The University of Jordan

School of Engineering
Department of Computer Engineering

IoT-based Green House Monitoring and Control System

Supervisor:
Dr. Samah Rahammneh

Author(s):
Yazan M. Hussein 0161943
Adnan El Hammoudi 0167968

May 27th, 2021

Submitted in partial fulfillment of the requirements of B.Sc. Degree in Computer Engineering

This page is intentionally left blank

ETHICAL STATEMENT

We, the undersigned students, certify and confirm that the work submitted in this project report is entirely our own and has not been copied from any other source. Any material that has been used from other sources has been properly cited and acknowledged in the report.

We are fully aware that any copying or improper citation of references / sources used in this report will be considered plagiarism, which is a clear violation of the Code of Ethics of the University of Jordan.

We further certify and confirm that we had no external help without the approval of our supervisor and proper acknowledgment when it is due. We certify and affirm that we never at any point commissioned a 3rd. party to do the work or any part of it on our behalf regardless of the amount charged or lack thereof. We also acknowledge that if suspected and thereafter proven that we commissioned a 3rd. party to do any part of this project that we risk failing the entire project.

We certify and confirm that all results presented in this project are true with no manipulation of data or fraud, that any statistics done, or surveys collected are conducted with the highest degree of scientific fidelity and integrity, and that if proven otherwise, we risk failing the entire project. We acknowledge that for any data collected, we have taken all the steps necessary in applying for proper authorizations if deemed necessary, and that all user data collected is subject to the utmost degrees of privacy and anonymity.

Adnan El Hammoudi	Yazan Hussein
6/16/2021	6/16/2021
	yazan mohammad

This page is intentionally left blank

SUPERVISOR CERTIFICATION

I hereby certify that the students in this project have successfully finished their senior year project and by submitting this report they have fulfilled in partial the requirements of B.Sc. Degree in Computer Engineering. ☐

I hereby certify that the students in this project have not completed their senior year graduation project and I do not approve that they proceed to the discussion. ☐

I suspect that the students have violated one or more of the clauses in the ethical statement and I suggest that an investigation committee look into the matter. ☐

Prof. / Dr. / Eng.

Signature:

This page is intentionally left blank

DEDICATION

This project is dedicated to our parents and families.

SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ACL	Access Control List
AWS	Amazon Web Services
CV	Cross-Validation
GLCM	Gray Level Co-occurrence Matrix
GUI	Graphical User Interface
IoT	Internet of Things
RPI	Raspberry Pi
SVM	Support Vector Machine

ABSTRACT

In Jordan, 3.5% of the population is in the agriculture sector, and with the small percentage of the populous, and to meet the increased need for food, a solution must be found, not to mention that a lot of sectors are inseparable from the agriculture field. From agriculture, we get various raw materials that enter other industrial fields. However, crops and various other plantations get destroyed due to different reasons such as natural phenomena like floods, famine, earthquake, drought, in addition to, infectious agents or pathogens.

Using the Internet of Things (IoT) benefits farmers by automation and control of the environment inside the greenhouse. Greenhouses are affected by various factors such as weather, plant diseases. These factors affect the yield negatively.

Several computer or microcontroller designs have been purposed for controlling and monitoring the greenhouse. With recent development in embedded systems, ease of access to the web and cloud systems newer and better solutions have been developed. These systems require a long period of experimentation since they change with climate, location, and plants being grown. Taking these factors into consideration, predicting the actual price for the project may be a challenge.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page is intentionally left blank

CHAPTER 1
INTRODUCTION

1.1 Problem Definition

Automation and monitoring in a greenhouse are some of the most important subjects in the agriculture sector, many researchers have focused their efforts on this topic especially with the current decline of interest in agriculture amongst the youth and the interest in other fields.

Some of the benefits of implementing automation in greenhouses are but not limited to:

• Maintaining the ideal environment inside a greenhouse: heat, humidity, light levels, and ventilation can be a real challenge to manage. However, with the help of automation we can control these factors and help ease the burden and challenges farmers face having manage them, we integrate automatic controls in our greenhouse system to keep the environment attuned to the requirements of the corps without the need for constant adjustments and monitoring of the system, it also helps in building a consistent and predictable growing cycle.

• Reduce energy costs: The biggest benefit of automating greenhouses is their ability to reduce the overall costs of production. Automated systems do not require production costs but overall, it is worth the investment, with technology improving and a variety of options being available, automation in greenhouses will ensure that the system is working as a cohesive unit which will save a lot in the overall production cost.

• Issues that may arise from location: an automated system can help farmers in locations where its climate is the main challenge for the growth of the crops. For example, farmers in the floriculture industry cannot be restricted or miss deadlines because the year’s spring has been colder than usual, or the light levels are lower than expected. An automated system will ensure crop growth on the farmer’s own terms regardless of what the weather situation is.

• Customization and modularity: a greenhouse can be tuned according to the farmer's preference and within any budget. Several options are available, and the level of automation can be improved upon constantly on demand. In agriculture, a system that is used in the optimization of crop growth is called protective cultivation, in which the soil and the climate ecosystems are controlled with hardware, the goal of protective cultivation is for plants do not have to grow in their home environment, the effects of protective cultivation can be summarized as [1]:

- Faster plant growth.
- The shorter span in harvest periods.
- Increase in plant lifespan.
- Improvement in the quality of yield.
- Sustainability in production.

Protected cultivation is practiced often in crop cultivation techniques, while the environment is partially or fully manipulated depending on the growth period to maximize the yield and reduce energy costs.

The most widely used protected cultivation method to achieve its goals is greenhouses, which can supply the preferred environment for optimum plant growth and yield [2].

Greenhouse technology has been used in multiple countries and there are an estimated 9 million acres of greenhouses worldwide [3] , possess multiple advantages that can enhance the agriculture experience [4]:

- Ability to plant regardless of the season throughout the entire year.
- Protect the plant against disastrous seasons like rains, storms, or frost.
- Increase in the yield.
- Reduce fertilizer waste.
- Pest and disease control
- Optimal for tissue culture plants
- Greenhouses cage carbon dioxide which leads to better photosynthesis
- Promotion of plant growth and ripening
- Vapor reduction
- Exclude UV rays that can be harmful to the plant.

The rising interest in greenhouses can be due to the increase of urbanization and lack of fertile land thanks to land degradation, usually greenhouses are made with plastic, glass or fiberglass, these materials are commonly used because of their ability to retain heat and transmit light, which results in increase of temperature inside the greenhouse [5].

According to a 2020 World Bank study [6] Fig.1. below shows the interest in agriculture have been dwindling, in 1991 44% of the world population was working in the agriculture sector, as of 2020 only 26% of the population do, this can be attributed to the increase of migration from the countryside to the cities, according to a UN World Urbanization study in 2017, 4.1 billion people were living in urban areas against 3.5 billion living in the countryside [7], and the increased interest in IT fields of studying over agriculture, to counter this we need to integrate IT into this sector, with the help of the digital revolution we can help fill the gap in the agriculture industry, a computer can easily copy mundane and repetitive tasks like watering, monitoring, cooling, warming, and curing plants. Creating an optimal environment for plant growth and maximize the yield requires an effective way to regulate temperature, humidity, airflow, nutrients, and carbon dioxide [8].

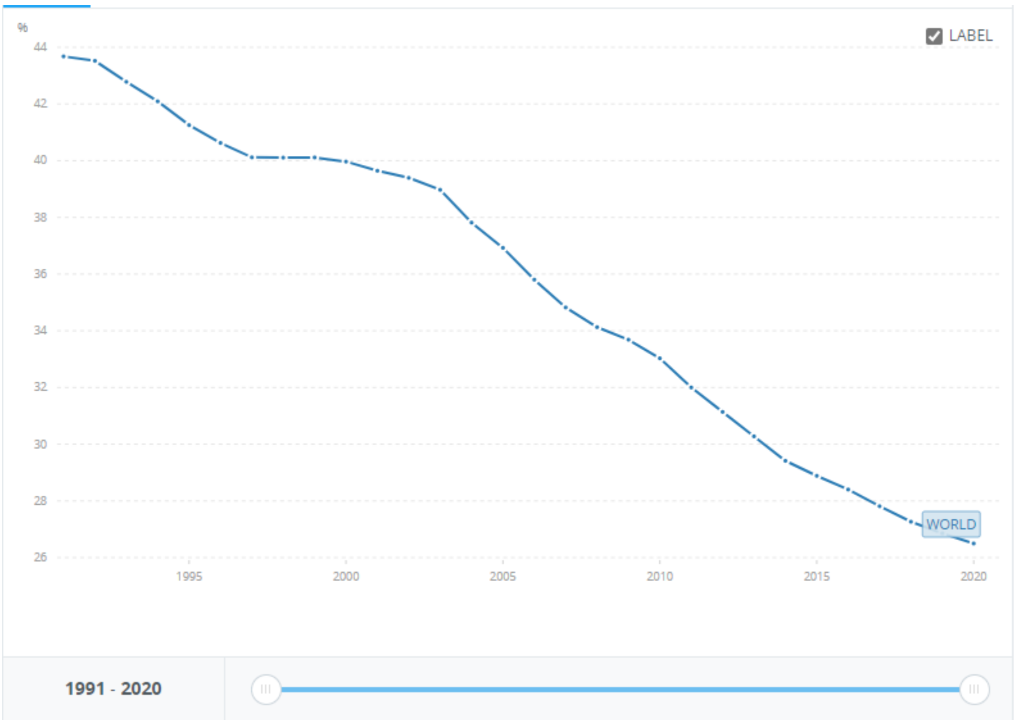

Figure 1 - Employment in agriculture (% of total employment) [6].

1.1.1 Monitoring and Automation

Monitoring means collecting data and processing it to produce information, verification of data can be achieved through analysis and stored in data storage then arranged so they can be useful later [9].

Quality can decrease if monitoring is not implemented in a system. System monitoring ensures efficiency, and problems can be identified early, and produce accurate data for decision making. In this project, we will monitor climate parameters inside the greenhouse, collect and analyse the data. Determining variables and comparing them to previously measured variables is called process control, with this, we create a system that can control these variables. Programmable logic controllers (PLCs) are control devices that have microprocessors and work with internal memory. PLCs are used in industrial applications to control machines and tools, since our project is small-scaled, we can use simpler PLCs like Arduino or RPI.

Applications of technology that produce and deliver goods/services with minimal human interactions, implementing automation technology improve efficiency [10], reliability, and speed of many tasks that would have been previously performed by humans. Automation usually is used to minimalize labor or completely substitute human interactions. These days, automation is used everywhere from the simplest to the more complex applications. Automation makes sure that techniques are used efficiently in the delivery of products and services. However, automation has its negatives, especially on employment and wages for some occupations using it. According to a 2019 World Bank's World Development Report [11], the positive economic effects in terms of new industries and jobs that are available outweigh the negatives [11].

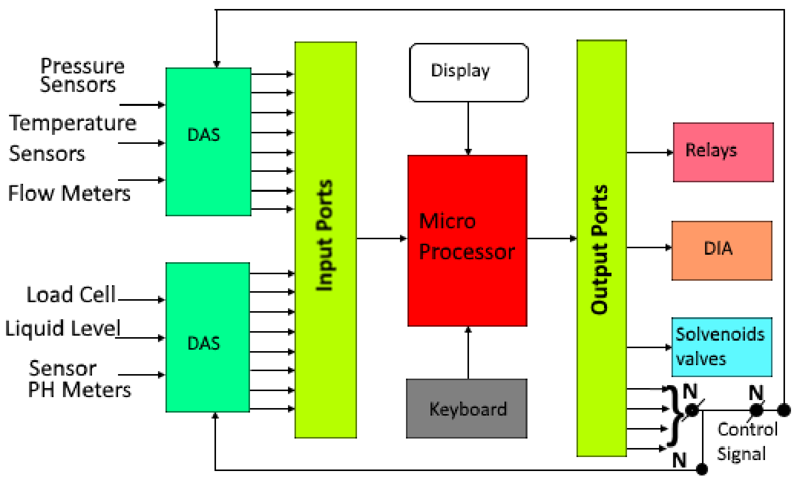

Figure 2- PLC Based Process Control[10]

1.1.2 Embedded Systems

An embedded system can consist of a single microcontroller to a full suite of micro-processors with peripherals and networks connections. It can have a Graphics User Interface (GUI) or a simple UNIX like command prompt terminal. All these factors depend on the application the system is being used for [12].

Embedded systems are managed usually by a microcontroller, DSP or even a micro-processor, these processing systems main job is to handle electric interfaces.

Fig.3. below shows the structure of an embedded system is basic and includes the following components [12]:

- sensors: used to measure process variables and are usually placed at the input port, and produce electrical signals, examples of them are magnetometer, cameras, and microphones.
- A\D converter: converts the analog signal sent from the sensor module into a digital signal.
- Processor\ASICs: process the data and produce the output to be stored in memory.
- D\A converter: converts digital signal sent from the processor to an analog signal.
- Actuator: impels process parameters and is usually connected to an output port and produce motion or heat examples of them are LED, speakers, motors

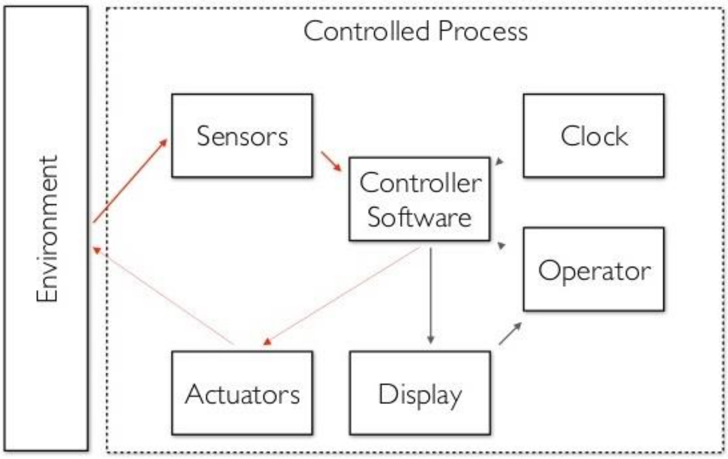

Figure 3- components of an embedded system.[12]

By using embedded systems, we can focus on the design flow while looking for the right iteration to improve performance, as of today there two types of embedded systems that are used in small scale projects and hobby activities, the Arduino UNO, and RPI. These two modules are extremely popular due to their simplicity and cheap price compared to their powerful function [13].

To measure variables in the environment we need sensors, by using sensors a microcontroller can detect what is happening in the surrounding environment and take proper action. In our application we will use the following sensors and modules:

1) Temperature and Humidity Sensor: Several sensors can detect temperature and humidity, i.e., DHT11 and DHT22. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air. the output is a digital signal on the data pin, The DHT11 is a basic to detect temperature and humidity, that this sensor has anti-interference characteristics and an elevated level of accuracy. Whereas the DHT22 sensor is less accurate but has a wider range of temperature/humidity degrees [14].
2) Gas detection sensor: the most used gas sensor is MQ2. It is a Metal Oxide Semiconductor, is known as Chemoreceptors, the recognition is based on the change of resistance of the sensing materials depending on the nature of the gas. It uses a simple integral potentiometer network; you can adapt the sensor sensitivity according to how accurately you want to detect gas[15].
3) LDR Light Sensor: it is indicating the intensity of light that converts this “light energy” into an electrical signal output whether visible or infra-red of the spectrum Light. It is more commonly known as “Photoelectric Devices” because it is converted light energy into electricity [16].
4) Servo motors: it works on the Pulse Width Modulation (PWM) principle, which means it is controlled by the period of the pulse applied to its control PIN to make an angle rotation. The servo motor consists of a DC motor. The servo will close and open the window of the greenhouse whenever the environment requires so, in rainy. On stormy days, the servo will close the window of the greenhouse to avoid flooding or damaging the plants inside the greenhouse. if there is gas or smoke detected or on hot days the servo will open the window to cool the house and reduce the saturation of harmful gases [17].

1.1.3 Cloud Computing

Cloud computing is the delivery of computing service over the "cloud" without the use of any management by the user. Services that are offered by the cloud are stored and executed on a physical server called a "data center." These data centers are maintained by a provider [18].

cloud computing is a big shift from the traditional way businesses think about IT resources. Some of the reason's organizations are turning to cloud computing [19]:

- cost: cloud resources can be purchased and used on a "pay-as-you-go", this eliminates the expenses of buying hardware and software, the cost of electricity, and the IT support for managing the infrastructure.
- Performance: cloud resources are easily upgradable to the latest generation of fast and efficient computing hardware.
- Productivity: end-user productivity will be enhanced because no software is installed, configured or upgradable on-land and services can be accessed anywhere.
- Reliability: cloud computing helps in data backup, recovery from disaster, and business continuity will be much easier.
- Security: cloud providers offer a set of policies and technology that help in protecting their user's data, apps, and infrastructure from potential threats.

Since we need a cloud platform to host a storage server, we will be using a Software as a Service (SaaS) model to be implemented in our project.

Table -1 below shows many big companies that offer cloud services, amazon, google, and Microsoft. All of them provide on-demand cloud computing services and APIs with "pay-as-you-go" pricing, that is you will only pay for the services that you will be using. All this includes computing power, storage, networking, database, deployment, management, mobile, dev. Tools, and finally tools that help in IoT development, we are going to discuss some of the top providers of cloud services in brief.

Table 1 - Top cloud providing platforms. [20]

	Amazon Web Services (AWS)	Microsoft Azure	Google Cloud
Launch Date	March 14'th, 2006	February 1'st, 2010	April 7'th, 2008
Number of Available Services as of 2020	283	261	115
Availability Zones	77	140	73
Market Share as of the end of 2020	33%	18%	9%
Website link	Aws.amazon.com	Azure.microsoft.com	Cloud.google.com

In our project we will use amazon web services as the cloud provider and since they offer free trial for students to use their platform as well as its advantage over other cloud providers as seen in Table -1.

1.2 Proposed Solution

In this project, we deliver an autonomous hardware structure that gives assistance to farmers in managing their crops. It makes the agriculture procedure easier and helps face the environmental challenges the world has faced. The greenhouse helps in keeping the temperature and the climate for optimal plant growth inside the house constant as much as possible also helps in early detection of diseases inside the greenhouse.

The project utilizes embedded systems, cloud computing and image recognition. The embedded system consists of RPI and Arduino with sensor modules attached to them to collect the data analyze them and make appropriate actions, then sends them over to the cloud platform through the RPI. The cloud will host a storage server that stores all the data collected from the greenhouse. Using image recognition, we can know what is happening inside the greenhouse most importantly what diseases are affecting the plants inside the greenhouse, constantly opening the door to check on the plants is a problem and can ruin the temperature inside the greenhouse so entering the greenhouse so opening and closing the windows inside the greenhouse is left for the microcontroller to decide.

1.3 Project Impact

Climate change and the rapid increase in population and dwindling resources of the world have all put pressure on the agriculture industry. The unpredictability in weather has motivated farmers to seek modern technologies that improve production and decrease cost. These days IoT can help farmers face these challenges. IoT helps as listed below [21]:

1. Maintaining micro-climate conditions: using sensors can allow farmers to collect data accurately and deliver them in real-time, temperature, humidity, light intensity, carbon dioxide levels, *etc*. these data are critical and prompts adjustments in Heating, ventilation, air conditioning, and Lighting settings to maintain optimal conditions for plant growth.
2. irrigation and fertilization enhancement: precision farming help farmers stay updated on the status of their crops, this helps to keep irrigation and fertilization on par with what the plant needs, too much fertilizer or water are as bad as too much.
3. Disease and infection control and containment: biggest challenge facing farmers is crop disease, crop margins are affected heavily with every outbreak, while chemical treatments are available applying them in the wrong time can be catastrophic, applying them too much raise safety, ecological, and cost concerns with the help of machine learning data in greenhouse environment, external weather and soil characteristics reveal existing risks of fungi or pests, with this information at hand farmers can treat the crop exactly when it needs treatment to ensure health and quality without too much chemical exposure. [22]

CHAPTER 2
RELATED WORK

Many studies have been made in this field. The main difference is the technology utilized, the protocols used in the IoT network, and the control method. Before the notion of cloud, the most popular method was using GSM networks. All modern systems use the RPI or an Arduino as the base for the system's controller.

2.1 Greenhouse Monitoring and Control System Based on Wireless Sensor Network.

Using wireless sensor networks the authors of [23] monitored and controlled the environment inside the greenhouse using temperature, humidity sensors. The temperature sensor monitors the temperature, if it goes above the optimal range of preset values the system will turn on a fan until the temperature reaches stability. regarding humidity, a spray device is activated if the humidity goes above the preset range. The sensor used to monitor temperature and humidity is DHT11 and SEN92355P respectively if the soil is dry, the system turns on the irrigation system. A gas FC-22-1 sensor is used to detect concentrations of CO2 in the greenhouse. In the case of a high percentage, the system turns on a ventilator to purify the air. A lighting system is attached in the case of poor lighting hours in the day. All the preset ranges for temperature, humidity, moisture is defined by the user and depend on the plant grown and climate requirements, the sensor node will read the different greenhouse parameters and control the device with temperature, humidity, Gas, and water sensors attached to it. The node communicates with the gateway via the Freakduino board using Chibi Wi-Fi. The node also transmits the acquired data via GSM module to the gateway in case of remote or long-distance monitoring, Local communication network used in the paper is Wi-Fi, it is implemented for the communication between the sensor node and the gateway in short distance measurements in the wireless monitoring system network, for Global communication GSM technology is used to transmit the acquisition data in a wide geographical area scenario, [23] is similar in the way that it implements automation using a microcontroller and a group of sensor modules working together to sustain a growth environment inside the greenhouse, in [23] they used GSM which we won't be using, GSM is relatively expensive to be used since you will have to count the expenses paid to the providers, using cloud infrastructure is much more efficient and cheaper, wireless sensors were not used in our project so the need for a Wi-Fi module and a gateway for the sensors will not be needed, another difference is that the gateway of the Arduino will be the RPI connected to it, the RPI will take the data from the Arduino and send them to the cloud so it can be displayed to the user.

2.2 Smart Indoor Vertical Farming Monitoring Using IoT.

In [24] the authors proposed a vertical farming method rather than a greenhouse; the sensors are used to collect desired data depending on the current situation. The sensors are connected to an analog input (AIN) of the BeagleBone Black (BBB) microcontroller, all the data gathered will be sent directly to the microcontroller. actions will be taken automatically once the input value from sensors is out of the preset acceptable range, i.e., the watering system will activate if the soil moisture level is low. the Wireless Module is coordinated with the microcontroller.

To provide 4G network coverage through GSM services. The SIM card module connected to the wireless module contains a SIM card. Global Positioning System (GPS) data will be retrieved from the Global Navigation Satellite System (GNSS) receiver inside the wireless module to locate the equipment in use, the microcontroller which is used as a central processing unit (CPU) to manage data received from the sensors via analog input (AIN) pin, these data will be uploaded directly to the cloud with network access supported by the Wireless Module. the watering system has been programmed to activate when the soil moisture level is below the specified preset level. The authors used the ThingSpeak IoT platform to store information and develop a web-based application. The application will be able to analyze the statistics in the cloud and visualize it in the form of graphs, charts, or tables for quality control. Even though the system is programmed to react automatically, it can be controlled by the user as well. Hence, there is a connection between the web-based application and the microcontroller. In [24] the authors used a cloud-based web interface that will display the data collected from the structure we will be taking values directly from the user but won’t be using a web interface, we see that a graphical user application that is installed directly into the user’s machine is more efficient since it cuts off the middle man that is the cloud provider also cloud-based applications are harder to debug and maintain a graphical interface created with Tkinter python library is easier to maintain and more efficient, we will be using an embedded system of sensors and modules that will collect data, in our project we will be using an RPI as a central processing unit that connects to the internet as well as not using any 4G GSM module or service, we won't be using any cultivation system because automated cultivation is too expensive to be implemented efficiently,Fig.4. shows the authors proposed system.

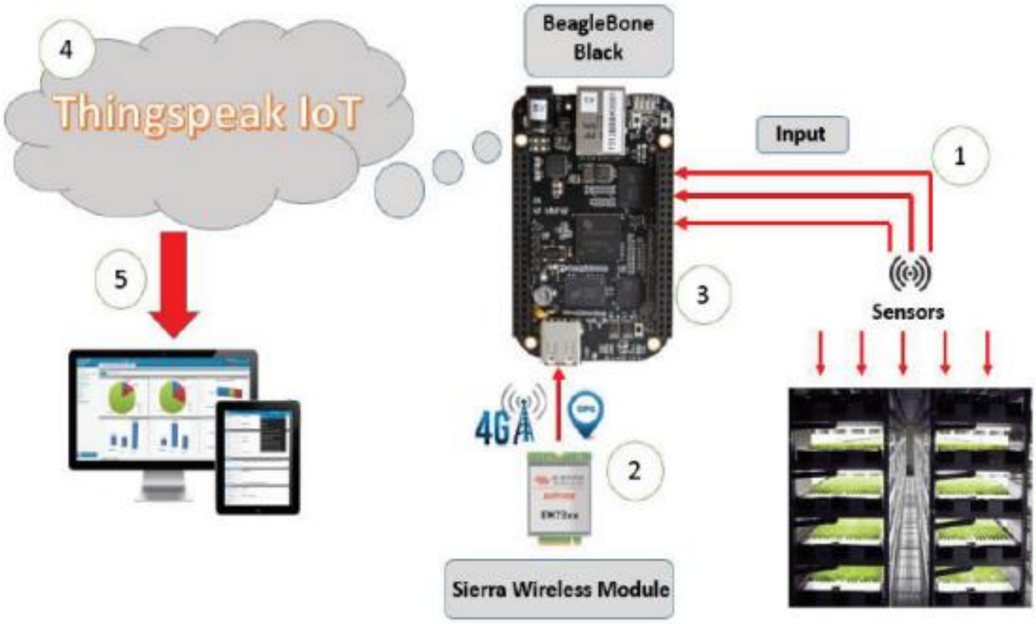

Figure 4- System design for [24] (Yap S. Chin)

2.3 Implementation of IoT With Image Processing in Greenhouse Monitoring System.

In [25] the authors proposed a system that detects plant diseases using image recognition and automates the greenhouse process, the authors used a RPI 3B+ as a controller to control the greenhouse and the actuator based on the input data gathered from the environment using various sensors and camera. The RPI can gather and process data and execute actions, the SP8266 is used to collect data from sensors, it is used to collect data from sensors and automate the greenhouse with the RPI. The 5MP RPI camera is used for capturing leaf images within a given period. It can capture up to (2592 x 1944) pixel images and 1080p30 video. A dedicated CSi interface, designed specifically for interacting with cameras, is implemented to interface it with the RPI. Other modules used in this project are Soil Humidity Sensor, Water Sensor, DHT11 Temperature, Humidity Sensor, the sensors are interconnected through WSN (Wireless Sensor Networks), they interact with a processing unit and the local network so that the data can be processed, analyze and stored, the sensors are connected to SP8266 NODE MCU board assess the condition of the greenhouse environment and sends information to the RPI which makes the decision by processing and comparing the data obtained from the sensors against the preset values. The controller then controls the actuator to influence the greenhouse environment e.g., a solid-state relay is used to turns on the water pump when the soil is dry or off when it is not. Similarly, the PI camera is connected to RPI to take pictures of leaves in the greenhouse for further disease identification and classification process. The values, which are acquired from sensors and cameras, are stored in the cloud and computer. Farmers or greenhouse workers can see proper real-time results from the cloud, our project implementation will be similar to [25] we will be using image recognition to detect plant disease using the MATLAB programming language, we will be using a cloud structure to store the data and a microcontroller to collect the data and analyze it the controller will be serially interfaced with the RPI however and won't be communicating wirelessly, since implementing a wireless interface is complicated, expensive and not worth it since it gives no advantages in the project, also the image recognition code will run on the user's computer and not on the RPI as running a MATLAB code on windows is better and faster, in our project we used an Arduino UNO and an off the shelf webcam instead of the RPI camera. Fig.5. shows the authors proposed system.

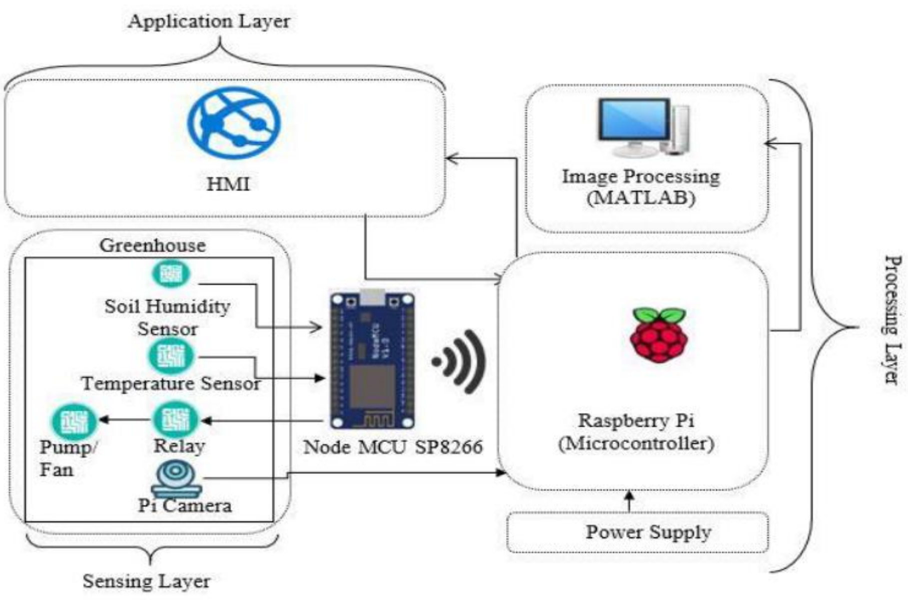

Figure 5 - system design for [25] (Tariku B. Wudneh)

CHAPTER 3
SOLUTION DESCRIPTION AND IMPLEMENTATION

The project is comprised of four main parts, first the Arduino system that is made of sensors and modules , that collect the data and automate the greenhouse climate, secondly the RPI that is connected to the Arduino and will collect the data and sends it to the cloud, also a camera is connected to the RPI , that will take a snapshot of the plant for disease detection, the cloud is to store all the output files from the Arduino sensors and RPI camera, finally, the end-user python interface to display all the data, We combine both the RPI and Arduino to have the best of both worlds, we can send the values from Arduino sensors and send them to the RPI to processed, saved and uploaded to the cloud to be viewed by the end-user using a executable python program.

3.1 Requirement Analysis

We can categorize system requirements like functional, software and hardware, system, and database requirements.

3.1.1 Functional Requirements

The system must capture data from the greenhouse environment , then send it to the cloud server to be displayed to the user on a python graphical user interface (GUI), the python GUI interface should also provide the Arduino system with the threshold value that are provided from the user, the python GUI interface should also notify the user in cases of emergenc, in cases of heavy rain the greenhouse should close all the windows and send a notification to the python GUI interface, the image recognition system should also identify cases of disease and display the diagnoses to the user.

3.1.2 Hardware/Software Technical Requirements

The system will consist of three main parts, the Arduino for the embedded system, the RPI for the image processing and web communication, and finally the cloud infrastructure to host the storage, the web storage will be hosted by amazon web services that provide free trial cloud platform for students, certain programming languages are needed to develop the project, the Arduino is programmed in a language that is similar to that of C, the Arduino and the RPI will communicate with each other with a python program that uses USB bus serial communication, also python is essential in the project as the serial communication between the RPI and Arduino and the upload and download of sensor values and is done by python. As for image detection we will need MATLAB environment and access to Statistics and Machine Learning Toolbox and Image processing toolbox.

3.1.3 System and User Interface

The main interaction with the system from the user will be using the python GUI interface, limiting human interaction inside the greenhouse is something to be desired as continuously opening will ruin the atmosphere inside, for the raspberry to send information to the cloud it will use a special python library called boto3.

3.1.4 Database Requirements

The Arduino records values from sensors and different warning messages every 5 minutes and sends them over to the RPI ,then the RPI will send different messages to the cloud, also any change in the greenhouse setting done in the python GUI interface should be sent periodically.

3.2 Design

The Arduino hardware section of the project is made from 2 parts; sensors and actuators are responsible for regulation inside the greenhouse, we collect data that will be compared to threshold data, and if any value falls below the threshold range, the appropriate actuator will be turned on to preserve the greenhouse's desired condition, the Arduino board will also read the sensor values and actuator actions and sends them to the RPI, and then the RPI will upload it to the cloud.

after collecting data, it should be sent to the RPI, this should be done using serial communication using UART meaning it has to be sent sequentially one bit at a time, UART protocol is an asynchronous multi-master protocol; meaning that both devices can send data whenever they want unlike slave-master protocols, Arduino Uno boards have one UART that can be used either with a USB cable or using RX/TX pins, on the RPI we can have as many serial devices connected on the USB ports or using GPIOs.

In this project we will be using a simple USB cable connected between both boards considering it is the easiest and least complicated way. on the RPI we will run a python program to read from the Arduino uno sensors and write them to the cloud platform using a data exchange interface.

after capturing sensor data, it should be uploaded to be displayed on the python GUI interface that, all the data will be added to a file and the file will be uploaded to what Amazon calls a “bucket”, all data must be assigned to a bucket and each bucket has a unique name and key and can be addressed by http://s3.amazonaws.com/{bucket}/{key} for the sake of simplicity, we will not assign access control list (ACL) or encryption to the files, a useful library to interface AWS is boto3 which works for S3, EC2, SES, SQS [26], Fig.6. shows how the entire system will interact with each other.

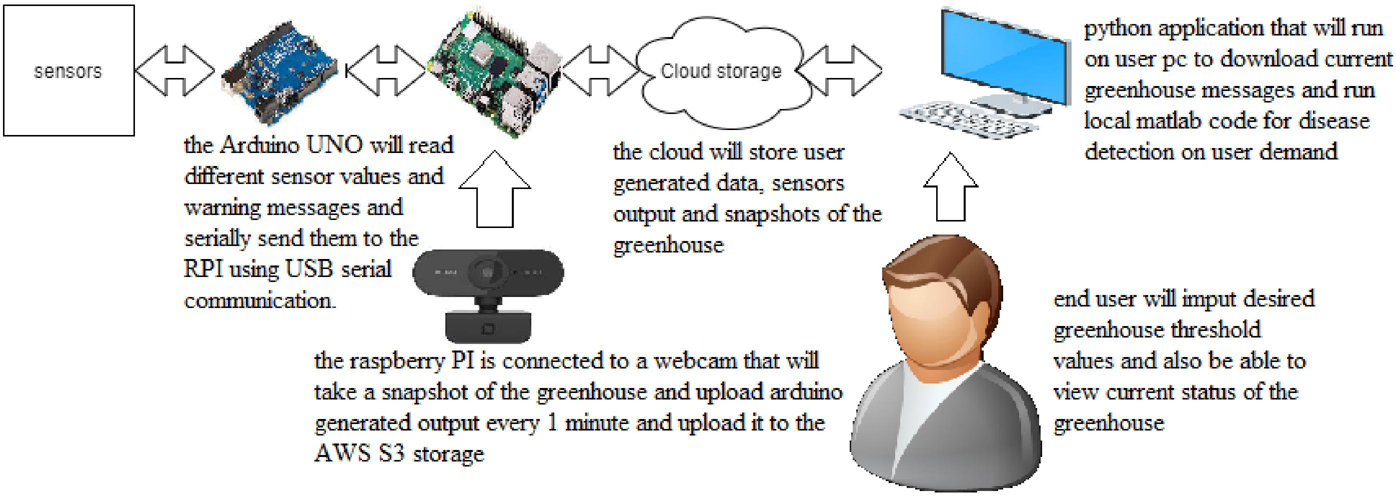

Figure 6 - greenhouse system design

3.2.1 Arduino UNO Hardware

Arduino UNO is a micro controller board that is perfect for hobbyists and small projects, it is cheap and easily available, we can attach different sensors to collect different data, Arduino is also open source based on the Microchip ATmega328P with I/O pins to attach different modules to, there exists 14 digital I/O pins and 6 analog I/O pins, the D2-gas, rain detection and LDR are connected to the analog input pins since they analyze analog data, while the DHT 11-humidity, lighting system and servo motors are connected to the digital output pins, Fig.7. and Fig.8. shows how the sensors are connected to the Arduino.

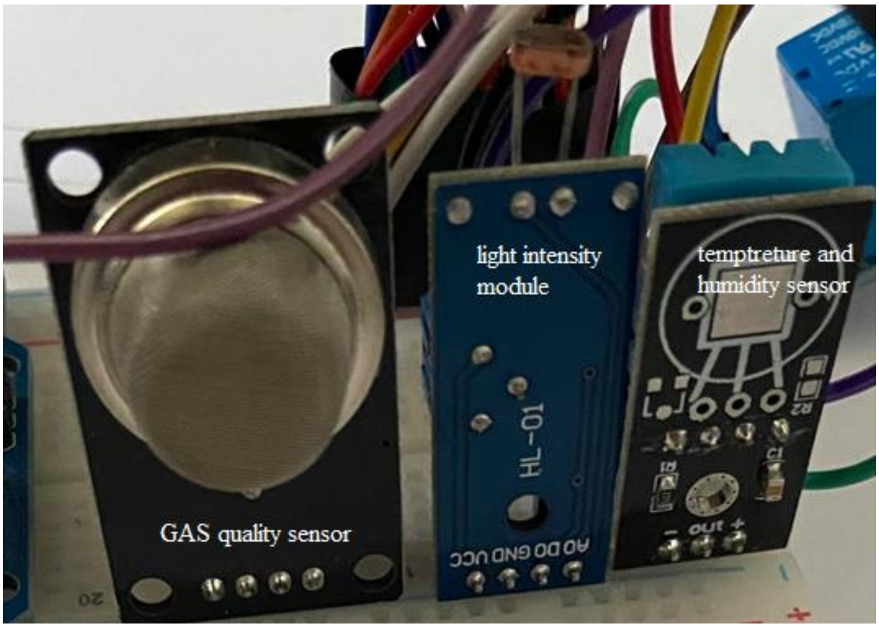

Figure 7- three main sensor modules

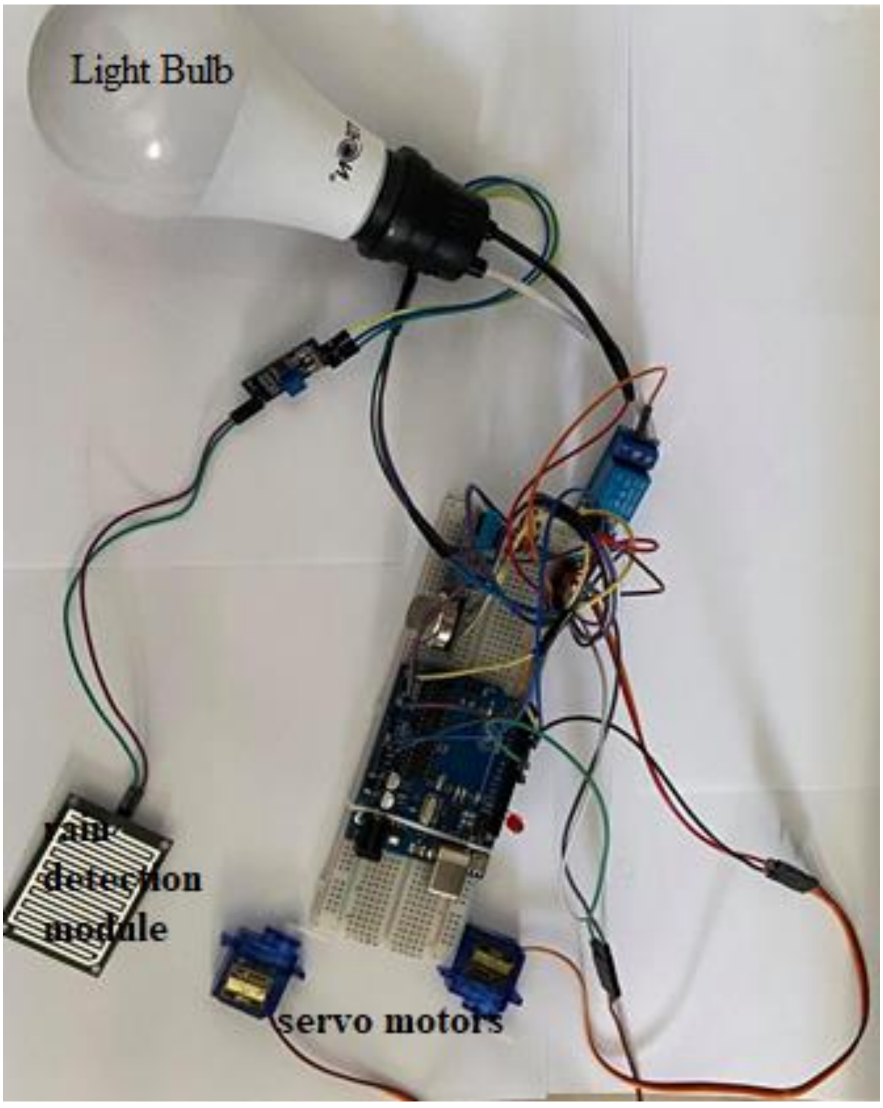

Figure 8- full Arduino circuit

3.2.2 Arduino UNO Software

Arduino software is written in language similar to C, the code in this project runs in a loop, the Arduino software will block until threshold values are provided, when the system starts it will still hold values from past runs until user provides values, when user provides values through a python application it will be read and processed, and all the current environment values will be sent serially, initially the windows are closed, if gas levels, humidity or temperature values are above threshold values the windows will open, if rain is detected the windows will lock to prevent flooding or gas levels, humidity or temperature, if it goes for the normal levels the windows will open again, finally at the end the Arduino will serially write all the sensor values and warning messages serially to the RPI, Fig.9. shows how the Arduino code will behave.

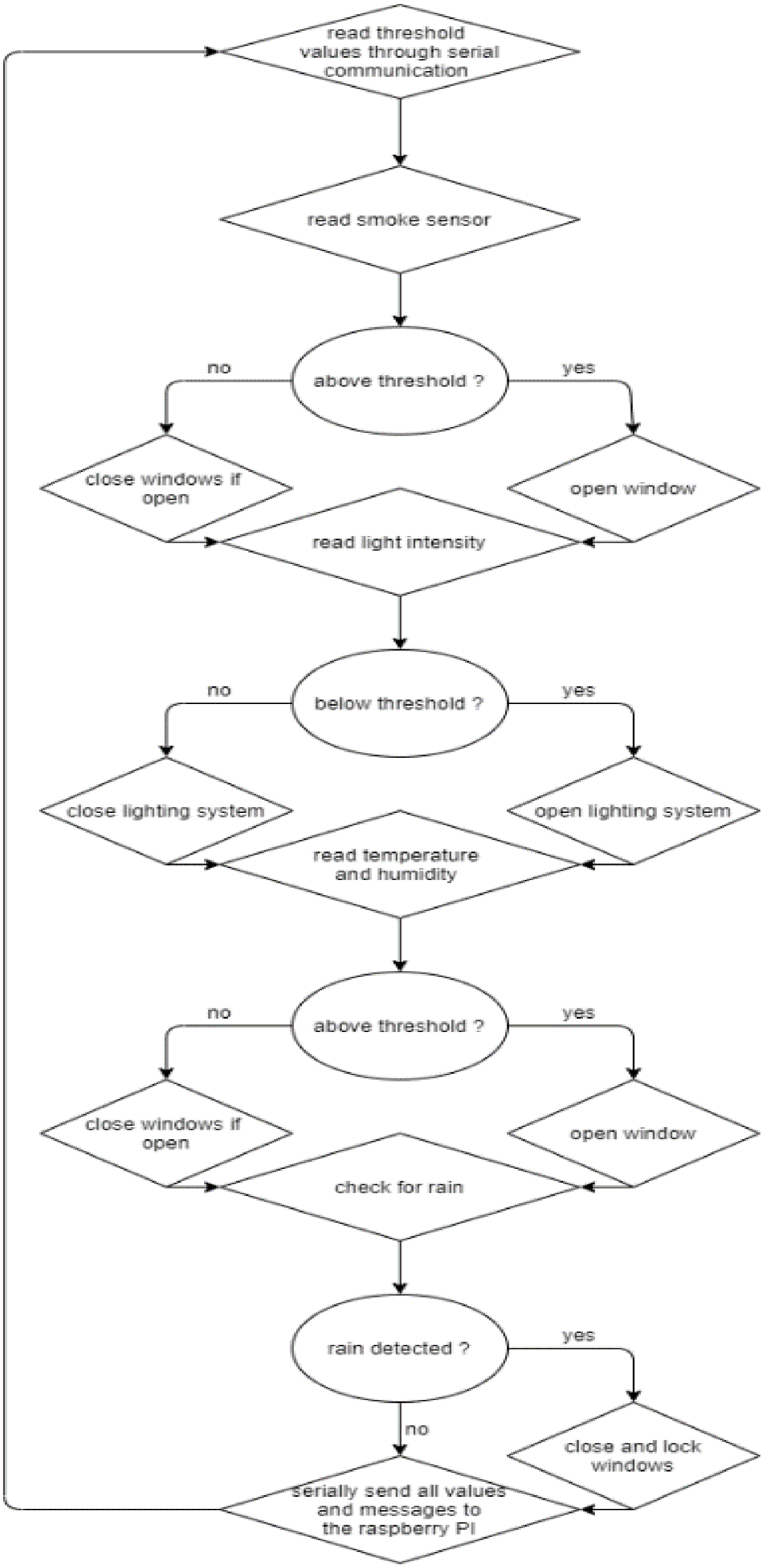

Figure 9- Arduino program flow chart

3.2.3 RPI Python Codes

The RPI main job is to:

1) Download threshold.txt that contains user generated data from AWS S3 bucket running download.py.
2) Serially send the values to the Arduino and read values from the Arduino and save them to output.txt file running tx_rx.py.
3) Take snapshot of the greenhouse using photo.py.
4) Upload snapshot and Arduino generated messages(output.txt) to the AWS S3 bucket running upload.py.

A simple bash script will run in a while loop that will run 5 different python codes that will do, Fig.10. shows what scripts the RPI will run.

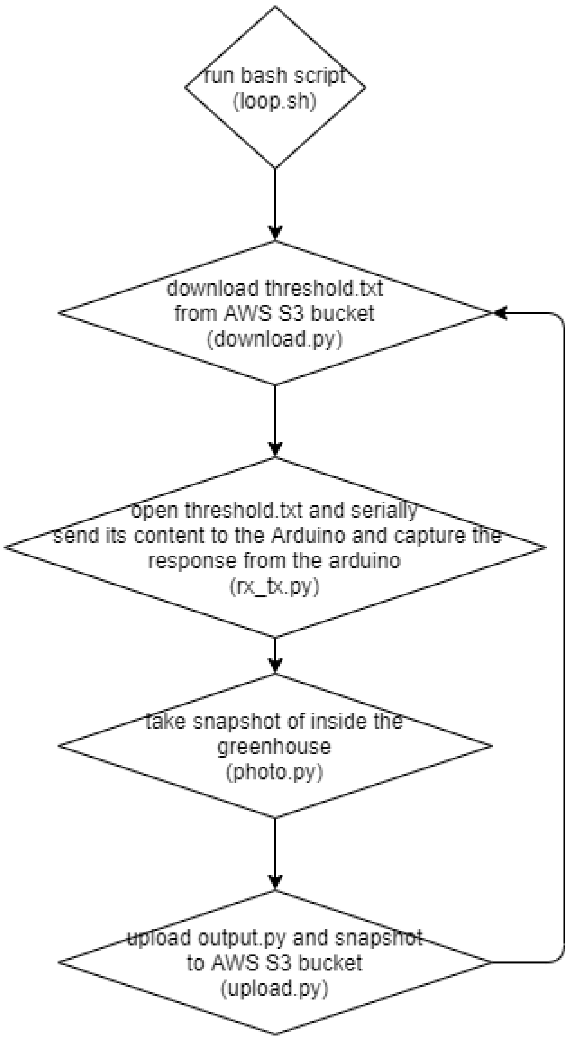

Figure 10- RPI code loop

3.2.4 Graphical User Interface Python Application

The python program is installed on the end user computer, it will be the main way for the user to interact with the greenhouse. From the GUI the user will input desired threshold values, the program will store these values to threshold.txt and uploads the file to S3 bucket, when the button submit values is pressed as shown in Fig.11.

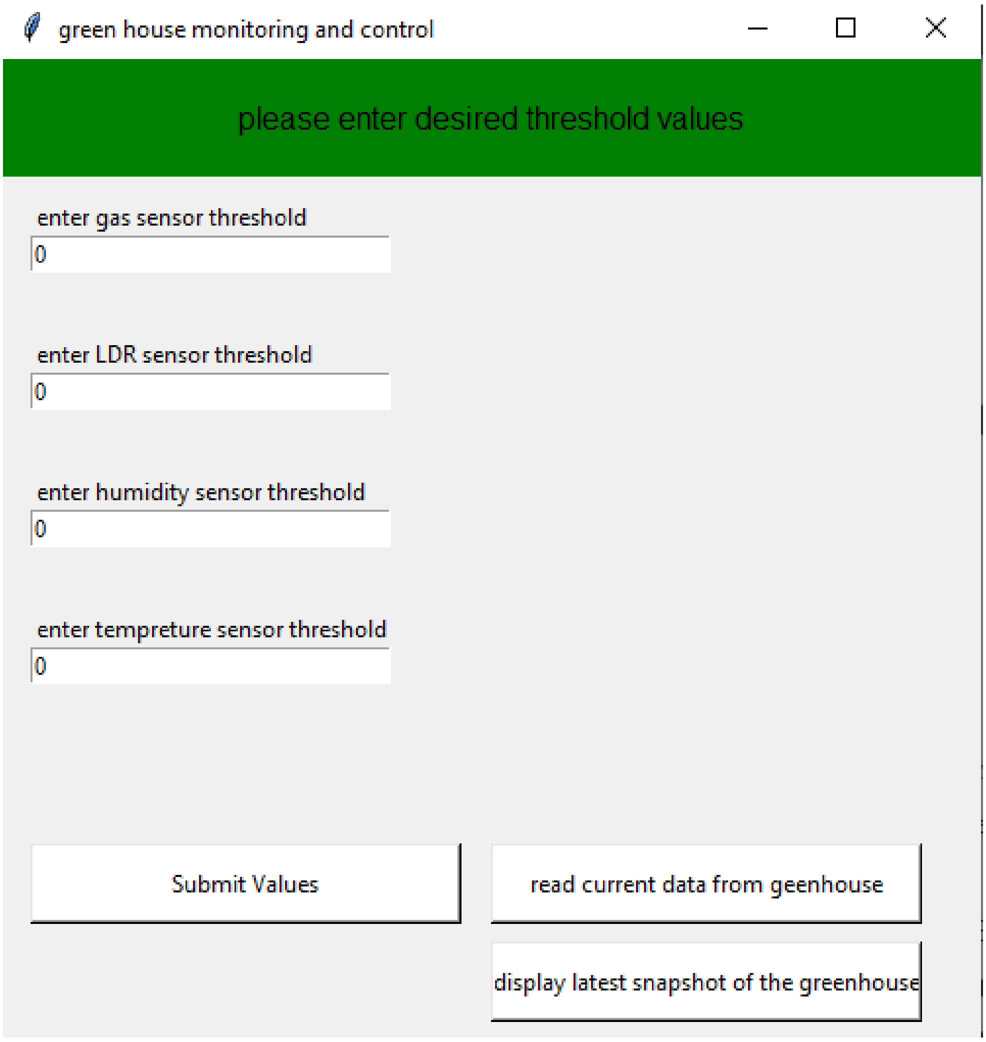

Figure 11- python GUI sending values to RPI interface

also, the GUI will display the latest greenhouse messages and greenhouse, when the user presses the read current data from greenhouse the application will download output.txt from S3 bucket and display its input to the user as shown in Fig.12.

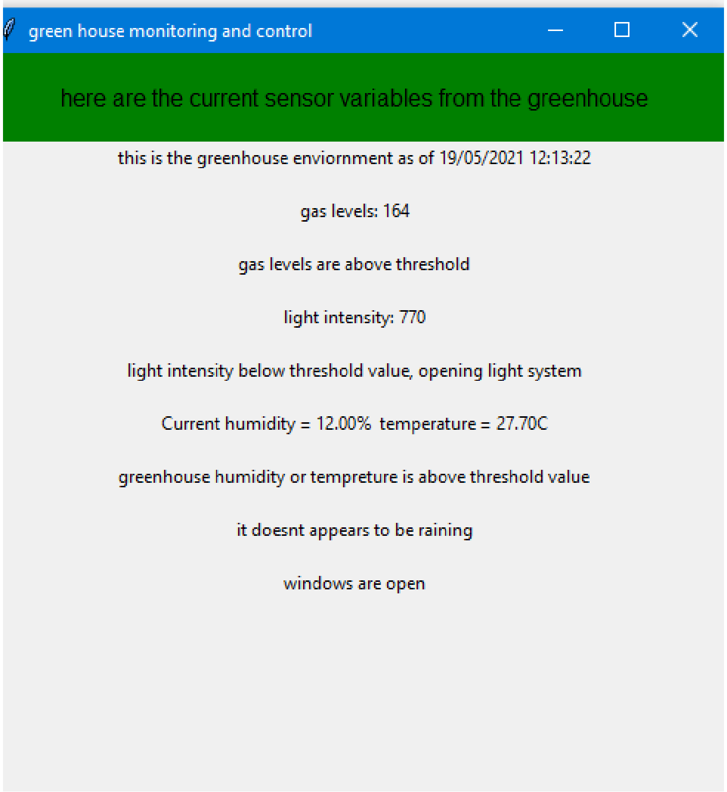

Figure 12- python GUI reading current greenhouse environmental values and messages

Also, the user can see the latest snapshot inside the greenhouse using the display latest snapshot button, once the button is pressed the snapshot is downloaded from the AWS S3 bucket, in this sub-window also the button diagnoses that when pressed will run MATLAB code to detect defected leaves in the greenhouse as shown in Fig.13.

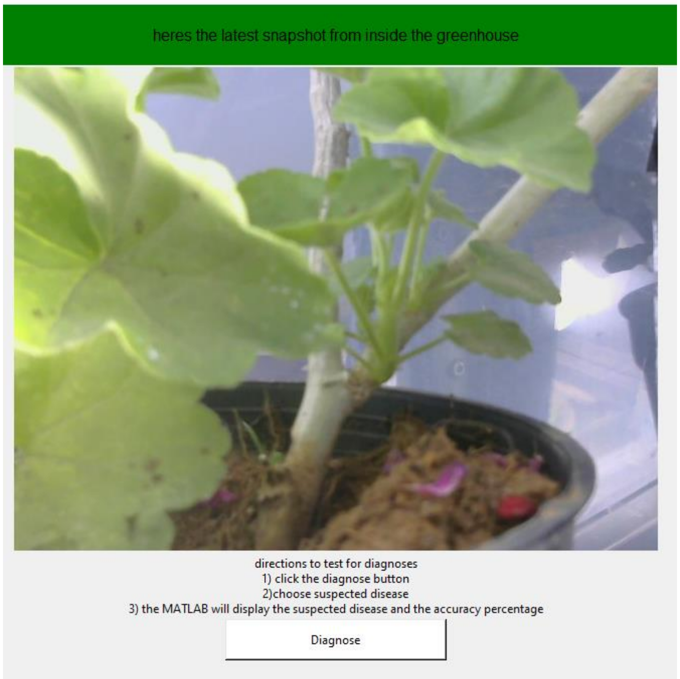

Figure 13- python GUI will display latest snapshot from the greenhouse and will use image recognition to detect diseases affecting the leaves of the plant.

3.3 Applying Image Processing Techniques and Artificial Neural Network to Detect Plant Diseases.

The present work recommends a system for identifying plant diseases early and accurately, using image processing techniques and artificial neural network. Agriculturalists suffer great obstacles in changing from one disease control policy to another. Depend On absolute naked eye to detect diseases can be costly, a range of plant diseases cause a significant risk to the farming sector by reducing the life of the plants.[27] the present work is targeted to develop a simple disease recognition system for plant diseases.

3.3.1 Proposed Methodology

The block diagram of the proposed system is shown in below Fig. 14. The step by step proposed approach consists of leaf image database collection, it is started with capturing the images, but in our project, we will use a Dataset obtained from “Mendeley Data” website [28], since we do not have these materials, in this work we will be focusing on five different diseases for different vegetables, then segmented the affected Area using automatic thresholding value after converted image to L*a*b* Color Space. Now we have the interest area texture to extract the features using GLCM method and make some statistical calculations After All, the feature values are fed as input to the SVM classifier to classify the given image.

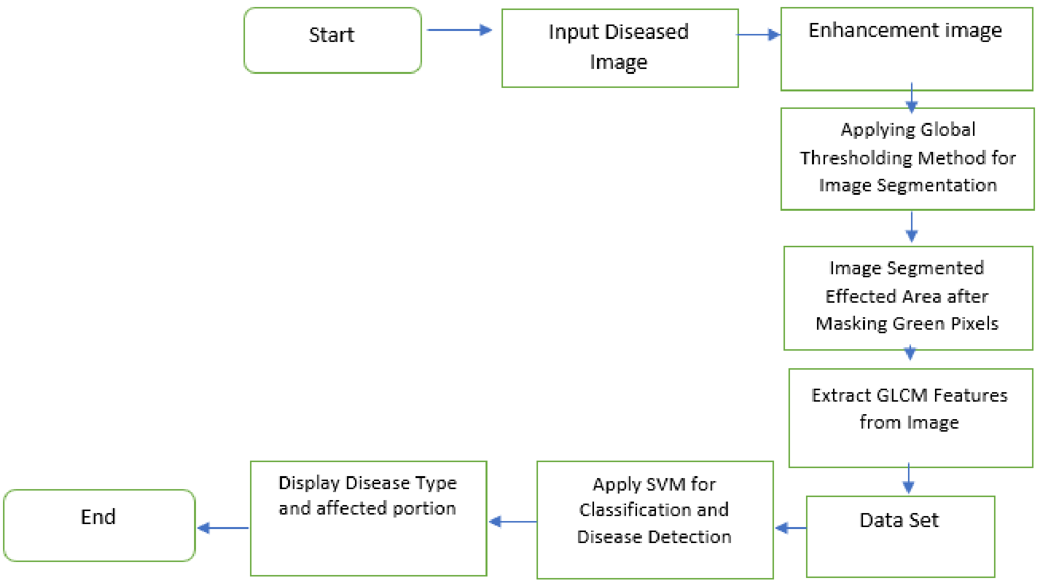

Figure 14- Framework of the proposed system

3.3.1.1 Image Acquisition

Image acquisition is the first step in the proposed method of digital image processing, and it is described as capturing the image through a digital camera and stores it in digital media for further MATLAB operations [29] . In this system, the available images from the digital camera or internet are also valid. Total 7028 image samples are captured of five classes shown in Fig.15, Pepper bell Bacterial spot, Potato Early blight, Potato Late blight, Tomato Bacterial spot, and Tomato Late blight diseases. Late blight and Early blight diseases are commonly found on tomato and potato plants [30], so, selecting the same disease for two different plants making some challenging for classification.

Plant Diseases	No. of Images used for Testing	No. of Images used for Training
Pepper bell Bacterial spot	970	22
Potato Early blight	970	30
Potato Late blight	970	30
Tomato Bacterial spot	2100	27
Tomato Late blight	1870	39
Total	6880	148

Figure 15- Plant diseases dataset

3.3.1.2 Image Preprocessing

Preprocessing stage is to enhance the image by removing noise and remove unwanted distortions, it does not add data, it positively enhances the image features that are important for further processing. The stored images are resized to a standard size(255x255).

Image enhancement is carried out for enhancing the contrast. The histogram equalization which distributes the intensities of the images is applied to the image to enhance the plant disease images. Fig.16. the contrast of the image is low in this case before applying Histogram Equalization. Moreover, as expected, the brighter pixel intensities were compressed which means it will be suitable for darkening.

Figure 16- Contrast Enhancement using Histogram Equalization.

3.3.1.3 Image Segmentation

Image segmentation is one of the essential approaches of digital image processing that used to simplify and change the representation of an image into something that is more meaningful and simpler to analyze, and to distinguish the object of interest from the background as shown in Fig.17, first it is good practice to select a color space before choosing the segmentation method since we act with color images. We select CIE L*a*b*, or CIELAB, The L* which represents a perceptual lightness, maximum value is 100 which represents white, and zero is the minimum value, which represents black. The a* and b* have no exact limits. Positive a* is red. Negative a* is green. Positive b* is yellow. Negative b* is blue. The Lab color space allows you to quantify the differences between points plotted in the color space correspond to visible differences between the colors plotted [31] . This means that the values give you an independent value representing that color. from this point, we can choose the threshold values for each color by visual inspection of the image based on your area of interest, in this technique, instead of using trial and error, we will use a graphical interface that using a slider to see the result immediately from MATLAB, called Color Thresholder in the image processing toolbox.

We will use Lab color space for each channel histogram to notify the peaks and valleys that represent each color in the a*, and b* channels histogram to allocate the cluster in the image, and masking the green pixels , by shifting the scroll of a* channel to zero to make sure we are masking all green pixels as shown in Fig.18.This is done in sense that the green colored pixels regularly represent the healthy parts of the leaf , and they do not add any valuable information to disease classification. Furthermore, this significantly reduces the processing time.

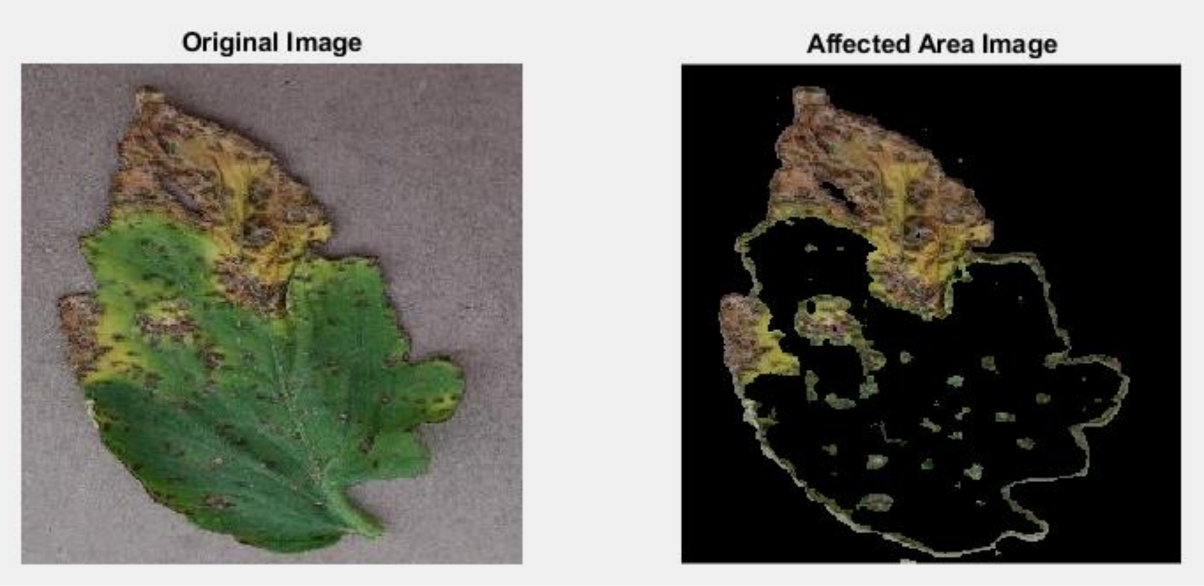

Figure 17 -Segmented affected Area for Tomato Bacterial Disease

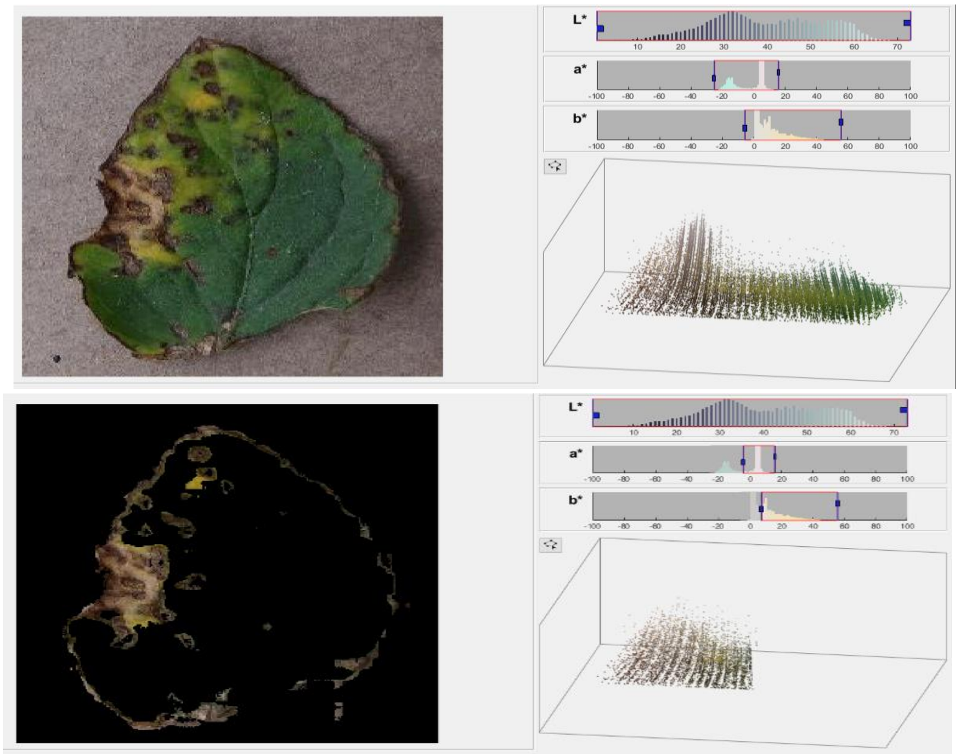
Figure 18 -Color Thresholder Application

3.3.1.4 Feature Extraction

Feature extraction is a process involving number of properties required to describe a large set of data accurately. Features are quantitative measures of texture that describe salient features in the image, there are two categories of features in the spatial domain; first-order statistical features and second-order statistical features, the first order depends only on pixel values and do not take the distribution of pixels into consideration, while the second-order statistical depends on both pixel values and the spatial inter-relationship. Statistical texture features are be given by gray level co-occurrence matrix (GLCM) [32], it is a common texture-based feature extraction method. The GLCM determines the textural correlation between pixels in distance and angle as show in Fig.19. The GLCM properties of an image are expressed as a matrix with the same number of rows and columns as the gray values in the image. The components of this matrix depend on the frequency of the two specific pixels. Both pixel pairs can vary depending on their neighborhood distribution. These matrix elements' values depending on the gray value of the rows and columns, the matrix would be larger If the intensity values increasing. This creates a time-consuming process load, the GLCM features used in this project are contrast, correlation, energy, homogeneity, mean, standard deviation, entropy, RMS, variance, smoothness, skewness, and inverse difference movement. The procedure in this work includes first, convert each segmented image to Grayscale image so we can calculate the co-occurrence matrix by "graycomatrix" MATLAB function, and extract the statistical texture features, and make a vector of features for every sample image for each class until the last one, now we must combine these vector feature in one big dataset and use it for future training.

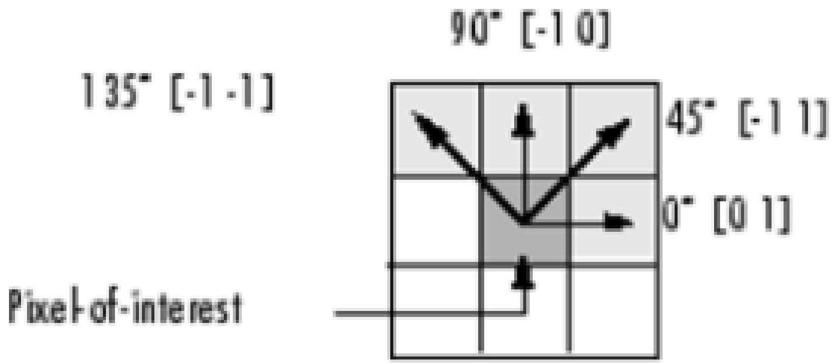

Figure 19 -GLCM depend on both pixel values and the spatial inter-relationship.

3.3.1.5 Training & Classification

the classification is performed through using Support Vector Machine (SVM). A support vector machine creates a hyper-plane or multi hyper-planes in all dimensional space, which can be utilized for classification, regression, and other different errands. SVM is supervised machine learning with related learning algorithms that examine the information and recognize patterns, used for classification analysis. Given a set of training data, and marked each sample belonging to one of the categories, Multiclass SVM aims to allocate labels to classes by applying SVM. The dominant approach to distinguish classes well is to convert the single multiclass problem into multiple two-class classification problems, this may also support the separation of classes, disease classification may face some challenges to recognize since it can mix up with other classes, this kind of problems can be eased by construct binary classifiers which distinguish between one of the labels and the rest (one vs all) shown in Fig.20(a), or among every pair of classes (one vs one) Fig.20(b). [33]

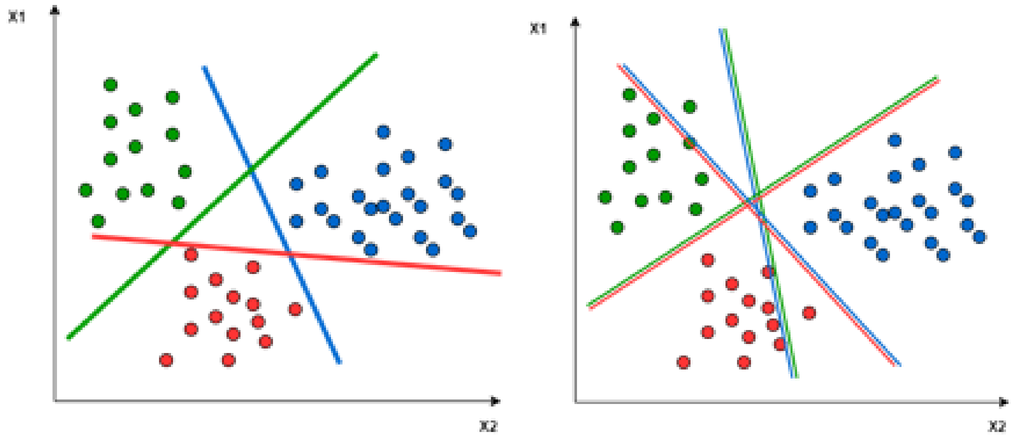

Figure 20 -(a) One-to-all approach (b) One-to-one approach [34].

Once extracting texture features and save it in (.mat) file, the file contains a (6880x13) matrix representing 6880 samples from all the five classes. Each row had 13 columns representing the 13 texture features for each sample image. Labeling the classes Done separately and the training dataset have been arranged, each class given a unique number (one, two, three, four, or five) which represented the class (*i.e.*, Tomato Late blight). "1" represented Pepper bell Bacterial spot disease, "2" represented Potato Early blight disease, "3" represented Potato Late blight disease, "4" represented Tomato Bacterial spot disease, "5" represented Tomato Late blight disease as shown if the Fig.21, then, a software program was created in MATLAB that would train the Classifier using the “Train_Feat.mat”, and then use the test texture feature vector that represent the image that would be predicted to perform the classification task as shown in Fig. 22.

label	Plant Diseases	No. of Samples
1	Pepper bell Bacterial spots	970
2	Potato Early blight	970
3	Potato Late blight	970
4	Tomato Bacterial spots	2100
5	Tomato Late blight	1870
	Total	6880

Figure 21 -labeling the arranged training dataset.

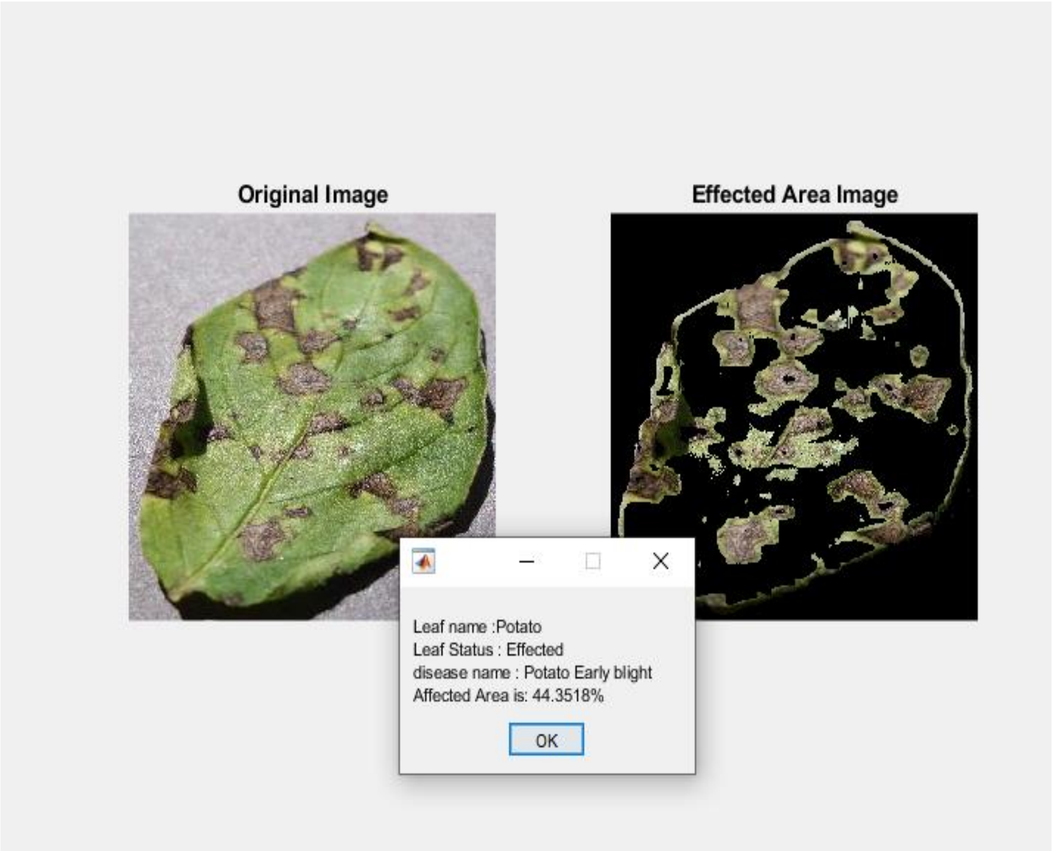

Figure 22 -prediction classification message box

3.4 Prototype IoT Smart Greenhouse

To test our system, we built a mini greenhouse and inserted the Arduino and the sensors inside as displayed in Fig.23, we fitted it with a window that was attached to 2 servo motors that will open if any of the humidity, temperature or gas levels crossed the threshold voltage and vice versa.

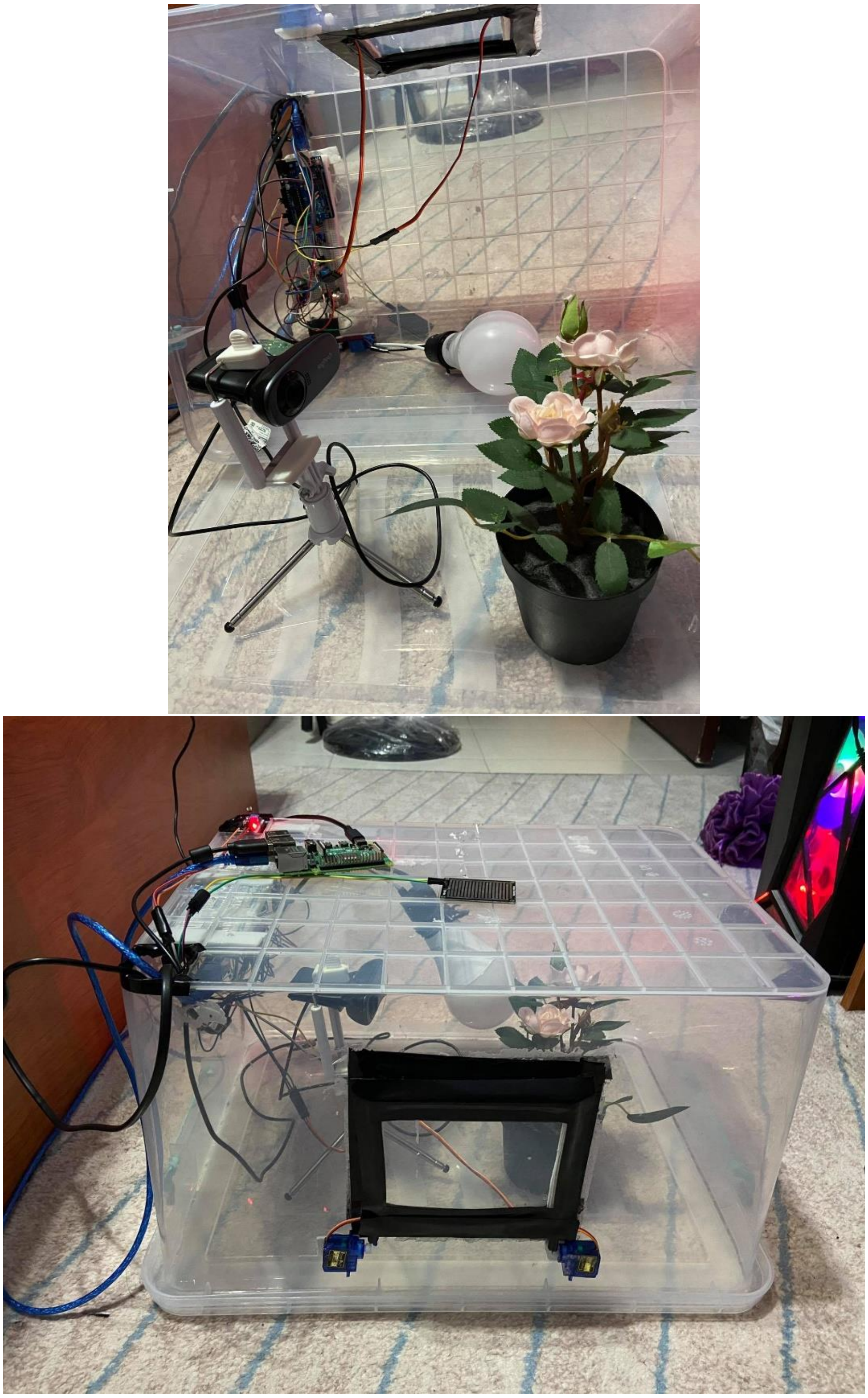

Figure 23 -the mini greenhouse built to test the system, it shows the window, webcam to capture the inside of the house.

The window will lock if any rain is detected using a rain detection sensor, also, in addition, the greenhouse has a lighting system that will light when the light intensity is lower than the threshold both the rain detection sensor and the light intensity sensor are located at the top of the house as shown in Fig.24. we placed the photoresistor module at the top of the greenhouse and placed it on a jet-black sheet so when the light system is open it does not interfere with its readings.

Figure 24 -the top of the mini greenhouse showing the rain detection sensor and the photoresistor module.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Greenhouse System Testing

We will test the greenhouse to see if the application is working, to verify that the system is working correctly we took a bottom-up approach, meaning we first test the Arduino system, then the RPI and cloud interfacing finally the GUI interface and image recognition

4.1.1 Arduino Testing

first, we worked on connecting the Arduino to the sensors correctly that is done by testing every sensor is tested individually to weed out hardware failure possibilities, when the sensors are confirmed to be working correctly all the Sensors are grouped together and a Arduino code is written to collect data and display it on the Serial monitor, next step is to connect the Arduino serially to the RPI so that rather than display the output to the serial monitor It gets sent to the RPI finally we need to test the Arduino's ability to receive a number and correctly assign it to its corresponding variable, the RPI will have to send variables in a start:NUM1:NUM2:NUM3:NUM4 format and the Arduino must correctly split it and assign ever NUMX variable to a correct corresponding variable and save it.

4.1.2 RPI Testing

First thing to test is the serial communication, simply attaching a LED to pin 13 and gnd pin in an Arduino can tell us when serial communication is happening, basically when the Arduino receives anything the LED will flash, after successfully validating that the Arduino is receiving python scripts for the RPI will be written to:

1) Download user generated data from S3 bucket using boto3 python library.
2) The data from the user will be written in a text file with a number in every new line so a python program that will read the text file and parse the numbers in it in a start: NUM1:NUM2: NUM3:NUM4 String format where "start" is the start bit that when read the Arduino will start measuring and sending messages to the RPI.
3) Take the generated String and serially send it to the Arduino.

4) After sending the string the raspberry will listen to the next 8 lines these lines represent the current greenhouse status and environment variables and write them to an output text file to be uploaded to S3 bucket using boto3 python library.
5) Take snapshot of inside the greenhouse and upload the picture to S3 bucket using boto3 python library.

Implementing modularity is a good way to avoid bugs and messy codes, we can write different python codes and link them together using bash scripts, implementing a bash script that runs an infinite while loop holds multiple benefits, if one program crashes the system will continue to run normally and the loop won't be broken, a python script would crash or serial communication will send or receive corrupted data and on the next iteration of the loop the program would work fine with no bugs, this is because the RPI computing power is not that powerful or problems in USB serial communication that is outside our powers that it sends or receives corrupted data.

4.1.3 Cloud Testing

The main use of cloud in this project is to be used as a storage to store the user generated data, output file that the RPI generated and the snapshot of inside the greenhouse or main method of accessing the S3 storage is the boto3 python library, boto3 library has functions that can download and upload files to AWS S3 bucket provided bucket id, bucket access key and secret access key.

4.1.4 User GUI Testing

The main way for the user to interact with the greenhouse is by the GUI, the GUI is written using python tkinter library, tkinter is the standard python GUI tool, we need the GUI to do the following:

1) Take user input, save to text file and upload to S3 bucket, this will be the same file the raspberry will download and serially send its content to the Arduino.
2) Download RPI generated text file that contains the Arduino generated messages and sensor values from S3 bucket.
3) Download snapshot from S3 bucket and display it to the user and run MATLAB code to detect disease from the snapshot of the plant leaves.

First step is to create the layout of the application and display the banners, next is to create the input boxes for the user to input the data, the main screen of the application is basically a form that takes user input from boxes and uploads it to the S3 bucket, after confirming that the main interface displays correct banners, saves the user input correctly and uploads it when a button is pressed, we next create secondary windows, these will be displaying the current status of the greenhouse and the image recognition part, for reading the values of the greenhouse when the read values button is pressed the output values will be downloaded from the S3 bucket and displayed to the user in a new window, finally in the main window there is a “disease detection” button, when pressed this button will download the greenhouse snapshot and a MATLAB code is called from python and the image will be segmented and the leaves of the plants inside the greenhouse will be tested for disease and the result will be displayed to the user.

4.1.5 Disease Detection Result and Testing

The classification performance of the SVM was compared to other machine learning classifiers to determine if SVM is the best technique. Classification Learner App in MATLAB was used to train the plant diseases with the same features used in the classifier.

To do the same training and testing in Classification Learner App. All the 22 machine learning algorithms in the Classification Learner App were selected and the 10-fold cross-validation (C.V) with one-to-one approach was used to set the training and testing data for the model. The 10-fold C.V technique was used to set 10% data for testing and 90% data for training. The 10-fold C.V also means that the training and testing process is repeated ten times. The results of the classification using the 22 machine learning algorithms are shown in Table 2. that the highest classification accuracy is 82.4%, which was produced by Cubic SVM, followed by Quadratic SVM with 81.7% and the third in rank is the Fine Gaussian SVM with 75.2%.

Table 2 -22 machine learning algorithms using 10-fold cross-validation.

Classifier	Classifier type	Accuracy (%)	Prediction speed (Objects/Seconds)	Training time (Seconds)
Decision Trees	Fine Tree	67.7	290000	1.5564017
	Medium Tree	63.5	270000	1.377095
	Coarse Tree	53.7	290000	1.2342071
Discriminant Analysis	Linear Discriminant	68.0	240000	0.6222891
	Quadratic Discriminant (Using Diagonal Covariance)	64.8	180000	1.0536454
SVM	Linear SVM	72.8	110000	19.262819
	Quadratic SVM	81.7	410000	190.18886
	Cubic SVM	82.4	46000	804.12537
	Fine Gaussian SVM	75.2	6500	14.577755
	Medium Gaussian SVM	73.3	8100	11.1521127
	Coarse Gaussian SVM	66.2	5500	12.299001
KNN	Fine KNN	68.2	12000	3.4496906
	Medium KNN	71.2	11000	3.3252121
	Coarse KNN	68.7	6500	4.7121804
	Cosine KNN	71.0	11000	4.1591818
	Cubic KNN	71.6	540	60.0427419
	Weighted KNN	72.2	11000	5.4975716
Ensemble classifiers	Boosted Trees	66.3	42000	23.9368547
	Bagged Trees	73.4	200000	17.4178874
	Subspace Discriminant	64.7	18000	9.3814331
	Subspace KNN	57.0	6900	12.862472
	RUS Boosted Trees	65.2	39000	25.2798088

In Cubic SVM classification, 776 images of Pepper bell Bacterial spots disease are correctly classified, and misclassified samples are 193. The correct classification rate for Pepper bell Bacterial spots disease is 80.1%. For Potato Early blight disease, 853 samples are correctly classified, and misclassified samples are 117. The correct classification rate for Potato Early blight disease is 87.9%. For Potato Late blight disease, 691 samples are correctly classified, and misclassified samples are 279. The correct classification rate for Potato Late blight disease is 71.2%. For Tomato Bacterial spots disease, 1896 samples are correctly classified out of 2100. The correct classification for Tomato Bacterial spots disease is 90.3%. For Tomato Late blight disease, 1455 samples are correctly classified out of 1870. The correct classification rate for

Tomato Late blight disease is 77.8%. The overall system classification rate for the above leaf samples is 82.4% and the error rate of the system is 17.6%. as shown in Fig.25.

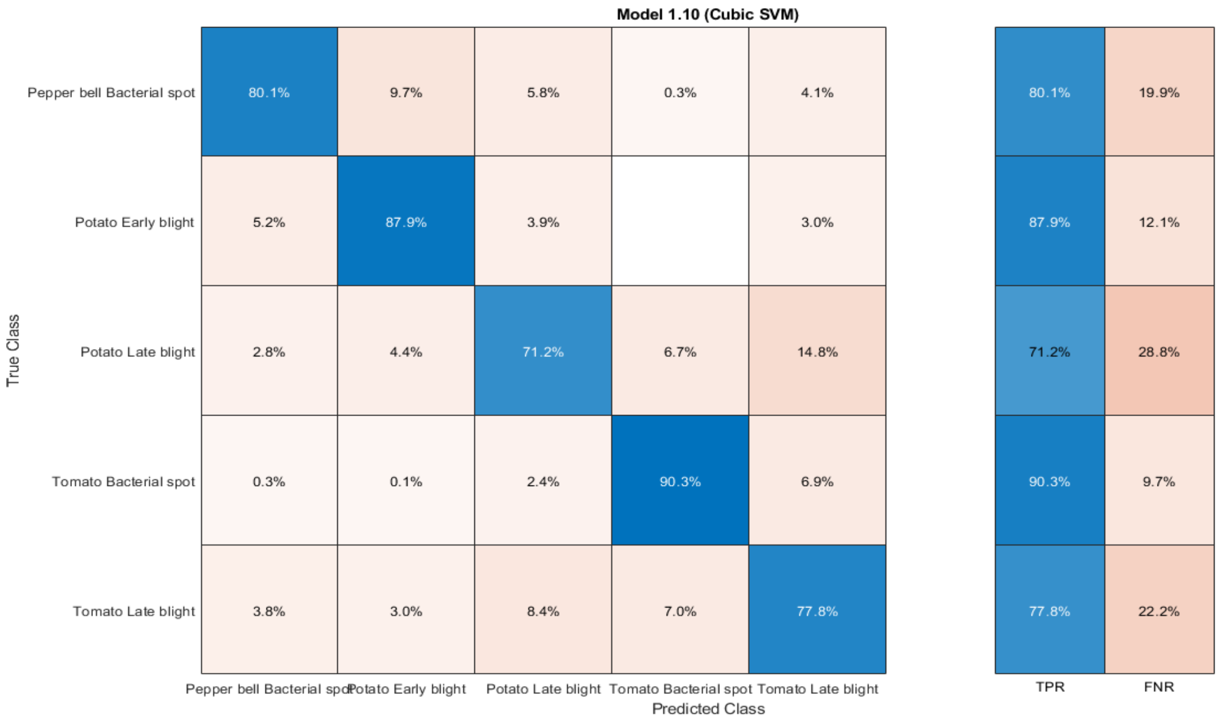

Figure 25 -Confusion Matrix for Cubic SVM Classification

4.2 Challenges

There is several problems that limit this project, for once the use of cheap hobbyist Arduino sensors while they may be good for trainees they don't provide precision and are very prone to sending corrupted data, this happens but not as frequently and for reasons beyond our control, secondly the poor processing capabilities of the RPI made it frustrating to debug problems, sometimes the OS would freeze and a reboot was necessary the use of custom made light Linux distros could solve this problem especially since Raspbian OS (RPI main OS) is filled with bloatware that could slow down the RPI, debugging the Arduino could be challenging especially since using Serial communication requires us to do so inside the RPI graphical interface and that was a very frustrating experience considering the 1 GB ram and the buggy Raspbian OS graphical interface, also a problem is when the user enters large values as threshold values while this doesn't brick the system it does create undesirable outputs, however the biggest problem facing the system is the USB serial communication where sometimes it can send or receive corrupted data that will leave the Arduino system acting randomly even though this happened rarely while testing the system, one old and very common problem in the RPI is the insatiability of the USB ports as it can randomly disconnect and reconnect devices, this happens due to faulty or outdated drivers or buggy newer distros, as for disease detection, the model has an accuracy of 82.4% this is because as mentioned before ,sometimes two plants can suffer the same disease and this can make the detection process harder on the classifier, also, we applied automatic segmentation technique and if the image was not taken in a good condition (*i.e.* the texture of the background is similar to a affected leaf texture) it's may affect the classification accuracy, also due to minimal knowledge about machine learning the image recognition requires longer time for development.

4.3 Final Product

the final deliverable product is a greenhouse system that can maintain the desired light intensity, temperature, humidity, and gas quality to maintain optimal plant growth, also using image recognition we can detect diseases and alert the farmer, all the relevant information for the greenhouse is in the cloud this makes accessing the greenhouse information available regardless of geographic location.

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

With the ease of accessing information technology, the low costs of internet connections and computer hardware and microcontrollers, the IoT is becoming more popular in engineering applications, it is now the go-to option in small, medium and even big agriculture applications [35], the model we introduced in this project is excellent for small and medium agriculture applications, the characteristics of the model we introduced in this project can be adapted by farmers that want to develop precision farming, although IoT has entered allot of fields some restrictions exist that prevent the agriculture sector to do so, high initial costs of precise and more accurate sensors for example, some agricultural countries may have unstable internet connections and this is essential for our model, also most workers in the agriculture sector are apprehensive when it comes to technology, A low cost IoT based solution is efficient in monitoring and controlling the environment inside the greenhouse, this solution is simple, portable and affordable and additional to that it also helps the farmers in identifying diseases using image processing.

5.2 Future Work

Improving the system can be done on many levels, for once a better embedded system with better and more accurate sensors can be used better and more heavier servos, also a cultivation system could be added to minimalize human interaction, one downside the Arduino UNO has is the lack of a debugger which makes checking scripts hard, the RPI is a good and powerful microcomputer, but a good upgrade is using a variant with more RAM albeit this will cost more. Also, we can develop our database for more plant disease identification.

REFERENCES

[1] N. Castilla, Greenhouse Technology and Management, 2nd ed. Oxfordshire: CABI, 2013.

[2] P. P. Reddy, Sustainable Crop Protection under Protected Cultivation. Springer Singapore, 2016.

[3] McNutty, Jennifer (3 November 2017). "Solar greenhouses generate electricity and grow crops at the same time, UC Santa Cruz study reveals". USC Newscenter. Santa Cruz: University of California. Retrieved 6 November 2017.

[4] M. K. Jha, S. S. Paikra, and M. R. Sahu, Protected Cultivation of Horticulture Crops. Educreation Publishing, 2019.

[5] A. K. Sharma, S. Wahab, and R. Srivastava, Agriculture Diversification: Problems and Perspectives. I.K. International Publishing House Pvt. Limited, 2010.

[6] *Employment in agriculture (% of total employment) (modeled ILO estimate) | Data*. (2020, January 1). Worldbank. https://data.worldbank.org/indicator/SL.AGR.EMPL.ZS

[7] Ritchie, H. (2018, June 13). *Urbanization*. Our World in Data. https://ourworldindata.org/urbanization

[8] A. Kumar, G. N. Tiwari, S. Kumar, and M. Pandey, "Role of Greenhouse Technology in Agricultural Engineering," Int. J. Agric. Res., vol. 1, no. 4, pp. 364–372, 2006

[9] R. C. Ward, J. C. Loftis, and G. B. McBride, Design of Water Quality Monitoring Systems. Wiley, 1990

[10] *PLC Based Process Control*. (2020, December 17). Daenotes. https://www.daenotes.com/

[11] the world bank. (2019). *World Development Report 2019 THE CHANGING NATURE OF WORK*. https://www.worldbank.org/en/publication/wdr2019

[12] *Embedded Systems*. (2020). Omni Sci. https://www.omnisci.com

[13] I. Ardiansah and S. H. Putri, "Perbandingan Analisis SWOT Antara Platform Arduino UNO Raspberry Pi," in Seminar Nasional MIPA, 2016, pp. 27–28.

[14] Dejan, "DHT11 & DHT22 Sensor Temperature and Humidity Tutorial," HowToMechatronics, Jan. 13, 2016. https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/ (accessed Dec. 19, 2020).

[15] "In-Depth: How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino," Last Minute Engineers, Aug. 14, 2018. https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/ (accessed Dec. 20, 2020).

[16] "Light Sensor including Photocell and LDR Sensor," Basic Electronics Tutorials, Aug. 15, 2013. https://www.electronics-tutorials.ws/io/io_4.html (accessed Dec. 20, 2020).

[17] Goel, A. (2018, October 23). *Servo Motor : types and working principle explained.* Engineering. https://engineering.eckovation.com/servo-motor-types-working principle-explained/

[18] "What is Cloud Computing? Everything You Need to Know," *SearchCloudComputing*. https://searchcloudcomputing.techtarget.com/definition/cloud-computing (accessed May 23, 2021).

[19] Cloud computing. (2021, January 6). In Wikipedia.

https://en.wikipedia.org/wiki/Cloud_computing

[20] Statista. (2020, September 14). *Cloud infrastructure services market share worldwide 2017-2020, by vendor*. https://www.statista.com/statistics/477277/cloud-infrastructure-services-market-share/

[21] H. (2020b, October 28). *4 Benefits of Smart Greenhouses and How to Get Started | BehrTech Blog*. BehrTech. https://behrtech.com/blog/4-benefits-of-smart-greenhouses-and-how-to-get-started/

[22] Luymes, G. (2017, November 20). *Crop theft a growing concern for some farmers*. Times Colonist. https://www.timescolonist.com/news/b-c/crop-theft-a-growing-concern-for-some-farmers-1.23099276

[23] Mekki, M., Abdallah, O., Amin, M. B. M., Eltayeb, M., Abdalfatah, T., & Babiker, A. (2015). Greenhouse monitoring and control system based on wireless Sensor Network. *2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE)*, 270–274. https://doi.org/10.1109/iccneee.2015.7381396

[24] Chin, Y. S., & Audah, L. (2017). Vertical farming monitoring system using the internet of things (IoT). *Vertical Farming Monitoring System Using the Internet of Things (IoT)*, 2–5. https://doi.org/10.1063/1.5002039

[25] Birhanu Wudneh, T. (2019). Implementation of IoT with Image Processing in Greenhouse Monitoring System. *International Journal of Innovative Technology and Exploring Engineering*, *8*(9), 509–514. https://doi.org/10.35940/ijitee.i7711.078919

[26] *boto: A Python interface to Amazon Web Services — boto v2.49.0*. (2021). Boto. https://boto.readthedocs.io/en/latest/

[27] "COVID-19 and the food and agriculture sector: Issues and policy responses," *OECD*. https://www.oecd.org/coronavirus/policy-responses/covid-19-and-the-food-and-agriculture-sector-issues-and-policy-responses-a23f764b/ (accessed May 12, 2021).

[28] M.-L. Huang and Y.-H. Chang, "Dataset of Tomato Leaves," vol. 1, May 2020, doi: 10.17632/ngdgg79rzb.1.

[29] R. C. Gonzalez, R. E. Woods, and B. R. Masters, "Digital Image Processing, Third Edition," *J. Biomed. Opt.*, vol. 14, no. 2, p. 029901, 2009, doi: 10.1117/1.3115362.

[30] "Late Blight: Symptoms, Treatment and Control," *Planet Natural*. //www.planetnatural.com/pest-problem-solver/plant-disease/late-blight/ (accessed May 12, 2021).

[31] "CIELAB color space," *Wikipedia*. May 06, 2021. Accessed: May 13, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=CIELAB_color_space&oldid=1021691209

[32] N. Aggarwal and R. Agrawal, "First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images," *J. Signal Inf. Process.*, vol. 03, Jan. 2012, doi: 10.4236/jsip.2012.32019.

[33] K. Varpa, H. Joutsijoki, K. Iltanen, and M. Juhola, "Applying One-vs-One and One-vs-All Classifiers in k-Nearest Neighbour Method and Support Vector Machines to an Otoneurological Multi-Class Problem," p. 6.

[34] baeldung, "Multiclass Classification Using Support Vector Machines | Baeldung on Computer Science," Oct. 07, 2020. https://www.baeldung.com/cs/svm-multiclass-classification (accessed May 20, 2021).

[35] WIRED. (2017, November 4). *Tokyo's Vertical Farms - The Future of Farming | WIRED*. YouTube. https://www.youtube.com/watch?v=pGtdoGXhjxQ

APPENDICES

Appendix A:

Libraries and Toolboxes Used:

Multiple libraries were used in the project as shown in Fig.26 these libraries provide useful functions and modules, these include:

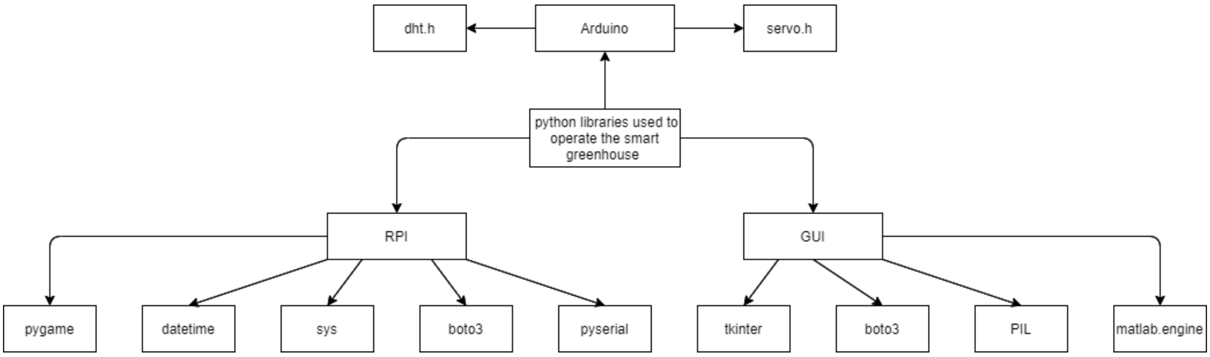

Figure 26 -libraries used in the project.

Libraries that will run on the RPI segment:

- Boto3: boto3 was used to access S3 bucket storage where the configuration and output values are stored.
- Pyserial: pyserial is used to access the serial port that is connected to the Arduino.
- Datetime: used to print the current date and time to the output file in the RPI.
- Sys: used to exit the script when all the response has been captured.
- Pygame: used to access the web cam and take a snapshot of inside the greenhouse.

Libraries used in the GUI segment:

- Tkinter: python standard GUI building tool.
- PIL: used to display images inside the GUI environment.
- Boto3: used to download output values and latest snapshot, also uploads threshold values to cloud.
- MATLAB. engine: standard MATLAB engine imported to python that will call a MATLAB script from python.

Libraries used in Arduino segment:

- Dht.h: Arduino library for the dht humidity and temperature sensor.
- Servo.h: Arduino library for the servo motors.

Toolboxes used in MATLAB segment:

- Image processing toolbox: used in segmentation and feature extraction.
- Statistics and machine learning toolbox: used in disease detection and classification.

Appendix B:

Code Segments:

scripts that will run on the RPI segment:

- Loop.sh: this bash script will run in a loop and will run the following scripts in order: download.py, tx_rx.py, photo.py, upload.py.
- Download.py: will download the threshold values from the S3 bucket.
- Tx_rx.py: will open the threshold.txt file read it and serially send its content in a start: NUM1:NUM2: NUM3:NUM4 format and reads the messages from the serial port and write it to the output.txt file.
- Photo.py: will take a snapshot of inside the greenhouse.
- Upload.py: will upload both the snapshot and output.txt to the S3 bucket.
- Secret.py: contains the secret access key and the access key for the S3 bucket.

The GUI segment:

- GUI.py: graphical user interface that will take the threshold values from the user, download output file from the S3 bucket and display its content, also will download the snapshot from the S3 bucket and display it to the user, also will run Multi_SVM.m on demand.

The disease detection segment:

- Diseases_Features.m: MATLAB script for establishing the Training Dataset.
- Multi_SVM.m: MATLAB script for disease detection affecting the leaves.
- TrainingSet : arranged Training set folder.
- TestSet : Testing set folder.
- Train_Feat.mat: Training Dataset file.
- Train_Feat.xlsx: Training Dataset file used in Classification Learner App.

Appendix C:

User Manual:

note that at least MATLAB 2018 and python 3.7 must be downloaded for the GUI script to run, also MATLAB engine must be installed on user PC, to do this open MATLAB and on the command prompt:

1) cd (fullfile(matlabroot,'extern','engines','python'))
2) system ('python setup.py install')

Also note that the directory of Multi_SVM.m must be changed inside GUI.py to point to the script location.

To run the system:

1) Connect the external power supply and USB connection to the Arduino and the other end to the RPI.
2) Run loop.sh on the RPI.
3) Open GUI on user PC.
4) Enter desired threshold values then click “submit” button.
5) To view latest status in greenhouse, click “read current data from greenhouse” button.
6) To display latest snapshot of inside the greenhouse, click “display latest snapshot” button.
7) For classification disease from image click” Diagnose” button then select the image.