*Optimal Energy Management of Mini Greenhouses*

1std Xiaogan Zhang  
*Electrical Engineering Nano*  
*University of Alberta*Edmonton, Canada  
 xiaogan@ualberta.ca  
  
 2nd Yining Jin  
*Electrical Engineering*   
*University of Alberta*Edmonton, Canada  
jyining@ualberta.ca  
  
 3rd Howard Lin  
*Electrical Engineering Nano*  
*University of Alberta*Edmonton, Canada  
 hlin2@ualberta.ca  
  
 4th Xiangye Chen  
*Electrical Engineering*   
 *University of Alberta* Edmonton, Canada  
 xiangye@ualberta.ca

*Abstract*—Our project is an energy-efficient mini greenhouse that smartly controls environmental conditions such as temperature, CO2 levels, and humidity. We utilize a Raspberry Pi 4 Model B as the central control unit. An attached camera module is used with artificial intelligence to detect plant growth status. We develop an android application to illustrate data from sensors and display AI-generated plant growth suggestions. All of the specifications from our COVID-19 scope update are met. Our achievements including: system controlling the hardware inside of the greenhouse, using camera module to synchronize pictures to the computer for processing, sending the received pictures into neural networks to classify the state of growth of mushrooms, and building an App on smartphones to monitor the growth of mushrooms, temperature, humidity, lighting, and CO2 concentration

# Introduction

Because of the problem of greenhouses having energy inefficiencies and a steadily growing demand for locally grown organic vegetables; our project is a result of these problems. Our capstone’s purpose is to design a box-sized greenhouse that will intelligently monitor the environment conditions for energy optimization and optimal plant development by utilizing artificial intelligence. In the project, the temperature, humidity, and density of CO2 will be adjusted according to the optimal growing conditions for our test subject. The greenhouse will have an intelligent monitoring system that manages the greenhouse’s energy consumption, and a control unit that controls the environment of the greenhouse. Lastly, a self-designed smartphone application will be used to set the value of each equipment inside the mini-greenhouse and will be able to provide the user with growth details. Concepts such as Deep Learning (DL) and Computer Vision (CV) methods are used to build the classifier to classify the plant growth stage. In our project, we utilize Deep learning as a type of Artificial Intelligence (AI) that imitates the way humans gain knowledge. Computer vision is also a field of AI that enables computers to obtain information from digital visual inputs, and act based on the information. Quantum computing is a type of computation that has the collective properties of quantum states, such as superposition, interference, and entanglement, to perform calculations. Adhering to our COVID-19 scope update; the IOT devices and creation of our complete greenhouse system has been canceled for the ECE 490/491 Capstone Project.

Figure 1. The Mini Greenhouse

Diagram

Description automatically generated

# Technical Content

To begin explaining the technical portion of our project the following block diagram in **Figure 2** details on a high level how our overall system is constructed. An appropriate schematic can be found at **Figure 3,4**. In this section of the report the technical details of our control system, artificial intelligence, smartphone application and how each component of our system communicates with each other.

## Hardware Design

For the design of our control system, our group utilized the Raspberry Pi 4B as the center for our control system. The Raspberry Pi monitored all our environmental conditions inside our greenhouse utilizing the sensors. The sensor that is being used for our design is an Adafruit SCD-30 NDIR Sensor. The Adafruit SCD-30 NDIR sensor is being used because of its capability to record temperature, CO2 density, and humidity. Which are all the current parameters our greenhouse ecosystem will utilize. The Raspberry Pi can utilize its ability to handle complex data algorithms which will be requirement to create our intelligent monitoring smart system that utilizes artificial intelligence. Another rationale behind the Raspberry Pi is that it will make one of our project goals more convenient because it supports Qiskit which is the quantum computing software development package. Quantum computing is a stretch goal of our group and will be discussed later. The capability that the Raspberry Pi can run Qiskit gives us the opportunity to remove an external computer that would be attached to the greenhouse that would have been running Qiskit originally. Furthermore, our overall design needs the inclusion of a camera. Another convenience of the Raspberry Pi is the ability to attach a camera module that will be used for monitoring the mushroom’s growth.

**A picture containing table

Description automatically generated**

A block diagram can be found in Figure 6 below and the circuit schematic can be found at the end of the report. How the system functions is that the Raspberry Pi will be receiving data values from our sensor. As the Raspberry Pi receives data values it will control the condition parameters by actuating the appropriate unit.   For example, according to the optimal temperature growth parameters set by the system, if the greenhouse environment drops below a certain temperature that will influence growth the Raspberry Pi will recognize that and turn on the heating unit of the greenhouse to return to optimal temperature. Using an IOT power relay, we can control outlet applications utilizing our Raspberry Pi. But as COVID-19 has removed the IOT devices from our project, these controlling aspects of the Pi are not implemented.

Figure 4: Schematic

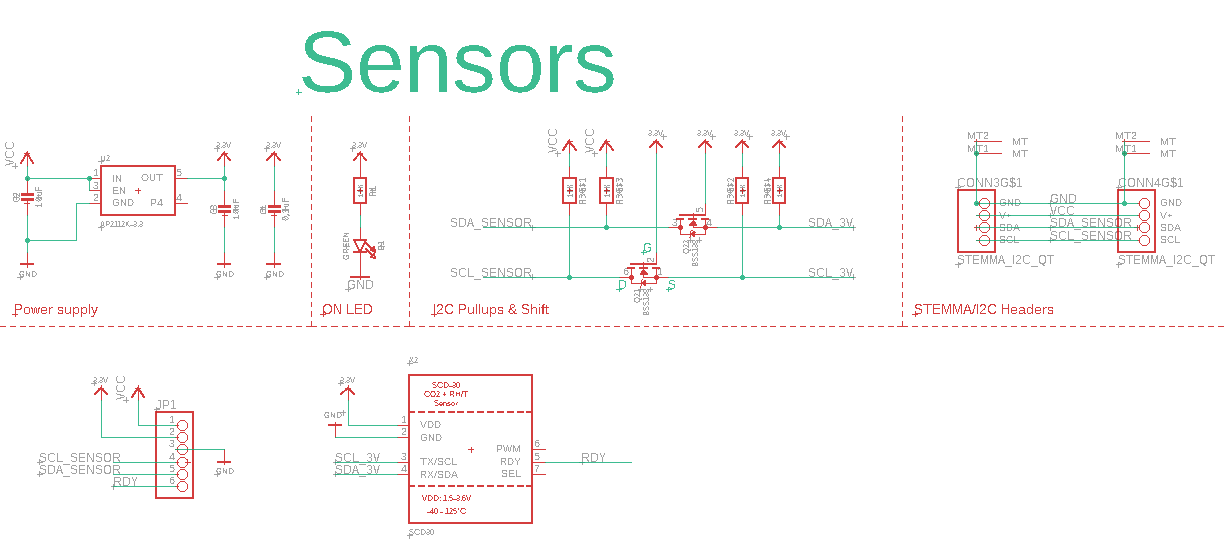
****

Figure 3: Schematic

Figure 2:Overall Block Diagram

For this project, the Raspbian OS went through an upgrade to a 64-bit system in order to support libraries that would be used for the software. Install the Adafruit SCD-30 NDIR sensor libraries on the Raspberry Pi in order to use the sensor's capabilities. Based on the provided example code found from the sensor’s datasheet, the Raspberry Pi can record and display the values that the SCD-30 sensor has recorded.

Diagram

Description automatically generated

In order to take pictures of our system, attach the Raspberry Pi camera module. To begin using the camera module, enable I2C in the Raspberry Pi configuration files. With these steps complete, the Raspberry Pi has the basic functionalities to act as a control system. Further elaborated later in this report is how the data communication process works, and how artificial intelligence works with the Raspberry Pi and its camera module.

## Artificial Intelligence

Artificial intelligence (AI) is used in this project, where the Raspberry Pi 4 Model B is used for object detection and related analysis, the Raspberry Pi Camera Module and the SCD-30-NDIR CO2 Temperature and Humidity Sensor are for environment data capture, and the Android application is used to visualize the outputs of AI analysis. In this project, AI is used to recognize the real-time number and size of the plants inside the greenhouse, analyze the real-time growth status of the plants, and adjust the optimal environment data according to the current growth stage.

The pipeline for our AI is divided into 3 stages. The first stage is to detect the plant inside the greenhouse. The second stage is to find the number and size of the plants. In the final stage, the growth stage of plants is determined. For the first stage of our AI, which is the object detection part, the Raspberry Pi takes pictures using the camera module on a fixed frequency. The pictures are saved and sent into our **pre-trained object detection mode**l. By using our model, the Raspberry pi can detect the plant in the picture with bounding boxes. The above-mentioned pre-trained model utilizes a widely used object detection architecture, YOLO v5 [5]. The detection structure provides reliable object detection performance. We use a small-scaled labeled dataset with around 300 labeled mushroom pictures to train the model set. As shown in **Figure 5**, the model converged successfully. After adjusting the batch size of model training to 4 and the number of epochs to 50, the accuracy is also satisfying. We train the model on a personal computer and then send the weight file to the Raspberry Pi so that the raspberry pi does not need to do the training. When choosing algorithms, we attempted both Hybrid Quantum-

Graphical user interface, chart

Description automatically generated

Figure 6: Example image of object detection output image

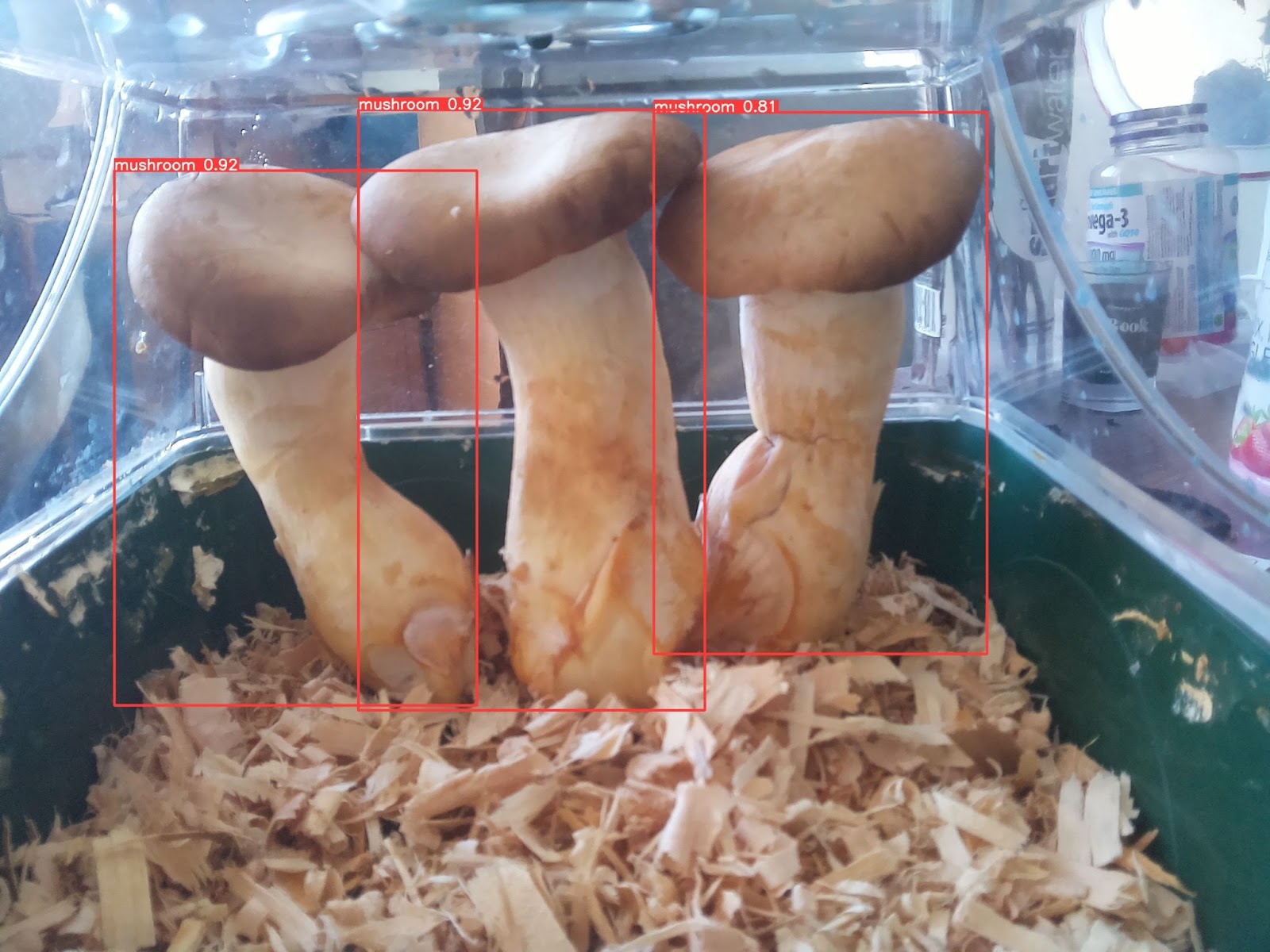
Figure 5:Object Detection training result.

Classical Neural Network and YOLO v5, as was required by the client. However, by comparison, the Hybrid Quantum-Classical Neural Network asks for working on remote quantum computers, which makes the design more complicated while not providing any speed improvement. For the above reason, we decide that the quantum method is not suitable for our small-scaled object detection project, and we decide to use YOLO v5. **Figure 6.** shows one of the pictures taken by the Raspberry Pi Camera Module and the corresponding output picture of our object detection model.

In the second stage of our AI, from the pictures detected in the first stage, the Raspberry Pi can determine the number of plants by simply counting the number of bounding boxes. As for the size, Since the distance between the camera and the plants is fixed, the real sizes of the plants are calculated by multiplying a selected factor by the sizes of the bounding boxes.

For the third stage of our AI, the determination is based on the size of the plants. We researched the best growth conditions of our test object, oyster mushrooms, and found the optimal environment for the mushrooms in different growth stages. With this knowledge in mind, the system can adjust the optimal environmental data based on the different growth stages of the plant. We separate the growth stage into the spawning, pinning, and fruiting stage based on the real size of the mushrooms. A text description of the plant is automatically generated based on the growth status. This information is to be sent to the application on the Android device to be made available to the user. The users can easily know how many plants are detected, what the growth stage is, and the optimal environment for this stage.

Figure 6: Example image of object detection output image



## Smartphone Application

Based on the requirements of our client, Dr. Hao Liang, our project develops an Android application using Android Studio to make it easier for our users to monitor and control this Mini greenhouse. For this application, we have four main sections, including Login & Register section, Monitoring & Controlling section, Information section, and Notification section. Java is used in this application. Users will need to connect the access point (AP) on our raspberry pi to get real-time connections. Though our greenhouse is mini-sized, and with a low amount of components, we decide to give it some privacy by developing a login page. It is because the mushroom growing environment in the greenhouse is flammable, where the planting base is usually wood chips. In this case, even the heat created by the Raspberry Pi can become dangerous. We design the first section: the login & register section, to protect the greenhouse from potentially unwanted actions from people other than our users. When a user opens our application for the first time, they are required to register an account by filling in the username and password, as well as agreeing to our user agreement, as shown in **Figure 7**. The username and password are stored in the database and can be used for future login. Once the user is registered and logged in, the user has access to the rest of our sections.

On the home page, as shown in **Figure 7**, which is the monitoring and controlling section, our user can find the output of the SCD-30-NDIR sensor, including the temperature, humidity, and concentration of CO2, which are the three main factors we looked at during the process of plant growth. The top three sliding bars are for the user to adjust the three factors manually. Users who want to grow based on their own can growing knowledge can adjust the bar to grow the plants in whatever way they want. The “Eco Mode” is for saving energy.

Figure 7:Smartphone Application

Graphical user interface, text, application, chat or text message

Description automatically generatedDiagram

Description automatically generatedGraphical user interface, text, website

Description automatically generatedGraphical user interface, text, application

Description automatically generated

However, in this mode, the environment inside the greenhouse depends more on the outside environment, because the potentially existing IoT devices inside the greenhouse would be barely on. In addition to the “Eco Mode”, we designed three more modes. The “Fast Mode” should make the temperature and humidity relatively high, which helps the plants to grow fast. The “Slow Mode” is the opposite of the fast mode, in which the temperature and humidity are relatively low. The “Stop Mode” is to shut down everything, including the Raspberry Pi, which may be used by users in extreme cases.

On the dashboard page, which is also the information section, as shown in **Figure 7,** users can find better visible information inside the greenhouse, including real-time pictures of the plants and a real-time AI-generated text description. Every time users click on the red ”I want to look at my plants” button, the app synchronizes a new real-time image of the plants taken by the camera inside the greenhouse. And by clicking the “tell me more about my plants” button, the Raspberry Pi analyzes the status of the plants and generates a description of their recent status, including their dimension and when to harvest, and shows the text description in the blue box.

On the third page, as shown in **Figure 7**, users can find the IP and port number used to connect with the Raspberry Pi, which is discussed further in the “Data Communication” section. By clicking the “Get IP Address button, the IP of the Raspberry Pi is received, and by clicking the “Start Connection” button, a socket communication is built between the Raspberry Pi and the App.

## Data communication

For data communication in our project, IP and port numbers are used to find the connection between the Raspberry Pi as the server and the Android application as the client. A static IP is set on our Raspberry Pi so that it shares the same IP address even after rebooting. Our socket communication uses Java on the Android device side and Python on the Raspberry Pi side. When the Raspberry Pi is started, a program runs to set the raspberry pi as a server. Our Raspberry Pi is set as an access point, and users can connect Android device to it through Wi-Fi. If future users want to add more than one Android devices to connect with our raspberry pi, they can simply keep the IP address unchanged and pick a new port number, which can be done on the notification page in our app.

# Future Directions

For our future directions our group is looking to expand its capabilities and reliabilities. Currently the greenhouse system is on a very miniature scale, our group is looking to expand the size of it to grow more potential plants. Since our AI system currently has its limitations. There are two main limitations. The first is that the system can only give an estimated number of plants. For example, in our design, due to budget limitations, we only have one camera on the left side of the greenhouse, and thus, if a plant on the right side is hidden by another one on the left side, it is likely that our systems object detection is unable to detect this. However, if this AI system is used in the industry or built into a product, we will consider using two or more cameras to take pictures from different angles, which should solve this problem. The second main limitation of our design is that we use mushrooms as the only test object, while real users may want to grow other plants. Our design can not only apply to mushrooms, vegetables, and flowers that grow in height. The design can also apply to some plants like tomatoes, which increase the size of the fruit as it grows. Because the algorithm can detect the part we want in bounding boxes, the system should also be able to detect the growth stage based on fruit size. Technically, applying this design to other plants can be solved by applying the corresponding plant datasets to train the model again so that the detection algorithm can also recognize other plants. Research on the other plants will also be required to provide the optimal environment for different growth stages.

For future plans on the Raspberry Pi, there have to be many improvements to improve our system. one limitation is the raspberry Pi's computational power due to its small compact system. If we expand the system to a larger system, we will have to switch to a stronger control unit. With the current operation, the Raspberry Pi is close to hitting its CPU limit and temperature, which will cause it to be a huge limitation for expanding the system.

# Conclusion

As per our requirements specifications report, our project satisfied the necessary requirements and achievements specified at the beginning of the report. The mini greenhouse that we will create for our capstone project will utilize mushroom as a test subject. As opposed to large modern industrial greenhouses that often have energy inefficiencies, our solution with our capstone project is to demonstrate an energy efficient smart management system that on a low-cost scale. Utilizing a sensor that can record CO2 density, temperature levels, and humidity levels and a Raspberry Pi the environmental conditions will be monitored constantly to provide details to a user through a smartphone application. Artificial intelligence techniques based on current mathematical models are applied to this project. Artificial intelligence in this system will determine the number and size of the current number of plants located inside the greenhouse. With the knowledge of the number and size, the AI will determine the plant’s growing stage and alter the greenhouse’s environment to the correct growing condition. This is all done by the intelligent monitoring system that was created utilizing a Raspberry Pi. Consumers can access all information and operate this greenhouse remotely using their mobile smartphones. The user-friendly mobile application that has been developed will informs consumers about plant status and greenhouse conditions. With the completion of all these tasks, any individual can replicate and build our capstone project’s mini greenhouse.

##### References

1. [1] Ahamed MS, Guo H, Tanino K. Energy saving techniques for reducing the heating cost of conventional greenhouses. Biosystems Engineering. 2019 Feb 1;178:9-33.K. Elissa, “Title of paper if known,” unpublished.
2. Agriculture and Agri-Food Canada, "Statistical Overview of the Canadian Ornamental Industry ± 2017," Dec. 05, 2018. [Accessed November 2020].
3. Deepa K, Rajendra CJ, Suresh DS, Smitha UV. Optimal Energy Management of Greenhouse by using ARM LPC1768.
4. “Hybrid quantum-classical Neural Networks with PyTorch and Qiskit,” *community.qiskit.org*.
5. Ultralytics. (n.d.). *Pytorch*. PyTorch. Retrieved April 8, 2022, from https://pytorch.org/hub/ultralytics\_yolov5/