Faculty of Electrical and Electronic Engineering Technology

DEVELOPMENT OF REMOTE MONITORING SYSTEM USING IOT SYSTEM FOR GREENHOUSE (G-REM)

MUHAMMAD SYAHMI BIN SALZAZARY

Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

2024

DEVELOPMENT OF REMOTE MONITORING SYSTEM USING IOT SYSTEM FOR GREENHOUSE (G-REM)

MUHAMMAD SYAHMI BIN SALZAZARY

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 and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF REMOTE MONITORING SYSTEM USING IOT SYSTEM FOR GREENHOUSE (G-REM)

Sesi Pengajian : 23/24

Saya MUHAMMAD SYAHMI BIN SALZAZARY 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)

Alamat Tetap: NO. 2 JLN TTJS 3/2,
TAMAN TUANKU JAAFAR,
71450 SEREMBAN,
NEGERI SEMBILAN

(COP DAN TANDATANGAN PENYELIA)

Tarikh: 11/1/2024

Tarikh: 14 JANUARI 2024

*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 Remote Monitoring System Using IoT System for Greenhouse (G-REM)” 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 : MUHAMMAD SYAHMI BIN SALZAZARY

Date : 11/1/2024

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 JANUARI 2024

Signature : ....

Co-Supervisor Name (if any) : ....

Date : ....

DEDICATION

I am dedicating this bachelor's degree project to my Creator, Allah s.w.t the Almighty, my steadfast support, the wellspring of my inspiration, wisdom, knowledge, and understanding. He is the one who has been my source of resilience, empowering me to complete this project during my academic journey. Additionally, I extend this dedication to my parents, Salzazary Bin Salzazary and Shimaliah Binti Baba, and my family who have provided unwavering support, encouraging me to persist in my endeavours. I am grateful to all my friends and lecturers for their consistent encouragement, guidance, and advice, contributing significantly.

ABSTRACT

In this era of globalization, technology is one of the best initiatives to improve the quality of an agriculture-based especially greenhouse-based product. As globalization continues to connect markets and worldwide consumers, utilizing technology will be crucial for maximizing sustainable and efficient greenhouse systems. Moreover, the main challenge that the Malaysian industry faces is to have proficient workers with specific skills in agriculture products. Based on their expertise, skills, and dedication, will play a vital role in ensuring the success of growth in the agriculture industry and ensuring it is always of high quality. It also maximizes productivity in any greenhouse area. The objective of this project is to develop an application of a greenhouse system for the agriculture industry based on mobile development, to monitor the concentration of reproduction parameters integrated with machine learning development to predict the condition of greenhouse planting. Also, to analyze the performance of a greenhouse system by a specific planting structure in the agriculture industry. This hydroponic development system used Android Studio software to develop mobile applications that involve specific elements to support the implementation of the entire greenhouse process. Additionally, by integrating machine learning, this system has a wide range of built-in applications that have the capability of collecting, exchanging, and analyzing data from specific crops, especially for low-crop greenhouse plants. This includes a user interface, image processing, and data analytics which can improve the efficiency of various aspects of the agriculture industry. Overall, this project introduces a user-friendly system that can be operated in real-time situations for all agriculturalists and can lead to advancement for specific greenhouse plant growth.

ABSTRAK

Pada era globalisasi ini, teknologi merupakan salah satu inisiatif terbaik untuk meningkatkan kualiti berasaskan pertanian terutamanya produk berasaskan rumah hijau. Memandangkan globalisasi terus menghubungkan pasaran dan pengguna di seluruh dunia, penggunaan teknologi akan menjadi penting untuk memaksimumkan sistem rumah hijau yang mampan dan cekap. Selain itu, cabaran utama yang dihadapi oleh industri Malaysia adalah untuk mempunyai pekerja mahir dengan kemahiran khusus dalam produk pertanian. Berdasarkan kepakaran, kemahiran, dan dedikasi mereka, akan memainkan peranan penting dalam memastikan kejayaan pertumbuhan dalam industri pertanian dan memastikan ia sentiasa berkualiti tinggi. Ia juga memaksimumkan produktiviti di mana-mana kawasan rumah hijau. Objektif projek ini adalah untuk membangunkan aplikasi sistem rumah hijau untuk industri pertanian berdasarkan pembangunan mudah alih, untuk memantau kepekatan parameter pembiakan yang disepadukan dengan pembangunan pembelajaran mesin untuk meramalkan keadaan penanaman rumah hijau. Juga, untuk menganalisis prestasi sistem rumah hijau oleh struktur penanaman tertentu dalam industri pertanian. Sistem pembangunan hidroponik ini menggunakan perisian Android Studio untuk membangunkan aplikasi mudah alih yang melibatkan elemen khusus untuk menyokong pelaksanaan keseluruhan proses rumah hijau. Selain itu, dengan menyepadukan pembelajaran mesin, sistem ini mempunyai rangkaian luas aplikasi terbina dalam yang mempunyai keupayaan untuk mengumpul, bertukar dan menganalisis data daripada tanaman tertentu, terutamanya untuk tumbuhan rumah hijau tanaman rendah. Ini termasuk antara muka pengguna, pemprosesan imej dan analisis data yang boleh meningkatkan kecekapan pelbagai aspek industri pertanian. Secara keseluruhannya, projek ini memperkenalkan sistem mesra pengguna yang boleh dikendalikan dalam situasi masa nyata untuk semua petani dan boleh membawa kepada kemajuan untuk pertumbuhan tumbuhan rumah hijau tertentu.

ACKNOWLEDGEMENTS

Bismillahirrahmanirrahim In the name of Allah S.W.T the Most Gracious, the Most Merciful, and to our prophet Muhammad S.A.W. Alhamdulillah. I am immensely thankful to announce the completion of my final year project after overcoming various challenges. I want to extend my deepest gratitude to Mohamad Haniff Bin Harun, my supervisor, and Siti Nur Suhaila Binti Mirin, my co-supervisor, for their invaluable guidance, wisdom, and unwavering patience throughout this journey. Their constant support, regular project updates, and willingness to dedicate time from his busy schedule were instrumental in helping me successfully finish my project on time. I truly appreciate their commitment and without their assistance, this achievement would not have been possible.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and my parents for the financial support through Bachelor Degree Projects 1 and 2 (BDP 1 and 2) which enabled me to accomplish the project. Not forgetting my colleagues, Hanis Saadah Binti Mohd Azmi, Siti Nor Saharah Binti Lahapi, Nurnajihah Hazirah Binti Azhari, Nurul Ezzah Hazirah Binti Elias, Aina Alisa Binti Ghazali, Mohamad Niza Shaffrel Bin Mohd Junaidi, Nor Hazimah Syahmimi Binti Sahfani, Izzul Hakim Bin Samsaiful, and Nazwa Najmuddin Bin Mohd Nizwaruddin for their willingness to share their thoughts and ideas regarding the project.

My highest appreciation goes to my parents Salzazary Bin Salleh, Shimaliah Binti Baba, and family members for their love and prayer during the period of my study. An honorable mention also goes to Hazim Hamizy Bin Badri Shah for all the motivation and understanding.

Finally, I would like to thank all the staff at the Sahas Nursery especially Muhammad Rusdan Bin Sahabudin, Aziera Syaliana Binti Mat Mokhtar, and Aisyah Syahirah Binti Mat Mokhtar who provided a service for consulting and guiding throughout this Bachelor Degree Project. Also, to all my colleagues and classmates, the faculty members, as well as other individuals who are not listed here for being cooperative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

$G - REM$	-	Greenhouse Remote Monitoring
AI	-	Artificial Intelligent
IoT	-	Internet of Things
pH	-	Potential Hydrogen
RFID	-	Radio Frequency Identification
5G	-	5 Generation mobile network
Zigbee	-	Zonal Intercommunication Global - standard
Soil EC	-	Soil Electrical Conductivity
LCC	-	Leaf Color Chart
CIELAB	-	Commission Internasionale de l'Eclairage
CO2	-	Carbon Dioxide
CNN	-	Convolutional Neural Network
RNN	-	Recurrent Neural Network
MSRCR	-	MultiScale Retinex with Color Restoration
R-CNN	-	Regions with Convolutional Neural Networks
U-Net	-	U-Shape encoder-decoder Network
IoU	-	Intersection over Union
TSSM	-	Two Spotted Spider Mites
YOLOv4	-	You Only Look One version 4
VGG	-	Visual Geometry Group
HMI	-	Human Machine Interface
SQL	-	Structured Query Language
PV	-	PhotoVoltaics
OpenCV	-	Open source Computer Vision library

LIST OF APPENDICES

APPENDICES	TITLE	PAGE

CHAPTER 1

INTRODUCTION

1.1 Background

Greenhouses in Malaysia play a crucial role in supporting agricultural production and addressing the unique challenges presented by the country's tropical climate. They are widely utilized for cultivating various crops, including vegetables, herbs, flowers, and ornamental plants. Greenhouses provide a controlled environment that effectively mitigates the adverse effects of high temperatures, humidity, and heavy rainfall, which are common in Malaysia. By offering shade, proper ventilation, and regulated irrigation, greenhouses enable growers to optimize growing conditions and safeguard plants from extreme weather conditions and pests.

The adoption of greenhouse cultivation in Malaysia facilitates the extension of growing seasons, enhances crop quality, boosts productivity, and reduces the need for imported agricultural products. Additionally, it opens avenues for implementing innovative agricultural techniques and cultivating specialty crops that are typically unsuited to the local climate.

1.2 Problem Statement

Agriculture in Malaysia is characterized by high productivity in a large area with a limited source of proficient workers in the specified scope of the agriculture industry to diversify horticulture. This creates a challenge when it comes to finding proficient workers within the specified scope of the agriculture industry. Moreover, the lack of the initiator of greenhouse technology led to the improvement of low land parameters through a control and treatment system. Low land parameters likely refer to areas with suboptimal soil quality, drainage, or other conditions that make it challenging to cultivate crops effectively.

1.3 Project Objective

The main aim of this project is to develop a system-based greenhouse that can empower every farmer with advanced tools and technology. Specifically, the objectives are as follows:

a) To develop an application of a greenhouse system for the agriculture industry based on mobile development.

b) To develop a machine learning to predict and analyze the performance of a greenhouse planting in the agriculture industry.

1.4 Scope of Project

The scope of this project is as follows:

a) The system can display and predict the condition of plant reproduction via image capture.

b) The software of MIT Apps Inventor will be used to program and design the interface of the layout for the application.

c) This project is dedicated to the field of herb plants in the greenhouse industry only in the context of low-land agriculture.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Over the years, before the advancement of greenhouse technology, agricultural practices heavily relied on traditional open-field cultivation methods. In the absence of a controlled greenhouse system, plants became increasingly vulnerable to the impacts of external weather conditions, pests, and diseases. The controlled environment of the greenhouses system is the cultivation of plants that were previously unsuitable for the local climate, expanding the range of agricultural production.

2.1.1 Optimized Greenhouse Environment and Resources Management

An optimized environment is characterized by the presence of intelligent systems that are capable of autonomously analyzing various parameters such as water availability, temperature, humidity, and soil pH. The exchange of human labor with computerized systems is legal, as it results in significant cost savings for commercial farms and improves overall crop yields. Nevertheless, contemporary IoT systems have distinct applications that necessitate the precise identification of sensors and configurations, timely acquisition and optimization of data, and rule-based control. The primary hurdle lies in transitioning from conventional mechanical farming methods to intelligent farming, as it demands substantial resources to accomplish AI optimization.

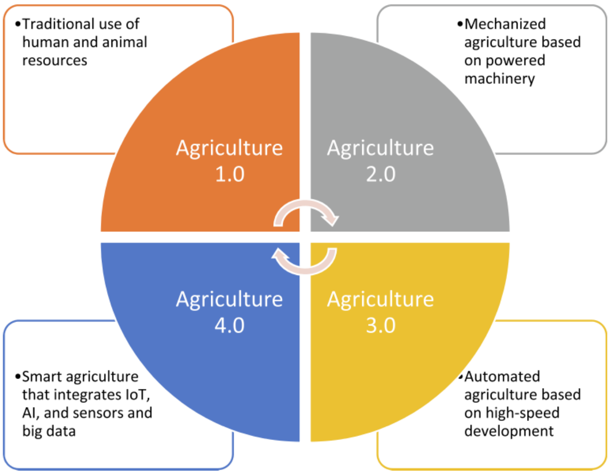


Figure 2.1: Agriculture and greenhouse complexity

Results have been observed in pilot studies that investigate the utilization of IoT in agriculture. It was confirmed that there exist multiple processes of IoT systems in modern greenhouses as well as open-field agriculture. The current discussion is primarily focused on four key areas of IoT application: green hardware in a border of sensors, ultra-low power microcontrollers green software (event prediction, data classification, data delivery, and big data analytics), green communication infrastructure technologies (future Internet 5G, Bluetooth, RFID, Ad hoc, and ZigBee), and green architecture in the cloud [1].

2.2 Studies Related to IoT

To create an IoT system in a greenhouse, it involves the use of sensors, networks, and automation to monitor and control environmental conditions. Sensors collect data on temperature, humidity, light, and soil moisture, which is transmitted and analyzed in real-time. Actuators and controllers adjust parameters like temperature and irrigation based on this data. The system allows for remote access and control, enabling operators to monitor and manage the greenhouse from anywhere. Implementing an IoT system improves crop quality, productivity, and resource efficiency in greenhouse farming [2].

2.2.1 Soil Property Analysis Using IoT

Inappropriate soil fertility ranges have a significant impact on crop growth and the agricultural ecosystem. To address this, soil fertility management practices aim to optimize the use of fertilizers, organic compost, and crop selection based on soil conditions. However, many farmers apply fertilizers without analyzing their soil nutrient status, leading to unbalanced agrochemical inputs that can contaminate agricultural resources. While laboratory soil analysis procedures can track soil fertility, they are inadequate for Indian farmers.

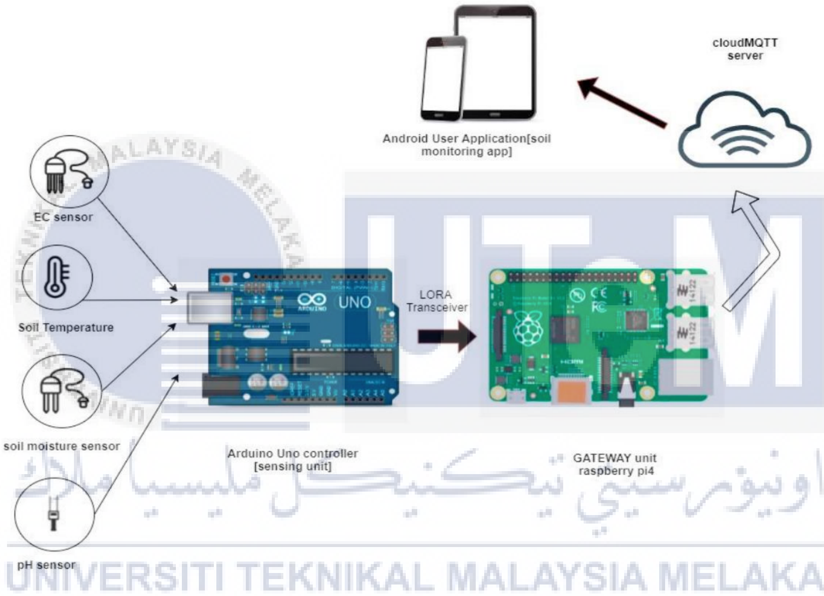


Figure 2.2: Flow diagram of soil property prediction system

Recent research has focused on developing on-field soil property monitoring systems, supported by internet-based soil fertility monitoring. As show in the figure 2.1, this system introduces a prototype of a smart soil fertility prediction system based on the Internet of Things (IoT), which was tested in a home gardening application to monitor spinach crop growth. This system promotes cost-effective direct field measurement in agriculture. Accuracy testing of the prototype involved comparing sensor results with laboratory analysis. The comparison revealed only slight variations in soil pH (0.7% difference) and soil EC (0.16% difference) between the sensor and laboratory values, demonstrating the prototype's reliability [3].

2.3 Studies Related to Mobile Development

The context of mobile development involves creating applications and software for mobile devices that enable greenhouse operators to remotely monitor and control various aspects of greenhouse operations. These mobile apps provide real-time access to data on temperature, humidity, lighting, irrigation, and ventilation, allowing operators to make informed decisions and take timely actions [4]. Mobile development also facilitates data collection, analysis, and integration with other smart technologies, enabling optimized resource allocation and improved efficiency in greenhouse management. Overall, mobile development empowers greenhouse operators by enhancing accessibility, control, and data-driven decision-making capabilities [5].

2.3.1 Leaf Color Detection

Detecting the nitrogen content in rice leaves is crucial for farmers to determine fertilizer application. Existing detection methods are highly dependent on environmental conditions and require specialized equipment. To address these limitations, a smartphone app was developed using a standard leaf color chart (LCC) to detect the color levels of rice leaves. The app successfully identified the rice leaf and LCC regions in images through color threshold segmentation. CIELAB histograms effectively extracted the color features of each region, and the color difference values calculated using the CIEDE2000 formula allowed differentiation of rice leaf color levels.

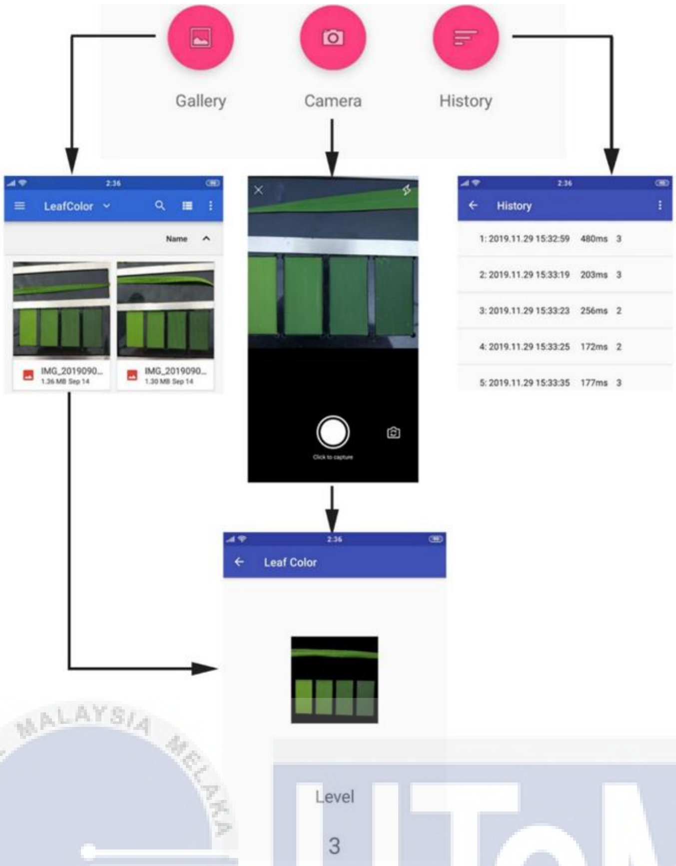


Figure 2.3: Interface of system application

Field testing showed 96% accuracy, with the app outperforming manual inspections by achieving over 92% accuracy in determining rice leaf color levels. The app processed leaf images in approximately 248 ms on a Xiaomi Mi5 smartphone and performed well on other smartphones. This smartphone app provides accurate, efficient, and cost-effective detection of rice leaf color levels, assisting farmers in making informed decisions about nitrogen fertilizer management for rice production [6].

2.3.2 Greenhouse Monitoring and Control System

ZigBee is a wireless communication technology that is commonly used in various applications, including monitoring and control of greenhouse systems. ZigBee-based systems enable the collection and transmission of data from sensors and actuators within the greenhouse environment. This data can include information about temperature, humidity, light levels, soil moisture, CO2 levels, and more.

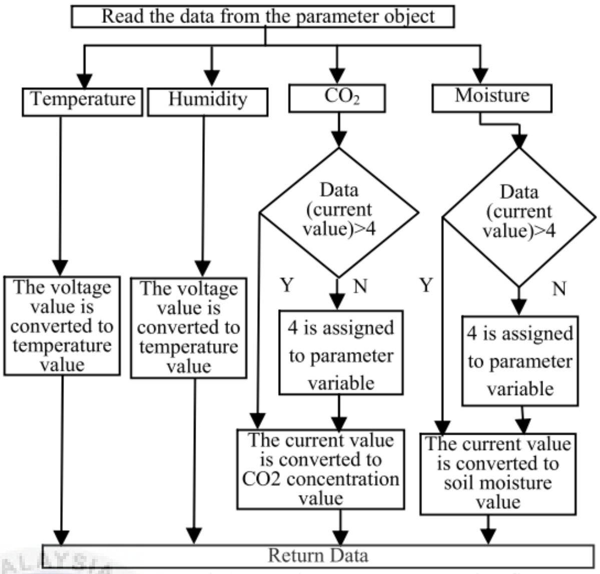


Figure 2.4: Parameter process flow

This system enables the monitoring of temperature, humidity, soil water content, and carbon dioxide concentration in the greenhouse. The collected data is saved to a database and the greenhouse monitoring and control system provides real-time monitoring of both the interior environment of the greenhouse and the external depending on weather conditions. The greenhouse remote monitoring and control software can enable serial communication control. When the system receives the data which are obtained from the greenhouse controller, the greenhouse system can receive and process the collected data, then save and display the data. Figure 2.5 below shows an experimental result by [7] about collected data of specific parameter.

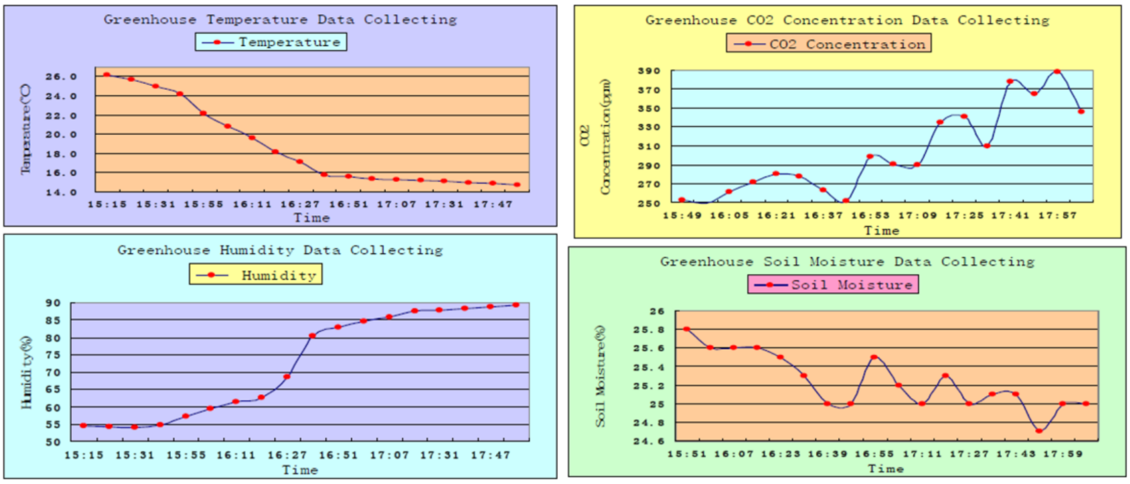


Figure 2.5: Collected data of parameter

2.4 Studies Related to Deep Learning

Deep learning is an advanced artificial intelligence method that utilizes neural networks to analyze extensive datasets. In the context of greenhouses, deep learning techniques, such as CNNs and RNNs, are employed to perform tasks like image recognition, object detection, and sensor data analysis [8]. These algorithms enable automated monitoring, early identification of problems, and optimization of environmental conditions to enhance plant growth [9]. Deep learning can also be applied to predict yields, diagnose diseases, and automate harvesting processes. By harnessing the power of deep learning, greenhouse operators can make informed decisions and enhance overall crop production.

2.4.1 Classification Soybean Seeds

The development of deep learning-based method for the online classification of soybean seeds to efficiently assess their quality. Initially, the used of multiscale Retinex with color restoration (MSRCR) algorithm to segment images of soybean seeds, even under uneven illumination. Subsequently, with constructing a convolutional neural network (CNN) with appropriate parameters to classify the soybean seeds into four categories.

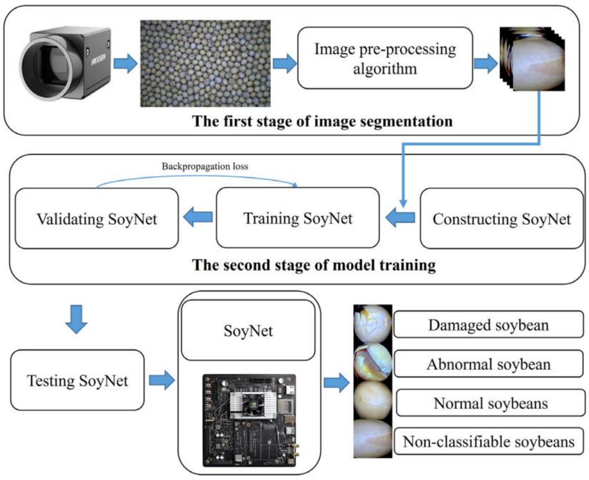


Figure 2.6: Process operation of soybean seed classification

The F-score for normal, damaged, abnormal, and non-classifiable soybeans achieved high accuracies of approximately 95.97%, 97.41%, 97.25%, and 96.14%, respectively. A successfully implemented this method on the NVIDIA Jetson TX2 platform, achieving an accuracy of 95.63% and an average classification time of 4.92 ms per soybean seed. These results meet the requirements for online soybean quality assessment [10].

2.5 Comparison Previous Studies

Obtaining access to comparisons with previous studies in a report serves several purposes. It provides background information on existing research, validates your findings, identifies research gaps, and strengthens your arguments. Comparing your results with previous studies helps establish context and enhances the credibility of your report. It also guides future research directions and contributes to the advancement of knowledge in greenhouse-related topics.

2.5.1 Machine Learning and Deep Learning

Table 1: Comparison of deep learning from previous studies

Title	Author's	Method
Using digital image processing for counting whiteflies on soybean leaves	Jayme Garcia Arnal Barbedo	This paper presents a novel system for accurately counting whiteflies on soybean leaves using digital image processing. The system offers an automated approach that improves efficiency compared to manual methods. The developed algorithm can detect and quantify both adult whiteflies and nymphs, and the paper evaluates its performance and suggests areas for future enhancement. Furthermore, the system can be adapted for other crops with minimal modifications, as it relies on commonly used image-processing operations, making it versatile and easily implementable in different image-processing software packages [11].
Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse	Wenyong Li, Dujin Wang, Ming Li, Yulin Gao, Jianwei Wu. Xinting Yang	This study aimed to develop a detection model for whitefly and thrips using images of sticky traps in greenhouse conditions. The researchers created an end-to-end model called TPest-RCNN, based on the Faster R-

		CNN architecture, using a transfer learning strategy. The TPest-RCNN model outperformed other approaches and achieved high accuracy in detecting multiple species of pests. The results showed that the model was robust and could accurately estimate the abundance of whitefly and thrips. The proposed method provides a valuable tool for rapid pest monitoring and population estimation in greenhouse agriculture [12].
Deep learning-based precision agriculture through weed recognition in sugar beet fields	Amin Nasiri, Mahmoud Omid, Amin Taheri-Garavand, Abdolabbas Jafari	This study focuses on weed control in agriculture using precision agriculture techniques. The U-Net architecture, a deep convolutional neural network, was employed to perform pixel-wise semantic segmentation of sugar beet, weed, and soil. The model was trained with a custom loss function to handle imbalanced data and small area segmentation issues, achieving high accuracy and intersection over union (IoU) scores. The results indicate that integrating CNN-based automatic weed detection into autonomous weed control robots can enhance the precision of herbicide application. [13]

Detecting two-spotted spider mites and predatory mites in strawberry using deep learning	Congliang Zhou, Won Suk Lee, Oscar E. Liburd, Ikbal Aygun, Xue Zhou, Alireza Pourreza, John K. Schueller, Yiannis Ampatzidis	This study presents the strawberries crop in the US are susceptible to damage from two-spotted spider mites (TSSM). This study explored the use of deep learning models to detect TSSM and predatory mites in strawberry fields using smartphone images. The YOLOv4 and Faster R-CNN models achieved high detection accuracy, with YOLOv4 performing best on iPhone XR images. Increasing the image size improved the detection performance of YOLOv4, and the model trained with 640 × 640 pixel images achieved a detection accuracy of 0.933. This smartphone-based method offers a quick and accurate way for growers to monitor mite populations in the field. [14]
Knowledge graph and deep learning based pest detection and identification system for fruit quality	DingJu Zhu, LianZi Xie, BingXu Chen, JianBin Tan, RenFeng Deng, Yongzhi Zheng, Qi Hu, Rashed Mustafa, Wanshan Chen, Shuai Yi,	This paper aimed on pests and diseases pose challenges in fruit cultivation, affecting both the quality and quantity of fruit. Manual identification and detection of pests and diseases require significant human and material resources. This article proposes an automated system based on Raspberry Pi to detect and identify pests and diseases in fruits like Longan and Lychee. The system utilizes

	KaiLeung Yung, Andrew W.H.IP	a knowledge graph, image processing, and a trained VGG-16 model to achieve 94.9% accuracy in pest identification. The results are broadcasted in real-time through a Bluetooth speaker. [15]

2.5.2 Mobile Development

Table 2: Comparison of mobile development from previous studies

Title	Author's	Method
Smart platform based on IoT and WSN for monitoring and control of a greenhouse in the context of precision agriculture	Hamza Benyezza, Mounir Bouhedda, Reda Kara, Samia Rebouh,	This paper presents an IoT-based intelligent platform for greenhouse monitoring and control. It utilizes a low-cost wireless sensor network and Raspberry Pi for data collection and processing. The platform incorporates a fuzzy logic controller to make optimal decisions for greenhouse management. Users can remotely monitor the greenhouse through an HMI, and data is stored in a MySQL database for analysis. The tests confirm the platform's effectiveness in enhancing greenhouse operations [16].
Application of an image and	Dan Jeric Arcega Rustia, Chien	This study presents an automated system for insect pest counting and

environmental sensor network for automated greenhouse insect pest monitoring	Erh Lin, Jui-Yung Chung, Yi-Ji Zhuang, Ju-Chun Hsu, Ta-Te Lin	environmental monitoring in greenhouses. It utilizes camera modules and a wireless sensor network to capture images and measure environmental parameters. An image processing algorithm accurately detects and counts insect pests on sticky traps. The system provides continuous and reliable pest count information, enabling timely pest management and environmental assessment. It offers a labor-saving alternative to manual counting and supports long-term pest behavior observations [17].
Development and design of mobile terminal APP for greenhouse environment control	Qiao Xiaohui, Du Shangfeng, He Yaofeng, Liang Meihui	This study aimed to address the need for remote greenhouse control, a mobile app for greenhouse environment control is proposed. Developed on the Android 2.3.3 platform, the app allows users to monitor real-time greenhouse parameters on their smartphones and remotely control the greenhouse equipment. This portable and flexible app enhances crop growth by providing an optimal environment, improving production efficiency and economic

		returns. Experimental results demonstrate the app's capability to obtain accurate real-time greenhouse data, enabling reliable monitoring and control with a user-friendly interface and strong expandability [18].
A new mobile application of agricultural pests' recognition using deep learning in cloud computing system	Mohamed Esmail Karar, Fahad Alsunaydi, Sultan Albusaymi, Sultan Alotaibi	This article discusses the development of a mobile application that uses artificial intelligence and deep learning for automatic pest classification in agriculture. The application utilizes a Faster R-CNN model integrated with cloud computing to accurately recognize and classify insect pests. It also provides a database of recommended pesticides associated with the detected pests. The study demonstrates high accuracy in pest recognition, outperforming other methods, and aims to implement the application for online pest recognition in different agricultural settings. [4]
Hybrid and organic photovoltaics for greenhouse applications	Luca La Notte, Lorena Giordano, Emanuele Calabro`, Roberto Bedini, Giuseppe	This paragraph discusses the exploration of agrivoltaics as a renewable technology to enhance greenhouse sustainability and reduce energy consumption. Agrivoltaics combines

	Colla, Giovanni Puglisi, Andrea Reale	farming and photovoltaics (PV) power generation. The effects of PV shading on plant growth and energy production are studied through simulations and experiments, highlighting the potential benefits of innovative PV systems in enhancing crop growth in hot and tropical regions. New PV technologies such as organic, dye-sensitized, and perovskite solar cells offer opportunities for integrating PV modules into greenhouses due to their semi-transparency and flexibility, allowing for optimized solar energy utilization [19].

2.6 Types of Mobile Applications for Greenhouse Development

Table 3: Types and purpose of mobile application

Types of mobile application for greenhouse	Purpose
Climate Monitoring and Control	Mobile applications of software development are designed to monitor and manage the environmental conditions of greenhouse systems remotely. These applications provide real-time data on temperature, humidity, light levels, CO_2 levels, and other relevant parameters, allowing growers to make

	informed decisions and adjust settings for optimal plant growth. This application provides growers with the ability to monitor and control the climate remotely, ensuring optimal growing conditions and improving crop productivity.
Irrigation and Fertilizer Management	Mobile applications of software development that are designed to assist growers in efficiently managing irrigation and fertilizer application for their plants. These applications provide a range of features to monitor, control, and optimize water and nutrient management within the greenhouse environment. By utilizing Irrigation and Fertilizer Management mobile applications, growers can enhance the efficiency of water and nutrient management, promote healthier plant growth, reduce resource wastage, and ultimately improve the overall productivity and sustainability of greenhouse operations.
Pest and Diseases Monitoring	Mobile applications of software development that are designed to help growers detect, identify, and manage pest and disease issues within the greenhouse environment. These applications provide real-time monitoring, data analysis, and recommendations to reduce the risks associated with pests and diseases. By utilizing Pest and Disease Monitoring mobile applications, growers can enhance their ability to detect, identify, and manage pest and disease issues in a timely manner. These applications provide valuable support for implementing effective pest and disease control measures, minimizing crop

	losses, and maintaining the overall health and productivity of greenhouse crops.
Crop Management and Tracking	Mobile applications of software development that are designed to assist growers in effectively managing and tracking their crops throughout the cultivation process. These applications provide features to monitor crop growth, record important information, and optimize several parameters to improve productivity. Also, these applications help ensure an efficient greenhouse system, a better resource budget, and effective tracking of critical information throughout the cultivation process in a greenhouse.
Inventory and Supply Chain Management	Mobile applications of software development that are designed to optimize the management of inventory and the supply chain processes within a greenhouse operation. These applications provide features to track inventory, manage the supply chain, monitor supply levels, and enhance overall efficiency in the greenhouse. Also, it helps to ensure the availability of necessary supplies, minimize stockouts, and optimize resources within the greenhouse system.
Knowledge and Resource Hub	Mobile application of software development that is designed to provide growers with a main platform for accessing important information, resources, and educational materials related to greenhouse operations. These applications offer a range of features to facilitate learning, knowledge sharing, and access to essential resources for greenhouse growers. Also, it

	serves as a valuable resource hub and encourages continuous learning, collaboration, and improvement within the greenhouse system.

2.7 Summary

By studying the works of previous researchers and relevant theories, valuable insights have been obtained for this project. This information clarifies the methods that previous researchers have employed. As a result, a comparative analysis of these methods has been conducted to highlight the similarities and differences. Furthermore, the underlying theories behind these components have been elucidated, providing a comprehensive understanding of how they contribute to the development and performance of the system.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will go over the project research and methodology by conducting all of the information, explanation, and overall strategy of the project. The method will be listed below which will be applied to this project for theoretical analysis. The development of the project sequence had been properly put in place to the concern of the suitable site in the greenhouse for the agriculture system.

The project flow has been depth discussed in the methodology part. A portion of the material explained is related to the development of machine learning and the development of mobile applications that are involved in completing the project. The purpose of this chapter is to provide additional information and confirmation of the way this project was carried out. The development of a remote monitoring system for the greenhouse involves art Artificial Intelligence (AI) and software implementation. This system is wearable and flexible by monitoring via smartphone through the application.

3.2 Project Flow

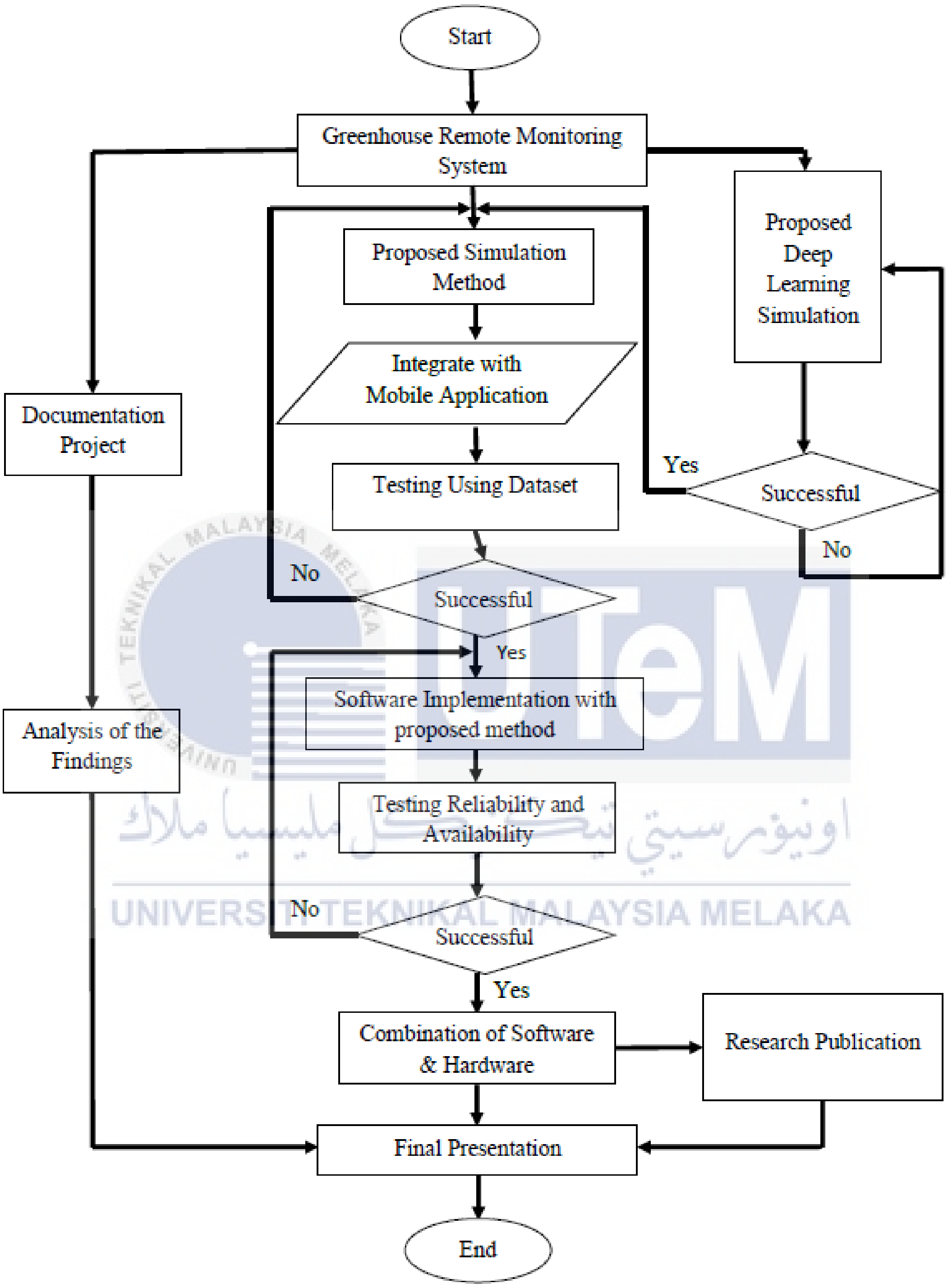


Figure 3.1: Project flowchart

3.3 Methodology

Developing an autonomous system for Greenhouse Remote Monitoring (G-REM) includes the major action of several systems. First, mobile development to develop an application as an instrument for users to inspect or monitor the data and condition from greenhouse sites. Second, machine learning based on Artificial Intelligence (AI) for image processing and image capture to develop a trained system that can predict and analyze every crop based on the conditions of every cultivation. Also, the data will be recorded in the user interface for users to follow up on any information.

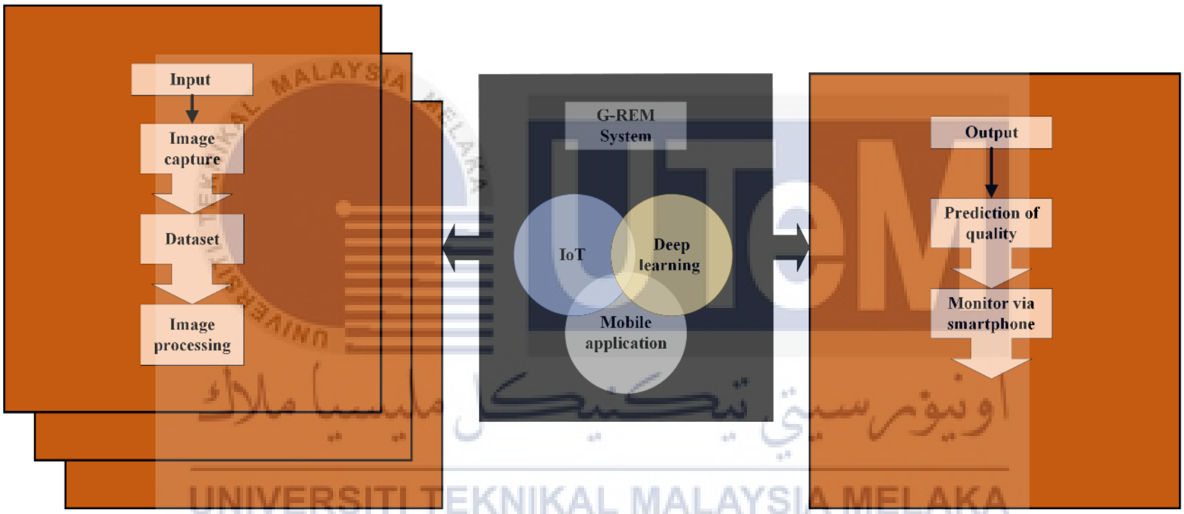


Figure 3.2: G-REM system operation block diagram

However, based on the objective and scope of this project idea, this system can only be executed for a greenhouse area in terms of low crops or herbs such as any plant that can be eaten that is identical to low crops of fruit or low crops vegetables.

Besides, the expected outcome that can be reached by the users is the cultivation of greenhouse crops can improve to the maximum efficiency if implementing this system. Users can define the conditions of every crop because this system can show the details of data from every low-crop plant. For example, users can define if there are any pests or diseases based on the monitoring system by image capture. Also, with machine learning that has been implemented in this system, it can define if there are any odds of strange conditions on those crops like the change of leaf colors, the black or white dots on the leaf, or any torn or damaged leaf.

Furthermore, users can reduce the waste of resource water and fertilizer by making an irrigation system and fertilizing system. This is based on the data that has been collected and recorded by the system which can give visual observations to the users to make any changes and improve the operation of the greenhouse system. Lastly, the monitoring system via smartphone embedded in this system to receive, record, and store data, users can monitor the system based on parameters that have been implanted in this system and reduce the manpower and waste energy.

3.4 Software Development

Software development in the context of greenhouse automation involves creating and implementing software solutions that enhance and standardize various processes within the greenhouse system. This includes developing an automation system software that enables for controlling and monitoring of planting reproduction such as the condition of the plants. The software is designed to provide precise monitoring of these environmental factors, ensuring optimal cultivation conditions for specific crops. Additionally, software development encompasses sensors and data collection devices to gather real-time information on environmental conditions and crop health. This data is then processed and analyzed in an application via smartphone for a user to make a decision and trigger an action to work autonomously for efficient greenhouse management.

Another crucial aspect of software development in greenhouse automation is the creation of user-friendly interfaces. These concepts allow users to interact with the system to monitor the data and record any information required. Also, advancing software development can provide visualizations, awareness, and notifications, to notify users about any oddness or differences in the greenhouse environment. Moreover, software development may involve integrating the automation system with external platforms or services, such as any sensing device or elements to enhance its functionality and enable perfect data exchange for improving productivity, optimizing resources, and enhancing overall management of the greenhouse environment.

3.5 Design and Description

The Greenhouse Remote Monitoring System (G-REM) implements several model designs particularly the Convolutional Neural Network (CNN) model to complete the project. Other than that, an open-source library is also being used to develop the model. Also, embedded with a mobile application for monitoring and tracking systems.

3.5.1 Mobile Development

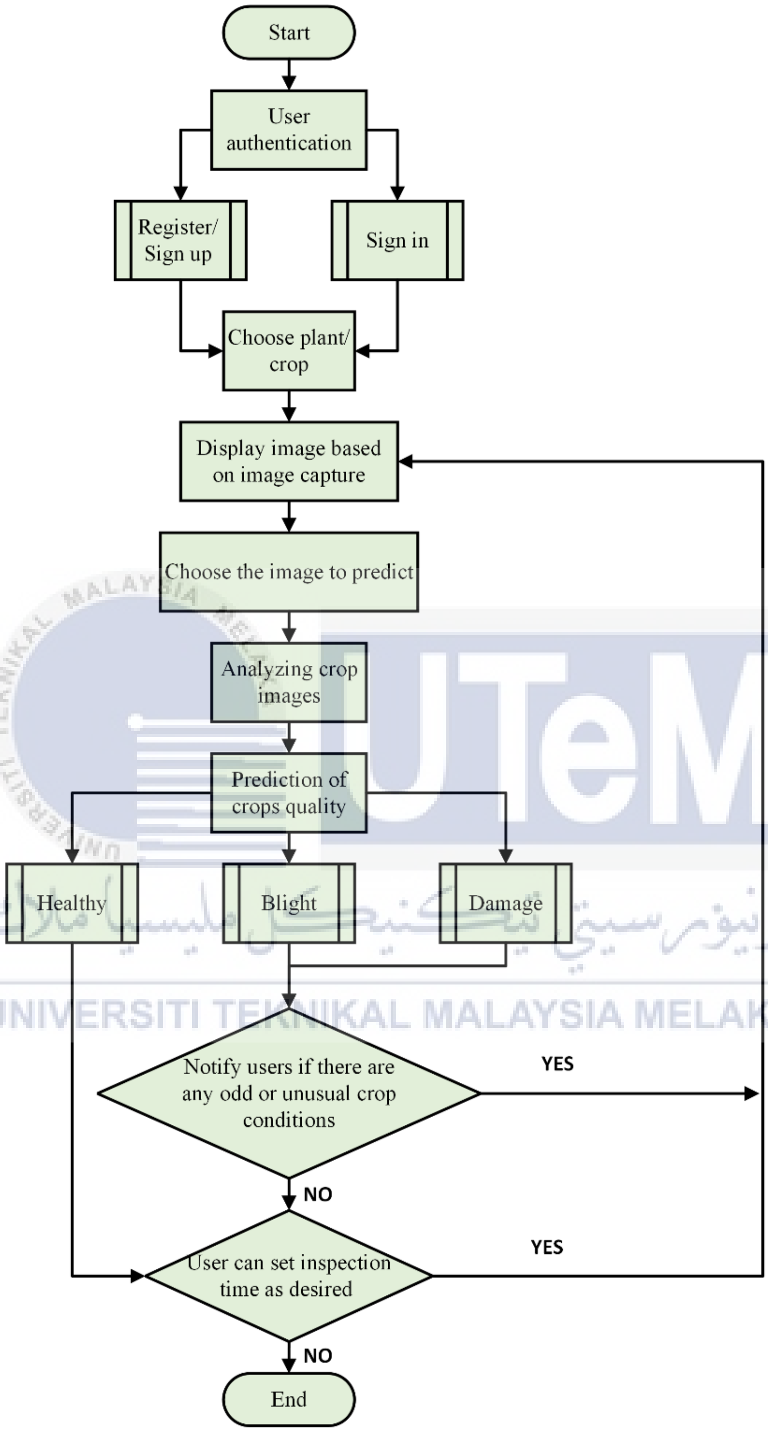


Figure 3.3: Mobile development flowchart

The integration of mobile applications with the greenhouse system can leverage the capabilities of both to enhance and innovate in the field of agriculture, especially in the environment of greenhouse technology. The implementation of the mobile application that will be developed in this greenhouse system, can provide a user-friendly interface for controlling and monitoring the greenhouse system via smartphone. This includes the features of specific environmental prediction to anticipate the condition of greenhouse plants. To some extent, users can easily manage irrigation and nutrient systems based on receiving and recording data. This means users will be more alert also can analyze plant cultivation based on system performance.

Figure 3.4: Mobile application development process

The combination of mobile applications with the greenhouse system can lead to the development of new technology that offers improvements in areas of automation, efficiency, and productivity in greenhouse operations of planting stage and treatment control. Also, can lead to optimized resource utilization, better crop management, enhanced yield, and reduced environmental impact due to decarbonization. Additionally, integrating the mobile application with advanced technologies like the Internet of Things (IoT), data analytics, deep learning, and machine learning can further augment the capabilities of the greenhouse system, enabling advanced monitoring, predictive analysis, and intelligent decision-making.

3.5.2 Application programming language (JavaScript)

Mobile applications that will be created for the greenhouse system use JavaScript as the programming language in Android Studio software and can be controlled via smartphone. This software provides the necessary tools, libraries, and frameworks to create strong features of applications only for the Android platform. This software can design and implement various functionalities related to greenhouse systems. For example, it can be created and designed to allow features for monitoring the prediction of crop conditions. Also, can allow the users to receive recorded data from time to time, based on image capture from the greenhouse system. It can serve users with insights and alerts regarding the conditions of the crops in the greenhouse and the overall greenhouse environment.

Figure 3.5: JavaScript

Furthermore, integrating this application with other technologies like IoT devices, and data analytics enhances the functionality and efficiency of the greenhouse system. The application can facilitate remote monitoring, data visualization, and analysis of parameters that notify users to undertake any decisions making and optimize the cultivation process in the greenhouse. Again, leveraging the capabilities of Android Studio and the programming languages it supports, can lead to the development of a powerful and user-friendly mobile application that fits the specific needs of the greenhouse system.

3.5.3 Image Processing Development

Image processing development by using Jupyter Notebook that is related to monitoring systems to improve greenhouse technology. Image processing can play a significant role in various aspects of greenhouse technology, such as monitoring, automation, prediction, and optimization. For instance, by analyzing specific parameter images of crops that have been captured from image capture based on the dataset, the Jupyter Notebook can serve as a platform to detect and analyze various visual indicators of plant health, such as leaf color, texture, shape, or disease. This can assist users in identifying the early signs and odds of each varying crop. Also, allows users to inspect if there are needs for intervention and improved crop management.

Figure 3.6: Jupyter Notebook Python

Jupyter Notebook can also be used to develop algorithms that detect and recognize diseases like pests or insects in specific scales to enable targeted pest management for cultivation strategies and reduce the use of pesticides. Additionally, by implementing image processing development in this system it can be utilized in greenhouse technology for crop cultivation analysis and perform an automated harvesting feature. Through analyzing images of crops over time, this open source can also measure growth parameters such as crop condition. This data provides valuable insights into crop management, allowing for the optimization and strategy of cultivation techniques. However, By leveraging the versatility and capabilities of Jupyter Notebook, innovative solutions can be developed to advance greenhouse technology and contribute to the sustainable and optimized cultivation of crops.

3.5.4 Deep Learning Development

In a deep learning system for developing greenhouse image processing, data is collected from the crops in the greenhouse environment by image capture. This data includes several types of images of leaf conditions such as the healthy leaves, blight leaves, and damaged leaves. The collected images are preprocessed to enhance their quality and remove noise. Also, can be trained on a labeled dataset of these images to learn patterns and features that distinguish different classes, such as healthy leaves, blight leaves, or damaged leaves. Once trained, the system can analyze the images and can start making predictions or classifications. This enables tasks and leads to overcome with automation systems such as disease detection, pest monitoring, and environmental analysis to improve greenhouse management and optimize crop health and productivity.

Figure 3.7: Python

Python's flexibility, extensive libraries, and community support make it a suitable language for developing both deep learning and machine learning models for image processing and image capture in developing greenhouse autonomous systems. Its wide adoption in the field of data science and Artificial Intelligence (AI) ensures access to a rich set of tools, resources, and documentation that can assist in creating innovative solutions for greenhouse technology. This can perform as an automation system and can automate tasks, provide data insights, and contribute to sustainable and efficient greenhouse practices.

3.5.5 TensorFlow

Tensorflow allows users to build a data input pipeline. Using this users can handle large datasets for the development of deep learning training by streaming the training samples from the dataset. The TensorFlow pipeline allows not only to stream the data for training but users can perform various transformations easily by performing any open source code such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and You Only Look Once (YOLO) based on deep learning requirement.

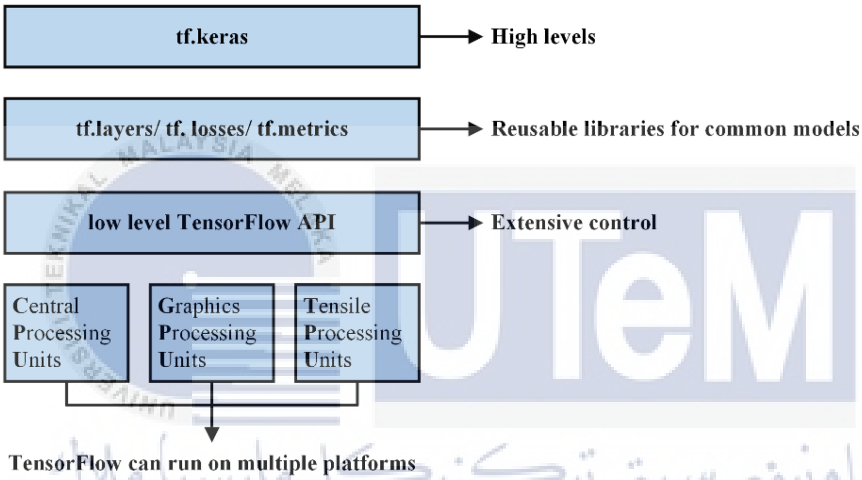


Figure 3.8: TensorFlow block diagram

G-REM project development uses TensorFlow as an open-source machine learning library that is aimed at simplifying the development and deployment of machine learning models, particularly deep learning. Offering flexibility, scalability, and support for Convolutional Neural Networks (CNN). TensorFlow enables users to build and train models for a wide range of tasks, including classification, regression, and clustering of the leaves. Its compatibility which includes CPUs and GPUs, as well as its support for distributed computing, makes it suitable for handling large-scale machine learning tasks. It is widely used in applications such as image recognition, image processing, and other areas of Artificial Intelligence (AI).

3.5.6 Prefetch

In the context of deep learning and machine learning, prefetch will be used to overlap the loading or preprocessing of data with the actual computation, improving overall efficiency. Especially when dealing with large datasets or complex preprocessing steps, the time it takes to load and process data can be significantly high.

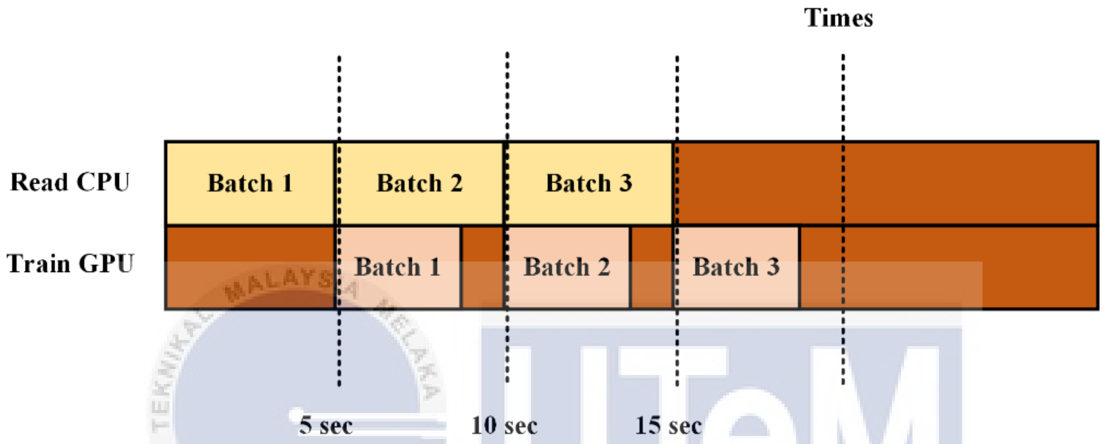


Figure 3.9: Prefetch process

The Greenhouse Remote Monitoring System (G-REM) consists of 300 datasets which are 100 datasets of healthy leaves, 100 datasets of blight leaves, and 100 datasets of damaged leaves. The datasets based on image capture help the G-REM system to analyze and perform a training method of Convolutional Neural Network (CNN). However, prefetching helps mitigate these problems by allowing the system to work on one part of the data while simultaneously loading or preparing the next set of data. Prefetching addresses this complication by decoupling the data loading and processing steps. Instead of waiting for the current batch of data to be fully loaded before starting the training, prefetching allows the model to start processing the current batch while the next batch is being loaded in the background. This overlapping of computation and data loading helps to keep the GPU or CPU busy, leading to improved training throughput.

3.5.7 Data Augmentation

This method is commonly used in machine learning, especially in the context of computer vision and deep learning, to artificially increase the size of a training dataset by applying various transformations to the existing data. G-REM's project goal of data augmentation is to improve the generalization and strength of a machine learning model by exposing it to a wider range of variations in the input data.

The process involves applying random transformations to the original training images or samples which can create new instances that are variations of the existing data. These transformations cover the scope of rotation to rotate the image by a certain angle, scaling for zooming in or out of the image, flipping acts as mirroring the image horizontally or vertically, translation for shifting the image along the x-axis or y-axis and modifying the color of the image to adjusting the several parameters such as brightness, contrast, or saturation.

3.5.8 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a specialized type of deep learning neural network employed in machine learning algorithms, specifically crafted for tasks related to image processing and computer vision. It proves highly effective in endeavors such as image classification, object detection, and segmentation. This G-REM system project works in handling data originating from image capture or other user-provided input models. The image dataset utilized for constructing this model consists of both testing and training data suitable for CNNs.

The training dataset encompasses up to 300 sample images, with each character represented by 32 kernels or layers. This dataset is instrumental in training a model to recognize leaves condition, incorporating these sample images for specific operations within systems.

The data includes sample images of types of leaf conditions such as healthy leaves, blight leaves, and damaged leaves. Following the completion of training, the trained CNN models become capable of classifying new, unseen leave images by inputting them into the network and utilizing the final layer's output for prediction. In a CNN's development, training data and testing data serve distinct purposes. The former is utilized to train the model, while the latter assesses the performance of the trained model. Notably, there exist several key differences between training data and testing.

There exist particular distinctions between training data and testing data in the context of Convolutional Neural Networks (CNN):

a) Purpose: The primary objective of training data is to facilitate the CNN in learning and adapting to the patterns and relationships within the data. Conversely, testing data serves the purpose of evaluating the CNN's performance, assessing its accuracy, and gauging its generalization capability.

b) Quantity: Training data is typically characterized by a larger quantity compared to testing data. This asymmetry arises from the need for the CNN to encounter numerous examples for effective learning, while a comparatively smaller number of examples suffices for evaluating the model's performance.

c) Diversity: Ensuring diversity in training data is crucial, as it should be representative of real-world data encountered by the CNN. Similarly, testing data should be diverse but distinct from the training data to accurately evaluate the CNN's generalization ability.

d) Labelling: Training data is conventionally labeled, encompassing both input data (images) and corresponding output labels (class labels). Testing data also bears labels, but these labels remain unused during training. Instead, they are employed solely to assess the accuracy of the CNN's predictions.

Figure 3.10: Sample of training dataset for healthy leaves

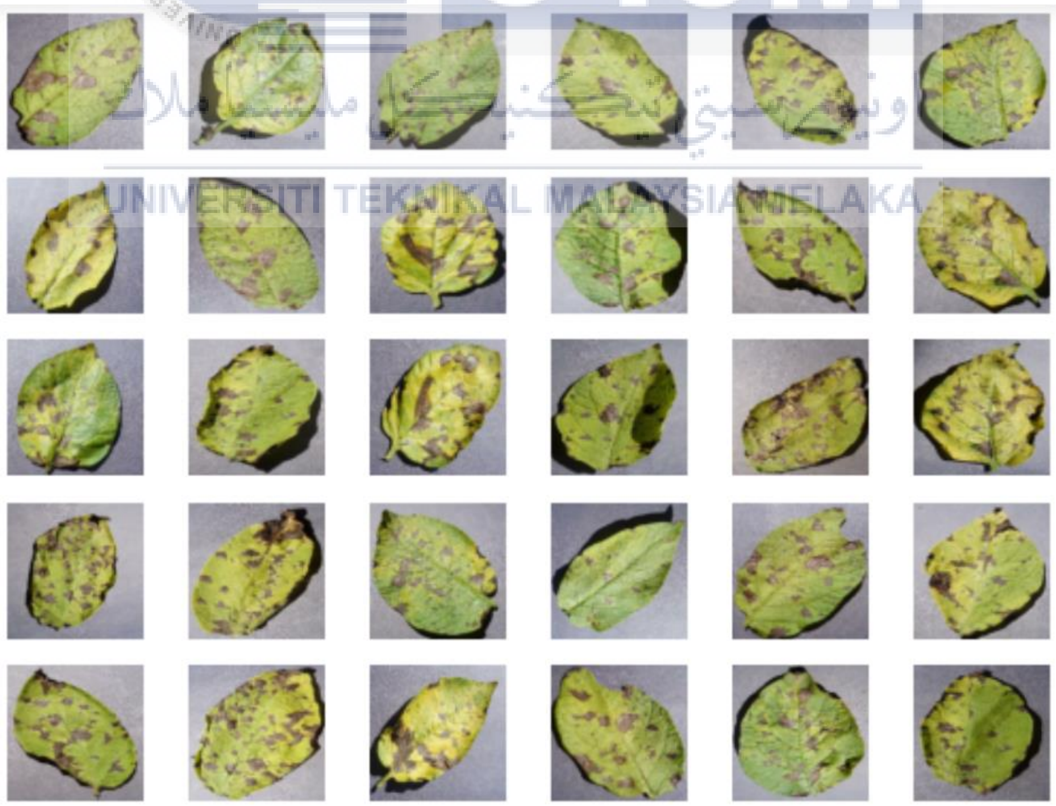


Figure 3.11: Sample of dataset for blight leaves

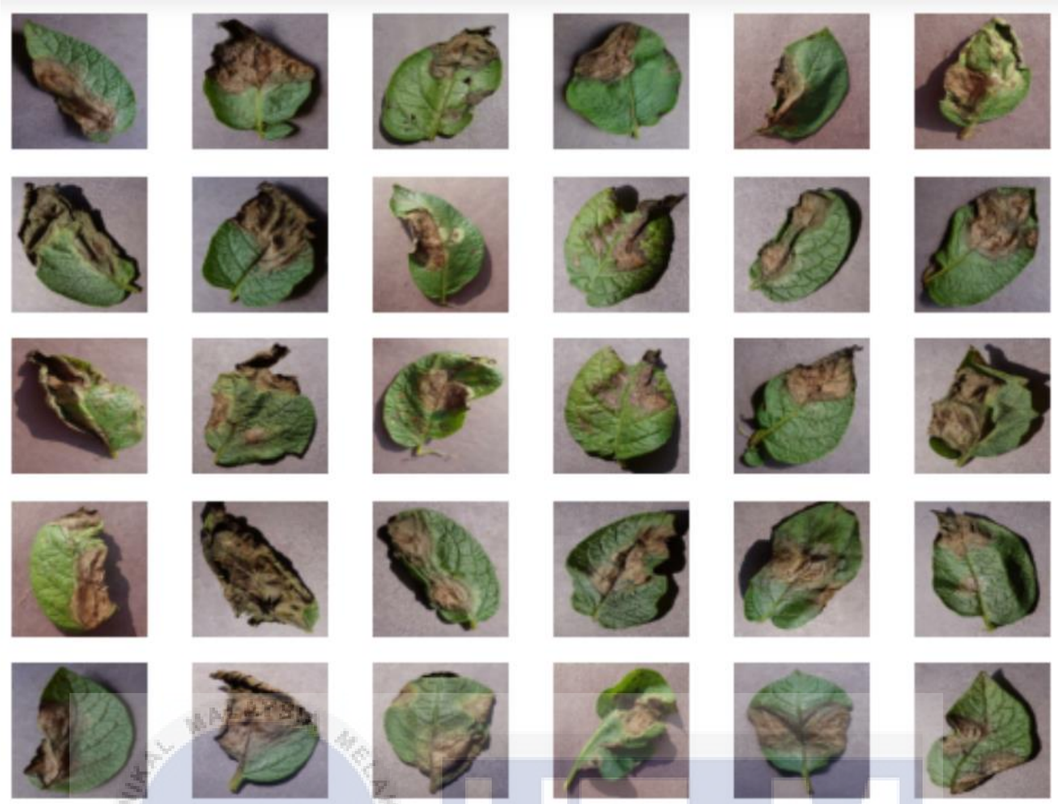


Figure 3.12: Sample of dataset for damaged leaves

Convolutional Neural Networks (CNN) are specifically crafted for processing data with a grid-like topology, such as images, making them highly adept at tasks involving pattern and feature identification within images. This is achieved through the application of convolutional filters to the input data, extracting and learning features from the image. Subsequently, there are more connected layers that follow these filters, utilizing the acquired features to make predictions or decisions regarding the input data. In certain scenarios, a combination of convolutional and pooling layers is employed. Convolution and pooling layers are incapable of performing classification with a fully connected neural network which is the highest priority.

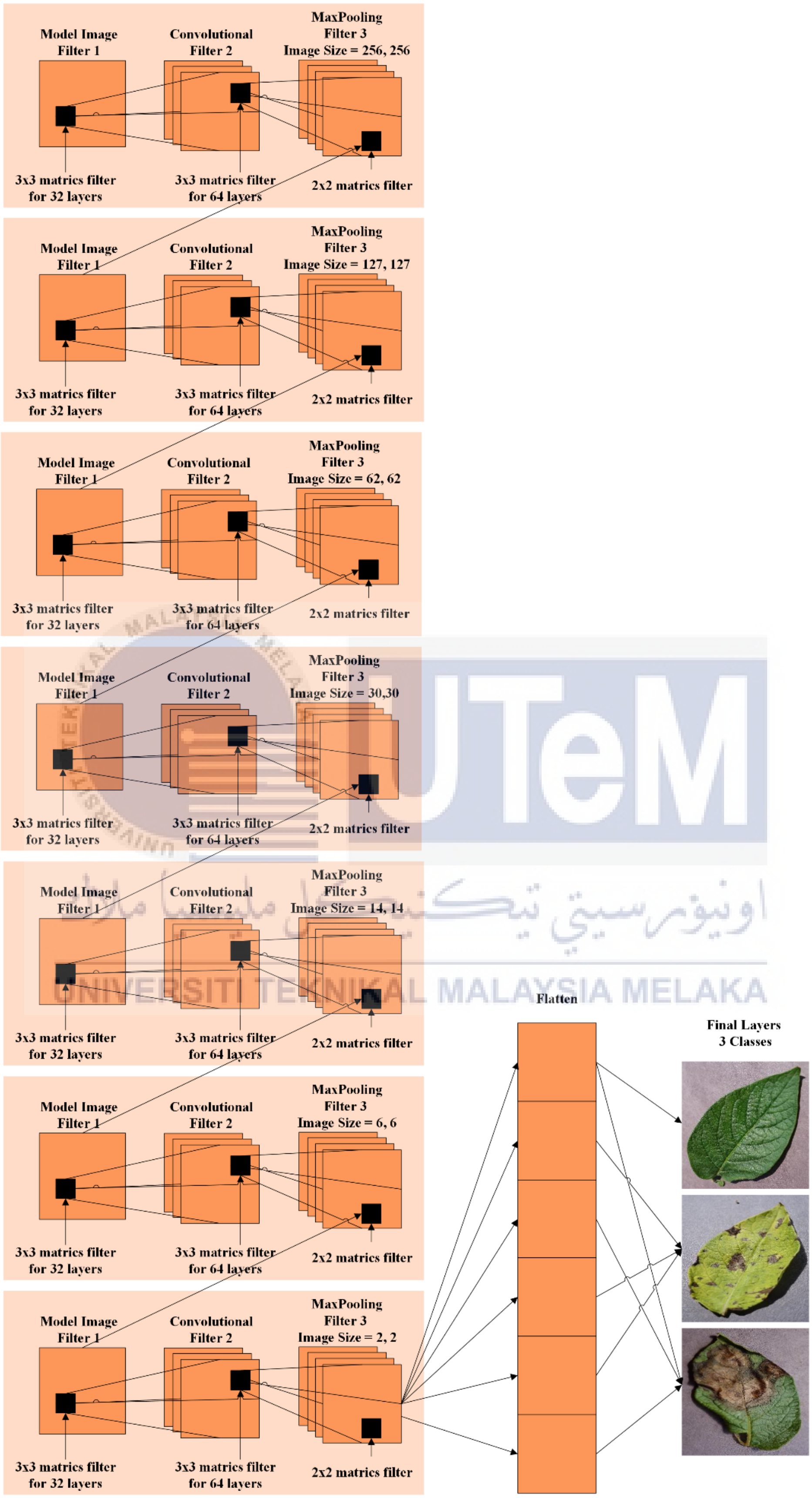


Figure 3.13: CNN structure of G-REM system

G-REM systems are using the deep learning of the Convolutional Neural Network (CNN) method that comprises several layers:

a) Input layer: This initial layer receives grayscale image data as input. Compiling the input image by using the image augmentation technique to fix and perform some action to the image. For example, rotation, flip, mirror, rescale, zoom in, zoom out, and contrast of the image.

b) Convolutional layers: These layers execute a convolution operation on the input data, utilizing a set of learnable filters of kernels (the size of matric in every layer) or weights to extract features, generating a set of feature maps. For example, the G-REM systems use using 3 x 3 matric of kernel size to improve the details for every layer.

c) Pooling layers: Employing techniques like "MaxPooling" or average pooling, these layers reduce the input data size by downsampling feature maps after filtering by the kernel layers part, decreasing spatial dimensions, and enhancing translation invariance.

d) Flatten layers: These layers establish dense connections, linking every neuron or filter in one layer to every neuron in the next. Flatten layers leverage features learned by convolutional and pooling layers to overcome the output based on the fully connected layers.

e) Output layer: This is the final layer that produces the CNN ultimate output, which may be a classification or envision. Just like the G-REM system output data the output of the final layer is to perform a prediction on the quality of potato leaves.

3.6 Summary

This chapter presents the development of a greenhouse automation system by developing mobile applications and image processing techniques. The methodology includes selecting a suitable site in the greenhouse, developing a software system for environmental control, solutions for automation, and user interaction. Mobile applications are designed to provide remote monitoring and control of the greenhouse system through smartphones. Image processing using OpenCV enables the analysis of crop images for plant health and pest detection. Besides, image processing of machine learning and deep learning algorithms implemented in Python facilitate the classification and prediction of crop conditions. The integration of these technologies aims to improve greenhouse operations, optimize resource utilization, and enhance crop management. Overall, the report emphasizes the role of technology in advancing greenhouse practices for sustainable and efficient cultivation.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This section presents the results and analysis of the development of an autonomous system for the greenhouse area. It highlights the capabilities of an autonomous system for greenhouse remote monitoring system (G-REM), with a specific focus on low-crop plants, particularly herbs. By utilizing the image capture, the system securely stores and efficiently manages data collected from sensors and cameras within the greenhouse area. This data can be easily accessed by users at any time, allowing users to remotely monitor and track environmental parameters and crop conditions. The system incorporates machine learning of Artificial Intelligence (AI) to analyze and inspect crop conditions based on planting reproduction. The goal of this autonomous greenhouse monitoring system is to enhance crop management and optimize greenhouse operations.

4.2 Results

The data from all of the software simulations that were recorded will be displayed in the mobile application via smartphone. The output is focused on the condition of the cultivation of the potato leaves which is between healthy, blighted, or damaged. From there, the application will be performed as a remote monitoring system for the Greenhouse Remote Monitoring System (G-REM) for tracking and monitoring purposes.

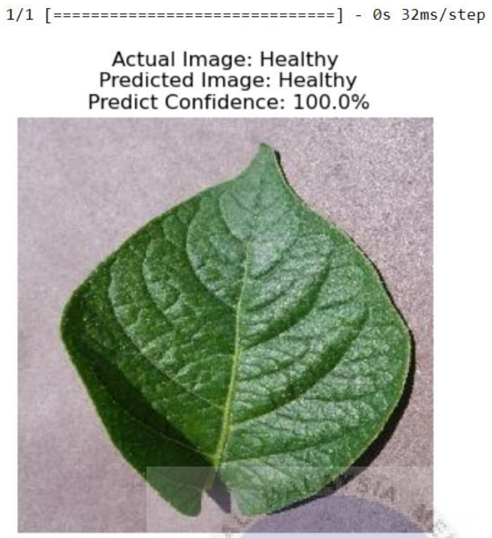
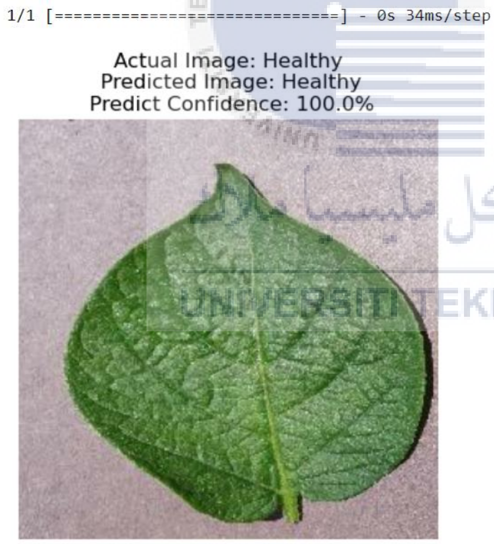
4.2.1 Result of Machine Learning System

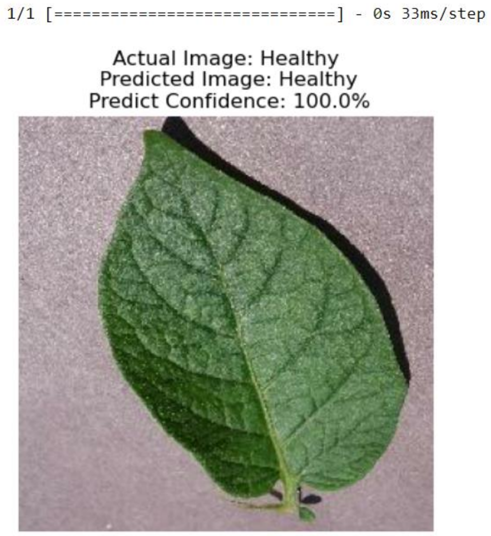
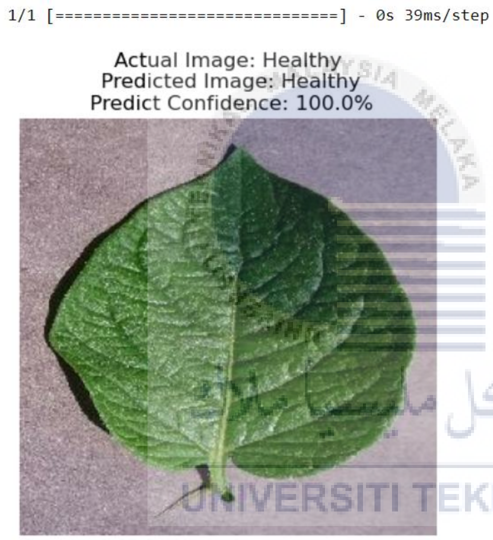
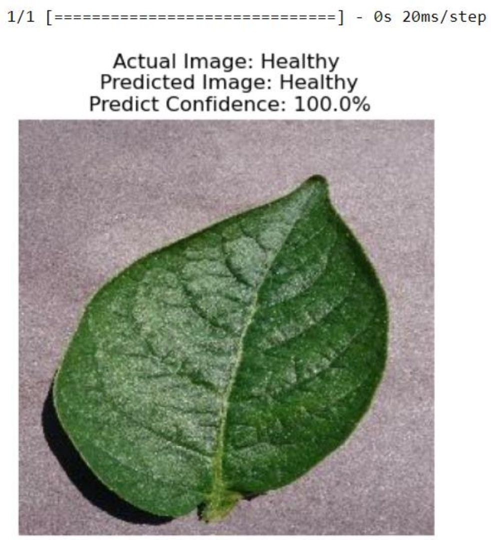
The development G-REM system for potato leaf conditions involves both deep learning and machine learning. The analyses encompass various aspects, including accuracy assessment, Convolutional Neural Network (CNN), data exploration, and image processing. Feature importance and interpretability are considered, along with robustness testing and integration with the system simulation. The goal is a comprehensive evaluation of user satisfaction and overall system effectiveness.

4.2.1.1 Prediction Samples of Healthy Leaves

The table below shows the output of G-REM systems that are consistent with 100% accuracy for the healthy leaves. To emphasize the quality and performance of the image processing and image capture method, a huge improvement needs to be made to provide an advantage especially if there is a big-scale dataset. This is the first five samples of predicted data of healthy leaves that had been observed through the G-REM system. The g-REM system was able to predict images that were accurate to the actual image after several times of training.

Table 4: 5 samples of predicted data from G-REM System for healthy leaves

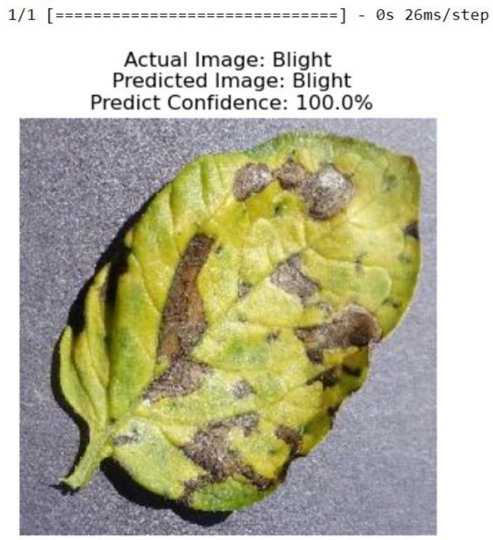
Type of class: Actual Image (Healthy)	Predict Image	Accuracy, %	Predict Time, ms
	Healthy	100%	32ms
	Healthy	100%	34ms

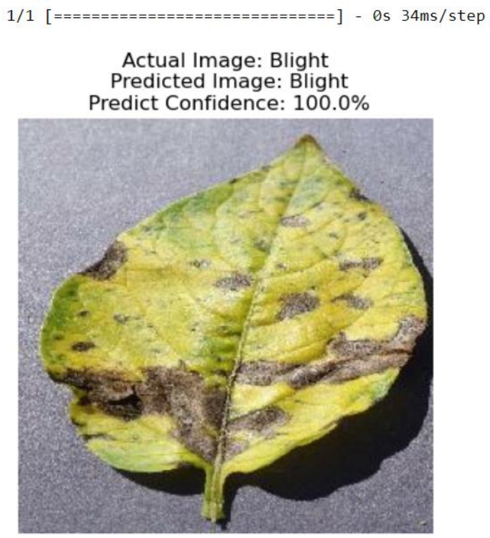
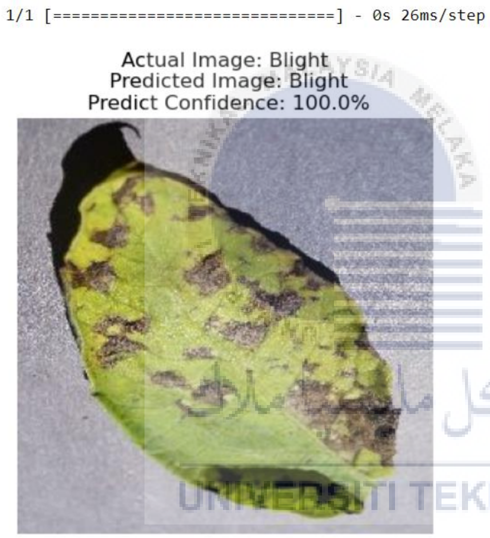
	Healthy	100%	33ms
	Healthy	100%	39ms
	Healthy	100%	20ms

4.2.1.2 Prediction Samples of Blight Leaves

The table below shows the output of G-REM systems that are almost consistent and that was close to 100% accuracy for the blight leaves. This is the first five samples of predicted data of healthy leaves that had been observed through the G-REM system. The g-REM system was able to predict images that were accurate to the actual image after several times of training.

Table 5: 5 samples of predicted data from G-REM System for blight leaves

Type of class: Actual Image (Blight)	Predict Image	Accuracy, %	Predict Time, ms
	Blight	100%	36ms
	Blight	100%	26ms

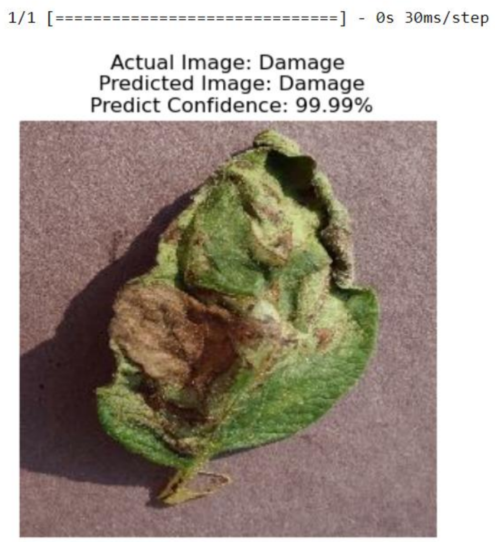
	Blight	100%	34ms
	Blight	100%	26ms
	Blight	99.99%	34ms

4.2.1.3 Prediction Samples of Damage Leaves

The table below shows the output of G-REM systems that are also almost consistent and that was close to 100% with 3 of them being 99.99% accurate in predicting the damaged leaves. So, this is the first five samples of predicted data of healthy leaves that had been observed through the G-REM system. The g-REM system was able to predict images that were accurate to the actual image after several times of training.

Table 6: 5 samples of predicted data from G-REM System for damage leaves

Type of class: Actual Image (Damage)	Predict Image	Accuracy, %	Predict Time, ms
	Damage	100%	36ms

	Damage	100%	26ms
	Damage	99.99%	34ms
	Damage	99.99%	26ms

1/1 [==============================] - 0s 33ms/step Actual Image: Damage Predicted Image: Damage Predict Confidence: 99.99% 	Damage	99.99%	34ms

4.2.1.4 Summary of CNN Model

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (32, 256, 256, 3)         0

 conv2d (Conv2D)             (32, 254, 254, 32)        896

 max_pooling2d (MaxPooling2  (32, 127, 127, 32)        0
 D)

 conv2d_1 (Conv2D)           (32, 125, 125, 64)        18496

 max_pooling2d_1 (MaxPoolin  (32, 62, 62, 64)          0
 g2D)

 conv2d_2 (Conv2D)           (32, 60, 60, 64)          36928

 max_pooling2d_2 (MaxPoolin  (32, 30, 30, 64)          0
 g2D)

 conv2d_3 (Conv2D)           (32, 28, 28, 64)          36928

 max_pooling2d_3 (MaxPoolin  (32, 14, 14, 64)          0
 g2D)

 conv2d_4 (Conv2D)           (32, 12, 12, 64)          36928

 max_pooling2d_4 (MaxPoolin  (32, 6, 6, 64)            0
 g2D)

 conv2d_5 (Conv2D)           (32, 4, 4, 64)            36928

 max_pooling2d_5 (MaxPoolin  (32, 2, 2, 64)            0
 g2D)

 flatten (Flatten)           (32, 256)                 0

 dense (Dense)               (32, 64)                  16448

 dense_1 (Dense)             (32, 3)                   195

=================================================================
Total params: 183747 (717.76 KB)
Trainable params: 183747 (717.76 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```


Figure 4.1: CNN model summary

The summary models from the G-REM system provide a concise overview of the model samples, making it easier for users to understand the structure of the neural network and check for any unexpected issues such as the size of layers or the total number of parameters. By using "model.summary()" It provides a textual representation of the model samples, including information about the layers, the number of parameters, and the output shapes. The "Total params:" is the number of the parameters which indicates the number of trainable parameters in each layer. That means the G-REM system indicates 183747 (717.76kb) or the total parameter or the size of the total layers. These parameters are the weights that the model of image learns during training.

```
Epoch 1/30
8/8 [==============================] - 13s 1s/step - loss: 1.1000 - accuracy: 0.3398 - val_loss: 1.1012 - val_accuracy: 0.2500
Epoch 2/30
8/8 [==============================] - 10s 1s/step - loss: 1.0544 - accuracy: 0.4062 - val_loss: 1.0013 - val_accuracy: 0.3125
Epoch 3/30
8/8 [==============================] - 10s 1s/step - loss: 0.7959 - accuracy: 0.6289 - val_loss: 0.6277 - val_accuracy: 0.8125
Epoch 4/30
8/8 [==============================] - 9s 1s/step - loss: 0.5020 - accuracy: 0.7383 - val_loss: 0.3267 - val_accuracy: 0.8125
Epoch 5/30
8/8 [==============================] - 10s 1s/step - loss: 0.3037 - accuracy: 0.9062 - val_loss: 0.0773 - val_accuracy: 0.9688
Epoch 6/30
8/8 [==============================] - 10s 1s/step - loss: 0.0584 - accuracy: 0.9883 - val_loss: 0.0095 - val_accuracy: 1.0000
Epoch 7/30
8/8 [==============================] - 10s 1s/step - loss: 0.0153 - accuracy: 0.9922 - val_loss: 0.0037 - val_accuracy: 1.0000
Epoch 8/30
8/8 [==============================] - 10s 1s/step - loss: 0.0071 - accuracy: 0.9961 - val_loss: 0.0098 - val_accuracy: 1.0000
Epoch 9/30
8/8 [==============================] - 9s 1s/step - loss: 0.0032 - accuracy: 1.0000 - val_loss: 2.0993e-04 - val_accuracy: 1.00
00
Epoch 10/30
8/8 [==============================] - 9s 1s/step - loss: 0.0016 - accuracy: 1.0000 - val_loss: 1.3803e-04 - val_accuracy: 1.00
00
Epoch 11/30
8/8 [==============================] - 9s 1s/step - loss: 6.2258e-04 - accuracy: 1.0000 - val_loss: 6.2271e-04 - val_accuracy:
1.0000
Epoch 12/30
8/8 [==============================] - 9s 1s/step - loss: 2.9006e-04 - accuracy: 1.0000 - val_loss: 0.0020 - val_accuracy: 1.00
00
Epoch 13/30
8/8 [==============================] - 9s 1s/step - loss: 4.3390e-04 - accuracy: 1.0000 - val_loss: 4.9157e-04 - val_accuracy:
1.0000
Epoch 14/30
8/8 [==============================] - 9s 1s/step - loss: 1.4791e-04 - accuracy: 1.0000 - val_loss: 6.1401e-05 - val_accuracy:
1.0000
Epoch 15/30
8/8 [==============================] - 9s 1s/step - loss: 1.6903e-04 - accuracy: 1.0000 - val_loss: 4.5373e-05 - val_accuracy:
1.0000
```


Figure 4.2: Output of system accuracy for Epoch 1 - 15

```
Epoch 16/30
8/8 [==============================] - 9s 1s/step - loss: 1.2322e-04 - accuracy: 1.0000 - val_loss: 5.7271e-05 - val_accuracy:
1.0000
Epoch 17/30
8/8 [==============================] - 9s 1s/step - loss: 1.1182e-04 - accuracy: 1.0000 - val_loss: 6.1820e-05 - val_accuracy:
1.0000
Epoch 18/30
8/8 [==============================] - 9s 1s/step - loss: 9.0564e-05 - accuracy: 1.0000 - val_loss: 5.4564e-05 - val_accuracy:
1.0000
Epoch 19/30
8/8 [==============================] - 9s 1s/step - loss: 7.5158e-05 - accuracy: 1.0000 - val_loss: 4.8035e-05 - val_accuracy:
1.0000
Epoch 20/30
8/8 [==============================] - 9s 1s/step - loss: 6.5163e-05 - accuracy: 1.0000 - val_loss: 4.4420e-05 - val_accuracy:
1.0000
Epoch 21/30
8/8 [==============================] - 9s 1s/step - loss: 5.5095e-05 - accuracy: 1.0000 - val_loss: 4.8985e-05 - val_accuracy:
1.0000
Epoch 22/30
8/8 [==============================] - 9s 1s/step - loss: 4.8330e-05 - accuracy: 1.0000 - val_loss: 5.1105e-05 - val_accuracy:
1.0000
Epoch 23/30
8/8 [==============================] - 9s 1s/step - loss: 4.1791e-05 - accuracy: 1.0000 - val_loss: 4.3993e-05 - val_accuracy:
1.0000
Epoch 24/30
8/8 [==============================] - 9s 1s/step - loss: 3.6036e-05 - accuracy: 1.0000 - val_loss: 4.1522e-05 - val_accuracy:
1.0000
Epoch 25/30
8/8 [==============================] - 9s 1s/step - loss: 3.2247e-05 - accuracy: 1.0000 - val_loss: 3.9672e-05 - val_accuracy:
1.0000
Epoch 26/30
8/8 [==============================] - 9s 1s/step - loss: 2.9104e-05 - accuracy: 1.0000 - val_loss: 3.4367e-05 - val_accuracy:
1.0000
Epoch 27/30
8/8 [==============================] - 9s 1s/step - loss: 2.5878e-05 - accuracy: 1.0000 - val_loss: 3.2893e-05 - val_accuracy:
1.0000
Epoch 28/30
8/8 [==============================] - 9s 1s/step - loss: 2.3160e-05 - accuracy: 1.0000 - val_loss: 3.9007e-05 - val_accuracy:
1.0000
Epoch 29/30
8/8 [==============================] - 10s 1s/step - loss: 2.0800e-05 - accuracy: 1.0000 - val_loss: 3.4854e-05 - val_accuracy:
1.0000
Epoch 30/30
8/8 [==============================] - 10s 1s/step - loss: 1.8541e-05 - accuracy: 1.0000 - val_loss: 3.3625e-05 - val_accuracy:
1.0000
```

Figure 4.3: Output of system accuracy for Epoch 16 - 30

The neural network undergoes training using the provided training set in the deep learning function. Subsequently, its performance is evaluated using the test set. The functions yield two metrics for each epoch: 'accuracy' represents the accuracy of predictions on the training set, while 'val accuracy' signifies the validation accuracy attained on the test set. However, the epoch that the G-REM system uses refers to one complete pass through the entire training dataset during the training phase of a neural network. The G-REM system trains up to 30 epochs in the context of training a model, the dataset is typically divided into batches, and during each epoch, the model iterates through all the batches, updating its weights based on the computed descent.

4.2.2 Result of Mobile Application

The development of a mobile application as a monitoring and tracking system device via smartphone. The G-REM system which can predict the condition of potato leaves and can display the data via smartphone can be a valuable tool for farmers and agriculturalists. The G-REM application can provide data insights into the health, blight, and damaged leaves of potato plants. This will allow users to reduce the time for actions to address any odd or potential issues.

4.3 Analysis

The G-REM system development delivers these analyses that can gain insights into the overall effectiveness, user satisfaction, and performance of both the mobile application and the underlying machine learning or deep learning model. Continuous monitoring, feedback collection, and iterative improvements will contribute to the long-term success of your system.

4.3.1 Analysis of Deep Learning and Machine Learning System

The G-REM system development is based on potato leaves condition and cultivation prediction system, the analysis depend on factors such as the size of the dataset, complexity of patterns, and interpretability requirements. Machine learning that has been developed offers interpretability and may be suitable for simpler cases with limited data, while deep learning development with its feature learning and ability to handle complex patterns like tone of color and disease, also excels when dealing with large, diverse datasets. A hybrid approach, combining the strengths of both machine learning and deep learning, could also be considered for optimal results.

4.3.1.1 Prediction Time

By referring to the table below, the G-REM system has recorded the first 30 random samples of the image model to determine and measure the prediction time that will be consumed to predict every model. Besides, prediction time known as inference time or latency that one of the most important aspects to consider when deploying machine learning models, including Convolutional Neural Networks (CNNs). It refers to the time it takes for the model to process input data and generate a prediction. The 30 random samples include the actual image of 10 healthy leaves, 10 blight leaves, and 10 damaged leaves. The estimated time for the G-REM system to predict 1 model of the image is between 30 milliseconds and 40 milliseconds which means the average time for 1 model of image can be predicted is 32.1 milliseconds.

Table 7: First 30 sample of image prediction time

No.	Image Cassification			Prediction Time, ms
	Healthy	Blight	Damage	
1.			✓	33ms
2.		✓		31ms
3.	✓			31ms
4.		✓		32ms
5.	✓			32ms
6.			✓	31ms
7.		✓		33ms

8.		✓		31ms
9.	✓			32ms
10.		✓		37ms
11.	✓			33ms
12.		✓		40ms
13.			✓	32ms
14.			✓	32ms
15.	✓			31ms
16.			✓	31ms
17.			✓	33ms
18.	✓			32ms
19.			✓	32ms
20.	✓			33ms
21.			✓	32ms
22.		✓		30ms
23.	✓			31ms
24.	✓			31ms
25.		✓		31ms
26.			✓	31ms

27.			✓	31ms
28.		✓		31ms
29.		✓		31ms
30.	✓			32ms

4.3.1.2 Graph Analysis of Accuracy

The graph that is shown is for training accuracy and validation accuracy. Based on the graph, it is represented by two lines, the blue line for training accuracy and the orange line for validation accuracy. The aim of performing the blue line (training accuracy) is to measure the accuracy of the model on the training dataset. It indicates how well the model is learning the patterns and features present in the training data. However, the orange line (validation accuracy) is to assess the model's performance on a separate dataset not used during training.

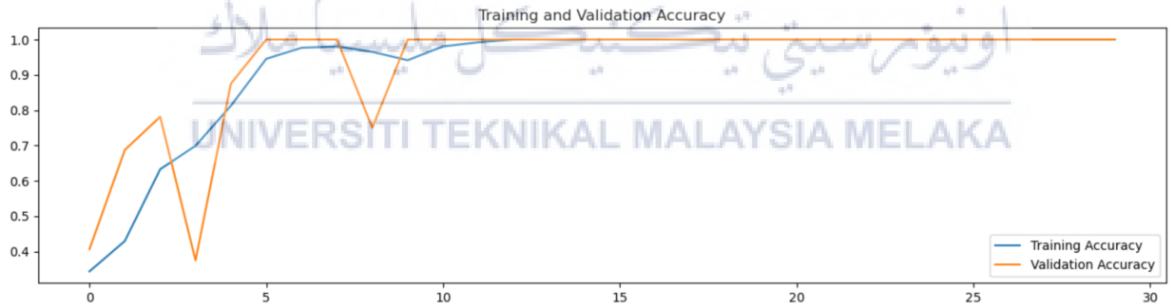


Figure 4.4: Graph of training accuracy and validation accuracy

4.3.1.3 Graph Analysis of Loss

The graph that is shown is for training loss and validation loss. Based on the graph, it is represented by two lines, the blue line for training loss and the orange line for validation loss. The aim of performing the blue line (training loss) is to measure the error between the predicted image and the actual image during the training phase. It reflects how well the model is learning from the training data. However, the orange line (validation loss) is to be computed on a separate dataset not used during training which is the validation set.

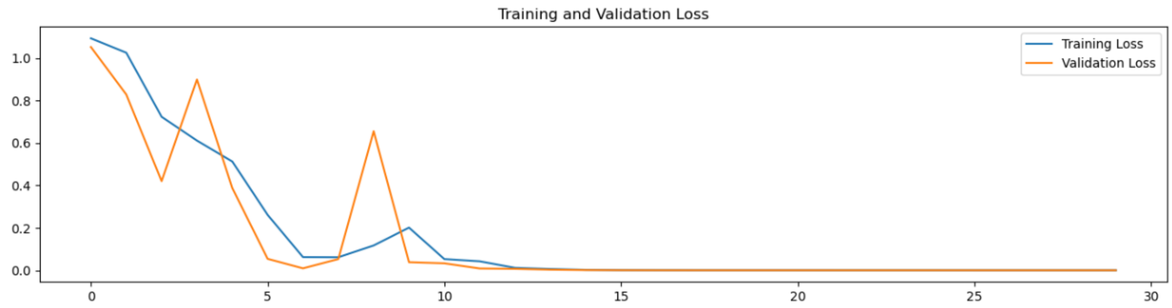


Figure 4.5: Graph of training loss and validation loss

4.4 Summary

This chapter presents the preliminary results and analysis of an autonomous system for greenhouse monitoring. The system, called G-REM (Greenhouse Remote Monitoring), utilizes cloud storage, mobile applications, and machine learning technology which is image processing and image capture to remotely monitor and track environmental cultivation and crop conditions in a greenhouse. The mobile application allows users to access, track, and record data and make informed decisions about crop management. Machine learning and deep learning algorithms, implemented using Jupyter Notebook and Python, enable image processing and image capture for analysis for identifying crop conditions whether it healthy, blight, or damaged by pests or diseases. The system aims to enhance crop management, optimize resource usage, and improve greenhouse operations for low-crop plants, particularly herbs. The report emphasizes the potential environmental impact of this system, including reduced waste of resources and energy, as well as improved efficiency in greenhouse cultivation.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

To sum up, the development of remote monitoring systems using IoT technology for greenhouses (G-REM) offers significant benefits. By monitoring the plant's cultivations that are embedded with advanced technologies, such as image processing, image capture, and data analytics, the system enhances productivity and ensures high quality and maximum efficiency. The project focuses on creating a user-friendly application that facilitates the tracking features for plant reproduction in the greenhouse.

This remote monitoring system supports sustainable agriculture processes and reduces waste of resources consumption leading to the growth and success of the greenhouse industry. By implementing the monitoring system, greenhouse operators can execute tracking systems, and control environmental parameters which resulting in improved crop quality, increased productivity of crops, and reduced operational costs of manpower. This system provides valuable data insights and enables measures of precautions to be taken to prevent crop damage.

Also, it allows users to access and control, enhancing convenience and flexibility. Moreover, the integration of Artificial Intelligence (AI) technology promotes sustainability by optimizing the parameters which will lead to the optimization of resource utilization and minimization of environmental impact against greenhouse [20]. Overall, the implementation of a deep learning-based remote monitoring system revolutionizes greenhouse operations, empowering the system with datasets from image capture, automation capabilities, and sustainable development.

5.2 Contribution of Research

Based on the chapter presented, the case studies demonstrate the expected capability of this autonomous system that will be applied in the greenhouse system. The selected area for the greenhouse system only covered low-crop plants, especially herbs. With the utilization of image capture, the data obtained from the dataset within the greenhouse and monitoring system can be securely stored, efficiently managed and easily accessed anytime by users.

The ability to access data remotely allows users to monitor and inspect overall environmental parameters and users can keep track of the crops from time to time if there are any odds or strange conditions. Also, implemented with machine learning development this system is trained to predict and define the condition of the crop and monitor specific parameters based on the types of crops that users want.

5.3 Future Works

The project of Greenhouse Remote Monitoring (G-REM) that has been created only covers the software system which means it only involves developing the system software for the development of machine learning and mobile applications. The goal is to perform a remote monitoring system that can predict and track the data from image capture. This system has been trained based on the insight dataset and will display the recorded data via smartphone.

There are various parts for contributing to the innovation of this project. Users can make an innovation in a part of hardware development to complete and make full approaches to emerge as a complete product for commercial.

However, by reforming in terms of deep learning, users can improve by making a specific parameter for collecting data. Such as soil moisture, temperature, pH value, and humidity for the greenhouse plants. Also, integrating with the real-time data for deep learning steaming would allow continuous monitoring and immediate responses to changes in plant conditions.

REFERENCES

[1] C. Maraveas, D. Piromalis, K. G. Arvanitis, T. Bartzanas, and D. Loukatos, "Applications of IoT for optimized greenhouse environment and resources management," *Comput Electron Agric*, vol. 198, Jul. 2022, doi: 10.1016/j.compag.2022.106993.

[2] S. Al-Naemi and A. Al-Otoom, "Smart sustainable greenhouses utilizing microcontroller and IOT in the GCC countries; energy requirements & economical analyses study for a concept model in the state of Qatar," *Results in Engineering*, vol. 17, no. January, p. 100889, 2023, doi: 10.1016/j.rineng.2023.100889.

[3] R. Aarthi, D. Sivakumar, and V. Mariappan, "Smart Soil Property Analysis Using IoT: A Case Study Implementation in Backyard Gardening," *Procedia Comput Sci*, vol. 218, no. 2022, pp. 2842–2851, 2023, doi: 10.1016/j.procs.2023.01.255.

[4] M. E. Karar, F. Alsunaydi, S. Albusaymi, and S. Alotaibi, "A new mobile application of agricultural pests recognition using deep learning in cloud computing system," *Alexandria Engineering Journal*, vol. 60, no. 5, pp. 4423–4432, 2021, doi: 10.1016/j.aej.2021.03.009.

[5] S. Ale *et al.*, "Smart Agricultural Technology Development and evaluation of a decision support mobile application for cotton irrigation management," *Smart Agricultural Technology*, vol. 5, no. June, p. 100270, 2023, doi: 10.1016/j.atech.2023.100270.

[6] M. Tao *et al.*, "Smartphone-based detection of leaf color levels in rice plants," *Comput Electron Agric*, vol. 173, no. April, 2020, doi: 10.1016/j.compag.2020.105431.

[7] Z. Jianjun, W. Xiaofang, W. Xiu, Z. Wei, and C. Jichen, "Greenhouse Monitoring and Control System Based on Zigbee."

[8] M. Nandhini, K. U. Kala, M. Thangadarshini, and S. Madhusudhana Verma, “Deep Learning model of sequential image classifier for crop disease detection in plantain tree cultivation,” *Comput Electron Agric*, vol. 197, no. March, p. 106915, 2022, doi: 10.1016/j.compag.2022.106915.

[9] J. Kotwal, Dr. R. Kashyap, and Dr. S. Pathan, “Agricultural plant diseases identification: From traditional approach to deep learning,” *Mater Today Proc*, vol. 80, pp. 344–356, 2023, doi: 10.1016/j.matpr.2023.02.370.

[10] W. Lin, L. Shu, W. Zhong, W. Lu, D. Ma, and Y. Meng, “Online classification of soybean seeds based on deep learning,” *Eng Appl Artif Intell*, vol. 123, no. April, p. 106434, 2023, doi: 10.1016/j.engappai.2023.106434.

[11] J. G. A. Barbedo, “Using digital image processing for counting whiteflies on soybean leaves,” *Journal of Asia-Pacific Entomology*, vol. 17, no. 4. pp. 685–694, 2014. doi: 10.1016/j.aspen.2014.06.014.

[12] W. Li, D. Wang, M. Li, Y. Gao, J. Wu, and X. Yang, “Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse,” *Comput Electron Agric*, vol. 183, no. December 2020, p. 106048, 2021, doi: 10.1016/j.compag.2021.106048.

[13] A. Nasiri, M. Omid, A. Taheri-garavand, and A. Jafari, “Sustainable Computing : Informatics and Systems Deep learning-based precision agriculture through weed recognition in sugar beet fields,” *Sustainable Computing: Informatics and Systems*, vol. 35, no. May, p. 100759, 2022, doi: 10.1016/j.suscom.2022.100759.

[14] C. Zhou *et al.*, “Smart Agricultural Technology Detecting two-spotted spider mites and predatory mites in strawberry using deep learning,” *Smart Agricultural Technology*, vol. 4, no. February, p. 100229, 2023, doi: 10.1016/j.atech.2023.100229.

[15] “Classification and detection of insects from field images using deep learning.”

[16] H. Benyezza, M. Bouhedda, R. Kara, and S. Rebouh, “Smart platform based on IoT and WSN for monitoring and control of a greenhouse in the context of precision agriculture,” *Internet of Things (Netherlands)*, vol. 23, no. May, p. 100830, 2023, doi: 10.1016/j.iot.2023.100830.

[17] D. J. A. Rustia, C. E. Lin, J. Y. Chung, Y. J. Zhuang, J. C. Hsu, and T. Te Lin, “Application of an image and environmental sensor network for automated greenhouse insect pest monitoring,” *J Asia Pac Entomol*, vol. 23, no. 1, pp. 17–28, Apr. 2020, doi: 10.1016/j.aspen.2019.11.006.

[18] Q. Xiaohui, D. Shangfeng, H. Yaofeng, and L. Meihui, “Development and design of mobile terminal APP for greenhouse environment control,” *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 822–825, 2018, doi: 10.1016/j.ifacol.2018.08.093.

[19] R. Bedini *et al.*, “Hybrid and organic photovoltaics for greenhouse applications,” vol. 278, no. September, 2020, doi: 10.1016/j.apenergy.2020.115582.

[20] S. Xia, X. Nan, X. Cai, and X. Lu, “Data fusion based wireless temperature monitoring system applied to intelligent greenhouse,” *Comput Electron Agric*, vol. 192, no. December 2021, p. 106576, 2022, doi: 10.1016/j.compag.2021.106576.

[21] D. J. Zhu *et al.*, “Knowledge graph and deep learning based pest detection and identification system for fruit quality,” *Internet of Things (Netherlands)*, vol. 21, no. November 2022, 2023, doi: 10.1016/j.iot.2022.100649.

[22] J. T. Shaw *et al.*, “A case study application of machine-learning for the detection of greenhouse gas emission sources,” *Atmos Pollut Res*, vol. 13, no. 10, p. 101563, 2022, doi: 10.1016/j.apr.2022.101563.

[23] A. Subeesh *et al.*, “Deep convolutional neural network models for weed detection in polyhouse grown bell peppers,” *Artificial Intelligence in Agriculture*, vol. 6, pp. 47–54, 2022, doi: 10.1016/j.aiia.2022.01.002.

[24] R. S. Dungan *et al.*, "Growing and non-growing season nitrous oxide emissions from a manured semiarid cropland soil under irrigation," *Agric Ecosyst Environ*, vol. 348, no. January, p. 108413, 2023, doi: 10.1016/j.agee.2023.108413.

[25] R. K. Jain, "Experimental performance of smart IoT-enabled drip irrigation system using and controlled through web-based applications," *Smart Agricultural Technology*, vol. 4, no. May 2022, p. 100215, 2023, doi: 10.1016/j.atech.2023.100215.

[26] T. Miao *et al.*, "Large differences in ammonia emission factors between greenhouse and open-field systems under different practices across Chinese vegetable cultivation," *Science of the Total Environment*, vol. 852, no. June, 2022, doi: 10.1016/j.scitotenv.2022.158339.

[27] "Selected Agricultural Indicators , Malaysia , 2022 Ξ Latest Release : 25 October 2022 Previous Release Share this release Download release," no. October 2022, pp. 1–5, 2023.

[28] R. S. Ã and F. Citro, "Decarbonization : Achieving near-total energy independence and near-total elimination of greenhouse emissions with available technologies," vol. 30, pp. 1–16, 2008, doi: 10.1016/j.techsoc.2007.10.006.

[29] G. Martinović and J. Simon, "Greenhouse microclimatic environment controlled by a mobile measuring station," *NJAS - Wageningen Journal of Life Sciences*, vol. 70, pp. 61–70, Dec. 2014, doi: 10.1016/j.njas.2014.05.007.

[30] L. He *et al.*, "IoT-based urban agriculture container farm design and implementation for localized produce supply," *Comput Electron Agric*, vol. 203, Dec. 2022, doi: 10.1016/j.compag.2022.107445.

[31] A. Subeesh *et al.*, "Deep convolutional neural network models for weed detection in polyhouse grown bell peppers," *Artificial Intelligence in Agriculture*, vol. 6, pp. 47–54, Jan. 2022, doi: 10.1016/J.AIIA.2022.01.002.

[32] M. Lazaar, S. Bouadila, S. Kooli, and A. Farhat, “Conditioning of the tunnel greenhouse in the north of Tunisia using a calcium chloride hexahydrate integrated in polypropylene heat exchanger,” *Appl Therm Eng*, vol. 68, no. 1–2, pp. 62–68, 2014, doi: 10.1016/j.applthermaleng.2014.04.014.

[33] L. J. Fleckenstein, N. A. Kring, T. W. Tierney, J. C. Fisk, B. C. Lawson, and A. J. Ray, “The effects of artificial substrate and stocking density on Pacific white shrimp (Litopenaeus vannamei) performance and water quality dynamics in high tunnel-based biofloc systems,” *Aquac Eng*, vol. 90, Aug. 2020, doi: 10.1016/j.aquaeng.2020.102093.

[34] T. Ha *et al.*, “Finite element model updating of multi-span greenhouses based on ambient vibration measurements,” *Biosyst Eng*, vol. 161, pp. 145–156, Sep. 2017, doi: 10.1016/j.biosystemseng.2017.06.019.

[35] J. He and L. Qin, “Growth and photosynthetic characteristics of sweet potato (Ipomoea batatas) leaves grown under natural sunlight with supplemental LED lighting in a tropical greenhouse,” *J Plant Physiol*, vol. 252, Sep. 2020, doi: 10.1016/j.jplph.2020.153239.

APPENDICES

Appendix A: Gantt Chart Bachelor Degree Project 1 (BDP 1) (Week 1 - Week 7)

DEVELOPMENT OF REMOTE MONITORING SYSTEM USING IOT SYSTEM FOR GREENHOUSE (G-REM)

NAME: MUHAMMAD SYAHMI BIN SALZAZARY
SUPERVISOR: SITI NUR SUHAILA BINTI MIRIN

Project Start: Mon, 3/20/2023
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END	DAYS
Chapter 1: Introduction					
Study and undertand the aim of the project		100%	3/20/23	3/26/23	7
Identify the problem statement		100%	3/27/23	4/2/23	7
Identify the objective and scope of the project		100%	4/3/23	4/9/23	7
Chapter 1 progress of report		100%	4/10/23	4/16/23	7
Chapter 2: Literature Review					
Research related journal and articles about this project		100%	4/17/23	4/23/23	7
Perform comparison table with previous research works		100%	4/24/23	4/30/23	7
Chapter 2 progress of report		100%	5/1/23	5/7/23	7
Chapter 3: Methodology					
Implement methodology with a flowchart		100%	5/8/23	5/14/23	7
Research all the hardware component use in this project		100%	5/15/23	5/21/23	7
Chapter 3 progress of report		100%	5/22/23	5/28/23	7
Chapter 4: Preliminary Result					
Design prototype for this project		100%	5/29/23	6/4/23	7
Implement coding for testing		100%	6/5/23	6/11/23	7
Chapter 4 and 5 progress of report		100%	6/12/23	6/18/23	7
Preparation for Final Presentation			6/19/23	6/26/23	8

Mar 20, 2023 | Mar 27, 2023 | Apr 3, 2023 | Apr 10, 2023 | Apr 17, 2023 | Apr 24, 2023 | May 1, 2023 | May 8, 2023

Appendix B: Gantt Chart Bachelor Degree Project 1 (BDP 1) (Week 8 - Week 14)

DEVELOPMENT OF REMOTE MONITORING SYSTEM USING IOT SYSTEM FOR GREENHOUSE (G-REM)

NAME: MUHAMMAD SYAHMI BIN SALZAZARY
SUPERVISOR: SITI NUR SUHAILA BINTI MIRIN

Project Start: Mon, 3/20/2023

Display Week: 8

TASK	ASSIGNED TO	PROGRESS	START	END	DAYS
Chapter 1: Introduction					
Study and undertand the aim of the project		100%	3/20/23	3/26/23	7
Identify the problem statement		100%	3/27/23	4/2/23	7
Identify the objective and scope of the project		100%	4/3/23	4/9/23	7
Chapter 1 progress of report		100%	4/10/23	4/16/23	7
Chapter 2: Literature Review					
Research related journal and articles about this project		100%	4/17/23	4/23/23	7
Perform comparison table with previous research works		100%	4/24/23	4/30/23	7
Chapter 2 progress of report		100%	5/1/23	5/7/23	7
Chapter 3: Methodology					
Implement methodology with a flowchart		100%	5/8/23	5/14/23	7
Research all the hardware component use in this project		100%	5/15/23	5/21/23	7
Chapter 3 progress of report		100%	5/22/23	5/28/23	7
Chapter 4: Preliminary Result					
Design prototype for this project		100%	5/29/23	6/4/23	7
Implement coding for testing		100%	6/5/23	6/11/23	7
Chapter 4 and 5 progress of report		100%	6/12/23	6/18/23	7
Preparation for Final Presentation			6/19/23	6/26/23	8

Week	M	T	W	T	F	S	S
May 8, 2023	8	9	10	11	12	13	14
May 15, 2023	15	16	17	18	19	#	#
May 22, 2023	22	#	24	#	#	#	#
May 29, 2023	29	30	31	1	2	3	4
Jun 5, 2023	5	6	7	8	9	10	11
Jun 12, 2023	12	13	14	15	16	17	18
Jun 19, 2023	19	#	21	#	#	#	#
Jun 26, 2023	26	#	28	29	30	1	2

Appendix C: Gantt Chart Bachelor Degree Project 2 (BDP 2) (Week 1 - Week 7)

DEVELOPMENT OF REMOTE MONITORING SYSTEM USING IOT SYSTEM FOR GREENHOUSE (G-REM)

NAME: MUHAMMAD SYAHMI BIN SALZAZARY
SUPERVISOR: MOHAMAD HANIFF BIN HARUN

Project Start: Mon, 10/9/2023

Display Week: 1

Oct 9, 2023 | Oct 16, 2023 | Oct 23, 2023 | Oct 30, 2023 | Nov 6, 2023 | Nov 13, 2023 | Nov 20, 2023 | Nov 27, 2023

TASK	ASSIGNED TO	PROGRESS	START	END	DAYS
PSM 2 Briefing		100%	10/9/23	10/15/23	7
Presentation BDP 1 to new supervisor		100%	10/16/23	10/22/23	7
Supervisor discussion		100%	10/23/23	10/29/23	7
Compenent purchases and material preparation		100%	10/30/23	11/5/23	7
Software implementation		100%	11/6/23	11/12/23	7
Consultation of software development		100%	11/13/23	11/19/23	7
Programming research and troubleshooting		100%	11/20/23	11/26/23	7
Implement coding for testing		100%	11/27/23	12/3/23	7
Simulation process for this project		100%	12/4/23	12/10/23	7
Finalizing prototype of simulation		100%	12/11/23	12/17/23	7
Chapter 4 and chapter 5		100%	12/18/23	12/24/23	7
Supervisor verification and final update		100%	12/25/23	12/31/23	7
Final report submission		100%	1/1/24	1/7/24	7
Preparation for Final Presentation			1/8/24	1/15/24	8

Appendix D: Gantt Chart Bachelor Degree Project 2 (BDP 2) (Week 8 - Week 14)

DEVELOPMENT OF REMOTE MONITORING SYSTEM USING IOT SYSTEM FOR GREENHOUSE (G-REM)

NAME: MUHAMMAD SYAHMI BIN SALZAZARY
SUPERVISOR: MOHAMAD HANIFF BIN HARUN

Project Start: Mon, 10/9/2023
Display Week: 8

TASK	ASSIGNED TO	PROGRESS	START	END	DAYS
PSM 2 Briefing		100%	10/9/23	10/15/23	7
Presentation BDP 1 to new supervisor		100%	10/16/23	10/22/23	7
Supervisor discussion		100%	10/23/23	10/29/23	7
Compenent purchases and material preparation		100%	10/30/23	11/5/23	7
Software implementation		100%	11/6/23	11/12/23	7
Consultation of software development		100%	11/13/23	11/19/23	7
Programming research and troubleshooting		100%	11/20/23	11/26/23	7
Implement coding for testing		100%	11/27/23	12/3/23	7
Simulation process for this project		100%	12/4/23	12/10/23	7
Finalizing prototype of simulation		100%	12/11/23	12/17/23	7
Chapter 4 and chapter 5		100%	12/18/23	12/24/23	7
Supervisor verification and final update		100%	12/25/23	12/31/23	7
Final report submission		100%	1/1/24	1/7/24	7
Preparation for Final Presentation			1/8/24	1/15/24	8