Grow, Plant. Grow!

Ryah Carpenteer
Computer Science Department
George Washington University
Washington DC, United States
rcarpenter349@gwu.edu

Jonathan Nguyen
George Washington University
Washington DC, United States
Jonathannguyen@gwu.edu

***Abstract*—Using computer vision, soil sensors, and environmental sensors we were able to get accurate data of plant health with the hopes of building a device with the potential of expanding to larger scale agriculture. To accomplish this, we sourced the sensors we felt were relevant, and set up a system to gather data, process it, and make it available to the user in a usable format. We used the 5-layer IoT architecture to design our system, incorporating OpenCV, MQTT, and Teachable Machine. To accomplish the Edge computing necessary, we used OpenCV. While we were not able to accomplish this project completely how we envisioned, we have provided a viable plant monitoring device that can be scaled up to meet the large agricultural needs.**

Keywords—IoT, MQTT, OpenCV, plant health, sustainable agriculture, smart farming

I. Introduction

With the increase in popularity of sustainable farming and healthy food, people have turned to new innovations to help advance these goals. The heightened demand for urban farming is a result of the sustainability movement in agriculture and provides a way to combat the competitive prices of land and water while still providing the market with the fruits and vegetables it demands. Some innovators have implemented vertical gardens or farms in order to decrease the water required to sustain the plants. Whether it is traditional field farming, contemporary urban farming, or simply a home garden, this project offers a solution to the challenges of plant growing in any setting. We use computer vision along with environmental and soil sensors to monitor plants in a way that takes the responsibility off the user. We visualize the sensor data in a way that allows the user to act upon it and stay updated on the health of their plant(s) without being an expert in the field.

II. Background and Literature Review

Sustainable farming is only achievable by adopting sustainable practices for farmers. Modern practices of farming fall short in two major aspects: Soil degradation and Water scarcity. Soil degradation has grown to be an issue because of the overuse of fertilizer application. Nutrients found in soil such as nitrogen and phosphorus can grow past their absorption limit if not monitored According to a study by the United Nations, 33% of the world's soils are degraded due to unsustainable farming and at a rate of 12 million hectares each year [1]. Water scarcity is an alarming issue outside of farming, yet farming accounts for 70% of global freshwater use, and many of these water resources are becoming scarcer [1]. Efficient sustainable farming practices will be able to address issues of overuse of water for irrigation, inefficient irrigation practices, and the overuse of chemical fertilizers.

With the growing availability of IoT practices, it is possible to develop an improved intelligence of farming by giving farmers the capability to monitor and control aspects of their farm like moisture level and soil health. These systems are being adapted in farmlands with embedded sensors for moisture monitoring and soil monitoring, and with systems that will water automatically. The systems can also set alerts and alarm conditions according to humidity and temperature [2,3]. Moreover, these systems can be accessed through a smartphone application that allows farmers to monitor their farm's air temperature, control water, and fertilizers through the smartphone application at any time. Benefits of this type of smartphone application are that this can reduce labor and operation cost and promote the quality of production [3]. Other innovations in IoT involved automatic temperature control systems that use smart wireless which and temperature sensors that automatics adjust the temperature of a greenhouse by turning on sprinklers when temperatures are too high[3].

III. Methods used for IoT device implementation

This device receives data from five sensors that each monitor their respective aspect of the plant health and/or environment. It collects information about the temperature and humidity of the environment as well as information about the soil the plant(s) reside in. A camera also collects information about the height of the plant(s) in order to complete the comprehensive health monitoring system. The results of the data collected are displayed on ThingSpeak.com in a user-friendly manner, and the results of the soil moisture sensor are setup to alert the user through text if the soil reaches a level of moisture below a set threshold.

The five-layer IoT architecture model was used in the design and implementation of this project. The bottom layer is the sensor-layer, then the communication layer, the processing-layer, the information-layer, and finally the business-layer. This model was chosen to maximize the utility the IoT device can offer as well as improve its scalability.

Within the sensor-layer there is all the plant monitoring sensors. These include the camera, soil moisture sensor, NPK sensor collecting the nitrogen, phosphorous, and potassium levels in the soil, and Sense HAT providing the temperature and humidity data. In the communication-layer the MQTT protocol sends the data collected to ThingSpeak.com. The camera data gets processed in the processing-layer before all the data is displayed to the user in the information-layer. ThingSpeak.com is the application through which the user interfaces with all data, and the text to the user through Twilio.com provides a second aspect to this informative layer. The potential this device has for scalability is the business-layer including its possible use in traditional or contemporary farming practices.

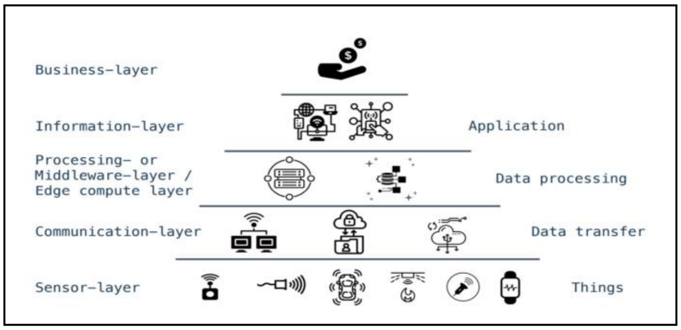
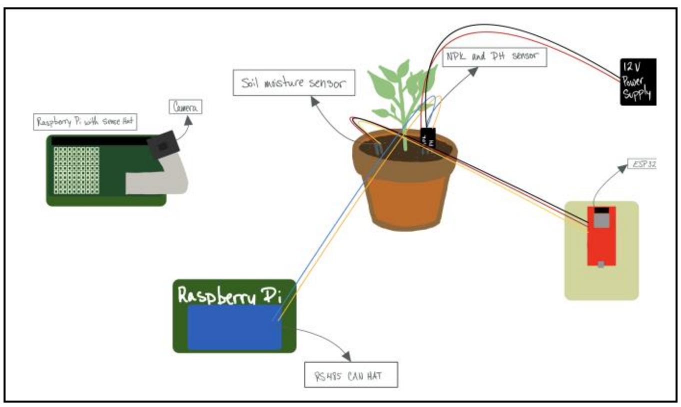


Image of 5 Layer model from CS3907 class lecture slides created by Professor Kartik Bulusu.



A. Sensors, Actuators, Microcontrollers and Single-board computer used

This device consists of a Raspberry Pi 3B+, ESP32 microcontroller (ESP32-WROOM), Pi NoIR Camera, Kuman Soil Moisture Sensor with comparator using a LM393 chip, NPK sensor (providing readings for nitrogen, phosphorous, and potassium levels), Waveshare RS485 CAN HAT, and a Raspberry Pi Sense HAT (used for temperature and humidity data collection).

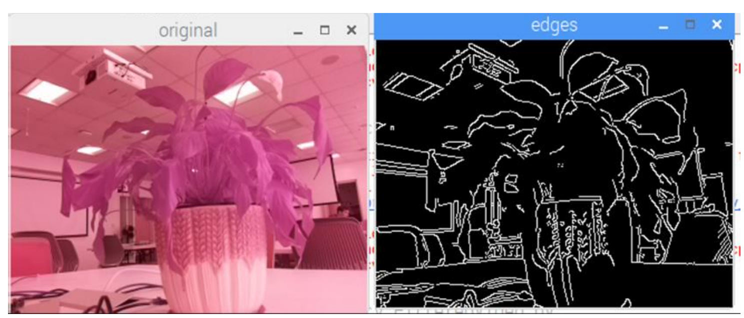
Device:	**Voltage range:**
Raspberry Pi 3B+	
ESP32-WROOM	
Pi NoIR Camera	
Kuman Soil Moisture Sensor	5V
NPK Sensor	12-24V
Sense HAT	3.3-5V
RS485 CAN HAT	3.3V

B. Firmware development using Python and relevent libraries

- Raspbian - RaspberryPi OS, Thonny IDE: used to developed python and micropython for the Pi.
- To determine height use libraries opencv, picamera to determine image edges and contours to estimate plant height.
- To read temperature and humidity used SenseHat libraries.
- To determine when the plant needs water – DigitalOutputDevice libraries
- To determine the NPK levels – UART and Modbus protocol were attempted.
- Sensors libraraies – cv: camera vision libraries that allow for computing edges and contours to determine plant height.
- ThingSpeak - used in code to create a MQTT server in which the Pi sends payloads of data info.

C. Edge computing section

For OpenCV, we used the libraries to determine an estimation of the height of the plant. In order to get the height , image had to separate the color or plant from the background. OpenCV have functions that get close gaps and detect edges by using Canny() . Canny() will take a gray scale image and draw the edges in white seen in figure below. To remove noise from the plant and foreground, we used cv2.dilate and cv2.erode.



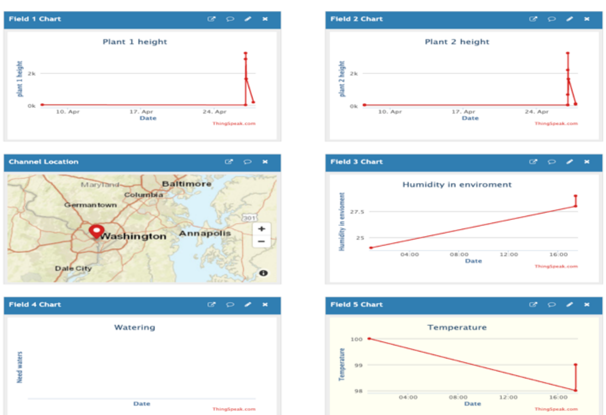
To get the height we can make contours around the image then from those decide which contour best fit the plant's description. To do this make a array of contours and use cv2.appoxPolyDP(). Below in the figure are the blue boxes that are the contours of the image. To differentiate the boxes to the plant height, the user will add an argument of the plant's width, then we'll create a boundary that no contour that is the plant is larger than the desired width.

Included in the data processing level in order to extract something meaningful from the camera data before submitting that information to the user.

IV. ISCUSSION OF YOUR IOT DEVICE AND RESULTS GENERATED USING THE DEVICE

For the soil moisture sensor, we ran tests to determine the appropriate moisture threshold below which the user is alerted to add water to the soil. As most plants dislike being constantly swamped in water, the threshold is at a relatively low moister level. We also performed many tests with OpenCV and Teachable Machine to calibrate the learning model used in our data processing. These tests then carried over when that set up was complete when we tested the implementation.

By gathering necessary data related to plant growth and environment, our device is able to provide a comprehensive report of plant health overtime. We achieved this by obtaining the data through relevant sensors, data processing, and presenting the results in an informative, user-friendly manner.



V. CONCLUSIONS, SUMMARY, FUTURE WORK

The aspects of this device that we successfully implemented provided a mostly comprehensive report of plant health for a user to view. It is clear from our work that this device is able to collect the necessary data and use it in a beneficial way that assists plant growers in every capacity.

Limitations of this project include not ideal sensor setups. Ideally this project can be run through one Raspberry Pi, however, in our implementation and with the sensors available to us two Pi's were required. This can be improved by upgrading the NPK sensor to one that is compatible with Raspberry Pi GPIO connections instead of RS485. Another downside to the current implementation is the limitations of the ESP32. Our team was unable to implement this aspect of the project given the unfamiliarity with the device. Furthermore, the low voltage available from this small device greatly restricts the sensor compatibility. Therefore, even with it up and running in the context of this project, it cannot support any of the sensors we had on its own. Even the small moisture sensor required power from the Pi. To improve the ESP32 limitations, we suggest the dedication to understanding its desired interaction and investing in the slightly more powerful versions of the microcontroller that exist. Limitation of knowledge of TensorFlow, didn't allow us to add a feature in which we trained a trainable machine to determine the identity of the plant.

Given the newness of this field and ongoing learning process we were not able to fully implement the device in the time allotted. However, we believe that the aspects we did achieve provide a strong foundation for this device in the field of plant health, and with more time and practice this device can be fully completed and perform the robust monitoring that we have described.

We believe this device has incredible potential for a scaled-up implementation outside of the classroom. The first step would be to address the limitations addressed earlier in this section. Then simply deploying multiple of these devices to farmers to set up in their soil and around their plants.

ACKNOWLEDGMENT

We would like to thank Professor Kartik Bulusu from the George Washington University Computer Science Department for all his help and inspiration in the ideation and implementation of this project.

REFERENCES

[1] *European Environment Agency. Agricultural Land: Nitrogen Balance. 2018. Available online: https://www.eea.europa.eu/airs/2018/natural-capital/agricultural-land-nitrogen-balance#tab-based-on-indicators (accessed on 26 March 2020).*

[2] IoT in Smart Agriculture - Mobiloitte Blog, 2018, [online] Available: http://www.mobiloitte.com/blog/iot-in-smartagriculture.

[3] B. Yimwadsana, P. Chanthapeth, C. Lertthanyaphan and A. Pornvechamnuay, "An IoT Controlled System for Plant Growth," *2018 Seventh ICT International Student Project Conference (ICT-ISPC)*, Nakhonpathom, Thailand, 2018, pp. 1-6, doi: 10.1109/ICT-ISPC.2018.8523886.