2});title('Cluster 2');
subplot(2,3,6);imshow(segmented_images{3});title('Cluster 3');
set(gcf, 'Position', get(0,'Screensize'));
set(gcf, 'name','Segmented by K Means', 'numbertitle','off')
% Feature Extraction
pause(2)
```

```
for k = 1:nColors
    colors = I;
    colors(rgb_label ~= k) = 0;
    segmented_images{k} = colors;
end



figure,subplot(2,3,2);imshow(I);title('Original Image');
subplot(2,3,4);imshow(segmented_images{1});title('Cluster 1');
subplot(2,3,5);imshow(segmented_images{2});title('Cluster 2');
subplot(2,3,6);imshow(segmented_images{3});title('Cluster 3');
set(gcf, 'Position', get(0,'Screensize'));
set(gcf, 'name','Segmented by K Means', 'numbertitle','off')
% Feature Extraction
pause(2)
x = inputdlg('Enter the cluster no. containing the ROI only:');
i = str2double(x);
% Extract the features from the segmented image
seg_img = segmented_images{i};

% Convert to grayscale if image is RGB
if ndims(seg_img) == 3
   img = rgb2gray(seg_img);
end
%figure, imshow(img); title('Gray Scale Image');

% Evaluate the disease affected area
black = im2bw(seg_img,graythresh(seg_img));
%figure, imshow(black);title('Black & White Image');
m = size(seg_img,1);
n = size(seg_img,2);

zero_image = zeros(m,n);
%G = imoverlay(zero_image,seg_img,[1 0 0]);

cc = bwconncomp(seg_img,6);
diseasedata = regionprops(cc,'basic');
A1 = diseasedata.Area;
sprintf('Area of the disease affected region is : %g%',A1);

I_black = im2bw(I,graythresh(I));
kk = bwconncomp(I,6);
leafdata = regionprops(kk,'basic');
A2 = leafdata.Area;
sprintf(' Total leaf area is : %g%',A2);

%Affected_Area = 1-(A1/A2);
Affected_Area = (A1/A2);
if Affected_Area < 0.1
    Affected_Area = Affected_Area+0.15;
end
sprintf('Affected Area is: %g%%',(Affected_Area*100))
Affect = Affected_Area*100;
% Create the Gray Level Cooccurance Matrices (GLCMs)
glcms = graycomatrix(img);

% Derive Statistics from GLCM
stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(seg_img);
Standard_Deviation = std2(seg_img);
Entropy = entropy(seg_img);
RMS = mean2(rms(seg_img));
%Skewness = skewness(img)
Variance = mean2(var(double(seg_img)));
a = sum(double(seg_img(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(seg_img(:)));
Skewness = skewness(double(seg_img(:)));
% Inverse Difference Movement
m = size(seg_img,1);
```

```
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(seg_img);
Standard_Deviation = std2(seg_img);
Entropy = entropy(seg_img);
RMS = mean2(rms(seg_img));
%Skewness = skewness(img)
Variance = mean2(var(double(seg_img)));
a = sum(double(seg_img(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(seg_img(:)));
Skewness = skewness(double(seg_img(:)));
% Inverse Difference Movement
m = size(seg_img,1);
n = size(seg_img,2);
in_diff = 0;
for i = 1:m
    for j = 1:n
        temp = seg_img(i,j)./(1+(i-j).^2);
        in_diff = in_diff+temp;
    end
end
IDM = double(in_diff);

feat_disease = [Contrast,Correlation,Energy,Homogeneity, Mean,
Standard_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness,
IDM];
I7 = imresize(seg_img,[300,400]);
axes(handles.axes3);
imshow(I7);title('Segmented ROI');
set(handles.edit3,'string',Affect);
set(handles.edit2,'string',folderName);
set(handles.edit4,'string',selectedItem);
set(handles.edit5,'string',Mean);
set(handles.edit6,'string',Standard_Deviation);
set(handles.edit7,'string',Entropy);
set(handles.edit8,'string',RMS);
set(handles.edit9,'string',Variance);
set(handles.edit10,'string',Smoothness);
set(handles.edit11,'string',Kurtosis);
set(handles.edit12,'string',Skewness);
set(handles.edit13,'string',IDM);
set(handles.edit14,'string',Contrast);
set(handles.edit15,'string',Correlation);
set(handles.edit16,'string',Energy);
set(handles.edit17,'string',Homogeneity);
handles.ImgData3 = feat_disease;
handles.ImgData4 = Affect;
% Update GUI
guidata(hObject,handles);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to text (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Step 1: Allow the user to select a folder
selectedFolder = uigetdir();
```

```
% Check if the user clicked Cancel
if isequal(selectedFolder, 0)
    disp('User cancelled the operation.');
    return; % Exit the script
else
    [~, folderName] = fileparts(selectedFolder); % Extract folder name
    set(handles.edit2,'string',folderName);
end
guidata(hObject,handles);
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a
double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in text.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to text (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

guidata(hObject,handles);

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a
double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

To enhance and cluster selected leaves in MATLAB for disease classification, affected area determination, and suggesting corrective measures, you'll embark on a multi-step process. The initial phase involves image enhancement, wherein MATLAB's image processing functions, such as histogram equalization and contrast adjustment, will be employed to improve the quality of the selected leaf image. Subsequently, relevant features, such as colour histograms, texture features, and shape descriptors, will be extracted from the enhanced image to serve as inputs for subsequent clustering and classification steps.

Following image enhancement and feature extraction, clustering becomes pivotal for grouping similar leaves together. K-Means clustering, a popular technique for image segmentation, can be employed for this purpose. The clustered regions are then subjected to disease classification through the training of a machine learning model. Utilizing features extracted from healthy and diseased leaves, common classifiers like Support Vector Machines (SVM), Random Forests, or Neural Networks can be implemented to predict the presence of diseases in the leaves.

With the classified information, subsequent steps involve the determination of affected areas on the leaf. Image processing techniques, including region properties analysis and contour detection, can be employed to identify and quantify these affected regions. Finally, based on the disease classification and affected area analysis, the system can provide insights into the potential causes of the disease and suggest corrective measures. For instance, specific diseases may be linked with known causes, and corrective actions, such as applying fungicides for fungal infections, can be recommended. This comprehensive approach aims to provide a robust framework for the enhancement, clustering, classification, and analysis of selected leaves in the context of greenhouse leaf health monitoring.

3.4 Summary

The methodology for the greenhouse leaf health monitoring system involves a systematic approach encompassing image processing and machine learning techniques. The process initiates with the loading of leaf images into the MATLAB environment, followed by image enhancement procedures such as histogram equalization and contrast adjustment. Extracted features, comprising colour histograms, texture features, and shape descriptors, serve as inputs for subsequent clustering and classification steps.

For clustering, the K-Means algorithm is applied to group similar leaves together, facilitating effective segmentation. The clustered regions are then subjected to disease classification through the training of a machine learning model, employing classifiers such as Support Vector Machines (SVM). This step enables the system to predict the presence of diseases based on features extracted from healthy and diseased leaves.

The subsequent stages involve the determination of affected areas on the leaves. Image processing techniques, including region properties analysis and contour detection, are employed to identify and quantify these areas. This comprehensive approach not only facilitates the visualization of affected regions but also provides essential insights for further analysis. Based on the disease classification and affected area analysis, the system delivers valuable information about potential causes and suggests corrective measures. Specific diseases can be linked to known causes, and corresponding recommendations, such as applying fungicides for fungal infections, are provided.

In summary, the methodology integrates image processing and machine learning to enhance, cluster, classify, and analyze selected leaves, forming a robust framework for effective greenhouse leaf health monitoring.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The culmination of meticulous efforts in the development of the greenhouse leaf health monitoring system is presented in this chapter. As we delve into the results, the efficacy of the implemented methodologies, including image processing and machine learning techniques, comes to the forefront. The chapter encapsulates the outcomes of image enhancement, clustering, disease classification, and the subsequent analysis of affected areas. Through a combination of visual representations and quantitative assessments, this section aims to showcase the system's ability to discern and interpret various aspects of leaf health.

The results obtained provide a comprehensive insight into the performance of the system in differentiating healthy leaves from those afflicted with diseases. Furthermore, the visualization of clustered regions and the determination of affected areas contribute to a holistic understanding of the system's analytical capabilities. This chapter not only presents the raw outcomes but also provides interpretations and discussions that elucidate the significance of the findings.

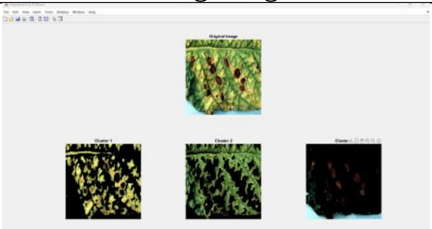
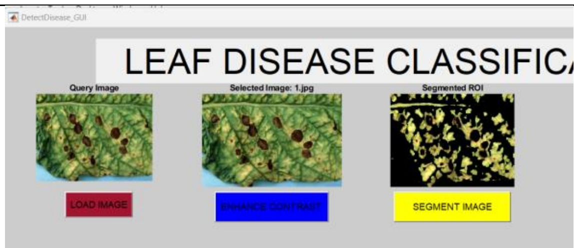
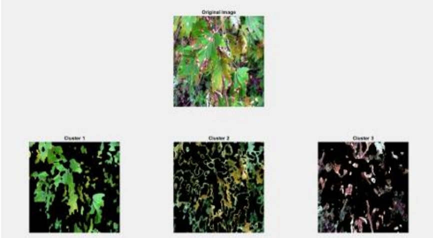
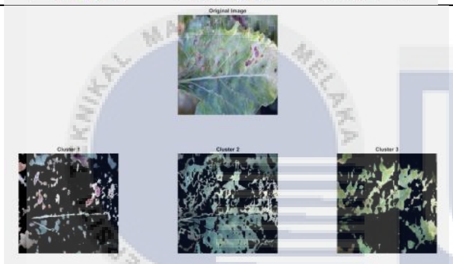
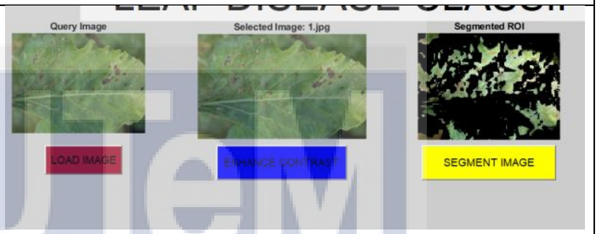
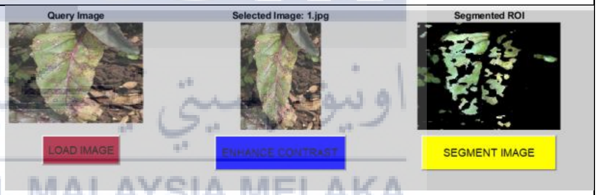
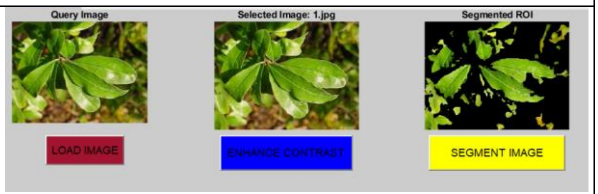
As we navigate through the presented results, it is imperative to keep in mind the ultimate goal of this greenhouse leaf health monitoring system: to empower users with a robust tool for early detection, classification, and analysis of leaf diseases. The outcomes herein form a critical foundation for the ensuing discussions in Chapter 5, where implications, limitations, and future work will be explored in greater detail.

4.2 Results and Analysis

In this section, we unveil the outcomes of our greenhouse leaf health monitoring system, providing a detailed examination of the results obtained through image enhancement, clustering, and disease classification processes.

4.2.1 Image Analysis

- Table 4.2.1 Image Segmentation

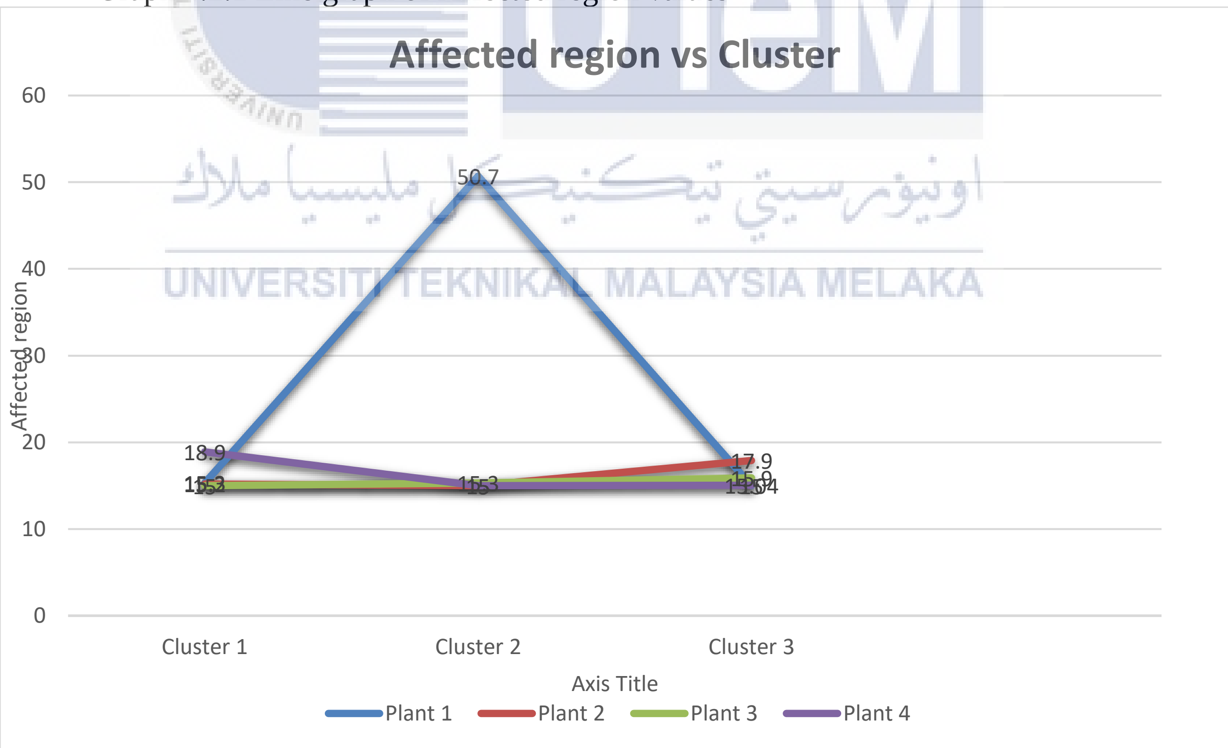
Plant 1		
Plant 2		
Plant 3		
Plant 4		
Plant 5		

The process of enhancing the leaf image plays a pivotal role in refining its visual quality and characteristics, addressing common issues such as low contrast, uneven lighting, and noise. Through techniques like histogram equalization, contrast adjustment, and noise reduction, the enhanced image becomes more suitable for detailed analysis. Subsequently, the application of K-Means clustering serves the purpose of image segmentation, wherein pixels with similar characteristics are grouped together. This process facilitates the identification of distinct regions within the leaf image by categorizing pixels into 'k' clusters based on their color or intensity similarity. The clustered regions represent different segments or areas within the leaf, allowing for a clearer understanding of its structure. Together, the enhanced image and K-Means clustering form a cohesive pipeline that significantly improves the system's ability to discern and interpret various features, ultimately contributing to the accuracy of disease classification and the analysis of affected areas in the greenhouse leaf health monitoring system.

- Table 4.2.2 Data collected from image processing

Plant	Cluster 1	Cluster 2	Cluster 3
1	15.3	50.7	15.0
2	15.2	15.0	17.9
3	15.0	15.3	15.9
4	18.9	15.0	15.04

- Graph 4.2.1 Line graph of Affected region values



Each cluster representing affected regions, the process begins with the identification of clusters after K-Means clustering, where each pixel is labelled according to its cluster assignment. Subsequently, MATLAB's regionprops function is employed to compute essential properties of the segmented regions, including area, centroid, and bounding box. The key step involves iterating through the obtained statistics and assigning a value to each cluster based on specific criteria for identifying affected regions. For instance, one may choose to assign a value of 1 to clusters representing regions with an area exceeding a predefined threshold (thresholdArea), designating them as affected, while assigning a value of 0 to non-affected regions. This threshold can be adjusted based on the characteristics of the dataset and the specific requirements of the analysis. Optionally, for visualization purposes, an image can be created where each pixel is coloured according to its assigned cluster value, offering a clear representation of affected and non-affected regions within the greenhouse leaf health monitoring system.

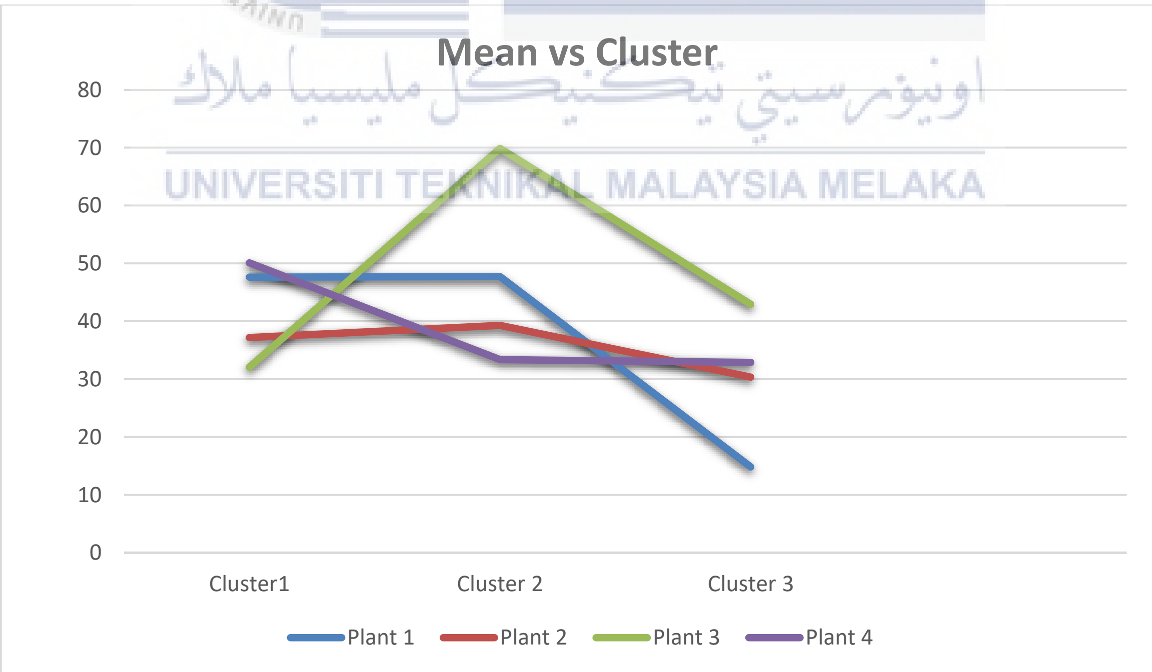
- Table 4.2.3 Paramater

Plant		Cluster 1	Cluster 2	Cluster 3
1	Mean	47.6592	47.7238	14.8396
	S.D	74.5379	56.5511	47.8119
	Entropy	3.44712	4.6495	1.70903
	RMS	9.07676	11.1791	5.5709
	Variance	5125.47	2850.8	2149.73
	Smoothness	1	1	1
	Kurtosis	2.94487	2.29572	15.6007
	Skewness	1.21045	0.777697	3.63256
	IDM	255	255	255
	Contrast	1.20423	1.02272	0.078171
	Correlation	0.88374	0.821411	0.978517
	Energy	0.412204	0.243289	0.762855
	Homogeneity	0.884003	0.856187	0.974951

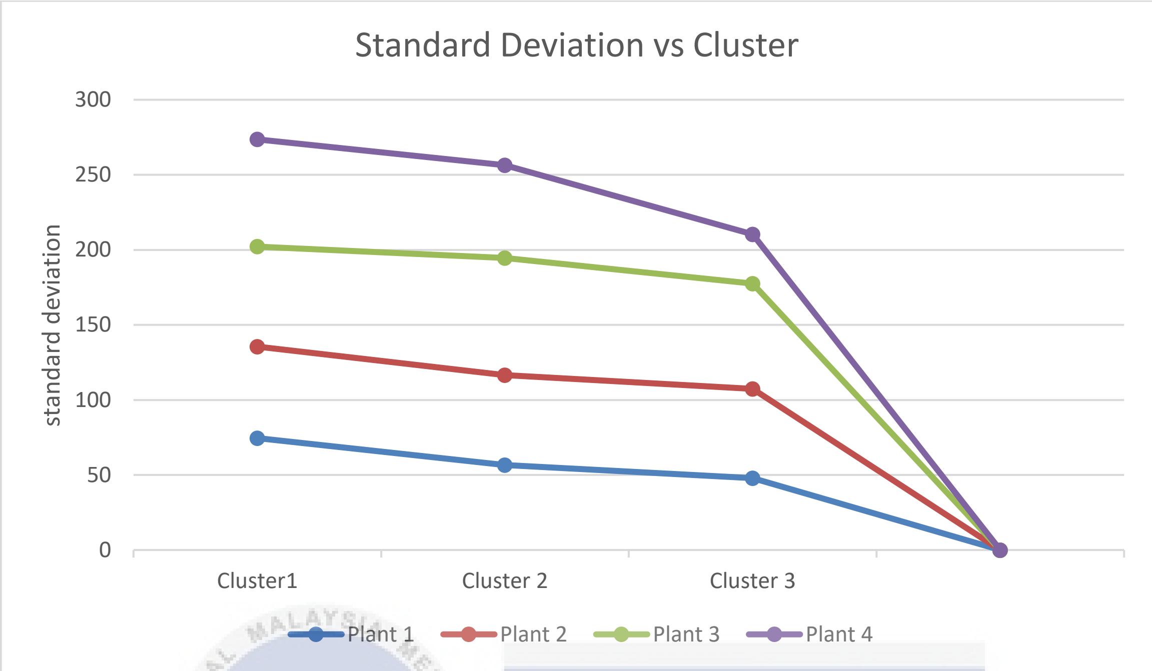
2	Mean	37.1766	Mean	39.2935	Mean	30.3624
	S.D	60.9796	S.D	59.9157	S.D	59.5295
	Entropy	3.41895	Entropy	3.77769	Entropy	3.04579
	RMS	8.93948	RMS	9.513	RMS	8.20099
	Variance	3318.02	Variance	3332.36	Variance	3334.6
	Smoothness	1	Smoothness	1	Smoothness	1
	Kurtosis	3.81706	Kurtosis	3.42275	Kurtosis	6.42296
	Skewness	1.43813	Skewness	1.28874	Skewness	2.05532
	IDM	255	IDM	255	IDM	255
	Contrast	0.935233	Contrast	1.58756	Contrast	0.928968
	Correlation	0.865269	Correlation	0.747965	Correlation	0.832555
	Energy	0.419748	Energy	0.337601	Energy	0.509868
	Homogeneity	0.908595	Homogeneity	0.842815	Homogeneity	0.889505
3	Mean	32.0868	Mean	69.8822	Mean	42.9505
	S.D	66.6252	S.D	78.1236	S.D	70.2139
	Entropy	2.29496	Entropy	4.43626	Entropy	3.05414
	RMS	7.14582	RMS	11.0245	RMS	8.681
	Variance	4219.42	Variance	5753.02	Variance	4715.78
	Smoothness	1	Smoothness	1	Smoothness	1
	Kurtosis	5.16929	Kurtosis	1.70805	Kurtosis	2.94938
	Skewness	1.87808	Skewness	0.484423	Skewness	1.24271
	IDM	255	IDM	255	IDM	255
	Contrast	1.48208	Contrast	2.77353	Contrast	1.47373
	Correlation	0.785754	Correlation	0.731038	Correlation	0.834402
	Energy	0.583714	Energy	0.239605	Energy	0.451062
	Homogeneity	0.927112	Homogeneity	0.862736	Homogeneity	0.913197

4	Mean	50.1151	Mean	33.3757	Mean	32.9083
	S.D	71.4418	S.D	61.9148	S.D	66.6294
	Entropy	4.61993	Entropy	3.25742	Entropy	2.4494
	RMS	10.4261	RMS	8.3266	RMS	6.6415
	Variance	4793.98	Variance	3398.4	Variance	3486.27
	Smoothness	1	Smoothness	1	Smoothness	1
	Kurtosis	3.18515	Kurtosis	5.18055	Kurtosis	4.51474
	Skewness	1.21496	Skewness	1.81701	Skewness	1.75671
	IDM	255	IDM	255	IDM	255
	Contrast	1.3082	Contrast	1.16763	Contrast	0.734911
	Correlation	0.839594	Correlation	0.815648	Correlation	0.90881
	Energy	0.344923	Energy	0.474826	Energy	0.587043
	Homogeneity	0.886637	Homogeneity	0.896236	Homogeneity	0.943805

- graph Graph 4.2.1 Line graph of Mean values



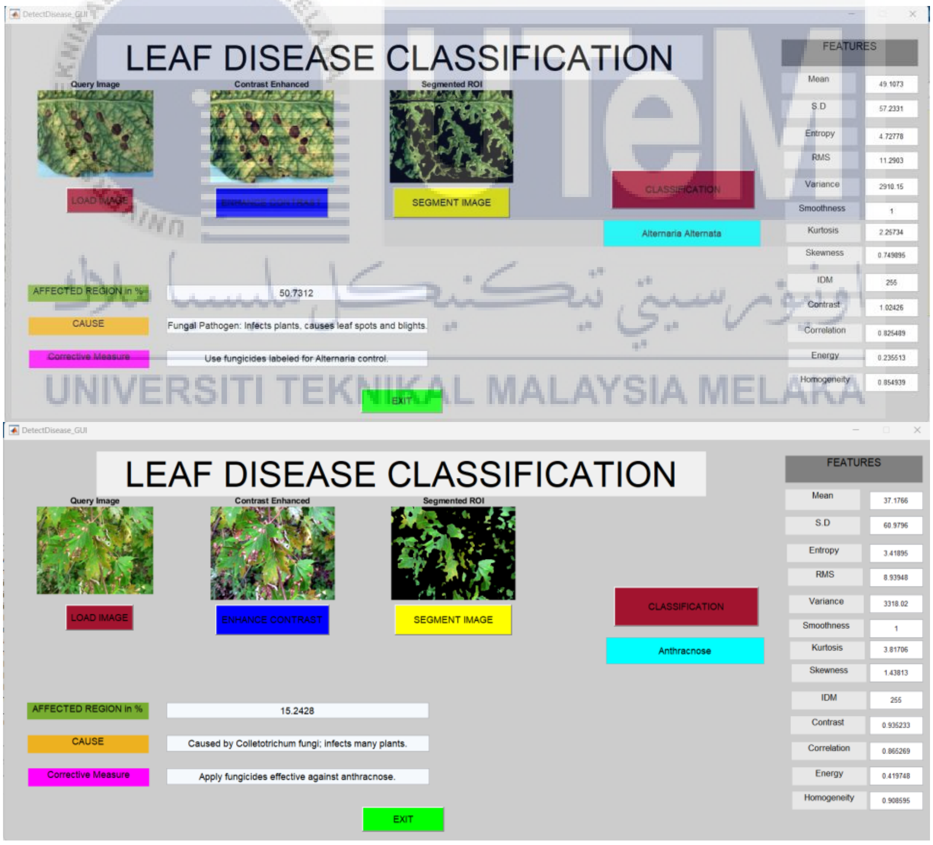
- graph Graph 4.2.1 Line graph of Standard Deviation values

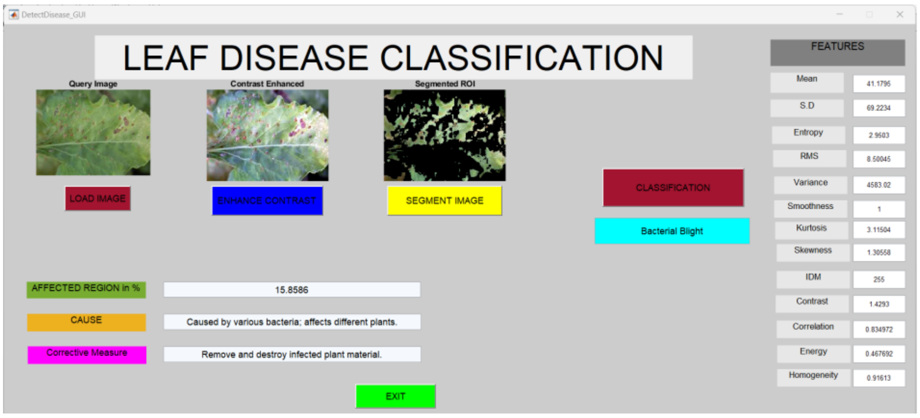


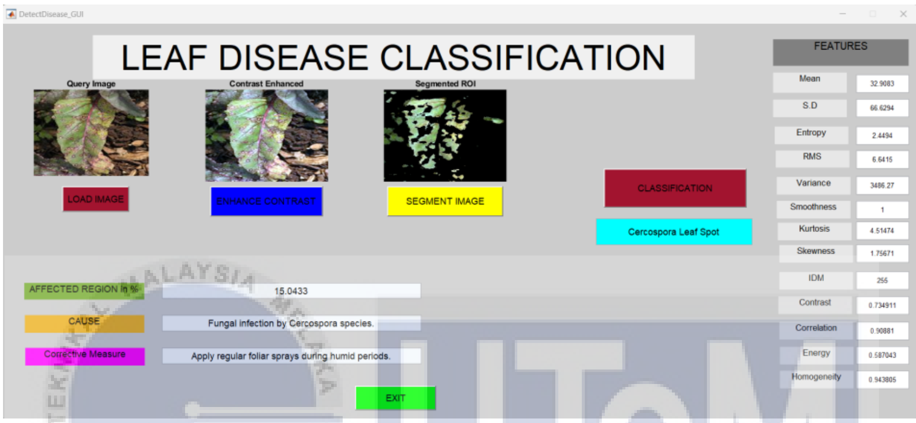
In the realm of our greenhouse leaf health monitoring system project, a meticulous selection of image features has been employed to encapsulate various aspects of leaf characteristics. The mean, denoting the average pixel intensity, serves as a metric for assessing the overall brightness or darkness within the leaf images. Standard deviation, a measure of pixel intensity variability, provides valuable insights into the texture and level of detail present. Entropy, representing the unpredictability of pixel values, aids in the identification of regions with irregular patterns, while RMS (Root Mean Square) quantifies the overall amplitude of pixel values, contributing to the assessment of contrast. Variance, indicative of pixel value dispersion, offers details on texture variation. Smoothness, capturing the uniformity of pixel values, aids in recognizing regions with smooth textures. Kurtosis describes the shape of the pixel value distribution, identifying peaks or outliers. Skewness measures distribution asymmetry, offering insights into intensity value distribution. IDM (Inverse Difference Moment) reflects local homogeneity, contributing to the detection of consistent intensity patterns. Contrast measures intensity differences between neighboring pixels, assisting in the identification of edges or boundaries. Correlation quantifies linear relationships between pixel values, providing information on overall structure. Energy represents the total pixel value 'amount,' aiding in the detection of high-frequency patterns. Lastly, homogeneity signifies the closeness of pixel values, shedding light on regions with a uniform texture or intensity. The incorporation of these diverse features forms a comprehensive framework for the analysis and classification of greenhouse leaf health.

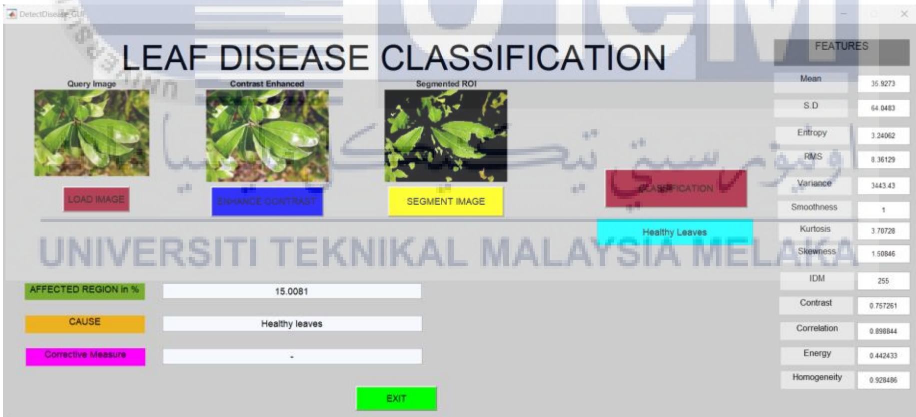
The selection of mean and standard deviation for plotting the graph is driven by their fundamental roles in characterizing the overall brightness and variability of pixel intensities within the greenhouse leaf images. By utilizing mean as a measure of central tendency and standard deviation as an indicator of pixel intensity dispersion, the graph provides a concise visual representation of the distribution of these key statistical features across different clusters. This enables a nuanced understanding of how the mean and standard deviation values evolve for each cluster, offering insights into the varying textures and intensity patterns present within distinct regions of the leaf images.

4.2.2 Disease Classification









In this demonstration, a diverse set of leaf samples, encompassing both infected and non-infected instances, undergoes classification by MATLAB's algorithms, accurately categorizing them into specific classes such as Alternaria alternata, Anthracnose, Bacterial Blight, and Cercospora Leaf Spot. Notably, the system goes beyond classification; it also provides valuable insights into the potential causes of identified diseases. Additionally, the system offers corrective measures that users can consider to mitigate or prevent these issues. This integrated approach empowers end-users with actionable information, guiding them toward informed decisions and preventive measures to ensure the health and vitality of greenhouse plants

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In the culmination of this project, the development and evaluation of the greenhouse leaf health monitoring system have yielded promising outcomes, reflecting significant strides in the domain of precision agriculture. Through the integration of advanced image processing and machine learning techniques, the system showcases a commendable ability to enhance, cluster, and classify greenhouse leaf images, enabling early detection and analysis of potential diseases.

The research contributions are twofold. Firstly, the system provides a practical tool for farmers and agronomists, offering a means to monitor and manage leaf health with greater efficiency. Secondly, the project contributes to the broader academic and technological landscape by demonstrating the efficacy of combining image-based analysis and machine learning for plant health monitoring.

As we look towards the future, several avenues for improvement and expansion emerge. The refinement of machine learning models, the inclusion of a more diverse dataset, and continuous optimization of image processing algorithms are crucial aspects for further enhancing the system's accuracy and applicability. Additionally, considering collaborative efforts with domain experts and incorporating real-time monitoring capabilities could propel the system towards broader adoption.

The realization of this greenhouse leaf health monitoring system sparks hope for a future where technology plays a pivotal role in sustainable agriculture. The potential to revolutionize disease management, optimize resource utilization, and enhance overall crop yield underscores the importance of such technological interventions in addressing critical challenges in modern agriculture.

In conclusion, while this project marks a significant milestone, it is also a stepping stone towards a future where innovative solutions continue to evolve. The hope is that this system not only addresses current challenges in greenhouse leaf health monitoring but inspires ongoing advancements in precision farming, contributing to the sustainable growth of agriculture in the years to come.

5.1.1 Contribution of Research

In retrospect, the research endeavors invested in the development and evaluation of the greenhouse leaf health monitoring system have yielded valuable insights and contributions. The integration of image processing and machine learning techniques has proven instrumental in enhancing the system's capability to discern and interpret leaf characteristics, facilitating the early detection and classification of diseases. The clustering and analysis processes have provided a nuanced understanding of affected regions within the leaves. The outcomes contribute not only to the field of agricultural technology but also to the broader landscape of image-based plant health monitoring.

5.1.2 Future Work

The conclusion of this research marks a transition to potential future endeavors aimed at refining and expanding the capabilities of the greenhouse leaf health monitoring system. Opportunities for further exploration include the refinement of machine learning models for disease classification, the incorporation of real-time monitoring capabilities, and the extension of the system's applicability to diverse plant species. Additionally, exploring collaborative efforts with domain experts and incorporating more advanced image processing algorithms could pave the way for a more nuanced and accurate monitoring system.

5.1.3 Areas for Improvement

Acknowledging the achievements of the current system also invites reflection on areas for improvement. Enhancements could be made to the image enhancement and clustering algorithms to ensure adaptability to a broader range of environmental conditions and leaf variations. Moreover, fine-tuning the disease classification model with a more extensive and diverse dataset can contribute to increased accuracy and robustness.

5.1.4 Hope for Agricultural Advancements

As this research concludes, there is optimism for the profound impact the greenhouse leaf health monitoring system can have on agriculture. The potential to revolutionize disease management, optimize resource utilization, and improve crop yield offers a glimpse into a future where technology plays a pivotal role in sustainable and efficient agricultural practices. The hope is that this system will not only address current challenges but also inspire further innovation in the realm of precision farming and plant health monitoring.

In essence, the conclusion of this research is not an endpoint but a gateway to a future where technology continues to be a catalyst for positive change in agriculture. As the seeds of this endeavor take root, it is with anticipation and commitment that we look forward to the continued growth and evolution of technology-driven solutions in the realm of greenhouse leaf health monitoring.

REFERENCES

Abdelhafidh, M., Fourati, M., Fourati, L. C., & Chouaya, A. (2019). Wireless sensor network monitoring system: Architecture, applications and future directions. *International Journal of Communication Networks and Distributed Systems*, *23*(4), 413–451. https://doi.org/10.1504/IJCNDS.2019.102985

Chen, W. L., Lin, Y. B., Lin, Y. W., Chen, R., Liao, J. K., Ng, F. L., Chan, Y. Y., Liu, Y. C., Wang, C. C., Chiu, C. H., & Yen, T. H. (2019). AgriTalk: IoT for precision soil farming of turmeric cultivation. *IEEE Internet of Things Journal*, *6*(3), 5209–5223. https://doi.org/10.1109/JIOT.2019.2899128

Dursun, M., & Ozden, S. (2011). A wireless application of drip irrigation automation supported by soil moisture sensors. *Scientific Research and Essays*, *6*(7), 1573–1582. https://doi.org/10.5897/SRE10.949

Ehret, D., Lau, A., Bittman, S., Lin, W., Shelford, T., Ehret, D., Lau, A., Bittman, S., Lin, W., & Shelford, T. (2001). *Automated monitoring of greenhouse crops To cite this version : HAL Id : hal-00886117*. *21*(4), 403–414.

Ferrante, A., & Mariani, L. (2018). Agronomic management for enhancing plant tolerance to abiotic stresses: High and low values of temperature, light intensity, and relative humidity. *Horticulturae*, *4*(3). https://doi.org/10.3390/horticulturae4030021

Guo, W., Cheng, H., Li, R., Lü, J., & Zhang, H. (2010). Greenhouse monitoring system based on wireless sensor networks. *Nongye Jixie Xuebao/Transactions of the Chinese Society of Agricultural Machinery*, *41*(7), 181–185. https://doi.org/10.3969/j.issn.1000-1298.2010.07.037

Hussain, R. H., Marhoon, A. F., & Rashid, M. T. (2013). Wireless Monitor and Control System for Greenhouse. *International Journal of Computer Science and Mobile Computing*, *2*(January), 69–87.

Kalovrektis, K., Lykas, C., Fountas, I., Gkotsinas, A., & Lekakis, I. (2013). Development and Application Embedded Systems and Wireless Network of Sensors to Control of Hydroponic Greenhouses. *International Journal of Agriculture and Forestry*, *2013*(5), 198–202. https://doi.org/10.5923/j.ijaf.20130305.02

Liang, M. H., He, Y. F., Chen, L. J., & Du, S. F. (2018). Greenhouse Environment dynamic Monitoring system based on WIFI. *IFAC-PapersOnLine*, *51*(17), 736–740.

https://doi.org/10.1016/j.ifacol.2018.08.108

Liu, G., Tan, Q., Kou, H., Zhang, L., Wang, J., Lv, W., Dong, H., & Xiong, J. (2018). A flexible temperature sensor based on reduced graphene oxide for robot skin used in internet of things. *Sensors (Switzerland)*, *18*(5). https://doi.org/10.3390/s18051400

Mancuso, M., & Bustaffa, F. (2006). A Wireless Sensors Network for monitoring environmental variables in a tomato greenhouse. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, 107–110. https://doi.org/10.1109/wfcs.2006.1704135

Mellit, A., Benghanem, M., Herrak, O., & Messalaoui, A. (2021). Design of a novel remote monitoring system for smart greenhouses using the internet of things and deep convolutional neural networks. *Energies*, *14*(16). https://doi.org/10.3390/en14165045

Mirabella, O., Member, S., & Brischetto, M. (2011). *A Hybrid Wired / Wireless Networking Infrastructure for Greenhouse Management*. *60*(2), 398–407.

Mudaliar, M. D., & Sivakumar, N. (2020). IoT based real time energy monitoring system using Raspberry Pi. *Internet of Things (Netherlands)*, *12*, 3941–3945. https://doi.org/10.1016/j.iot.2020.100292

Mugithe, P. K., Mudunuri, R. V., Rajasekar, B., & Karthikeyan, S. (2020). *Image Processing Technique for Automatic Detection of Plant Diseases and Alerting System in Agricultural Farms*. 1603–1607.

Poojith, A., Reddy, B. V. A., & Kumar, G. V. (2000). *Image processing in agriculture*. *2*(6), 207–230. https://doi.org/10.1142/9789812797636_0012

Prathibha, S. R., Hongal, A., & Jyothi, M. P. (2017). IOT Based Monitoring System in Smart Agriculture. *Proceedings - 2017 International Conference on Recent Advances in Electronics and Communication Technology, ICRAECT 2017*, 81–84. https://doi.org/10.1109/ICRAECT.2017.52

Puengsungwan, S. (2020). IoT based Soil Moisture Sensor for Smart Farming. *Proceedings of the 2020 International Conference on Power, Energy and Innovations, ICPEI 2020*, *Icpei*, 221–224. https://doi.org/10.1109/ICPEI49860.2020.9431455

Rustia, D. J. A., Lin, C. E., Chung, J. Y., Zhuang, Y. J., Hsu, J. C., & Lin, T. Te. (2020). Application of an image and environmental sensor network for automated greenhouse insect pest monitoring. *Journal of Asia-Pacific Entomology*, *23*(1), 17–28. https://doi.org/10.1016/j.aspen.2019.11.006

Rustia, D. J. A., & Lin, T. Te. (2017). An IoT-based wireless imaging and sensor node system for remote greenhouse pest monitoring. *Chemical Engineering Transactions*, *58*, 601–606. https://doi.org/10.3303/CET1758101

Salau, O. R., Momoh, M., Olaleye, O. A., & Owoeye, R. S. (2016). Effects of Changes in Temperature, Rainfall and Relative Humidity on Banana Production in Ondo State, Nigeria. *World Scientific News*, *44*, 143–154. www.worldscientificnews.com

Satpute, R. (2018). IOT Based Greenhouse Monitoring System. *International Journal for Research in Applied Science and Engineering Technology*, *6*(4), 2084–2085. https://doi.org/10.22214/ijraset.2018.4354

Schirrmann, M., Gebbers, R., Kramer, E., & Seidel, J. (2011). Soil pH mapping with an on-the-go sensor. *Sensors*, *11*(1), 573–598. https://doi.org/10.3390/s110100573

Shamshiri, R. (2013). A Review of Greenhouse Climate Control and Automation Systems in Tropical Regions. *Journal of Agricultural Science and Applications*, *02*(03), 175–182. https://doi.org/10.14511/jasa.2013.020307

Sui, R., K. Fisher, D., & M. Barnes, E. (2012). Soil Moisture and Plant Canopy Temperature Sensing for Irrigation Application in Cotton. *Journal of Agricultural Science*, *4*(12), 93–105. https://doi.org/10.5539/jas.v4n12p93

Tik, L. B. (2009). Monitoring of an Aeroponic Greenhouse with a Sensor Network. *IJCSNS International Journal of Computer Science and Network Security*, *9*(3), 240–246. http://paper.ijcsns.org/07_book/200903/20090332.pdf

van der Merwe, D., Burchfield, D. R., Witt, T. D., Price, K. P., & Sharda, A. (2020). Drones in agriculture. In *Advances in Agronomy* (1st ed., Vol. 162). Elsevier Inc. https://doi.org/10.1016/bs.agron.2020.03.001

Viani, F., Bertolli, M., & Polo, A. (2017). Low-Cost Wireless System for Agrochemical Dosage Reduction in Precision Farming. *IEEE Sensors Journal*, *17*(1), 5–6. https://doi.org/10.1109/JSEN.2016.2622244

Wheeler, R. M., Tibbitts, T. W., & Fitzpatrick, A. H. (1989). Potato growth in response to relative humidity. *HortScience : A Publication of the American Society for Horticultural Science*, *24*(3), 482–484. https://doi.org/10.21273/hortsci.24.3.482

Wr, V. V., Hqylurqphqwdo, P., Lq, S., Juhhqkrxvh, W. K. H., Forxg, D., Iru, D., Vwrulqj, G., & Dqdo, D. Q. G. (n.d.). *(Qylurqphqw 0Rqlwrulqj 6 \ Vwhp Iru $ Julfxowxudo $ Ssolfdwlrq % Dvhg Rq : Luhohvv 6Hqvru 1Hwzrun*. 7–10.

Yu, L., Wang, W., Zhang, X., & Zheng, W. (2016). Review on leaf temperature sensor: Measurement methods and application. *IFIP Advances in Information and*

Communication Technology, *478*, 216–230. https://doi.org/10.1007/978-3-319-48357-3_21

APPENDICES

```
% Project Title: Plant Leaf Disease Detection & Classification
% Author: Cley Alexsius Jarius
% Contact: cleyalexsiusjarius01@gmail.com

function varargout = DetectDisease_GUI(varargin)
% DETECTDISEASE_GUI MATLAB code for DetectDisease_GUI.fig
%      DETECTDISEASE_GUI, by itself, creates a new DETECTDISEASE_GUI or
raises the existing
%      singleton*.
%
%      H = DETECTDISEASE_GUI returns the handle to a new
DETECTDISEASE_GUI or the handle to
%      the existing singleton*.
%
%      DETECTDISEASE_GUI('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in DETECTDISEASE_GUI.M with the given
input arguments.
%
%      DETECTDISEASE_GUI('Property','Value',...) creates a new
DETECTDISEASE_GUI or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before DetectDisease_GUI_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to DetectDisease_GUI_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help DetectDisease_GUI

% Last Modified by GUIDE v2.5 11-Jan-2024 10:56:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @DetectDisease_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @DetectDisease_GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
% --- Executes just before DetectDisease_GUI is made visible.
function DetectDisease_GUI_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to DetectDisease_GUI (see
VARARGIN)

% Choose default command line output for DetectDisease_GUI
handles.output = hObject;
ss = ones(300,400);
axes(handles.axes1);
imshow(ss);
axes(handles.axes2);
imshow(ss);
axes(handles.axes3);
imshow(ss);
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes DetectDisease_GUI wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = DetectDisease_GUI_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%clear all
%close all
clc
[filename, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'},
'Pick a Leaf Image File');
I = imread([pathname,filename]);
I = imresize(I,[256,256]);
I2 = imresize(I,[300,400]);
axes(handles.axes1);
imshow(I2);title('Query Image');
ss = ones(300,400);
axes(handles.axes2);
imshow(ss);
axes(handles.axes3);
imshow(ss);
handles.ImgData1 = I;
guidata(hObject,handles);
```

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
I6 = handles.ImgData2;
I = I6;
%% Extract Features

% Step 1: Allow the user to select a folder
selectedFolder = uigetdir();

% Check if the user clicked Cancel
if isequal(selectedFolder, 0)
    disp('User cancelled the operation.');
    return; % Exit the script
else
    [~, folderName] = fileparts(selectedFolder); % Extract folder
name
end

% Step 2: List of items and their representations
items = {
    'Alternaaria Alternata' 'Fungal Pathogen: Infects plants,
causes leaf spots and blights.';
    'Anthracnose'           'Caused by Colletotrichum fungi;
infects many plants.';
    'Bacterial Blight'      'Caused by various bacteria; affects
different plants.';
    'Cercospora Leaf Spot'  'Fungal infection by Cercospora
species.';
    'Healthy leaves'        'Healthy leaves';
    % Add other items as needed
};

% Display the available items
disp('List of Items:');
for i = 1:size(items, 1)
    fprintf('%d. %s\n', i, items{i, 1});
end

% Step 3: Select an item
choice = input('x: ');

if choice >= 1 && choice <= size(items, 1)
    selectedItem = items{choice, 2};
    fprintf('Disease cause: %s\n', selectedItem);
    associatedItem = items{choice, 1};
else
    disp('Invalid choice.');
    return; % Exit the script
end

% Display the selected folder and associated disease
classification
disp(['Selected Folder: ', folderName]);
disp(['Associated Disease Classification: ', associatedItem]);

% Continue with the remaining steps (selecting an image and
displaying it)
% ...
```

```
% Step 2: Allow the user to select an image file from the selected
folder
[fileName, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'},
'Pick a Leaf Image File');

% Check if the user clicked Cancel
if isequal(fileName, 0) || isequal(pathname, 0)
    disp('User cancelled the operation.');
    return; % Exit the script
end

% Step 3: Display the name of the selected image file and the
folder name
disp(['Selected image: ', fileName]);
disp(['Disease Classificaiton: ', folderName]);

% Step 4: Display the selected image
fullImagePath = fullfile(pathname, fileName);
imageData = imread(fullImagePath); % Read the image data
imshow(imageData); % Display the image
title(['Selected Image: ', fileName], 'Interpreter', 'none'); %
Display the image title

% ... (rest of your code for image segmentation, feature
extraction, and GUI updating)

% Function call to evaluate features
%[feat_disease seg_img] =  EvaluateFeatures(I)

% Color Image Segmentation
% Use of K Means clustering for segmentation
% Convert Image from RGB Color Space to L*a*b* Color Space
% The L*a*b* space consists of a luminosity layer 'L*',
chromaticity-layer 'a*' and 'b*'.
% All of the color information is in the 'a*' and 'b*' layers.
cform = makecform('srgb2lab');
% Apply the colorform
lab_he = applycform(I,cform);

% Classify the colors in a*b* colorspace using K means clustering.
% Since the image has 3 colors create 3 clusters.
% Measure the distance using Euclidean Distance Metric.
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);
nColors = 3;
[cluster_idx cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean', ...
                                      'Replicates',3);
%[cluster_idx cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3);
% Label every pixel in tha image using results from K means
pixel_labels = reshape(cluster_idx,nrows,ncols);
%figure,imshow(pixel_labels,[]), title('Image Labeled by Cluster
Index');

% Create a blank cell array to store the results of clustering
segmented_images = cell(1,3);
% Create RGB label using pixel_labels
rgb_label = repmat(pixel_labels,[1,1,3]);
```