REAL-TIME DATA ANALYSIS AND VISUALIZATION OF SMART AQUAPONICS SYSTEM VIA CONVOLUTIONAL NEURAL NETWORK USING INTERNET-OF-THINGS TECHNOLOGY

A Project Study Presented to the Faculty of

Electronics Engineering Department

College of Engineering

Technological University of the Philippines

In partial fulfillment of the requirements for the Degree of

Bachelor of Science in Electronics Engineering

Submitted by:

Fernandez, Maria Jasmine B.

Macahis, Jeanella A.

Mesias, Ma. Shia Jeremy G.

Pinili, Ralfritz Mari C.

Ymson, Vennazelle P.

Advisers:

Engr. Edmon O. Fernandez

Engr. Jomer V. Catipon

August 2023

APPROVAL SHEET

This project study entitled "**REAL-TIME DATA ANALYSIS AND VISUALIZATION OF SMART AQUAPONICS SYSTEM THROUGH USING INTERNET-OF-THINGS TECHNOLOGY VIA CONVOLUTIONAL NEURAL NETWORK,**" has been prepared and submitted by the following proponents:

FERNANDEZ, Maria Jasmine B.
MACAHIS, Jeanella A.
MESIAS, Ma. Shia Jeremy G.
PINILI, Ralfritz Mari C.
YMSON, Vennazelle P

In partial fulfillment of the requirements for the **Degree of Bachelor of Science in Electronics Engineering** is hereby recommended for approval:

ENGR. EDMON O. FERNANDEZ
Project Adviser

ENGR. JOMER V. CATIPON
Project Co-Adviser

ENGR. JAY FEL C. QUIJANO
Panel Member

ENGR. NILO M. ARAGO
Panel Member

ENGR. TIMOTHY M. AMADO
Panel Member

ENGR. VILLAMOR M. AMON
Panel Member

ENGR. EDMUND G. MONILAR
Panel Member

Accepted and approved in partial fulfillment of the requirements for the Degree of Bachelor of Science in Electronics Engineering.

ENGR. TIMOTHY M. AMADO
Head, ECE Department

ENGR. NILO M. ARAGO
Dean, College of Engineering

ABSTRACT

The rise in the global population has led to an increasing challenge to meet the demand for food production. Traditional farming methods can be labor-intensive and require significant amounts of land and water required making it challenging to implement in urban or densely populated areas. Smart Aquaponics system provides an innovative and sustainable approach to address these issues while requiring less water, generates less waste, and can be implemented in a smaller area. The deployment of numerous sensors and actuators in the system has made it easier to analyze and monitor different parameters necessary in the cultivation. Specifically, this study focused on utilizing Internet of Things technology to deploy a real-time data dashboard using an IoT cloud platform responsible for storing and synchronizing the data in real-time. The web application allowed the end-users to control, monitor, and modify the system remotely. Furthermore, to improve the effectiveness and efficiency of monitoring the system, machine learning such as Convolutional Neural Network was utilized for plant disease monitoring, canopy area detection, and fish length detection. By incorporating different approaches of new technologies in the system, it showed a positive effect on monitoring which results in better production. The result of achieving the objectives of this study provided a reliable and accurate tool for users to monitor their systems and optimize their yields through a data-driven approach to aquaponics, which led to increased production, reduced costs, and improved sustainability.

ACKNOWLEGDEMENT

The proponents of this study extend their deepest gratitude and appreciation to those who have been part of this long journey.

First and foremost, we express our gratitude to our research advisers, Engr. Edmon O. Fernandez and Engr. Jomer V. Catipon, who provided insight and expertise that greatly assisted us throughout the research process. Their lines of communication were always open for inquiries and consistently provided guidance to steer us in the right direction.

We are also thankful to the faculty members of the ECE department for their constant feedback and recommendations, which contributed to the improvement and success of this study. Most specially to our Capstone Adviser and College Dean, Engr. Nilo M. Arago, for his dedicated efforts in sharing his knowledge and expertise on the research subject.

We owe a significant debt of gratitude to our parents, who have been our biggest source of strength, for their unwavering support and continuous encouragement throughout our academic journey. We sincerely thank them for expecting nothing less than excellence from us.

Furthermore, we would like to express our appreciation to our friends and newfound friends in our deployment site in Kalayaan, Laguna, particularly to Josh Campomanes and his family, for their hospitality and allowing us to stay for one and a half years.

Lastly, we are profoundly grateful to God Almighty for granting us the strength, knowledge, and opportunities to undertake this study and persevere until its satisfactory completion. Without His blessings, this achievement would not have been possible.

TABLE OF CONTENTS

APPROVAL SHEET ii

ABSTRACT ii

ACKNOWLEGDEMENT iii

LIST OF FIGURES ix

LIST OF TABLES x

Chapter 1: THE PROBLEM AND ITS SETTING

1.1. Introduction 11

1.2. Background of the Study 14

1.3. Research Gap 17

1.4. Research Objectives 17

1.5. Significance of the Study 18

1.6. Scope and Limitations 20

1.7. Definition of Terms 20

Chapter 2: RELATED LITERATURE

2.1. Applications of Internet of Things (IoT) 22

2.1.1. Industrial Internet of Things 23

2.1.2. Internet of Medical Things 24

2.1.3. Smart cities and homes 24

2.1.4. Smart Agriculture 25

2.2. Internet of Things (IoT) Layers 26

2.2.1. Perception Layer 26

2.2.2. Network Layer 27

2.2.3. Application Layer 27

2.3. IoT Communication Modules in Smart Aquaponic System 28

2.3.1. Wi-Fi 28

2.3.2. Raspberry Pi 4 30

2.4. Image Processing 32

2.4.1. Image Processing in Plant Monitoring 32

2.4.2. Image Processing in Fish Monitoring 33

2.5. Convolutional Neural Networks 34

2.5.1. YOLOv4 35

2.5.2. Mask R-CNN 36

2.5.3. VGGNet 37

2.5.4. GoogleNet 38

2.6. Data dashboard components 39

2.6.1. User Interface Design 40

2.6.2. Remote Control Application 41

2.6.3. Remote Monitoring Interfaces 42

Chapter 3: METHODOLOGY

3.1. Research Design... 43

3.1.1. Block Diagram 44

3.2. Research Process Flow. 45

3.3. Development of IoT-based Architecture for Smart Aquaponics System 45

3.3.1. Materials and Equipment... 46

3.3.2. Establishing the IoT Connectivity 47

3.4. Development of Plant and Fish monitoring system via Convolutional Neural Network.. 48

3.4.1. Preparation of Datasets 48

3.4.1.1. Bounding Box... 49

3.4.1.2. Polygon Annotation 49

3.4.2. Implementing the different CNN models 49

3.4.2.1. Plant and Fish Growth Monitoring 50

3.4.2.2. Plant Disease Detection 50

3.4.2.3. Plant and Fish Harvestability.. 50

3.4.3. Calibration and Testing 51

3.5. Development of Web Application for Smart Aquaponics System 51

3.5.1. Front-end development 52

3.5.2. Back-end development ... 57

3.5.3. Integrating the algorithm to the web application. ... 58

3.6. Server Testing and User Acceptance ... 58

3.7. Statistical Analysis ... 60

Chapter 4: DATA AND RESULTS

4.1. Convolutional Neural Network Testing ... 61

4.1.1. Object Detection mAP Graph ... 61

4.1.2. Detection Testing ... 65

4.2. Web Camera Computation ... 67

4.2.1. Lettuce Measurements ... 68

4.2.2. Tilapia Measurements ... 69

4.3. Comparison Between Manual and Smart Measurement ... 71

4.3.1. Lettuce Measurements ... 71

4.3.2. Tilapia Measurements ... 71

4.4. Project Evaluation by end-users ... 72

4.4.1. Functionality and Efficiency ... 72

4.4.2. Usability and Aesthetics ... 73

4.4.3. Portability and Maintainability ... 73

4.4.4. Reliability ... 74

Chapter 5: CONCLUSION AND RECOMMENDATION

5.1. Summary of Findings 75

5.2. Conclusion 76

5.3. Recommendations 78

References 79

ANNEXES

ANNEX I: YOLOv4 Detection Codes 88

ANNEX II: Web Development Codes 97

ANNEX III: Camera Calibration Codes 116

ANNEX IV: User’s Manual 129

ANNEX V: Documentation 134

ANNEX VI: Thesis Defense 139

ANNEX VII: Student’s Profile 142

LIST OF FIGURES

.

Figure 1. Block Diagram 44

Figure 2. Research Process Flow 45

Figure 3. Internet-of-Things connectivity 48

Figure 4. Overall Framework of the System 51

Figure 5. Web Application flowchart 52

Figure 6. Log in Page 53

Figure 7. Sign up Page 53

Figure 8. Home Page 54

Figure 9. About Page 55

Figure 10. Contact Page 56

Figure 11. Firebase's Realtime Database Environment 57

Figure 12 (a) (b). Average Loss and mAP Graph for Romaine Lettuce 62

Figure 13. (a) (b) Average Loss and mAP Graph for Infected Lettuce 63

Figure 14. (a) (b) Average Loss and mAP Graph for Nile Tilapia 64

Figure 15. Detection Testing for Romaine Lettuce 65

Figure 16. Detection Testing for Infected Lettuce 66

Figure 17. Detection Testing for Nile Tilapia 66

Figure 18. Graphical Data for Lettuce Measurements 69

Figure 19. Graphical Data for Tilapia Measurements 70

LIST OF TABLES

Table 1. Application of Internet of Things 26

Table 2. Internet of Things (IoT) Layers 28

Table 3. IoT Communication Modules in Smart Aquaponic System 31

Table 4. Image Processing 34

Table 5. The Input-Process-Output of the system 43

Table 6. Ideal Measurements of Lettuce and Tilapia 50

Table 7. Survey Questionnaire 59

Table 8. Tabulated Data for Manual Computation (Lettuce) 68

Table 9. Tabulated Data for Smart Computation (Lettuce) 68

Table 10. Tabulated Data for Manual Computation (Tilapia) 69

Table 11. Tabulated Data for Smart Computation (Tilapia) 70

Table 12. Statistical Analysis Between Manual and Smart Measurement of Lettuce 71

Table 13. Statistical Analysis Between Manual and Smart Measurement of Nile Tilapia 71

Table 14. Results of Functionality and Efficiency 72

Table 15. Results of Usability and Aesthetics 73

Table 16. Results of Portability and Maintainability 73

Table 17. Results of Reliability 74

Chapter 1

THE PROBLEM AND ITS SETTING

This chapter presents a brief introduction, background of the study, the objectives that the researchers aim to fulfil, the research gap, the study's scope and limitations, and its significance.

1.1. Introduction

As the world continues to progress, rapid urban and rural development, increasing population and heightened economic status are beyond subjugation which has resulted in the increase in waste generation worldwide. Every person holds accountability upon producing massive waste and thus, a contributor to the issue. One of the major problems in the world today is the increase of solid waste mainly attributed to population growth, which is causing deterioration of the environment.

According to the World Bank, the global human population generates an enormous amount of waste, estimated at approximately 2 billion tons or about 4.5 trillion pounds per year. On average, each person produces about 0.74 kilograms of waste daily, with individual figures varying between 0.11 and 4.54 kilograms [1]. Although accounting for only 16 percent of the global population, high-income countries contribute approximately 34 percent of the world's total waste, equivalent to 683 million tons. In contrast, the East Asia and Pacific region is responsible for nearly a quarter, amounting to 23 percent, of the overall waste generated. Projections indicate that by 2050, garbage production in Sub-Saharan Africa is expected to more than triple from its present levels, while waste generation in South Asia is anticipated to more than double [2]. In this regard, the United States is responsible for most of the world's waste, producing roughly 250 million tons each year, or around 4.4 pounds of trash per person every day. Apart from the United States, China, Brazil, Japan, and Germany are also prominent as the world's largest waste generators [3]. In developing countries, particularly among urban communities facing poverty, the adverse impacts of inadequate waste management are more severe compared to those in developed nations. In low-income countries, it is prevalent for over 90% of waste to be disposed of in unregulated landfills or openly burned, without proper regulation or control measures in place [4]. Countries' trash cans continue to fill up ceaselessly as they leap into the spiral of urbanization and urge industrialization to become more affluent. The constant expansion in the volume of solid waste generated per day is worsened, making it increasingly difficult to find dumping sites.

In the Philippines, waste generation surged from 10.6 million tons in 2012 to a predicted doubling rate by 2025 [5], demanding the implementation of expanded disposal facilities. According to official government data, the daily waste production in the Philippines amounts to approximately 43,684 tons, with around 4,609 tons specifically consisting of plastic waste [6]. Solid waste management is still a big issue in the country, particularly in urban areas such as Metro Manila. The solid waste management in the country faces challenges related to improper waste disposal, insufficient waste collection systems, and a shortage of disposal facilities. According to the Solid Waste Management Division (SWMD) data, the residents of cities collectively generate approximately 10,000 tons of garbage daily [6]. According to the provided data, Quezon City stands out as the top producer of garbage, generating approximately 3,600 tons per day. Following closely behind are Manila and Caloocan, producing approximately 1,200 and 913 tons per day, respectively [7]. The excessive amount of garbage generated by these cities largely affects the cities' congested landfills and dumpsites, especially during natural disasters, which frequently result in waste dump landslides. Landslides of garbage dumps have destroyed homes beneath loads of waste in recent years and it is often the poorest who reside close to landfills that suffer.

Quezon City is the largest of the nine cities and eight municipalities that make up Metro Manila, with a total area of 15,106 hectares and a population of 2.3 million people [8]. The city is accountable for a quarter of the estimated daily trash produced in Metro Manila. For nearly 30 years, the Payatas dumpsite in Quezon City has been collecting Metro Manila's municipal solid waste. It covers 13 hectares and has become the Philippines' largest open landfill. This rubbish dumpsite was infamously known for its tragic trash slide last 10th of July 2000, which killed roughly 200 residents that resided near the area [9]. The site was taken over by the Quezon City government in 2001 with the purpose of reshaping it into a controlled dump per the approval of Republic Act 9003 or Ecological Solid Waste Management of 2000.

The city has a plan to transform the Payatas open dumpsite into a regulated waste disposal facility by implementing innovative strategies. One of these initiatives is the establishment of a smart greenhouse system, which aims to revitalize the area and turn it into an enhanced and rehabilitated space. Among the pioneering projects is the Smart Aquaponics System. The City Government collaborated with government organizations like the Department of Science and Technology (DOST) and the

Technological University of the Philippines (TUP) to develop and execute the project. The DOST has allocated a budget of PHP2.49 million to support the establishment of a smart greenhouse system at the Payatas Controlled Disposal Facility (PCDF) [10]. Under their partnership, they intend to promote the use of aquaponics technology, a system that combines the concepts of aquaculture with hydroponics which allows fish and vegetables to grow in an integrated environment.

Over the years, aquaponics is becoming popular because of its natural and recirculating system. It is a combination of hydroponics and aquaculture which means cultivation of plants and fish farming. A cycle of nutrient-rich effluent waste of fish coming from the fish tanks will be used to fertilize the plants in the production bed. This is a good combination of two systems that benefit each other while also taking proper care of waste disposal. Aquaponics systems are one of the great alternatives in culture farming, where this can be effective in conservation of water because of its recirculating system between the fish tank and plant bed. This technology can also produce high-quality vegetables considerably faster than standard plant farming techniques. It is one of the most efficient ways of meeting the growing need for food, and it may also aid in environmental rehabilitation.

1.2. Background of the Study

Aquaponics is a system where it can produce vegetables and fish production in a recirculating cycle and this can be a very good alternative for basic food production. To enhance the production process, different techniques of monitoring and controlling systems were used to effectively manipulate and make the process much faster and efficient. One of the examples shows that developing a website application in the system can be very effective in monitoring and observing processes. By this, numerous studies around the world developed new technologies.

Majority of the studies conducted involved the creation of a web page that enables users to monitor sensors and detect sudden changes in plant cultivation parameters. This web page offers real-time data visualization and a user interface that presents data through graphs. The system described in the study of Kyaw and Ng is divided into two sections. The first section includes an Arduino Mega, a Grove-Mega shield, and a relay board. The Arduino Mega, equipped with 54 input/output pins, facilitates communication with the sensors and actuators. The second section consists of a Raspberry Pi 3 model B and a camera module [11]. In this system, the web application displays and records sensor values, allowing for remote monitoring and control of the actuators. Users can also modify threshold values for each sensor through the web application. Additionally, a mobile application version of this system was developed specifically for the Android platform.

In the architecture described in the study of Khaoula et al., the central control system utilizes NodeMCU, which is equipped with a Wi-Fi interface for internet connectivity. The proposed system consists of four layers: the Physical layer, Gateway layer, Middleware layer, and Application layer [12]. The software development involves establishing communication from the gateway to the application layer. The NodeMCU microcontroller is responsible for connecting devices in the physical layer, collecting data from sensors, and sending control signals to the relay. The collected sensor data is periodically transmitted to the cloud using the MQTT protocol. To ensure secure connectivity to the cloud, a client certificate is employed, providing authentication and encryption. The final layer of the architecture focuses on the user interface, which subscribes to MQTT topics to retrieve sensor data and alarm information from the MQTT broker. It also offers monitoring and analytics capabilities to the user.

In an application named Aquadroid, although smartphones were used to connect, control, and monitor the system, the project used an Arduino Yun to serve as their microcontroller. In creating a design for the web application, the MIT App inventor is used to develop an Android application that will interact between the sensors to monitor pH and temperature level to act accordingly to the abnormalities and sudden changes to the parameters [13].

Another study used a Packetduino and was LoRaWAN-enabled where it is connected to a gateway that will provide a long scale of data transmission within a 15km range [14]. Using Android Studio, a mobile application shows real-time water parameter graph readings through LoRaWAN. All inadequacies of the relevant research were considered and used to advance the Water Monitoring and Correction System [15].

In other studies for fish cultivation, the proponents used PHP codes to transmit data to the user interface that is connected to the system server. HTML, CSS, and Javascript were used for the graphical interface design of their web application. The output on the UI were the Weight Prediction Tab, Water Quality Monitoring, and Data Compilation (where the graph was displayed) [16]. This study shows the convenient use of IoT in monitoring and controlling processes using web applications.

1.3. Research Gap

In Smart Aquaponics System studies, a graphical user interface was added to provide a data visualization for the system. For most instances, the web application was only added for viewing and updates. This project is mainly focused on developing an interactive application which will include the control for the actuators and smart monitoring via image processing to provide a data-driven visualization. Further, studies cited above only focus on either the optimal growth of plants or fish alone. None of the web applications monitored the data sets from both cultures.

1.4. Research Objectives

This study aimed to develop a data-driven graphical user interface of the Smart Aquaponic System for real-time data analysis and visualization of the status of crops and fishes, this project aims to:

1. To create a program that will collect the datasets from the sensors through Internet of Things (IoT) technology.
2. To develop a plant and fish growth monitoring system, harvest prediction, and disease identification via Convolutional Neural Network.
3. To design a web application which consists of all visual aspects and cloud server of the data dashboard for the IoT-based aquaponics set-up.
4. To test and deploy the web application of the aquaponic system.

1.5. Significance of the Study

This study will determine the utilization of advanced technologies and the Internet of Things (IoT) to develop an IoT-based integrated sensor module for monitoring smart aquaponics system parameters. It will contribute to the recovery and rehabilitation of Payatas and help improve household and community livelihoods.

The combination of plant production and fish farming in closed aquaponic systems significantly reduces the environmental impact. It can run with minimal waste; hence, they have no noticeable effects on the soil, provided no additional land is required for installation. However, aquaponics can serve purposes other than food production in urban environments. For instance, it can be used as an educational tool in schools, for interior greening (giving a better climate in public buildings and residences), and as a unit in social institutions. This method can play an essential part in smart cities' future environmental and socio-economic sustainability. Moreover, these cities plan to transform all knowledge hubs and information and communication technologies (ICT) into innovation poles to boost socio-economic development. Aquaponics is crucial in enabling local, pesticide-free, nutritious, and fresh produce with short supply chains in urban areas.

By this means, aquaponics addresses various sustainable development goals. First is Goal 2: End hunger, guarantee food security, enhance nutrition, and promote sustainable agriculture. Aquaponics may grow food efficiently and create a respectable income. Additionally, it encourages rural development and environmental conservation. It also addresses Goal six (6): Ensure that everyone has access to water and sanitation and offer sustainable water resource management. Lastly, sustainable water management is made possible by reusing and purifying water to be safe for fish and plants. Goal nine (9): Construct a resilient infrastructure, create sustainable industrialization that benefits everyone, and foster innovation. The 1H203 package delivers an infrastructure that is basic and modelable. Furthermore, it is constructed entirely of renewable, recyclable materials. Goal twelve (12): Develop sustainable consumption and manufacturing habits. Aquaponics does permit sustainable fish and vegetable consumption habits. It allows for sustainable fish and vegetable consumption habits. Goal fourteen (14): To sustainably conserve and develop the oceans, seas, and marine resources for sustainable development. The installation of aquaponics demonstrates the possibility of symbiosis between aquaculture and market gardening. Goal fifteen (15): Preserve and restore terrestrial ecosystems while guaranteeing their sustainable utilization. It is preventing and reversing land degradation and stopping biodiversity loss. Aquaponics is also a method for fish conservation.

This study will also contribute to the Harmonized National Research and Development, particularly in Section III: Agriculture, Aquatic and Natural Resources (AANR), Aquatic R&D Agenda, Item Seven: “Mechanization and automated systems for feeding, water and culture management, and post-production” In addition, it will serve as a guide for future researchers in Electronics and Communications Engineering. That will guide further research development connected to the Aquaponics system IoT-based integrated sensor module for monitoring smart aquaponics system parameters.

1.6. Scope and Limitations

This project will be deployed in Brgy. Payatas, Quezon City where it will be funded by the Department of Science and Technology (DOST). The researchers proposed one (1) model of the smart aquaponics system and one (1) IoT-based website-application for monitoring and controlling the system. Since it is a facility-based system, the study is limited to creating a web application what will be connected through a monitor for public data viewing.

This study emerged to generate a smart aquaponics system with the utilization of advanced technologies and the Internet of Things (IoT) that will develop an IoT-based integrated sensor module for monitoring smart aquaponics system parameters. This study is Also limited to monitoring lettuce plants (Lactuca sativa) and nile tilapia (Oreochromisniloticus). The system is also designed to monitor and automate light intensity, air and humidity, pH value, temperature, dissolved oxygen, and water temperature.

1.7. Definition of Terms

Smart Aquaponic System. It is an aquaponics, a combination of cultivating plants and fish farming with an addition of a system that will process real time data analysis and visualization of the status and crops and fishes.

Image Processing. Refers to the process of performing operations such as analyzing, enhancing and reconstructing images of the plants and fishes that were captured by the camera for the extraction of useful information from it.

Internet of Things (IoT). It is a device used for acquiring information without the need of human intervention. Monitoring smart aquaponics system parameters will be done with the help of IoT-based integrated sensors.

IoT Communication Modules. Bluetooth, Wi-Fi, and Zigbee; These are the wireless technologies that made the Internet of Things a reality. Through these modules, device-to-device communication became feasible and they will be used for remotely accessing and controlling the smart aquaponics system.

User Interface Design. A web application to monitor and visualize sensors and changes among the parameters in the smart aquaponics system. Real time data of the status of crops and fishes will be collected using an interactive application via image processing.

Remote Control Applications. Refers to the modules used in the collection of sensor data and automating system operations. It provides information about the sensors and then sent for interpretation of data gathered.

Remote Monitoring Interfaces. Data gathered from the sensors and microcontrollers were sent here to be monitored and interpreted. This will keep the system functioning in real time.

Chapter 2

REVIEW OF RELATED LITERATURE

` This chapter presents the related literature and studies the researchers used.

2.1. Applications of Internet of Things (IoT)

The Internet of Things (IoT) technology encompasses a wide range of applications; thus, its usage is expanding rapidly. Dr. P. Muthumari et al. describe the Internet of Things as an organization of interconnected, web-associated objects capable of acquiring and communicating information throughout a remote organization without the need for human intervention. IoT devices find utility in monitoring and controlling mechanical, electrical, and electronic systems within a wide range of structures, including public and individual, modern or residential establishments. This encompasses in-home computerization and building robotization or automation frameworks [17]. In basic terms, IoT technology can be described as the interlinking of individuals, computers, and objects [18]. The term "Internet of Things" was initially introduced by Kevin Ashton, the executive director of the Auto-ID Centre at MIT, in 1999. Since then, extensive endeavors have been undertaken in the past few decades to expand the notion, encompassing not only the development of infrastructure and technology but also addressing the social, legal, and ethical implications associated with it [19] [20].

Since the internet's early beginnings in 1989, there has been an existence of connecting "Things" on the internet. The Trojan Room coffee pot is likely the first of its kind [21]. In 1990, John Romkey pioneered the creation of the initial

Internet-connected "device," which allowed for remotely controlling the operation of a toaster using the Internet [22]. On the other hand, in 1994, Steve Mann innovated the WearCam, a cutting-edge technology that harnessed the capabilities of a 64-processor machine to achieve remarkable nearly real-time performance [18]. A decade later, a significant milestone in smart device applications was reached with the introduction of a large-scale implementation of an item identification system utilizing Radio Frequency Identification (RFID) technology [23]. These are some Internet of Things applications that are widely utilized in different fields: Industrial Internet of Things, Internet of Medical Things, Smart cities and homes, Smart Grids and Smart Metering, and Smart Agriculture.

2.1.1. Industrial Internet of Things

According to the definition provided by the Congressional Research Service, the Industrial Internet of Things (IIoT) facilitates communication and information sharing among interconnected machines within a manufacturing facility, with the objective of improving efficiency, productivity, and overall performance [24]. The machines within the Industrial Internet of Things (IIoT) framework possess the ability to track, monitor, and distribute workloads among themselves. They can also detect signs of wear and tear to prevent failures and maintain uninterrupted production [25]. IoT applications span a wide range, encompassing tasks such as corrosion detection within refinery pipes and the generation of real-time production data [24].

2.1.2. Internet of Medical Things

The integration of IoT technology within the healthcare sector, known as the Internet of Medical Things (IoMT), has facilitated the capturing and transmission of patient health data from devices like heart monitors and pacemakers to healthcare providers. This data is sent across multiple networks for monitoring, analysis, and remote configuration. Furthermore, wearable IoT devices track physical activity, vital statistics, and sleep patterns [24]. IoMT's primary role is to ensure the accessibility of information.

2.1.3. Smart cities and homes

Smart Cities represent a category of applications that exert a notable influence on society. With the global urban population experiencing rapid growth, cities are witnessing economic expansion. Within the "smart city" domain, IoT devices and systems play a pivotal role in sectors such as utilities, transportation, and infrastructure. By employing the Internet of Things, utilities can create smart grids and meters for electricity, water, and gas. Sensors collect and share customer usage data, enabling the central control system to optimize real-time production and distribution to meet demand [24]. Furthermore, enhancing waste disposal can be achieved through the installation of sensors in garbage bins. This approach eliminates the need for inefficiently collecting every dumpster, as only filled containers will be assessed for collection [26].

The range of IoT devices commonly found in households and buildings, including smart appliances, smart TVs, smart thermostats, network-connected light bulbs, outlets, door locks, and home security systems, falls under the "smart home" category. These smart home IoT devices can be connected to a unified network and conveniently operated remotely through the internet using a smartphone or computer [24].

2.1.4. Smart Agriculture

The agricultural sector plays a crucial role in the world as it involves the production of food and beverages through the cultivation of various crops. However, due to factors like population growth, resource depletion, pollution, and a shortage of human labor, managing the agriculture industry has become increasingly challenging. Leveraging automation presents a viable solution to enhance the efficiency of the agricultural industry. Implementing IoT infrastructure can prove advantageous in various aspects such as crop status tracking and livestock management. To monitor the status of crops, a combination of climatic sensors, water or moisture level sensors, chemical concentration or acidity sensors, and visual sensors can be utilized. Additionally, automated systems for dispensing water and fertilizer are implemented within the boundaries of the plantation. Implementing such an intelligent system offers various benefits, including the optimization of fertilizer and water usage, as well as mitigating the impact of climate-related challenges on crop yield [29].

__Table 1__. Application of Internet of Things

AUTHOR	YEAR	TITLE	RELEVANT FINDINGS	RELATIONSHIP TO THE STUDY
A. Abdul Qawy, E. Magesh, and S. Tadisetty,	2015	The Internet of Things (IoT): An Overview	The definition of the IoT and its relevance of it in the modern world.	Both studies used the IoT
E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, and B. Stiller	2022	Landscape of IoT security	How the monitoring system defines and works	Both studies have real-time monitoring system
L. Horwitz	2020	Internet of Things (IOT) products & solutions	How the IoT system helps in waste disposal	Both studies focused on improving the use of the IoT on the sensors
A. D. Jurcut, P. Ranaweera, and L. Xu	2019	A state-of-the-art review of the Internet of Things (IoT) History, Technology and fields of deployment	Insights regarding the aquaponics system in terms of the functionality in monitoring the fish and plants	Both studies are smart farming which is IoT operated

2.2. Internet of Things (IoT) Layers

The architecture of the Internet of Things (IoT) can be conceptualized as a layered structure. The fundamental three-layered architecture comprises the perception layer, network layer, and application layer [30].

2.2.1. Perception Layer

The first layer is the perception layer, which gathers data from the physical world. Its primary role involves converting data into signals that can be transmitted through networks and interpreted by applications [31]. The physical layer comprises sensors that detect and gather information about the surrounding environment. These sensors are utilized to detect specific physical parameters or identify other intelligent objects within the environment [30].

2.2.2. Network Layer

The network layer serves as the hub for data transfer, collecting the data generated by various devices and facilitating its transmission and processing [31]. It enables communication between these devices, as well as with other smart objects, servers, and network devices. Additionally, it oversees the transmission and processing of sensor data [30].

2.2.3. Application Layer

The application layer serves as the interface where data is utilized and users interact with it. Unlike the other layers of the IoT architecture, the application layer focuses on leveraging the data collected through IoT for end-user purposes, rather than contributing to the development of the architecture itself [31]. The responsibility of the application layer lies in providing the user with application-specific services [30].

Table 2. *Internet of Things (IoT) Layers*

AUTHOR	YEAR	TITLE	RELEVANT FINDINGS	RELATIONSHIP TO THE STUDY
P. Sethi and S. R. Sarangi	2017	Internet of Things: Architectures, Protocols, and Applications	How the IoT will be implemented and how the data will be collected by layer	Both studies used a layer that helps on with collections
M. Farhan, M.Pu'ad, et al	2019	IoT based water quality monitoring system for aquaponics	How the IoT will be implemented and how the data will be collected by layer	Both studies used a layer that helps on with collections

2.3. IoT Communication Modules in Smart Aquaponic System

The agricultural process has been significantly impacted by the emergence of the Internet of Things (IoT), specifically through the adoption and implementation of hydroponic and aquaponic systems [32]. The implementation of Internet of Things (IoT) communication modules in smart aquaponic systems allows for enhanced monitoring, control, and optimization of the system. These communication modules serve as the backbone of connectivity, facilitating seamless data exchange between various components of the aquaponic system and the central control unit.

2.3.1. Wi-Fi

IEEE 802.11, commonly referred to as "Wi-Fi," is a globally accepted wireless media utilized for transmitting and receiving various types of data, signals, and instructions. This wireless technology operates within the frequency range of 2.4GHz to 60GHz. It provides data rates ranging from 1Mb/s to 54Mb/s, with potential speeds reaching up to 6.75Gb/s. [18]. Nowadays, Wi-Fi is a standard feature found in contemporary devices such as cellphones, PDAs, tablets, computers, and more. The range and speed of Wi-Fi connections are contingent upon the transmitter's capabilities. Various IEEE 802.11 standards, collectively referred to as Wi-Fi standards, have been proposed to cater to different networking applications [18].

Wi-Fi devices have grown in popularity over time due to their easy installation and maintenance. Wi-Fi networks are now prevalent in public spaces, schools, universities, hospitals, and more. Such a Wi-Fi reach would be an extra benefit for the Internet of Things, which requires its network to be ubiquitous. Implementing such a situation utilizing regular wired infrastructure is laborious. Wi-Fi devices might be installed in an isolated region, regardless of topographical or environmental restrictions. This is a limitation concerning wired media or RFID since additional devices need the installation of new equipment. However, it is straightforward to add extra devices without installing new hardware in a Wi-Fi setting. This promising characteristic of Wi-Fi makes its application in IoT situations feasible [18].

Wi-Fi technology operates within the unlicensed 2.4GHz band spectrum, which is known for its high density of other devices. The coverage range of Wi-Fi equipment is influenced by the surrounding environment, typically reaching around 45m indoors and approximately

90m outdoors. To extend the range, additional repeaters and access points need to be deployed. Considering these aspects, Wi-Fi plays a pivotal role in IoT design, given its installation and maintenance requirements. However, Wi-Fi also introduces considerations related to security, reliability, and interoperability [18].

2.3.2. Raspberry Pi 4

The study of Farhan et al. shared that their design was using Raspberry Pi as the system's central processing unit to a cloud service for data storage and display and utilizing Wi-Fi to send data [31]. Due to its fast processor and built-in Bluetooth and Wi-Fi components, Raspberry Pi was set up as the system's main controller [11]. By incorporating Wi-Fi functionality, the Raspberry Pi enabled seamless connectivity to the internet, allowing the system to establish a connection with a data server for storing the system's settings values [34].

Additionally, the Raspberry Pi is equipped with a high-definition multimedia interface connection, which can be utilized to connect a Camera Module for live streaming on visual display devices. Once the data is collected from the sensor reader, it is stored and processed either in the Raspberry Pi itself or in a web server database, enabling its presentation on a live streaming monitoring website [34]. Furthermore, it was utilized for the creation and hosting of the web application, which functions as the graphical user interface (GUI) for the system. The GUI records system events, displays real-time and past sensor data, and provides a means to compare them.

***Table 3**. IoT Communication Modules in Smart Aquaponic System*

AUTHOR	YEAR	TITLE	RELEVANT FINDINGS	RELATIONSHIP TO THE STUDY
P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy.	2014	A state-of-the-art review of the Internet of Things (IoT) History, Technology and fields of deployment	Developed tools that will become responsible for the IoT communication modules	Both studies executed IoT Communication Modules
M. Farhan, M.Pu'ad, et al	2020	IoT based water quality monitoring system for aquaponics	Developed a remote monitoring system using an IoT.	Both studies developed a monitoring system that uses an IoT and Raspberry Pi at the same time
T. Y. Kyaw and A. K. Ng	2017	Smart Aquaponics System for Urban Farming	Developed a graphical user interface for the system, a web application was created and hosted on a Raspberry Pi	Both studies developed graphical user interface for the system that uses an IoT and Raspberry Pi at the same time
A. Dutta, et al.	2018	IoT based Aquaponics Monitoring System	Developed an IoT based aquaponics system that use Wi-Fi and Raspberry Pi to connect the system to the web	Both studies developed a system that uses an IoT and Raspberry Pi at the same time
Sunardi, A. et al.	2021	IoT Application on Aquaponics System Energy Optimizatio n	Developed a design system that uses IoT application on aquaponics cultivation system with web interface technology consisting of Raspberry Pi and more.	Both studies developed a system that uses an IoT and Raspberry Pi at the same time

2.4. Image Processing

Image processing serves as a highly effective method for enhancing raw images obtained from external sources such as cameras, satellite sensors, space probes, aircraft, and more. Through the application of image processing techniques, the quality of the original image can be significantly improved, while also preparing it for machine interpretation [35].

2.4.1. Image Processing in Plant Monitoring

The study of Pavel et al. utilizes an image processing system that has been set up to identify and categorize the affected plant disease. Four steps make up this process: picture capture and preprocessing; segmentation of the affected area; feature extraction; and classification utilizing a multi-class support vector machine approach [36]. The study categorizes plant diseases by collecting photos using a Raspberry Pi 3 IoT-enabled device with critical environmental factors from various agricultural sites. Images are submitted to the database through this IoT device, and after being resized to 256x256 and going through a preprocessing step, the affected part is segmented using the K-mean clustering technique. Multi-class SVM is applied with obtained features to classify. In addition, the availability of adequate data allows the researchers to properly build, analyze, and monitor the categorization with a 97.33 percent accuracy rate.

In the study of Porob, several image processing techniques can be used to determine the health of the plants [37]. The study was more focused on utilizing Android to detect disease on plant leaves. The condition may be detected using a variety of image-processing approaches. Using image processing, a useful description of the plant's health may be obtained that may be essential for additional actions. In this study, SVM and K-means were utilized and in comparison, to other algorithms, the SVM algorithm yielded better results.

2.4.2. Image Processing in Fish Monitoring

The goal of the study of Rodriguez et al. is to enhance the management of fish farms and aquariums by proposing a non-intrusive method to measure fish size in aquatic environments [38]. The study eliminates the requirement for fish marking methods or in-person fish observation by combining stereo imaging with computer vision techniques to recognize and quantify fish in photographs. By various tests with live fish, the proposed technique's accuracy and effectiveness have been evaluated. The outputs of this approach have shown a huge amount of potential, as it allowed the researchers to obtain the fish size in the image with a low error rate.

In the research of Petrellis, morphological feature extraction was carried out using non-invasive image processing and deep learning techniques. The initial approach for estimating the morphological features of the fish was based on image processing techniques. An average length error approximation of 4.93 percent was yielded considering the estimation of the relative fish length, height, and area [39].

Table 4. Image Processing

AUTHOR	YEAR	TITLE	RELEVANT FINDINGS	RELATIONSHIP TO THE STUDY
S. N. Pauzi, M. G. Hassan, N. Yusoff, N. H. Harun, A. H. Abu Bakar, and B. C. Kua	2021	A review on image processing for fish disease detection	Showed the advantage of image processing on the interpretation of raw data images	Both studies used image processing for the detection of the fish diseases
. M. I. Pavel, S. M. Kamruzzaman, S. S. Hasan and S. R. Sabuj	2019	An IoT Based Plant Health Monitoring System Implementing Image Processing	Showed how to employ technology capable of identifying and categorizing plant diseases with accuracy.	Both studies used image processing and IoT devices that will help on the detection of plant diseases
S. Porob	2017	Plant Health Monitoring Using Digital Image Processing	Showed how plant diseases are monitored	Both studies have plant health monitoring
A. Rodriguez, A. J. Rico-Diaz, J. R. Rabuñal, J. Puertas, and L. Pena	2015	Fish monitoring and sizing using computer vision	Developed a system that help to monitor the fish	Both studies monitored the conditions of the fishes using a live monitoring system
N. Petrellis	2021	Measurement of fish morphological features through image processing and deep learning techniques	Developed system that help to monitor the fish	Both studies monitored the conditions of the fishes via image processing and deep learning

2.5. Convolutional Neural Networks

Convolutional Neural Network (CNN) is a class of artificial neural networks that has become dominant in various computer vision tasks. It possesses the capability to process input images, dynamically allocate significance through learnable weights and biases to different elements or objects within the image, and effectively discern and distinguish between them [40].

2.5.1. YOLOv4

In the studies of Chen and Wang et al., the YOLOv4 model demonstrates enhanced precision in its prediction outcomes when it comes to identifying and locating objects [41] [42]. This model is composed of distinct components, namely the backbone, the neck, and the head. To train the object detection capabilities, the model utilizes the CSPDarknet53 as its underlying architecture. In addition, the neck of the model incorporates spatial pyramid pooling (SPP) and path aggregation network (PANet) to extract diverse layers of feature maps. Finally, the primary head of the network is referred to as the YOLO head.

In the research conducted by Chen et al., the YOLOv4 model showcased superior performance compared to the Faster R-CNN and SSD models in accurately identifying three distinct types of pest diseases [41]. The other models exhibited inconsistencies in predicting these pest diseases at varying scales. In summary, the YOLOv4 model excelled in both localizing pests of different scales within the image and achieving the highest accuracy in predicting the corresponding pest diseases.

Moreover, Yang et al. states that with the help of a single neural network, the high-precision target recognition algorithm in YOLOv4 can accurately anticipate the target's position and other features for the entire image [43]. Unfortunately, even though YOLOv4 was utilized to identify objects, they found out that it also has poor bounding box orientation and finds it challenging to tell between overlapping detected items. But compared to the previous models of YOLO, the model of YOLOv4 has faster detection speed and has good accuracy.

2.5.2. Mask R-CNN

In terms of detection and image processing, CNN is widely used for image segmentation, image classification, and object recognition. CNN is an algorithm language regarding image segmentation, and it uses a deep neural network where it performs sophisticated computations on large amounts of data for object detection. CNN strength is about its convolution layer where its valuable task is to detect [44]. In plant disease detection, using a different CNN architecture shows a high achieving levels of classification accuracy. It is widely used in agricultural applications because of its high accuracy in crop recognition, plant disease detection, and it is said that this method is a reliable tool for monitoring the growth of greenhouse lettuce [45]. The CNN model demonstrates remarkable performance and again, serves as a reliable tool for effectively monitoring the growth of lettuce in a greenhouse environment [46].

Different CNN architectures and approaches can be used in plant and fish monitoring. In the study Chen et al., Mask R-CNN is used for segmentation of image by mask region-based using a bounding box [41].

Determination of fish contour was generated from this method using binary mask and the annotated image is used to calculate and measure fish dimensions. This helps focus on fish parts such as the caudal fin, dorsal fins, and mouth of the fish. In the study of Lu et al., bounding-box regression, object classification and mask prediction are used to separate objects from the background for classification [47]. And this uses instance segmentation from applied Mask R-CNN that allows the system to segment and detect lettuce with a clear contour of its canopy area. To get the object detection process using Mask R-CNN, first was to get datasets of the image to be measured. Using bounding box label datasets for object segmentation. This will extract the contour features of the object then calibrate the system to train the program [48]. After calibrating, the object from the inserted data sets can be detected using this method. And actual object detection can be achieved with great accuracy.

2.5.3. VGGNet

For object detection and landmark annotation on a test image, the VGG16 CNN architecture was employed. This 16-layer CNN design incorporates 3x3 convolutional filters, which contribute to enhancing the network's depth. The utilization of VGG16 demonstrated a noteworthy advancement in achieving accurate image identification, particularly on a large scale [49]. In the study of Chen et al., the algorithm was trained to produce eight (8) pairs of coordinates of landmarks and the datasets were used to the VGG16 CNN with utilization of Keras and Tensorflow [41].

Furthermore, the VGG16 CNN model was suggested to detect plant diseases using the public dataset from PlantVillage with an accuracy of 91.83%, but when using the other dataset, it resulted in a 92% accuracy [51]. Coulibaly et al. recommended utilizing transfer learning to identify mildew infections in pearl millet and with the VGG16 CNN model, which was pre-trained using the public dataset. The output in this study has delivered a respectable outcome with a 94.5% recall rate and a 95% accuracy rate [50]. Nonetheless, in a study conducted by Fuentes et al., using the identical algorithm model trained with transfer learning on the same public dataset, an accuracy of 90.4% was achieved [51].

2.5.4. GoogleNet

In the research conducted by Hanbay and Turkoglu, various approaches were employed to evaluate the efficacy of powerful deep neural network architectures in tackling the task of plant disease identification [52]. A subset of the ImageNet database is being utilized to train these deep learning algorithms and classify images into 1000 object categories using a pre-trained network [53]. One of these architectures that they have trained is GoogleNet. On the other hand, the GoogleNet architecture is a significantly 22 layers deep and larger architecture, but still considerably lesser number of parameters which is based on a network-in-network approach, it includes architectural modules that extract different feature points using multiple convolutions running in parallel and in the form of the inception modules. The inception module captures a variety of characteristics simultaneously by using parallel 1 x 1, 3 x 3, and 5 x 5 convolutions as well as a max-pooling layer. To reduce dimensionality, 1 x 1 convolutions are added before the previously described 3 x 3, 5 x 5 convolutions (and also after the max-pooling layer) since the amount of related computation needs to be kept in check in order to provide practical implementation. The outputs from these parallel layers are concatenated in a filter concatenation layer. It is important to note that the GoogLeNet architecture utilized in the experiments consists of a total of nine inception modules, each following a similar structure. [54].

Moreover, disease areas were identified by using publicly available PlantVillage dataset using a variety of filters. However, they were able to classify the plant diseases by using the same dataset with a 99.35% accuracy. The precision (P), recall (R), F1 score, and overall accuracy of the performance were measured [55]. But when they tried it with other sets of images, the accuracy was low [55]. The number of modified pictures, minibatch sizes, weight variations, and bias learning rate are significant variables that influence the effectiveness of deep architectures [54]. This will extract the contour features of the object then calibrate the system to train the program.

2.6. Data dashboard components

A data dashboard is a visual representation of data that provides an overview of key metrics and performance indicators. It allows users to monitor and analyze data in a clear and concise manner. Generally, it includes charts and graphs, data tables, alerts and notifications, real-time data updates and among others.

2.6.1. User Interface Design

Developing a graphical user interface for the system, a web application was created and hosted on a Raspberry Pi and different programming languages were used in the study of Kyaw and Ng such as Python, Javascript, Node.js and more [11]. They used Circular Gauge View as their open-source framework and Bootstrap for the front-end framework. A notification system such as Google Firebase was used to provide an up-to-time database for storing and displaying the current and previous data sensor values.

The software on the internet of things application on the study of Hassan et al. was based on Raspberry Pi3, it is designed to monitor and control the input and output of the system so that the system may function properly. The website was programmed using PHP and it utilizes Apache as their web server and MySQL operates as a database server. Otherwise, the website application can serve as a user of data values from sensors processed by the Raspberry Pi, as well as a command indicator for the output to be operated [45].

The research conducted by Tolentino et al. has a website application that displays not just the status of the most recent data, but also the numerical values for all those parameters, as well as for determining fish status such as their average length and weight. It was created using different programming languages including JavaScript and PHP code that delivers data from the database to the user interface as new data is entered, and HTML, CSS, and JavaScript scripts for website application's dynamic design and graphs [16].

All sensors in the study of Zhang et al. were also connected to the Raspberry Pi microcontroller. Afterwards, the system parameters were successfully shown via Liquid Crystal Display and Internet of Things. A relay has also been used to control the lighting. The acquired values from the Temperature & Humidity sensor and the pH sensor module were sent to the data server, which first stored the values of the three parameters in a MySQL database and then displayed the sensor values in a continuous JavaScript Object Notation format in the web portal until the system components and server were up and running [46]. The Internet of Things has been made possible due to the Raspberry Pi. It was possible to monitor the readings from anywhere via the Internet of Things, and it also provided a graphical and analytical view of the system parameters that characterize the IoT Based Aquaponics Monitoring System.

2.6.2. Remote Control Application

The IoT-based aquaponics system comprises a typical aquaponics system with sensors and a microcontroller for collecting sensor data and automating system operations. The system employed NodeMCU, a microcontroller that collects sensor data from the fish tank and hydroponics grow bed [12]. Three modules are running that control the rest of the aquaponics systems. All of this is simply configured using a simple key value configuration file and a basic configuration module. The first module is the light module, which produces a twelve-hour day-night cycle by monitoring the present time zone's wall clock and turning the LED lights on for a set period. The pump module is the second module, and it's in charge of running the water pumps 45 minutes out of every hour. The temperature module is the final and most difficult module. It reads the system's temperature sensors and determines whether to heat or cool [45].

2.6.3. Remote Monitoring Interfaces

All the data gathered in the sensors including the pH value, temperature and water level that was measured and displayed, was transmitted using the Web Socket protocol. It establishes secure connections to the server and keeps the system functioning in real-time [48]. However, the values which are gathered from the sensors in the study of Tolentino et al. are sent to the Arduino Mega. The data is then sent through LoRaWAN IoT Protocol to the Raspberry Pi, which finally sends it to the Web Application for monitoring. Data transfer using LoRa modules is ideal for aquaculture monitoring since it lowers production costs and allows for longer-distance connection [16].

Chapter 3

METHODOLOGY

This chapter presents the research methods including research design, research process flow, procedures, statistical analysis, and the project work plan.

3.1. Research Design

This illustration below shows the flow of how the data have been gathered, analyzed, and solved.

Table 5. *The Input-Process-Output of the system*

INPUT	PROCESS	OUTPUT
Hardware Requirements: • Raspberry Pi 4B • USB Cameras • System Monitor **Software Requirements:** • HTML, Java and CSS • Raspberry Pi OS • Python • Google Collab • Firebase	1. Establish the IoT connection 2. Develop a program for growth monitoring, harvest prediction and plant disease detection. 3. Calibrate the camera for testing. Compare the accuracy for each model. 4. Design a webpage and program the database. 5. Integrate all systems and test the application	A data dashboard that includes all the necessary information and algorithms for monitoring of the greenhouse.

The study used a Developmental Research design which is a scientific and systematic process of designing specific programs and products in order to attain a specific goal [49]. By the name itself, in doing this research, it is expected to create and develop a new procedure, techniques and technology that will contribute to the body of knowledge. For most cases, developmental research aligned with computer applications can be considered as the frontline of technological advancement. Specifically meant for this study, the proposed research aims to develop a web application to provide the graphical user interface of the system. It includes an interactive data dashboard with graphs and other real time data and live monitoring page.

3.1.1. Block Diagram

In order to establish the IoT connection between the cameras and the web application, the block diagram is given below:

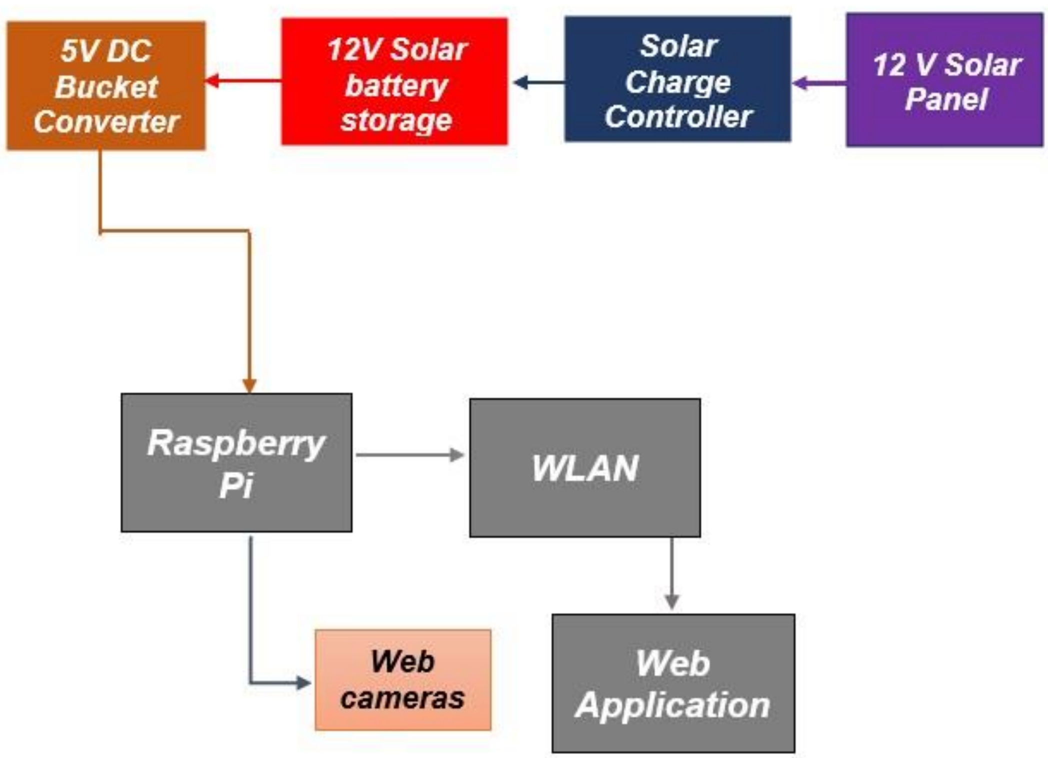


Figure 1. *Block Diagram*

3.2. Research Process Flow.

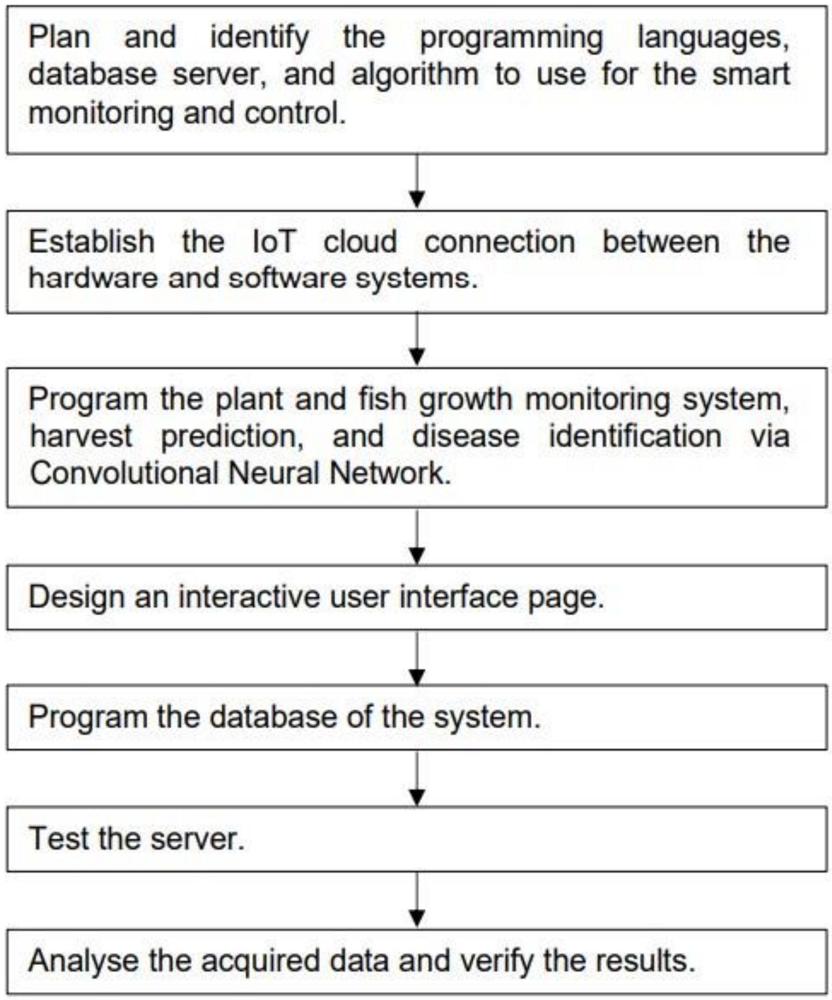


Figure 2. *Research Process Flow*

3.3. Development of IoT-based Architecture for Smart Aquaponics System

Through Internet of Things technology, data flows from input devices attached to "things" through a network and eventually on to a corporate data center or the cloud for processing, analysis and storage. Specifically for this study, two cameras was deployed in the system and is connected through a microcontroller.

3.3.1. Materials and Equipment

The data transmission and reception between the system and the Web App was be connected through Raspberry Pi 4. Specifically for the software side, this microcontroller ran the collected data from the camera and direct it to the software application. Each data was then be processed and uploaded to the cloud. The data obtained was used to continuously monitor the environment. The following materials was used:

a. Raspberry Pi 4 Model B

The 4GB version of Raspberry Pi 4 Model B is one of the newest members of the fourth generation Raspberry Pi family. Compared to the Raspberry Pi 3 Model B+ of the previous generation, it delivers revolutionary gains in CPU speed, multimedia capabilities, memory, and networking, while retaining backwards compatibility and similar power consumption. The primary characteristics of this device are a powerful 64bit quad-core dual-display capability through two 4K displays and a CPU hardware video decoding at up to 4Kp60, micro-HDMI connectors, and up to 4GB of RAM,

b. System Monitor and Peripherals

The required components and devices that to monitor and interact with a system for this project was system monitor, keyboard, and mouse, Once connected through the RPi, it was act as an instant set-up where the programmed systems was deployed and the user can browse through the web application.

c. Low-cost USB Camera

USB Cameras are imaging devices that utilize USB 2.0 or USB 3.0 technology to transfer image data. These cameras are specifically designed to seamlessly connect with dedicated computer systems, leveraging the ubiquitous USB technology found on most computers. The study will use a low-cost web camera with 2 megapixels image resolution, 25 fps/30 fps video frame rate, and effective resolution of 1932(H) × 1088(V).

3.3.2. Establishing the IoT Connectivity

Connectivity allows the Internet of Things to communicate with gateways, apps, servers, routers, and other IoT devices. By connecting to an IoT gateway or other edge device, where data is transferred to the cloud for analysis, IoT devices exchange the data they gather. These devices interact with one other and take action based on the information they share. Through this connection, such devices are able to carry out the tasks and perform the functions they were designed for. In order to establish communication between the acquired camera data and the web application, an IoT cloud server will be used. In setting up this database, the microcontroller must be connected to the internet.

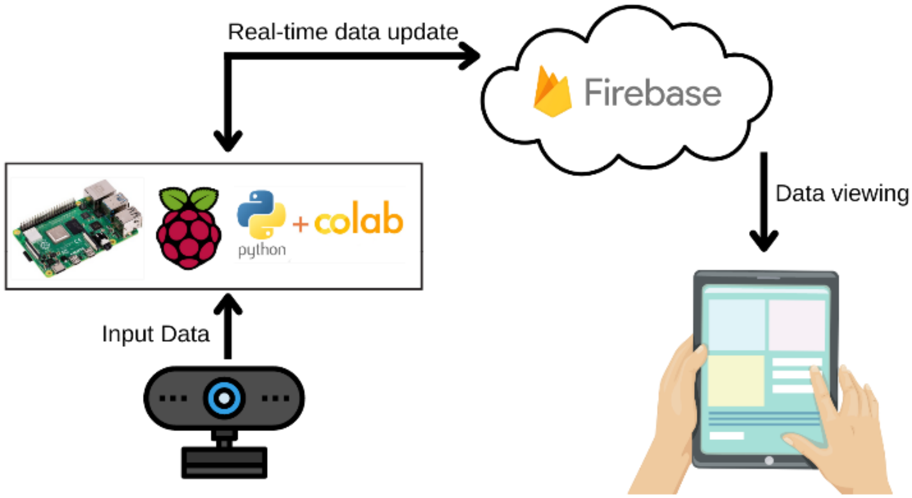


Figure 3. *Internet-of-Things connectivity*

3.4. Development of Plant and Fish monitoring system via Convolutional Neural Network

This study aims to integrate convolutional neural networks (CNN) to monitor the status of lettuce grown and tilapia cultivates in the aquaponics system. The approach consists of multi-stage procedures, including plant and fish growth monitoring, harvestability status, and plant disease detection.

3.4.1. Preparation of Datasets

Machine learning algorithms rely on labeled datasets to train computer vision models for object detection. To teach the model to identify objects in a similar way to humans, pre-labeled images are required. These labeled images serve as examples for the model to learn from. Through this, the model draws conclusions and learn to detect objects based on the patterns it has discovered within the labeled images.

3.4.1.1. Bounding Box

In this study, bounding boxes was used as annotation markers, represented by rectangular shapes drawn around objects in the images. To facilitate this, the study was utilize OpenLabelling, an open-source image and video labeling tool. OpenLabelling enables the annotation of each image by generating a corresponding .txt file. Within this text file, each line describes a bounding box, providing the necessary coordinates and dimensions for accurate object localization.

3.4.1.2. Polygon Annotation

Polygon annotation uses multiple vertices or x,y coordinates to effectively map the shape of complex shapes. Depending on the purpose of the dataset, polygon annotation was significantly enhance the accuracy of a CV model when dealing with real-world irregular and dynamic shapes. In this study, the VGG Image Annotator (VIA) was utilized as a manual annotation software for images, audio, and video. It operates directly within a web browser, without the need for any installation or setup process. This makes it easily accessible to use for annotating the data in the study.

3.4.2. Implementing the different CNN models

For this study, three CNN models was used to train the data for each system requirement. The model that was yield the highest accuracy result was then be integrated to the proposed aquaponics system.

3.4.2.1. Plant and Fish Growth Monitoring

In an aquaponics system, two cultures are being monitored. In order to ensure their growth in a closed water system, the system was collating the growth rate of both plants and fishes. The proposed CNN model was identifying the canopy area of the lettuce and the length of the fish. The data was stored and recorded in the cloud, and then later was analyzed and can be viewed graphically on the web application.

3.4.2.2. Plant Disease Detection

Even in a fully controlled and automated greenhouse system, grown lettuce were subjected to certain plant diseases, hence, a system to identify diseases early on was necessary. The primary objective was to develop a program to identify the infected plants that were induce in the greenhouse system.

3.4.2.3. Plant and Fish Harvestability

The ideal measurements of the harvestable plant and tilapia below as follows:

Table 6. Ideal Measurements of Lettuce and Tilapia

Lettuce	
Height	30 cm
Diameter	20 cm
Tilapia	
Ideal Weight	200 – 300 g
Length	16 – 20 cm

3.4.3. Calibration and Testing

The cameras were placed and calibrated at a certain distance to ensure that the accurate area and length can be measured. The process was tested using the USB cameras where it was calibrated to compute the camera parameters. On this step, the distance of the camera from the subject was tested. Same test images will be employed on the three models to compare the accuracy and analyze the results from the pre-trained models.

3.5. Development of Web Application for Smart Aquaponics System

Innovated for this project is a web application that shows the data dashboard where all the collected inputs can be viewed graphically and interactively. The web app also allows the user to control the different actuators manually such as the air and water pumps, fans and grow lights. Finally, a live monitoring can be viewed using the web app to ensure that the status of the plant can be monitored, and the user can act based on the system requirements.

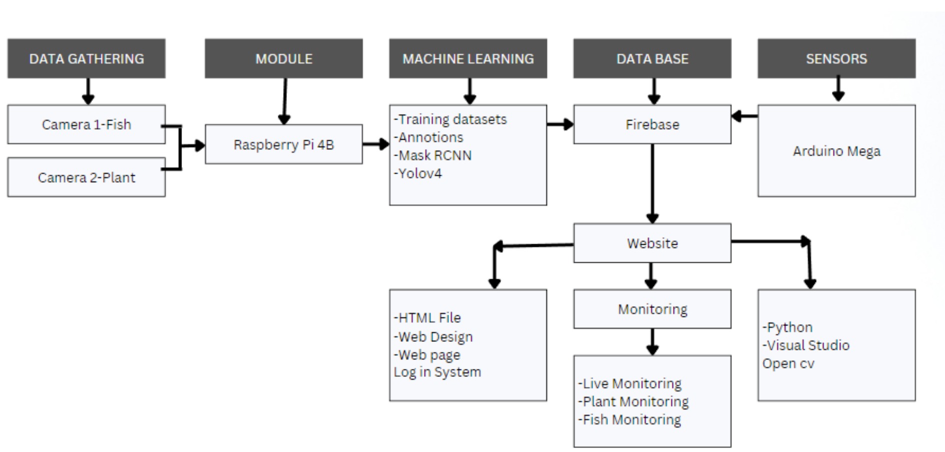


Figure 4. *Overall Framework of the System*

3.5.1. Front-end development

In the dynamic design aspect of this study, HTML, CSS, and JavaScript (JS) was utilized as the programming languages. HTML serves as the foundation for website structure, while CSS is employed to control the presentation, formatting, and layout of the web pages. JavaScript plays a crucial role in controlling the behavior and interactivity of various elements on the website. By combining these programming languages, the study aims to create an engaging and interactive user interface.

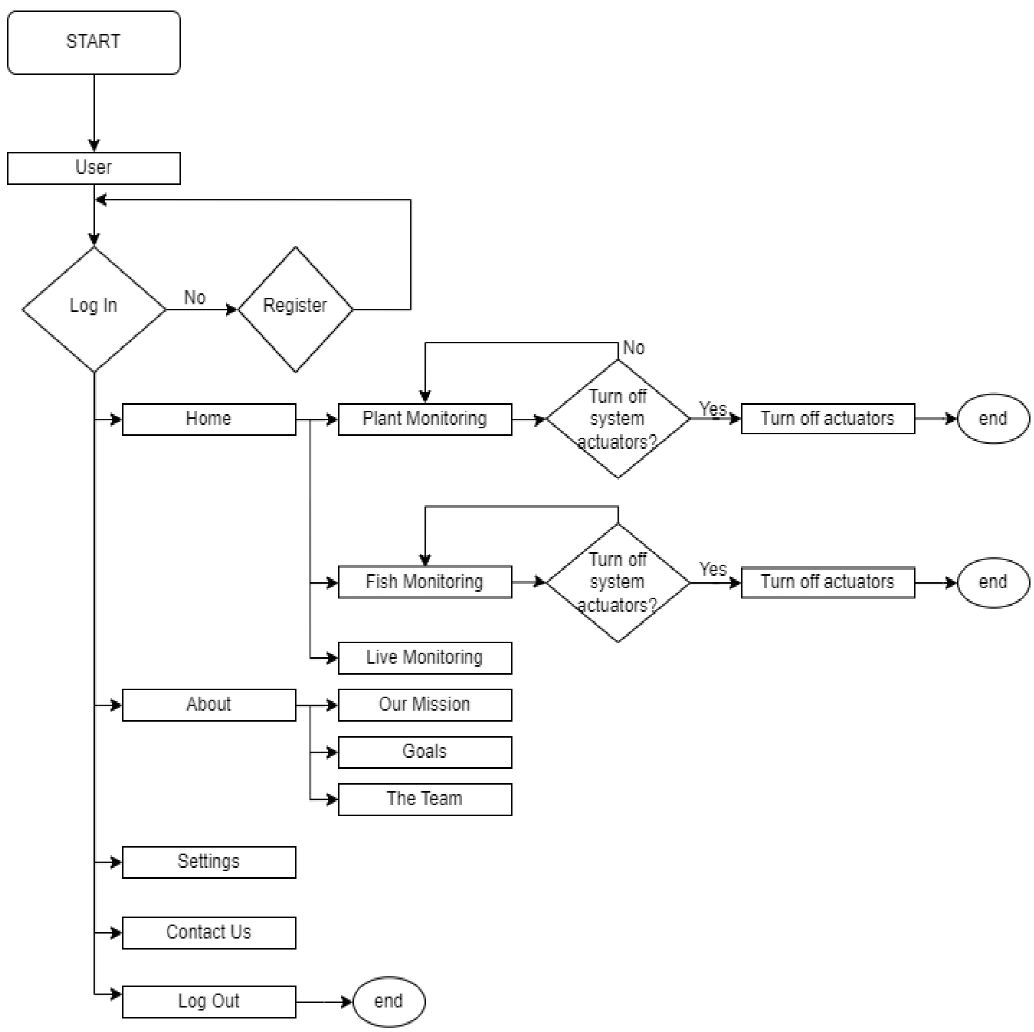


Figure 5. *Web Application flowchart*

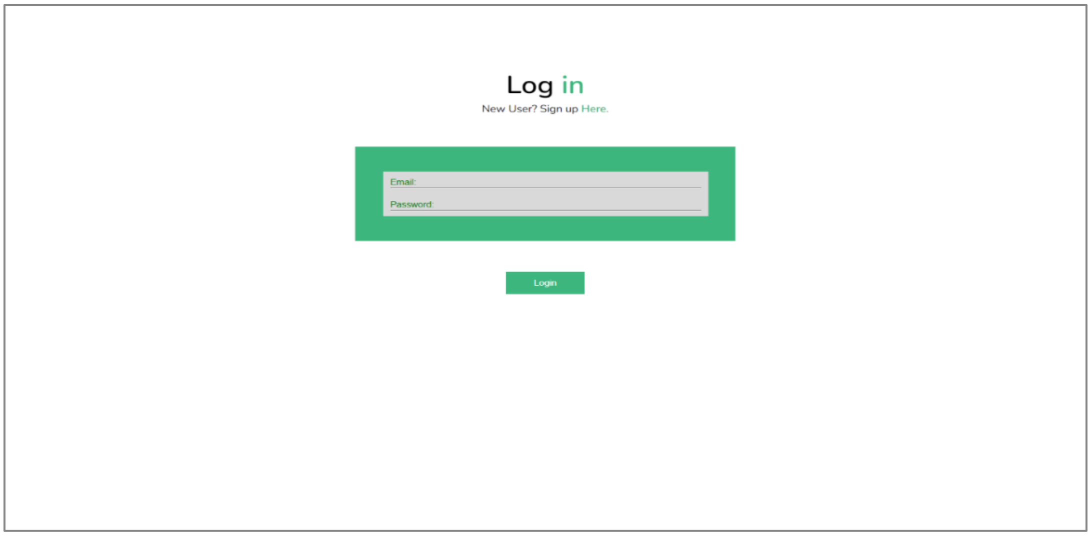


Figure 6. *Log in Page*

Users access the web application by signing in through their accounts on the login page. The login page acts as a crucial entry point, granting authorized users the ability to harness the full potential of the web application.

Figure 7. *Sign up Page*

Prior to gaining access to their account through the sign in process, users need to complete the registration on the sign-up page. Registration serves as the initial step for users to create their account and establish a unique identity within the web application.

Figure 8. *Home Page*

On the home page of the web application, users are greeted with a comprehensive and informative overview of the smart aquaponics system. This overview serves as a concise introduction, providing users with a clear understanding of the system's functionalities, benefits, and purpose.

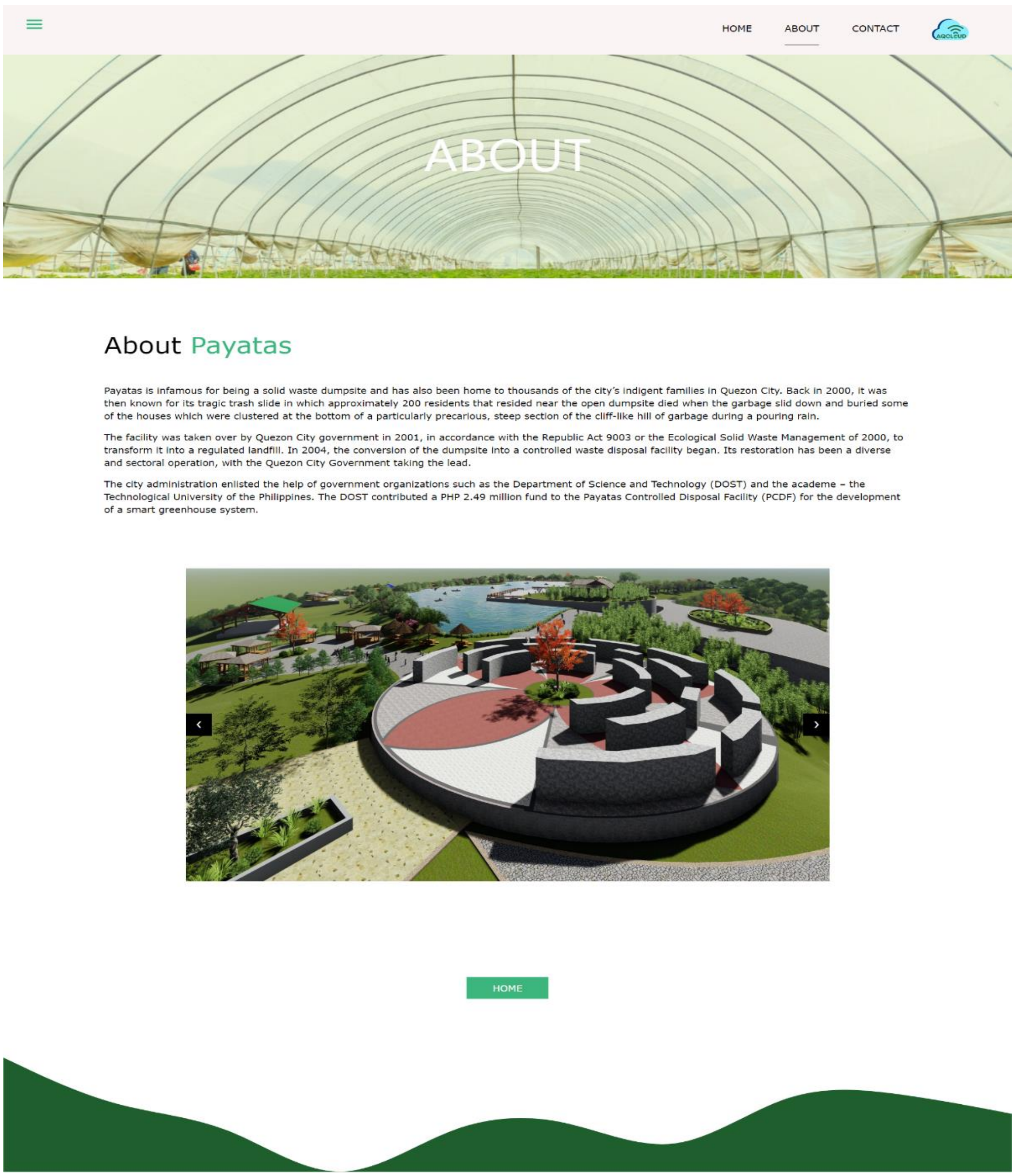


Figure 9. *About Page*

On the about page of the web application, users can explore a comprehensive overview that delves into the rich history of Payatas. By perusing this content, users can gain a deeper understanding of the community's past, its journey of progress, and the factors that have shaped its present-day landscape.

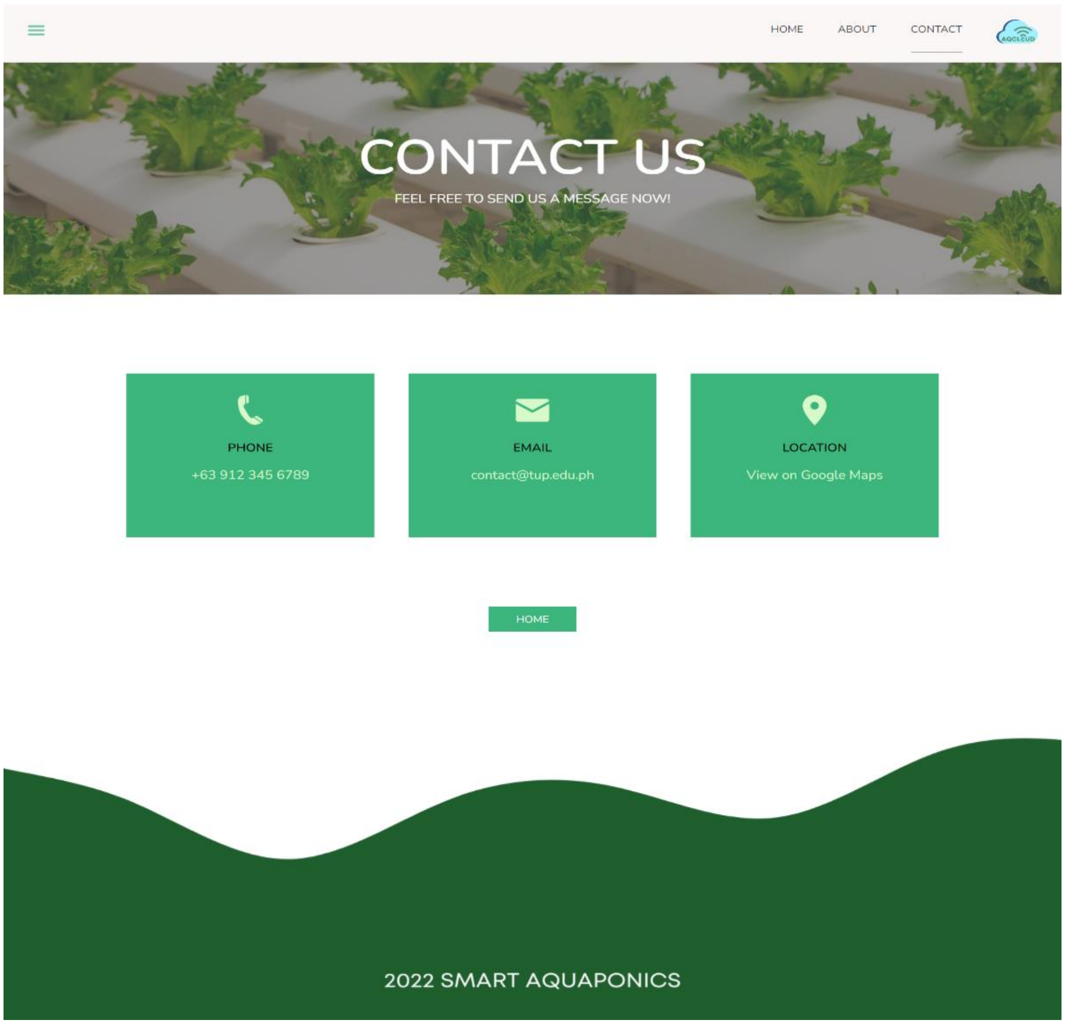


Figure 10. *Contact Page*

The contact page of the website provides users with essential contact information to connect with the smart aquaponics team. This information includes the telephone number, email address, and deployment location where users can easily reach out to the team.

3.5.2. Back-end development

The system database was implemented using Firebase, which is a comprehensive set of hosting services suitable for various types of applications.

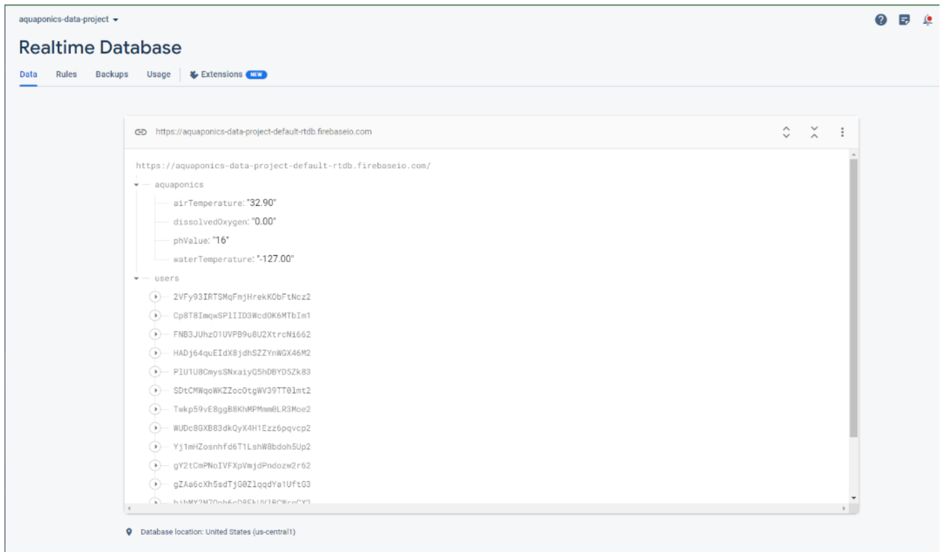


Figure 11. *Firebase's Realtime Database Environment*

It supports features such as content hosting, social authentication, notifications, and even real-time communication servers. In order to establish a connection with the cloud server, the API key generated from the programmed codes is necessary as it offers a unique identifier and authentication to securely communicate with the Firebase services.

The system must record the abnormalities and recovery events in the database. For the live monitoring, the cultivation of plants and fishers was continuously monitored through the camera modules. All these data sets were saved in the database of the proposed web application. The results from the processed image were displayed in the web application.

3.5.3. Integrating the algorithm to the web application.

When the application is established, the proposed algorithm should be integrated to the system. With this, the data from the processed image can be viewed and monitored through graphical representation on the data dashboard. In doing this, there are several steps necessary. First, the algorithm must be tested to ensure its accuracy and efficiency. Once the algorithm is ready, it integrates into the backend of the web application. To integrate the algorithm the appropriate APIs must be identified for the system to receive input data from the frontend.

3.6. Server Testing and User Acceptance

The evaluation form for the project was built upon the principles of ISO 25010, titled "Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models." This standard offers guidelines and criteria for evaluating and managing the quality of software products, covering diverse quality attributes.

Table 7. *Survey Questionnaire*

Introduction: The researchers involved in this study are required to administer a questionnaire to evaluate diverse factors and parameters, entitle *"Smart Aquaponics System Utilizing Internet-of-Things Technology via Convolutional Neural Network"*.					
Instructions: Please rate each criterion and choose one answer if your assessment meets this rating. Select one best answer from the following options. **1** – Strongly Disagree, **2** – Disagree, **3** – Neutral, **4** – Agree, **5** – Strongly Agree**.**					
Survey Statement	**Rating**				
	1	**2**	**3**	**4**	**5**
Functionality and Efficiency					
1. The Smart Aquaponics system outperforms the traditional approach in terms of efficiency.					
2. The web application provides correct and accurate results.					
3. The web application presents real-time data visualization of essential parameters.					
4. The system prevents unauthorized access.					
5. The system can be continuously monitored and operated 24 hours.					
Usability and Aesthetics					
6. Navigating and understanding the web application is user-friendly and straightforward.					
7. The web application includes an interactive data dashboard.					
8. The web application has an aesthetically pleasing user interface.					
9. It is not necessary to constantly supervise the entire system during operation.					
10. Operating the entire system is simple and straightforward.					
Portability and Maintainability					
11. The system does not require a special handling during the operation.					
12. The application system requires minimal effort for maintenance.					
13. The system can be adapted to the community.					
14. As long as you have internet-connected devices, the system remains continuously accessible and available.					
15. The performance of the web application or software system is hindered in the event of hardware component failure.					
Reliability					
16. The web application ensures user safety and provides reliable functionality.					
17. The system possesses comprehensive project information.					
18. In the event of a failure, the system is capable of data restoration.					
19. The web application can be open on other devices and still function correctly.					
20. The system provides guides that are easy to understand.					

3.7. Statistical Analysis

In this study, descriptive statistical analysis and an unpaired t-test was employed. The unpaired t-test is a statistical analysis method utilized to compare the means of two independent groups. Its purpose is to determine whether a significant difference exists between the means of these two groups. This technique is commonly employed in research, experiments, and studies to assess and compare the effects of different treatments, interventions, or conditions on outcome variables.

On the other hand, descriptive statistics are utilized to provide quantitative descriptions of data in a manageable and concise manner. They are used to present basic characteristics of the data in a study, offering simple summaries about the sample and the measurements taken. Descriptive statistics aid in summarizing and organizing data, making it easier to interpret and understand the information gathered.

The following formula was used:

$$\boldsymbol{Test\ Statistic\ of\ Unpaired\ Samples = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{s^2\left(\frac{1}{n^1} + \frac{1}{n^2}\right)}}}$$

$$\boldsymbol{where\ s^2 = \frac{\sum_{i=1}^{n_1}(x_i - \overline{x}_1)^2 + \sum_{j=1}^{n_2}(x_j - \overline{x}_2)^2}{n_1 + n_2 - 2}}$$

Chapter 4

DATA AND RESULTS

This chapter presents the gathered data, analysis and interpretation of the results and discussion.

4.1. Convolutional Neural Network Testing

This testing attempts to ascertain the CNN model's performance in precisely processing and interpreting graphical information, so verifying its accuracy, resilience, and adaptable competencies could contribute to more dependable as well as effective models in future advancements.

4.1.1. Object Detection mAP Graph

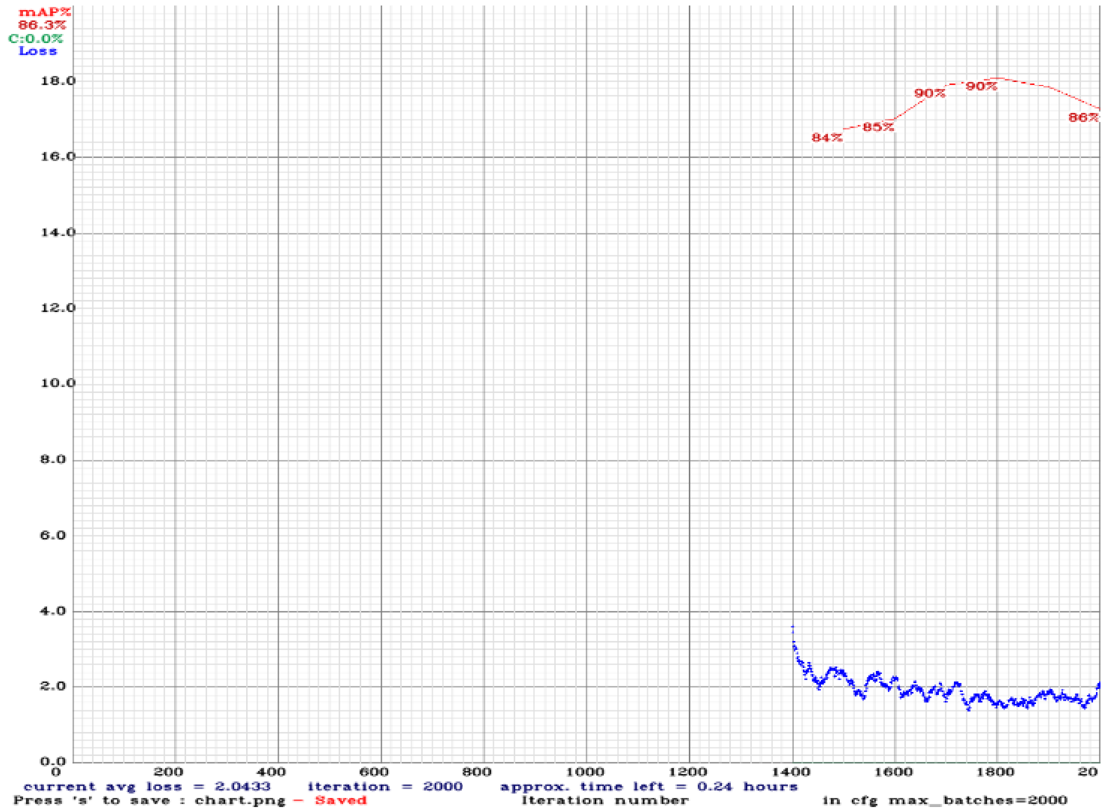
The object detection mAP (mean average precision) graph is a visual representation of the YOLOv4 model's accuracy at various levels of detection precision and recall. In YOLOv4, the mAP value is computed by calculating the average precision (AP) over various recall levels and averaging those values among all classes in the dataset.

```
 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
20
 detections_count = 87, unique_truth_count = 29
class_id = 0, name = romaine_lettuce, ap = 86.32%        (TP = 24, FP = 3)

 for conf_thresh = 0.25, precision = 0.89, recall = 0.83, F1-score = 0.86
 for conf_thresh = 0.25, TP = 24, FP = 3, FN = 5, average IoU = 77.92 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.863195, or 86.32 %
Total Detection Time: 10 Seconds
```

(a)



(b)

Figure 12 (a) (b). *Average Loss and mAP Graph Detection Time for Romaine Lettuce*

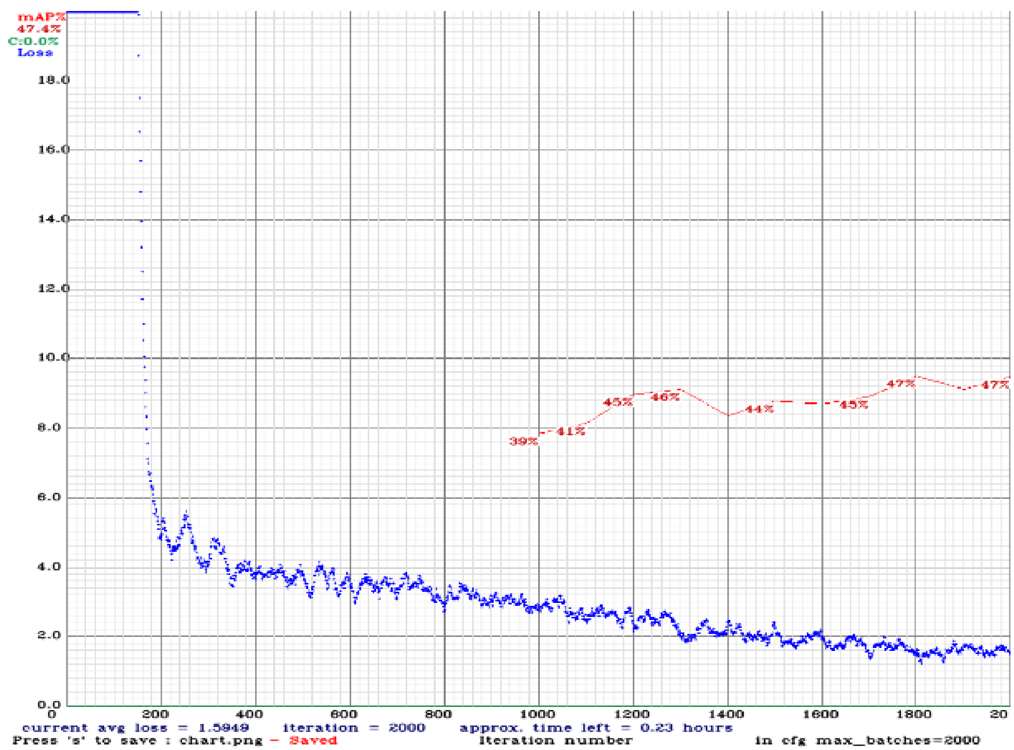
The average loss value for the detected romaine lettuce was calculated to be 2.0433. Moreover, the detection of romaine lettuce showcases a mean average precision (mAP) of 86.32% at an intersection-over-union (IoU) threshold of 0.50. The low average loss value suggests that the system is successful in minimizing losses and identifying romaine lettuce with a relatively small degree of damage or quality issues. Furthermore, the high mAP value indicates that the detection algorithm is robust and capable of accurately recognizing romaine lettuce.

```
calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
16
 detections_count = 135, unique_truth_count = 43
class_id = 0, name = Infected, ap = 47.37%       (TP = 15, FP = 9)

 for conf_thresh = 0.25, precision = 0.62, recall = 0.35, F1-score = 0.45
 for conf_thresh = 0.25, TP = 15, FP = 9, FN = 28, average IoU = 45.25 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.473667, or 47.37 %
Total Detection Time: 11 Seconds
```

(a)



(b)

Figure 13. (a) (b) *Average Loss and mAP Graph and Detection Time for Infected Lettuce*

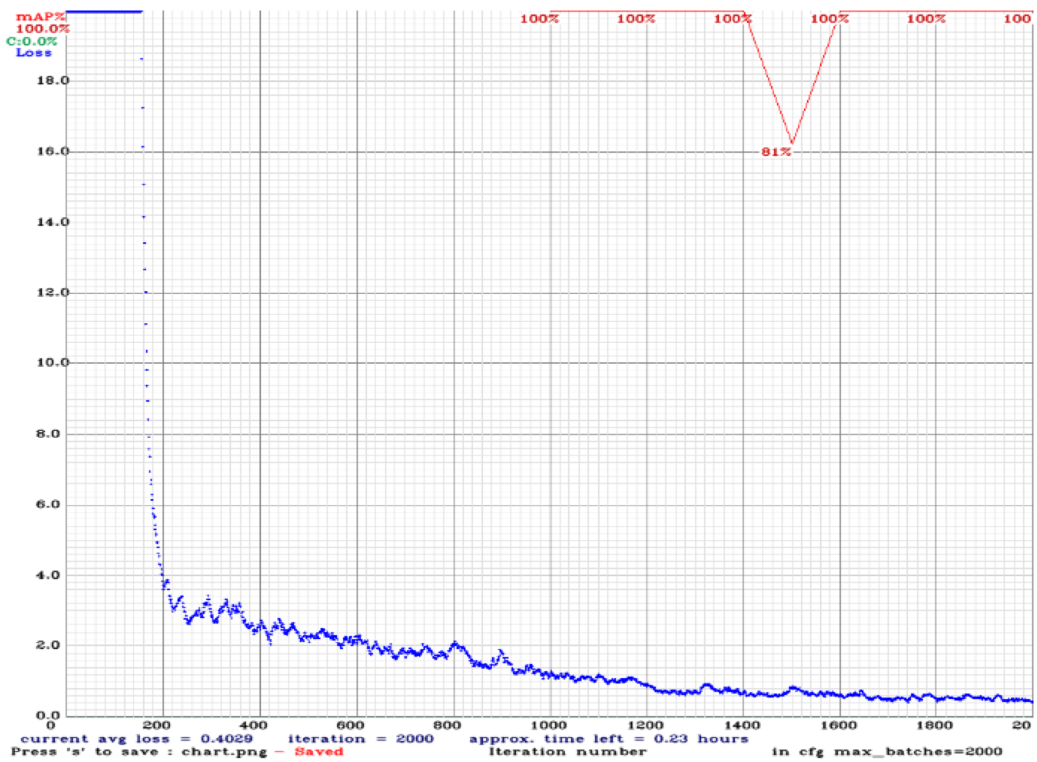
The average loss value for the detected infected romaine lettuce is determined to be 1.5949. Furthermore, the detection of the infected romaine lettuce yields a mean average precision (mAP) of 47.37% when considering an intersection-over-union (IoU) threshold of 0.50. A mAP of 47.37% suggests that the detection system successfully identifies and localizes infected romaine lettuce with a moderate level of accuracy, considering the specific IoU threshold.

```
 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
16
 detections_count = 17, unique_truth_count = 14
class_id = 0, name = nile_tilapia, ap = 100.00%        (TP = 14, FP = 1)

 for conf_thresh = 0.25, precision = 0.93, recall = 1.00, F1-score = 0.97
 for conf_thresh = 0.25, TP = 14, FP = 1, FN = 0, average IoU = 85.57 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 1.000000, or 100.00 %
Total Detection Time: 9 Seconds
```

(a)



(b)

Figure 14. (a) (b) *Average Loss and mAP Graph and Detection Time for Nile Tilapia*

The average loss value for the detected nile tilapia is 0.4029. Moreover, the detection of nile tilapia showcases an exceptional mean average precision (mAP) of 100% at an intersection-over-union (IoU) threshold of 0.50. These results highlight the outstanding performance and effectiveness of the detection system for nile tilapia.

The figures above depict the correlation between the average loss and the number of iterations. More specifically, they illustrate the detection of romaine lettuce, infected lettuce, and nile tilapia over a span of 2000 iterations. The graph exhibits a noticeable pattern where the average loss decreases as the number of iterations increases.

4.1.2. Detection Testing

The trained models were utilized to conduct detection testing at the deployment area. With the YOLOv4 algorithm, the deployed models were able to produce results that correspond to the calculated mean average precision (mAP).

Figure 15. *Detection Testing for Romaine Lettuce*

The figure above shows the results of the YOLOv4 detection of romaine lettuce. Most romaine lettuce in the images produced high accuracy rate, aligning closely with the calculated mean average precision (mAP).

Figure 16. *Detection Testing for Infected Lettuce*

The figure displayed above illustrates the outcomes of the YOLOv4 detection algorithm applied to infected romaine lettuce. The average accuracy rate observed in the images closely corresponded with the calculated mean average precision (mAP), indicating a moderate alignment between the two.

Figure 17. *Detection Testing for Nile Tilapia*

The figure displayed above illustrates the results of the YOLOv4 detection for nile tilapia. The average accuracy rate observed in the images closely matched with the calculated mean average precision (mAP) which highlighted a strong correlation between the two metrics.

Each figure shows the different accuracy rates for each input. This was evaluated to assess the performance of the algorithm in object detection tasks. The high accuracy rate from the testing signifies that YOLOv4 performs well in object detection, accurately identifying and localizing objects across various scenarios, including different object sizes, orientations, and environmental conditions.

4.2. Web Camera Computation

To evaluate the effectiveness of the image processing program employed in the plant growth monitoring system, a comparison was conducted by contrasting the results obtained from the program with manually computed data. Twelve (12) sample sizes for lettuce and fifteen (15) tilapias were tested for 35 days. The surface area measurements from the program were compared with the manual measurements as shown in the table and graph below.

4.2.1. Lettuce Measurements

Table 8. *Tabulated Data for Manual Computation (Lettuce)*

MANUAL COMPUTATION													
layer	Lettuce Number	0-14 days			14-21 days			21-28 days			28-35 days		
		L (cm)	W (cm)	A (cm²)	L (cm)	W (cm)	A (cm²)	L (cm)	W (cm)	A (cm²)	L (cm)	W (cm)	A (cm²)
1	1	5	6.5	32.5	8.5	9	76.5	17.5	13	227.5	26.5	21	556.5
	2	6.5	3	19.5	9	6	54	18	12.5	225	25.5	22	561
	3	7	5	35	9	8.5	76.5	16.5	15.5	255.8	21	22	462
	4	4	7	28	10.5	9	94.5	17	14	238	24	21.5	516
2	5	5	6	30	11	9	99	16	14.5	232	24	24	576
	6	5	7	35	9	9.5	85.5	15.5	14	217	20.5	23.5	481.8
	7	6.5	3	19.5	9.5	6	57	15	13.5	202.5	20	22	440
	8	7	4.5	31.5	10	8	80	17	16	272	24	22	528
3	9	3.5	6	21	8.5	11.5	97.75	16.5	14	231	22.5	21	472.5
	10	4.5	5.5	24.75	9.5	11	104.5	16	18.5	296	24	27.5	660
	11	6	4.5	27	9	9	81	18	14	252	25	22	550
	12	7.5	3	22.5	10	9.5	95	16.5	16.5	272.25	21	27	567
	6.5	**5.63**	**5.08**	**27.19**	**9.46**	**8.83**	**83.44**	**16.63**	**14.67**	**243.42**	**23.17**	**22.96**	**530.90**

Table 9. *Tabulated Data for Smart Computation (Lettuce)*

COMPUTATION USING WEBCAMERA													
layer	Lettuce Number	0-14 days			14-21 days			21-28 days			28-35 days		
		L (cm)	W (cm)	A (cm²)	L (cm)	W (cm)	A (cm²)	L (cm)	W (cm)	A (cm²)	L (cm)	W (cm)	A (cm²)
1	1	5.01	6.48	32.46	8.56	9.12	78.06	17.04	13.02	221.86	26.05	21.03	547.83
	2	6.79	3.95	26.82	9.16	6.19	56.70	18.06	13.04	235.50	25.09	22.11	554.74
	3	6.98	5.55	38.739	9.29	8.94	83.05	16.08	16.05	258.08	21.04	22.05	463.93
	4	4.01	7.23	28.992	10.24	9.13	93.49	17.05	14.07	239.89	24.01	20.01	480.44
2	5	4.95	6.05	29.95	11.49	9.16	105.25	16.03	16.03	256.96	24.13	24.05	580.32
	6	5.21	7.98	41.58	9.31	9.49	88.35	15.07	15.08	227.26	21.03	23.1	485.79
	7	6.32	3.85	24.332	9.74	6.29	61.26	15.05	13.02	195.95	20.01	22.02	440.62
	8	7.05	4.05	28.55	10.68	8.19	87.47	17.05	16.05	273.65	24.1	22.12	533.09
3	9	3.44	6.85	23.564	8.56	11.47	98.18	16.02	15.02	240.62	23.12	21.03	486.21
	10	4.86	5.28	25.66	9.64	11.19	107.87	16.01	19.04	304.83	24.06	27.08	651.54
	11	6.15	4.65	28.60	8.95	8.49	75.99	18.04	14.05	253.46	25.02	22.07	552.19
	12	7.25	3.89	28.20	10.19	9.57	97.52	16.04	17.01	272.84	21.08	27.02	569.58
	6.5	**5.67**	**5.48**	**29.79**	**9.65**	**8.94**	**86.10**	**16.46**	**15.12**	**248.41**	**23.23**	**22.81**	**528.86**

The tables above shows the gathered length, width, and area per lettuce head. In manual measurements, a ruler was used to identify the sizes and was measured every Wednesday, for five weeks. At the same time, smart measurement was programmed to detect the lettuces heads and measure its length and width for five weeks.

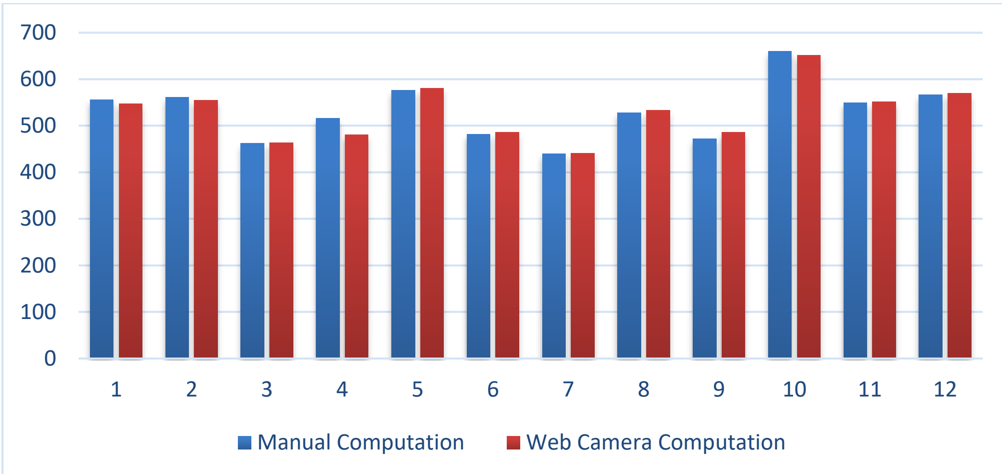


Figure 18. *Graphical Data for Lettuce Measurements*

4.2.2. Tilapia Measurements

Table 10. *Tabulated Data for Manual Computation (Tilapia)*

MANUAL COMPUTATION				
Fish Number	0-14 days (in cm)	14-21 days (in cm)	21-28 days (in cm)	28-35 days (in cm)
1	3.5	4.5	6.5	10
2	3.4	4.2	7	9
3	3.3	4.5	6.5	10
4	3.5	4.4	8	9
5	3.5	4.1	7	8
6	3.6	4.5	7	9.5
7	3.6	4.3	6	8
8	3.4	4.3	5	8.5
9	3.5	4.5	7.5	9
10	3.5	4.1	6.5	8.5
11	3.4	4.3	7	10.0
12	3.5	4.4	6.5	11
13	3.5	4.3	8	12
14	3.5	4.3	7	12
15	3.3	4.1	6.5	11
				9.7

Table 11. *Tabulated Data for Smart Computation (Tilapia)*

COMPUTATION USING WEBCAMERA				
Fish Number	0-14 days (in cm)	14-21 days (in cm)	21-28 days (in cm)	28-35 days (in cm)
1	3.49	4.89	6.41	10.11
2	3.38	4.71	7.12	9.52
3	3.57	4.66	6.49	10.01
4	3.47	4.92	7.98	9.65
5	3.39	4.63	7.01	8.56
6	3.51	4.73	6.98	9.46
7	3.31	4.91	6.41	8.78
8	3.33	4.98	4.95	8.65
9	3.58	4.86	7.38	9.04
10	3.59	4.72	6.48	8.75
11	3.41	4.96	7.02	10.8
12	3.36	4.81	6.48	11.96
13	3.35	4.73	8.01	12.01
14	3.52	4.79	7.09	11.56
15	3.54	5.06	6.54	12.58
				10.094

The tables above shows the gathered length of random tilapia fish. In manual measurements, a ruler was used to identify the sizes and was measured every Wednesday, for five weeks. At the same time, smart measurement was programmed to detect the fish length. Due to the turbid water and difficulty in catching the tilapia, each random sample were fished out of the tank for manual and smart measurement.

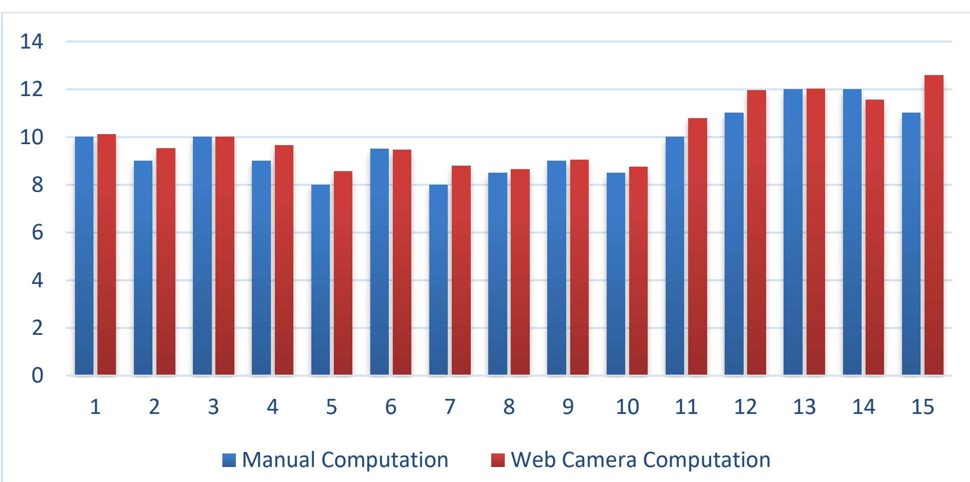


Figure 19. *Graphical Data for Tilapia Measurements*

4.3. Comparison Between Manual and Smart Measurement

To compare manual and smart measurement methods using an unpaired t-test, data from both measurement methods are collected independently. The following data were derived:

4.3.1. Lettuce Measurements

Table 12. *Statistical Analysis Between Manual and Smart Measurement of Lettuce*

GROUP	MANUAL MEASUREMENT	SMART MEASUREMENT
Mean	530.8953	528.8588
Standard Deviation	61.049	59.5211
Standard Error of the Mean	17.6234	17.1822
N	12	12

The statistical analysis of the data revealed that the two-tailed P value is 0.9348. According to conventional criteria, this indicates that the observed difference is not statistically significant. The confidence interval calculation resulted in a mean difference of 2.036950 between Manual and Smart Measurement. The 95% confidence interval for this difference ranges from -49.007855 to 53.081755. The intermediate values used in these calculations include a t-value of 0.0828, a degree of freedom (df) of 22, and a standard error of difference of 24.613. These findings provide insights into the statistical significance and precision of the computed values.

4.3.2. Tilapia Measurements

Table 13. *Statistical Analysis Between Manual and Smart Measurement of Nile Tilapia*

GROUP	MANUAL MEASUREMENT	SMART MEASUREMENT
Mean	9.7000	10.0960
Standard Deviation	1.3202	1.3627
Standard Error of the Mean	0.3409	0.3519
N	15	15

The statistical analysis of the data revealed that the two-tailed P value is 0.4257. According to conventional criteria, this indicates that the observed difference is not statistically significant. The confidence interval calculation resulted in a mean difference of -0.3960 between Manual Computation and Smart Computation. The 95% confidence interval for this difference ranges from -1.3995 to 0.6075. The intermediate values used in these calculations include a t-value of 0.8083, a degree of freedom (df) of 28, and a standard error of difference of 0.490. These findings provide insights into the statistical significance and precision of the computed values.

4.4. Project Evaluation by end-users.

Upon deploying the system, the researchers were able to conduct a project evaluation among different Aquaponics owner as technical consultant.

4.4.1. Functionality and Efficiency

Table 14. Results of Functionality and Efficiency

ITEM	WEIGHTED AVERAGE
1. The smart aquaponics system outperforms the traditional approach in terms of efficiency.	5.0
2. The web application provides correct and accurate results.	4.8
3. The web application presents real-time data visualization of essential parameters.	4.8
4. The system prevents unauthorized access.	4.8
5. The system can be continuously monitored and operated 24 hours.	5.0
AVERAGE	**4.88**

In terms of system functionality and efficiency, the evaluation revealed that the system successfully met the specified requirements and performed intended tasks effectively. The respondents Strongly Agreed that the system meets the functionalities intended for this project.

4.4.2. Usability and Aesthetics

Table 15. Results of Usability and Aesthetics

ITEM	WEIGHTED AVERAGE
6. Navigating and understanding the web application is user-friendly and straightforward.	4.8
7. The web application includes an interactive data dashboard.	4.8
8. The web application has an aesthetically pleasing user interface.	4.6
9. It is not necessary to constantly supervise the entire system during operation.	4.6
10. Operating the entire system is simple and straightforward.	4.8
AVERAGE	**4.72**

In terms of usability and aesthetics, the evaluation indicated that the system was user-friendly, allowing the users to operate the interface and perform tasks without difficulties, and also it provides an aesthetically pleasing and visually appealing experience for user.

4.4.3. Portability and Maintainability

Table 16. Results of Portability and Maintainability

ITEM	WEIGHTED AVERAGE
11. The system does not require a special handling during the operation.	4.6
12. The application system requires minimal effort for maintenance.	4.2
13. The system can be adapted to the community.	4.0
14. As long as you have internet-connected devices, the system remains continuously accessible and available.	4.0
15. The performance of the web application or software system is hindered in the event of hardware component failure.	4.0
AVERAGE	**4.16**

In terms of portability and maintainability, 40% of respondents agreed that a web application can be accessed on different devices while maintaining its functionality. This implies cross-device compatibility, where the application adapts to various devices and screen sizes without issues, allowing users to switch devices without compromising functionality.

4.4.4. Reliability

Table 17. *Results of Reliability*

ITEM	WEIGHTED AVERAGE
16. The web application ensures user safety and provides reliable functionality.	4.6
17. The system possesses comprehensive project information.	4.6
18. In the event of a failure, the system is capable of data restoration.	4.6
19. The web application can be open on other devices and still function correctly.	4.6
20. The system provides guides that are easy to understand.	4.6
AVERAGE	**4.72**

In terms of reliability, 60% of respondents strongly believe that web applications can be accessed on different devices while maintaining full functionality, indicating a high level of confidence in cross-device compatibility.

Chapter 5

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

This chapter presented the summary, conclusion, and recommendation of the study.

5.1. Summary of Findings

The main objectives of this study were to establish IoT connectivity, develop an image processing-based system for monitoring plant and fish growth, design a web application for data visualization, and deploy the system. The findings of study are as follow:

The researchers successfully implemented an IoT system using sensors and webcams for data collection and monitoring. The web cameras were used to collect images and conduct real-time video monitoring, whereas a stable camera installed for plant growth tracking and a submerged camera for fish observation. The Raspberry Pi played a crucial role in data collection, analysis, and experimental control. The connection was completed it was connected to a backend server, which enables storage for real-time data viewing using the web application.

Using a CNN model, particularly the YOLOv4, the system was able to calculate the plant's surface areas, enabling tracking and measurement of plant and fish growth over time. Further, the system was able to detect whether a lettuce head was infected. Further, the proposed system showed no significant difference when compared with the Manual Computation. In numbers, the measurement of average canopy area of the plant using Smart Computation recorded a $530.8958 cm^2$ compared with the Manual Computation which the yield is at 528.8589 cm^2. For the fish monitoring, the measurement of average length of the fish using Smart Computation recorded a 10.02 cm compared with the Manual Computation which the yield is at 9.7 cm.

A web application was also deployed in order to continuously monitoring growth and optimal environment of the system, the system yielded a better quality of crops and faster growth of the fish. This platform provides the general and basic information, as well as the data dashboard where the real-time data and analysis was presented.

In terms of the web application's functionality, all variables from the survey showed significant results which translates to the technical expert's satisfaction in using the system. The survey shows that the combination of Raspberry Pi, web camera, and image processing provided a reliable and efficient means of monitoring and analyzing plant growth in a systematic manner.

5.2. Conclusion

After collecting the data, the researchers drawn the following conclusions:

1. The system enabled IoT connectivity through modules, sensors, webcams, Raspberry Pi 4B, and Arduino Mega for data collection and analysis. Furthermore, the establishment of a connection to a backend server enabled the storage of data and facilitated real-time data viewing through the web application. This integration of technologies and the efficient connection to the backend server resulted in a robust and functional IoT system for data collection, monitoring, and real-time data visualization.
2. The implementation of the CNN model YOLOv4 allowed the system to accurately calculate the surface areas of plants, enabling effective tracking and measurement

of both plant and fish growth over time. Additionally, the system demonstrated the capability to detect lettuce head infections. In terms of its measurement calculations, the proposed system exhibited no significant difference when compared to the Manual Computation method. These results indicate the reliability and accuracy of the proposed system, as it achieved measurements comparable to the manual approach.

3. The deployment of a web application for continuous monitoring of the system's growth and optimal environment resulted in improved crop quality and accelerated fish growth. It facilitated real-time data visualization and allowed for the monitoring of processed image data through graphical representations. Generally, it served as a comprehensive platform, offering essential information about the system and presenting real-time data and analysis through a user-friendly data dashboard. This enabled users to easily access and interpret the collected data."
4. Using Likert scale, the data from the survey shows that the technical experts strongly agree with the functionality, usability, portability, and reliability. This feedback validated the effectiveness and practicality of the system in meeting the requirements of technical experts, reinforcing the value of the integrated technologies for precise and comprehensive plant growth monitoring and analysis.

In the end, the study demonstrated that the combination of Raspberry Pi, web camera, and image processing techniques provided a reliable and efficient approach for monitoring and analyzing plant growth in a systematic manner.

5.3. Recommendations

The following recommendations are provided as possible courses of action in light of the actual results and conclusions drawn from the study:

1. Add more cameras to have a better-quality picture and to easily detect the plants and fishes.
2. Provide a better device that can capture clear images even in murky and turbid water conditions.
3. Implement an energy-efficient alternative, including the utilization of sustainable energy sources such as solar power, to supply the intelligent aquaponics system with power.
4. Integration of a nutrient management system for automated dosing and monitoring of nutrients. To help prevent the possible potential risk of imbalance nutrients.

References:

[1] S. Kaza, L. Yao, P. Bhada-Tata, V. F. Woerden, and K. Ionkova, *What a Waste 2.0: A global snapshot of Solid Waste Management to 2050*. Washington, DC: World Bank Group, 2018.

[2] World Bank Group, "Global waste to grow by 70 percent by 2050 unless urgent action is taken: World Bank Report," *World Bank*, September 24, 2018. [Online]. Available: https://www.worldbank.org/en/news/press-release/2018/09/20/global-waste-to-grow- by-70-percent-by-2050-unless-urgent-action-is-taken-world-bank-report.[Accessed: June 6, 2022].

[3] Dispatch, "'Throwing 'Away' to somewhere: The Global Community is overwhelmed by a growing trash problem," *The NYU Dispatch*, February 10, 2020. [Online]. Available: https://wp.nyu.edu/dispatch/2020/02/10/throwing-away-to-somewhere-the-global- community-is-overwhelmed-by-a-growing-trash-problem/. [Accessed: June 6, 2022].

[4] World Bank Group, "Solid Waste Management," *World Bank*, May 31, 2022. [Online]. Available: https://www.worldbank.org/en/topic/urbandevelopment/brief/solid-waste-management. [Accessed: June 6, 2022].

[5] L. A. Guerrero, G. Maas, and W. Hogland, "Solid waste management challenges for cities in developing countries," *Waste Management*, vol. 33, no. 1, pp. 220–232, 2013.

[6] B. Sarmiento, "Plastic trash from the 'Sachet economy' chokes the Philippines' seas," *Mongabay Environmental News*, October 30, 2018. [Online]. Available:

https://news.mongabay.com/2018/10/plastic-trash-from-the-sachet-economy-chokes- the-philippines-seas/. [Accessed: June 6, 2022].

[7] A. Masigan, “A solution to Metro Manila’s Garbage Problem ,” *BusinessWorld*, June 30, 2019. [Online]. Available: https://www.bworldonline.com/editors-picks/2019/06/30/239416/a-solution-to-metro-manilas-garbage-problem/. [Accessed: June 6, 2022].

[8] Mitsubishi Securities : Clean Energy Finance Committee, “Project Design Document for PNOC EC Payatas Landfill Gas to Energy Project in the Philippines Mitsubishi Securities Clean Energy Finance Committee,” no. March, p. 7, 2004.

[9] A. Pagano, “‘Promised Land’ garbage landslide kills at least 200 in the Philippines,” *World Socialist Web Site*, July 21, 2000. [Online]. Available: https://www.wsws.org/en/articles/2000/07/phil-j21.html. [Accessed: May 1, 2022].

[10] M. C. Arayata, “Dost allots P2.4-M for Payatas Disposal Facility,” *Philippine News Agency,* November 15, 2021. [Online]. Available: https://www.pna.gov.ph/articles/1159828. [Accessed April 30, 2022].

[11] T. Y. Kyaw and A. K. Ng, “Smart Aquaponics System for Urban Farming,” Energy Procedia, vol. 143, pp. 342–347, 2017, doi: 10.1016/j.egypro.2017.12.694.

[12] T. Khaoula, R. A. Abdelouahid, I. Ezzahoui, and A. Marzak, “Architecture design of monitoring and controlling of IoT-based aquaponics system powered by solar energy,” Procedia Comput. Sci., vol. 191, pp. 493–498, 2021, doi: 10.1016/j.procs.2021.07.063.

[13] K. T. Lapuz and R. J. Corvera, “AQUADROID : AN APP FOR AQUAPONICS

CONTROL AND MONITORING AQUADROID : AN APP FOR AQUAPONICS CONTROL AND MONITORING College of Engineering , Technological University of the Philippines , Manila , Philippines Allen De Guzman College of Engineering , Technolo," no. March, 2018.

[14] K. S. T. Lean et al., "HyLo: Implementation of LoRaWAN in an Automated Hydroponics System," 2021 IEEE Int. Conf. Mob. Networks Wirel. Commun. ICMNWC 2021, pp. 4–9, 2021, doi: 10.1109/ICMNWC52512.2021.9688540.

[15] L. K. Tolentino et al., "IoT-based automated water monitoring and correcting modular device via LoRaWAN for aquaculture," Int. J. Comput. Digit. Syst., vol. 10, no. 1, pp. 533–544, 2021, doi: 10.12785/IJCDS/100151.

[16] L. K. S. Tolentino et al., "Development of an IoT-based Intensive Aquaculture Monitoring System with Automatic Water Correction," Int. J. Comput. Digit. Syst., vol. 10, no. 1, pp. 1355–1365, 2021, doi: 10.12785/ijcds/1001120.

[17] P. Muthumari, A. M. Kulkarni, and C. M. Kulkarni, "IoT Application in Smart Libraries," Int. J. Res. Libr. Sci., vol. 7, no. 3, p. 113, 2021, doi: 10.26761/ijrls.7.3.2021.1432.

[18] P. Suresh, J. V. Daniel, V. Parthasarathy and R. H. Aswathy, "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment," *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, 2014, pp. 1-8, doi: 10.1109/ICSEMR.2014.7043637.

[19] K. Gusmeroli, S., Haller, S., Harrison, M., Kalaboukas, K., Tomasella, M., Vermesan, O., & Wouters, Vision and challenges for realizing the internet of

things, vol. 1, no. APRIL. 2009.

[20] L. M. Camarinha-Matos, J. Goes, L. Gomes, and J. Martins, “Contributing to the internet of things,” IFIP Adv. Inf. Commun. Technol., vol. 394, no. January, pp. 3– 12, 2013, doi: 10.1007/978-3-642-37291-9_1.

[21] V. Gupta, D. G. Simmons, and S. Labs, “Building the Web of Things with Sun SPOTs,” Access, p. 375, 2010.

[22] J. Romkey, “The Toast of The IoT,” IEEE Consum. Electron. Mag., vol. 6, no. 1, pp. 116–119, 2017.

[23] R. Heuser, “Radio Frequency,” Chronic Total Occlusions A Guid. to Recanalization, no. June, pp. 121–135, 2009, doi: 10.1002/9781444308037.ch19.

[24] A. Abdul Qawy, E. Magesh, and S. Tadisetty, “The Internet of Things (IoT): An Overview,” Int. J. Eng. Res. Appl., vol. 5, no. 12, pp. 71–82, 2015, [Online]. Available: https://crsreports.congress.gov

[25] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, and B. Stiller, “Landscape of IoT security,” Comput. Sci. Rev., vol. 44, no. May, p. 100467, 2022, doi: 10.1016/j.cosrev.2022.100467.

[26] L. Horwitz, “Internet of things (IOT) products & solutions,” *Cisco*, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html. [Accessed: June 6, 2022].

[27] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, “Internet of things security: A top-down survey,” Comput. Networks, vol. 141, no. March, pp. 199–221, 2018, doi: 10.1016/j.comnet.2018.03.012.

[28] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, “Internet of Things

security: A survey," J. Netw. Comput. Appl., vol. 88, no. April, pp. 10–28, 2017, doi: 10.1016/j.jnca.2017.04.002.

[29] A. D. Jurcut, P. Ranaweera, and L. Xu, Introduction to IoT Security, no. April 2020. 2019. doi: 10.1002/9781119471509.w5gref260.

[30] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," J. Electr. Comput. Eng., vol. 2017, 2017, doi: 10.1155/2017/9324035.

[31] M. Farhan, M.Pu'ad, et al, "IoT based water quality monitoring system for aquaponics," Journal of Physics: Conference Series ICTEC 2019, pp. 3-4, 2019, doi:10.1088/1742-6596/1502/1/012020

[32] [T. Khaoula, R. A. Abdelouahid, I. Ezzahoui, and A. Marzak, "Architecture design of monitoring and controlling of IoT-based aquaponics system powered by solar energy," *Procedia Comput. Sci.*, vol. 191, pp. 493–498, 2021, doi: 10.1016/j.procs.2021.07.063

[33] A. Dutta, et al., "IoT based Aquaponics Monitoring System," 1st KEC Conference Proceedings, vol. 1, pp. 75-77, 2020, ISBN 978-9937-0-4872-9

[34] J. Y. Lu, C. L. Chang, and Y. F. Kuo, "Monitoring Growth Rate of Lettuce Using Deep Convolutional Neural Networks," An ASABE Meeting Presentation, 1900341. doi: 10.13031/aim.201900341

[35] . S. N. Pauzi, M. G. Hassan, N. Yusoff, N. H. Harun, A. H. Abu Bakar, and B. C. Kua, "A review on image processing for fish disease detection," J. Phys. Conf. Ser., vol. 1997, no. 1, 2021, doi: 10.1088/1742-6596/1997/1/012042.

[36] M. I. Pavel, S. M. Kamruzzaman, S. S. Hasan and S. R. Sabuj, "An IoT Based

Plant Health Monitoring System Implementing Image Processing," *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019, pp. 299-303, doi: 10.1109/CCOMS.2019.8821782.

[37] S. Porob, “Plant Health Monitoring using Digital Image Processing,” vol. 3, no.7, pp. 147–151, 2017.

[38] A. Rodriguez, A. J. Rico-Diaz, J. R. Rabuñal, J. Puertas, and L. Pena, “Fish monitoring and sizing using computer vision,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9108, pp. 419–428, 2015, doi: 10.1007/978-3-319-18833-1_44.

[39] N. Petrellis, “Measurement of fish morphological features through image processing and deep learning techniques,” Appl. Sci., vol. 11, no. 10, 2021, doi: 10.3390/app11104416.

[40] A. Patil and M. Rane, “Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition,” Smart Innov. Syst. Technol., vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.

[41] J. W. Chen, et al, “A Smartphone-Based Application for Scale Pest Detection Using Multiple-Object Detection Methods,” Electronics 2021, 10, 372. doi: 10.3390/electronics10040372

[42] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” aeXiv.2003.10934v1, no. April, 2020

[43] B. Yang, Z. Gao, Y. Gao, and Y. Zhu, “Rapid Detection and Counting of Wheat Ears in the Field Using YOLOv4 with Attention Module,” Agronomy 2021, 11, 1202. doi: 10.3390/agronomy11061202.

[44] B. Tugrul, E. Elfatimi, and R. Eryigit, “Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review,” Agriculture 2022, 12, 1192. doi: 10.3390/agriculture12081192

[45] S. M. Hassan, A. K. Maji, M. Jasinki, Z. Leonowicz, and E. Jasinska, “Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach,” Electronics 2021, 10, 1388. doi: 10.3390/electronics10121388

[46] L. Zhang, Z. Xu, D. Xu, J. Ma, Y. Chen, and Z. Fu, “Growth monitoring of greenhouse lettuce based on a convolutional neural network,” Horticulture Research (2020) 7:124 doi: 10.1038/s41438-020-00345-6

[47] J. Y. Lu, C. L. Chang, and Y. F. Kuo, “Monitoring Growth Rate of Lettuce Using Deep Convolutional Neural Networks,” An ASABE Meeting Presentation, 1900341. doi: 10.13031/aim.201900341

[48] Z. Wei, B. Zhang, and B. Zhang, “Object Dimension Measurement Based on Mask R-CNN,” An ASABE Meeting Presentation, 1900341. doi: 10.13031/aim.201900341

[49] P. Sahu, A. Chug, A. P. Singh, D. Singh, and R. P. Singh, “Deep Learning Models for Beans Crop Diseases: Classification and Visualization Techniques,” International Journal of Modern Agriculture, Volume 10, No.1, 2021 ISSN: 2305-7246

[50] Coulibaly, S.; Kamsu-Foguem, B.; Kamissoko, D.; Traore, D., “Deep neural networks with transfer learning in millet crop images.Deep neural networks with transfer learning in millet crop images.,” IComput. Ind. 2019, 108, 115–120. doi: 10.1016/j.compind.2019.02.003

[51] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park., "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition," Sensors in Agriculture 2017, 17(9), 2022. doi: 10.3390/s17092022

[52] M. Türkoglu and D. Hanbay, "Plant disease and pest detection using deep learning-based features," Turkish Journal of Electrical Engineering and Computer Sciences, vol 27, no. 3, doi: 10.3906/elk-1809-181

[53] A. Patani, I. Pardikar, P. Doshi, S. Rodge, and S. Khachane, "Plant Leaf Recognition and Disease Detection Using GoogLeNet," JETIR May 2021, Volume 8, Issue 5 ISSN-2349-5162

[54] S. P. Mohanty, D. P. Hughes, and M. Salathe, "Using Deep Learning for Image-Based Plant Disease Detection," Front. Plant Sci. 7:1419. doi: 10.3389/fpls.2016.01419

[55] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant Disease Detection and Classification by Deep Learning," Plants 2019, 8, 468, doi: 10.3390/plants8110468

[56] K. Neupane and F. Baysal-Gurel, "Automatic Identification and Monitoring of Plant Diseases Using Unmanned Aerial Vehicles: A Review," Remote Sens. 2021, 13, 3841. doi: 10.3390/rs13193841

ANNEXES

ANNEX I: YOLOv4 Detection Codes

```
#mount drive
%cd ..
from google.colab import drive
drive.mount('/content/gdrive')
# this creates a symbolic link so that now the path /content/gdrive/My\ Drive/ is equal to /mydrive
!ln -s /content/gdrive/My\ Drive/ /mydrive

# list the contents of /mydrive
!ls /mydrive

#Navigate to /mydrive/yolov4
%cd /mydrive/yolov4

!git clone https://github.com/AlexeyAB/darknet

# change makefile to have GPU and OPENCV enabled
# also set CUDNN, CUDNN_HALF and LIBSO to 1
%cd darknet/
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
!sed -i 's/LIBSO=0/LIBSO=1/' Makefile

# build darknet
!make

# Clean the data and cfg folders first except the labels folder in data which is required
%cd data/
!find -maxdepth 1 -type f -exec rm -rf {} \;
%cd ..
%rm -rf cfg/
%mkdir cfg

!unzip /mydrive/yolov4/obj.zip -d data/

# Copy the yolov4-custom.cfg file so that it is now in /darknet/cfg/ folder
!cp /mydrive/yolov4/yolov4-custom.cfg cfg

# verify if your custom file is in cfg folder
!ls cfg/

# Copy the obj.names and obj.data files from your drive so that they are now in /darknet/data/
folder
!cp /mydrive/yolov4/obj.names data
!cp /mydrive/yolov4/obj.data  data
# verify if the above files are in data folder
!ls data/
```

```
# Copy the process.py file to the current darknet directory
!cp /mydrive/yolov4/process.py .

# run process.py ( this creates the train.txt and test.txt files in our darknet/data folder )
!python process.py

# list the contents of data folder to check if the train.txt and test.txt files have been created
!ls data/

# Download the yolov4 pre-trained weights file
!wget
https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.
137

# train your custom detector! (uncomment %%capture below if you run into memory issues or
your Colab is crashing)
# %%capture
!./darknet detector train data/obj.data cfg/yolov4-custom.cfg yolov4.conv.137 -dont_show -map

#to restart training your custom detector where you left off(using the weights that were saved
last)
!./darknet detector train data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-
custom_last.weights -dont_show -map

#Check Performance (define helper function imShow)
def imShow(path):
  import cv2
  import matplotlib.pyplot as plt
  %matplotlib inline

  image = cv2.imread(path)
  height, width = image.shape[:2]
  resized_image = cv2.resize(image,(3*width, 3*height), interpolation = cv2.INTER_CUBIC)

  fig = plt.gcf()
  fig.set_size_inches(18, 10)
  plt.axis("off")
  plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
  plt.show()

#Training Chart (only works if the training does not get interrupted)
imShow('chart.png')

#You can check the mAP for all the saved weights to see which gives the best results ( xxxx here
is the saved weight number like 4000, 5000 or 6000 and so on )
!./darknet detector map data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-
custom_4000.weights -points 0

#Test your Custom Object Detector (set your custom cfg to test mode)
%cd cfg
!sed -i 's/batch=64/batch=1/' yolov4-custom.cfg
```

```
!sed -i 's/subdivisions=16/subdivisions=1/' yolov4-custom.cfg
%cd ..

# run your custom detector on an image with this command (upload an image to your google drive to test, the thresh flag sets the minimum accuracy required for object detection)
!./darknet detector test data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-custom_best.weights /mydrive/mask_test_images/image1.jpg -thresh 0.3
imShow('predictions.jpg')

#run detector on images captured by webcam for your custom YOLOv4 trained model
from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode

def take_photo(filename='photo.jpg', quality=0.8):
  js = Javascript('''
    async function takePhoto(quality) {
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);

      const video = document.createElement('video');
      video.style.display = 'block';
      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);
      div.appendChild(video);
      video.srcObject = stream;
      await video.play();

      // Resize the output to fit the video element.
      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.
      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');
      canvas.width = video.videoWidth;
      canvas.height = video.videoHeight;
      canvas.getContext('2d').drawImage(video, 0, 0);
      stream.getVideoTracks()[0].stop();
      div.remove();
      return canvas.toDataURL('image/jpeg', quality);
    }
    ''')
  display(js)
  data = eval_js('takePhoto({})'.format(quality))
  binary = b64decode(data.split(',')[1])
  with open(filename, 'wb') as f:
    f.write(binary)
```

```
  return filename

from IPython.display import Image
try:
  filename = take_photo()
  print('Saved to {}'.format(filename))

  # Show the image which was just taken.
  display(Image(filename))
except Exception as err:
  # Errors will be thrown if the user does not have a webcam or if they do not
  # grant the page permission to access it.
  print(str(err))
!./darknet detector test data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-custom_best.weights photo.jpg -thresh 0.5
imShow('predictions.jpg')

# run your custom detector on a video with this command (upload a video to your google drive to test, the thresh flag sets the minimum accuracy required for object detection).This saves the output video with the detections in your output path
!./darknet detector demo data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-custom_best.weights -dont_show /mydrive/mask_test_videos/test3.mp4 -i 0 -out_filename /mydrive/mask_test_videos/finalresult.avi

# Run Detector on a Live Webcam
# Code from theAIGuysCode Github (https://github.com/theAIGuysCode/YOLOv4-Cloud-Tutorial/blob/master/yolov4_webcam.ipynb)
# Adjusted for my custom YOLOv4 trained weights, config and obj.data files
# import dependencies

from IPython.display import display, Javascript, Image
from google.colab.output import eval_js
from google.colab.patches import cv2_imshow
from base64 import b64decode, b64encode
import cv2
import numpy as np
import PIL
import io
import html
import time
import matplotlib.pyplot as plt
%matplotlib inline

# import darknet functions to perform object detections
from darknet import *
# load in our YOLOv4 architecture network
network, class_names, class_colors = load_network("cfg/yolov4-custom.cfg", "data/obj.data", "/mydrive/yolov4/training/yolov4-custom_best.weights")
width = network_width(network)
height = network_height(network)
```

```
# darknet helper function to run detection on image
def darknet_helper(img, width, height):
  darknet_image = make_image(width, height, 3)
  img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
  img_resized = cv2.resize(img_rgb, (width, height),
                              interpolation=cv2.INTER_LINEAR)

  # get image ratios to convert bounding boxes to proper size
  img_height, img_width, _ = img.shape
  width_ratio = img_width/width
  height_ratio = img_height/height

  # run model on darknet style image to get detections
  copy_image_from_bytes(darknet_image, img_resized.tobytes())
  detections = detect_image(network, class_names, darknet_image)
  free_image(darknet_image)
  return detections, width_ratio, height_ratio

# function to convert the JavaScript object into an OpenCV image
def js_to_image(js_reply):
  """
  Params:
          js_reply: JavaScript object containing image from webcam
  Returns:
          img: OpenCV BGR image
  """
  # decode base64 image
  image_bytes = b64decode(js_reply.split(',')[1])
  # convert bytes to numpy array
  jpg_as_np = np.frombuffer(image_bytes, dtype=np.uint8)
  # decode numpy array into OpenCV BGR image
  img = cv2.imdecode(jpg_as_np, flags=1)
  return img

# function to convert OpenCV Rectangle bounding box image into base64 byte string to be
overlayed on video stream
def bbox_to_bytes(bbox_array):
  """
  Params:
          bbox_array: Numpy array (pixels) containing rectangle to overlay on video stream.
  Returns:
        bytes: Base64 image byte string
  """
  # convert array into PIL image
  bbox_PIL = PIL.Image.fromarray(bbox_array, 'RGBA')
  iobuf = io.BytesIO()
  # format bbox into png for return
  bbox_PIL.save(iobuf, format='png')
  # format return string
  bbox_bytes = 'data:image/png;base64,{}'.format((str(b64encode(iobuf.getvalue()), 'utf-8')))
```

```
  return bbox_bytes

# JavaScript to properly create our live video stream using our webcam as input
def video_stream():
  js = Javascript('''
    var video;
    var div = null;
    var stream;
    var captureCanvas;
    var imgElement;
    var labelElement;

    var pendingResolve = null;
    var shutdown = false;

    function removeDom() {
       stream.getVideoTracks()[0].stop();
       video.remove();
       div.remove();
       video = null;
       div = null;
       stream = null;
       imgElement = null;
       captureCanvas = null;
       labelElement = null;
    }

    function onAnimationFrame() {
      if (!shutdown) {
        window.requestAnimationFrame(onAnimationFrame);
      }
      if (pendingResolve) {
        var result = "";
        if (!shutdown) {
          captureCanvas.getContext('2d').drawImage(video, 0, 0, 640, 480);
          result = captureCanvas.toDataURL('image/jpeg', 0.8)
        }
        var lp = pendingResolve;
        pendingResolve = null;
        lp(result);
      }
    }

    async function createDom() {
      if (div !== null) {
        return stream;
      }

      div = document.createElement('div');
      div.style.border = '2px solid black';
      div.style.padding = '3px';
```

```
  div.style.width = '100%';
  div.style.maxWidth = '600px';
  document.body.appendChild(div);

  const modelOut = document.createElement('div');
  modelOut.innerHTML = "<span>Status:</span>";
  labelElement = document.createElement('span');
  labelElement.innerText = 'No data';
  labelElement.style.fontWeight = 'bold';
  modelOut.appendChild(labelElement);
  div.appendChild(modelOut);

  video = document.createElement('video');
  video.style.display = 'block';
  video.width = div.clientWidth - 6;
  video.setAttribute('playsinline', '');
  video.onclick = () => { shutdown = true; };
  stream = await navigator.mediaDevices.getUserMedia(
      {video: { facingMode: "environment"}});
  div.appendChild(video);

  imgElement = document.createElement('img');
  imgElement.style.position = 'absolute';
  imgElement.style.zIndex = 1;
  imgElement.onclick = () => { shutdown = true; };
  div.appendChild(imgElement);

  const instruction = document.createElement('div');
  instruction.innerHTML =
      '<span style="color: red; font-weight: bold;">' +
      'When finished, click here or on the video to stop this demo</span>';
  div.appendChild(instruction);
  instruction.onclick = () => { shutdown = true; };

  video.srcObject = stream;
  await video.play();

  captureCanvas = document.createElement('canvas');
  captureCanvas.width = 640; //video.videoWidth;
  captureCanvas.height = 480; //video.videoHeight;
  window.requestAnimationFrame(onAnimationFrame);

  return stream;
}
async function stream_frame(label, imgData) {
  if (shutdown) {
    removeDom();
    shutdown = false;
    return '';
  }
```

```
      var preCreate = Date.now();
      stream = await createDom();

      var preShow = Date.now();
      if (label != "") {
        labelElement.innerHTML = label;
      }

      if (imgData != "") {
        var videoRect = video.getClientRects()[0];
        imgElement.style.top = videoRect.top + "px";
        imgElement.style.left = videoRect.left + "px";
        imgElement.style.width = videoRect.width + "px";
        imgElement.style.height = videoRect.height + "px";
        imgElement.src = imgData;
      }

      var preCapture = Date.now();
      var result = await new Promise(function(resolve, reject) {
        pendingResolve = resolve;
      });
      shutdown = false;

      return {'create': preShow - preCreate,
              'show': preCapture - preShow,
              'capture': Date.now() - preCapture,
              'img': result};
    }
    ''')

  display(js)

def video_frame(label, bbox):
  data = eval_js('stream_frame("{}", "{}")'.format(label, bbox))
  return data

# start streaming video from webcam
video_stream()
# label for video
label_html = 'Capturing...'
# initialze bounding box to empty
bbox = ''
count = 0
while True:
    js_reply = video_frame(label_html, bbox)
    if not js_reply:
        break

    # convert JS response to OpenCV Image
    frame = js_to_image(js_reply["img"])
```

```
    # create transparent overlay for bounding box
    bbox_array = np.zeros([480,640,4], dtype=np.uint8)

    # call our darknet helper on video frame
    detections, width_ratio, height_ratio = darknet_helper(frame, width, height)

    # loop through detections and draw them on transparent overlay image
    for label, confidence, bbox in detections:
      left, top, right, bottom = bbox2points(bbox)
      left, top, right, bottom = int(left * width_ratio), int(top * height_ratio), int(right *
width_ratio), int(bottom * height_ratio)
      bbox_array = cv2.rectangle(bbox_array, (left, top), (right, bottom), class_colors[label], 2)
      bbox_array = cv2.putText(bbox_array, "{} [{:.2f}]".format(label, float(confidence)),
                        (left, top - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                        class_colors[label], 2)

    bbox_array[:,:,3] = (bbox_array.max(axis = 2) > 0 ).astype(int) * 255
    # convert overlay of bbox into bytes
    bbox_bytes = bbox_to_bytes(bbox_array)
    # update bbox so next frame gets new overlay
    bbox = bbox_bytes
```

ANNEX II: Web Development Codes

INDEX

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>AQCLOUD</title>

   <link rel="stylesheet" href="style.css">
   <link rel="preconnect" href="https://fonts.googleapis.com">
   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
   <link
href="https://fonts.googleapis.com/css2?family=Nunito:wght@300;400;600&display=swap"
rel="stylesheet">
</head>

<body>
   <p class="login-lbl">Log <span>in</span></p>
   <p class="login-signup">New User? Sign up <a href="signup.html" class="login-
a">Here.</a></p>

   <div class="login-card">
       <div id="form_content_inner_container">
                   <input type="email" id="email" placeholder="Email:">
                   <input type="password" id="password"
placeholder="Password:">
               </div>
   </div>

   <div class="center">
       <button class = login-btn onclick="login()"> Login</button>
   </div>
</body>

 <!-- The core Firebase JS SDK is always required and must be listed first -->
   <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-app.js"></script>

   <!-- TODO: Add SDKs for Firebase products that you want to use
       https://firebase.google.com/docs/web/setup#available-libraries -->
   <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-auth.js"></script>
   <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-database.js"></script>

   <!-- Our script must be loaded after firebase references -->
   <script src="index1.js"></script>

</html>
```

LOG IN AND SIGN UP

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>AQCLOUD</title>

   <link rel="stylesheet" href="style.css">
   <link rel="preconnect" href="https://fonts.googleapis.com">
   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
   <link
href="https://fonts.googleapis.com/css2?family=Nunito:wght@300;400;600&display=swap"
rel="stylesheet">
</head>

<body>
   <p class="signup-lbl">Sign <span>up</span></p>
   <p class="signup-login">Existing User? Log in <a href="index.html" class="signup-
a">Here.</a></p>

   <div class="signup-card">
     <div id="form_content_inner_container">
                <input type="text" id="full_name" placeholder="Full name:">
                <input type="email" id="email" placeholder="Email:">
                <input type="password" id="password"
placeholder="Password:">
              </div>
   </div>

   <div class="center">
       <button class = signup-btn onclick="register()">Register</button>
   </div>
</body>

 <!-- The core Firebase JS SDK is always required and must be listed first -->
   <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-app.js"></script>

   <!-- TODO: Add SDKs for Firebase products that you want to use
       https://firebase.google.com/docs/web/setup#available-libraries -->
   <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-auth.js"></script>
   <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-database.js"></script>

   <!-- Our script must be loaded after firebase references -->
   <script src="index1.js"></script>
</html>
```

JS FILE

```
document.getElementById('menu-icon').addEventListener('click', () => {
    document.getElementById('sidebar').style.display = "block";
    document.getElementById('sidebar').style.zIndex = "999";
});

document.getElementById('menu-icn').addEventListener('click', () => {
    document.getElementById('sidebar').style.display = "none";
    document.getElementById('sidebar').style.zIndex = "none";
});

// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyABJLUeCY0rOAzF9Nbb2SQ7ujA7pOf8IAY",
  authDomain: "aquaponics-data-project.firebaseapp.com",
  databaseURL: "https://aquaponics-data-project-default-rtdb.firebaseio.com",
  projectId: "aquaponics-data-project",
  storageBucket: "aquaponics-data-project.appspot.com",
  messagingSenderId: "933099964610",
  appId: "1:933099964610:web:f5621605fa6962c20d535b",
  measurementId: "G-6W6KTPFEN5"
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
// Initialize variables
const auth = firebase.auth()
const database = firebase.database()

// Set up our register function
function register () {
  // Get all our input fields
  email = document.getElementById('email').value
  full_name = document.getElementById('full_name').value
   password = document.getElementById('password').value

  // Validate input fields
  if (validate_email(email) == false || validate_password(password) == false) {
    alert('Please fill out all fields.')
    return
    // Don't continue running the code
  }
  if (validate_field(full_name) == false) {
    alert('Please fill out all fields.')
    return
  }

  // Move on with Auth
  auth.createUserWithEmailAndPassword(email, password)
  .then(function() {
```

```
    // Declare user variable
    var user = auth.currentUser

    // Add this user to Firebase Database
    var database_ref = database.ref()

    // Create User data
    var user_data = {
      email : email,
      full_name : full_name,
      last_login : Date.now()
    }

    // Push to Firebase Database
    database_ref.child('users/' + user.uid).set(user_data)

    // Done
    alert('Registration Successfull!')
  })
  .catch(function(error) {
    // Firebase will use this to alert of its errors
    var error_code = error.code
    var error_message = error.message
    alert(error_message)
  })
}
// Set up our login function
function login () {
  // Get all our input fields
  email = document.getElementById('email').value
  password = document.getElementById('password').value

  // Validate input fields
  if (validate_email(email) == false && validate_password(password) == false) {
    alert('Email or Password is Incorrect.')
    return
    // Don't continue running the code
  }

  auth.signInWithEmailAndPassword(email, password)
  .then(function() {
    // Declare user variable
    var user = auth.currentUser

    // Add this user to Firebase Database
    var database_ref = database.ref()

    // Create User data
    var user_data = {
      last_login : Date.now()
    }
```

```
    // Push to Firebase Database
    database_ref.child('users/' + user.uid).update(user_data)

    // Done
        document.location.href = 'home.html'
    alert('User Logged In Successfully!')
  })
  .catch(function(error) {
    // Firebase will use this to alert of its errors
    var error_code = error.code
    var error_message = error.message
    alert(error_message)
  })
}

// Validate Functions
function validate_email(email) {
  expression = /^[^@]+@\w+(\.\w+)+\w$/
  if (expression.test(email) == true) {
    // Email is good
    return true
  } else {
    // Email is not good
    return false
  }
}

function validate_password(password) {
  // Firebase only accepts lengths greater than 6
  if (password < 6) {
    return false
  } else {
    return true
  }
}

function validate_field(field) {
  if (field == null) {
    return false
  }

  if (field.length <= 0) {
    return false
  } else {
    return true
  }
}
```

CSS

```
body {
   font-family: 'Nunito', sans-serif;
   padding: 0px;
   margin: 0px;
}

/* common */
.center {
   text-align: center;
}

video {
   width: 100%;
}

button:hover {
   cursor: pointer;
}

nav > .d-flex  {
   padding: 1%;
   position: fixed;
   background-color: #FAF5F5;
   z-index: 99;
   width: 100%;
}

nav > div > .nav-list {
   text-align: right;
   padding-right: 50px;
}

nav > .nav-list > div > a {
   font-size: 20px;
}

a {
   text-decoration: none;
   color: black;
}

nav {
   background-image: url('img/background.png');
   height: 100vh;
   background-repeat: no-repeat;
   background-size: 100%;
}
```

```
nav.about {
   background-image: url('img/about us.png');
   height: 70vh;
   background-repeat: no-repeat;
   background-size: 100%;
}

nav.contact {
   background-image: url('img/contact us.png');
   height: 70vh;
   background-repeat: no-repeat;
   opacity: 0.8;
   background-color: grey;
   background-blend-mode: multiply;
}

nav.monitoring {
   background-image: url('img/monitoring.png');
   height: 70vh;
   background-repeat: no-repeat;
   background-size: 100%;
}

.nav-text {
   color: white;
   text-align: center;
   padding-top: 150px;
}

.nav-text > h1 {
   font-size: 80px;
   margin: 0;
}

.nav-text > p {
   font-size: 20px;
   margin: 0;
}

.header {
   font-size: 40px;
}

.header > span {
   color: #3DB67D;
}

.active {
   border-bottom: 1px solid black;
}
```

```css
.nav-list > div > a {
   margin-right: 50px;
   margin-top: 15px;
}

.sidebar {
   background-color: #D8FACA;
   width: 25%;
   height: 100vh;
   position: fixed;
}

.sidebar > ul {
   padding-left: 20px;
}

.nav-active {
   list-style-type: none;
   border-top-left-radius: 25px;
   border-bottom-left-radius: 25px;
   padding-left: 15px;
   margin-top: 10px;
   margin-left: -15px;
}

.nav-active {
   background-color: white;
}

ul {
   list-style-type: none;
}

.sub-menu {
   margin-left: 20px;
}

.d-flex {
   display: flex;
}

.menu-icon {
   width: 25px;
   padding-left: 20px;
   margin-top: 15px;
}

.menu-icon-about {
   width: 45px;
   padding-left: 20px;
   margin-top: 6px;
```

```
}

.menu-icon:hover, .menu-icon-about:hover {
   cursor: pointer;
}

.img-bg {
   background-image: url('img/tilapia.png');
   height: 300px;
   background-repeat: no-repeat;
   background-size: cover;
}

.last {
   background-image: url('img/last.png');
   min-height: 100vh;
   background-repeat: no-repeat;
   background-size: cover;
   background-position: center;
}

.container {
   padding: 3% 10%;
}

.container-small {
   padding: 3% 2%;
}

.container-small > .row > .col {
   padding: 1%;
}

.container > h1 {
   text-align: center;
}

.row {
   display: flex;
}

.col {
   flex: 50%;
   padding: 2%;
}

#scroll {
   width: 750px;
   display: flex;
   overflow-x: scroll;
}
```

```
.col > .card > #scroll {
   width: 180px;
   height: 190px;
}

.card {
   display: flex;
   flex-wrap: nowrap;
}

.col1 {
   flex: 100%;
}

.col-4 {
   flex: 33.33%;
   padding: 2%;
   text-align: center;
}

.col-6 {
   flex: 50%;
   padding: 2%;
   text-align: center;
}

.about-last {
   min-height: 70vh !important;
}

.text-left {
   text-align: left;
}

.icons {
   width: 70%;
}

.description {
   width: 70%;
   margin: 0 auto;
}

img {
   width: 100%;
}

.col > p {
   text-align: justify;
   line-height: 32px;
```

```
}

.logo {
   text-align: center;
}

.logo > img {
   width: 50%;
}

.w-50 {
   width: 50%;
}

.home-icon {
   width: 10%;
   height: 10%;
   margin-top: 13px;
   margin-right: 15px;
}

.nav-logo {
   width: 8%
}

.logo-nav {
   width: 100%;
}

.nav-list > .d-flex {
   margin-left: auto;
}

.sidebar > .menu-icn {
   padding: 5% 5%;
}

.mt-60 {
   margin-top: 60px;
}

.mt-100 {
   margin-top: 100px;
}

.mb-500 {
   margin-bottom: 500px;
}

.login-lbl, .signup-lbl {
   text-align: center;
```

```
    margin-top: 100px;
    font-weight: 900;
    font-size: 40px;
    margin-bottom: 0px;
}

.login-lbl > span, .signup-lbl > span {
    color: #3DB67D;
}

.login-a, .signup-a {
    color: #3DB67D;
    font-weight: 700 !important;
}

.login-signup, .signup-login {
    text-align: center;
    margin: 0px;
    margin-bottom: 50px;
}

.login-card, .signup-card {
    background-color: #3DB67D;
    padding: 40px;
    width: 30%;
    text-align: center;
    margin: 0 auto;
}

.login-card > div > p, .signup-card > div > p {
    color: white;
    text-align: left;
}
::placeholder {

  color: green;
  border-bottom-style:groove ;
}

.login-card > div > input, .signup-card > div > input {
    width: 96%;
    background-color: #D9D9D9;
    border: none;
    padding: 10px;
}

.login-btn, .signup-btn, .view-btn{
    background-color: #3DB67D;
    border: none;
    padding: 10px 40px;
    color: white;
```

```css
    margin-top: 50px;
}

video {
    margin: 0 auto;
}

.contact-card {
    background-color: #3DB67D;
    padding: 10% 5%;
}

.contact-card > img {
    width: 50px;
}

.contact-card > .contact-lbl {
    font-weight: bold;
}

.contact-card > .contact-description {
    font-weight: light;
    color: white;
}

.contact-detail {
    margin-top: 15px;
    color: #D8FACA;
    font-size: 20px;
}

.plant-monitoring-nav {
    background-color: #F3FAE7;
    padding: 15px 40px;
    max-height: 50px;
    position: absolute;
    width: 94%;
}

.fish-monitoring-nav {
    background-color: #E9F4FB;
    padding: 15px 40px;
    max-height: 50px;
    position: absolute;
    width: 94%;
}

.live-monitoring-nav {
    background-color: #3EDAD8;
    padding: 15px 40px;
    max-height: 50px;
```

```css
    position: absolute;
    width: 94%;
}

.plant-monitoring-nav > div > a > div, .fish-monitoring-nav > div > a > div, .live-monitoring-nav
> div > a > div {
    max-height: 50px;
    padding: 0px 15px;
}

.plant-monitoring-nav > div > a > div > p {
    color: #3DB67D;
}

.fish-monitoring-nav > div > a > div > p {
    color: #3498DB;
}

.live-monitoring-nav > div > a > div > p {
    color: #286291;
}

.monitoring-icon {
    width: 50px;
    margin-right: 10px;
}

.plant-tab-active {
    background-color: #F3FAE7;
    margin-top: 20px;
    max-height: 70px !important;
}

.fish-tab-active {
    background-color: #E9F4FB;
    margin-top: 20px;
    max-height: 70px !important;
}

.live-tab-active {
    background-color: #3EDAD8;
    margin-top: 20px;
    max-height: 70px !important;
}

.plant-tab-active > img, .fish-tab-active > img, .live-tab-active > img {
    width: 70px;
}

.plant-tab-active > p {
    font-size: 20px;
```

```
    border-bottom: 1px solid #3DB67D;
    font-weight: bold;
}

.fish-tab-active > p {
    font-size: 20px;
    border-bottom: 1px solid #3498DB;
    font-weight: bold;
}

.live-tab-active > p {
    font-size: 20px;
    border-bottom: 1px solid #286291;
    font-weight: bold;
}

.graph {
    width: 80%;
}

.header-blue > span {
    color: #3498DB;
}

.header-blue, .header-dark-blue {
    font-size: 40px;
}

.header-dark-blue > span {
    color: #3EDAD8;
}

.tilapia {
    height: 300px;
    object-fit: cover;
}

@media only screen and (max-width: 500px) {
    .nav-list {
        display: none;
    }

    nav {
        height: 30vh;
    }

    .last {
        min-height: 90vh;
    }
```

```
    .nav-text {
        padding-top: 100px;
    }

    .nav-text > h1 {
        font-size: 35px;
    }
    .nav-text {
        padding-top: 100px;
    }

    .col-4 {
        margin: 0 auto;
        text-align: center;
        flex: 100%;
    }

    .row {
        width: 100%;
    }
}

@media only screen and (min-width: 500px) {
    .last {
        min-height: 150vh;
    }

    .col-4 {
        margin: 0 auto;
        text-align: center;
        flex: 100%;
    }

    .row {
        width: 100%;
    }
}

@media only screen and (max-width: 727px) {
    .nav-list {
        display: none;
    }

    nav {
        height: 40vh;
    }

    .row {
        display: inline-block;
    }
```

```
  .icons {
    width: 30%;
  }

  .sidebar {
    width: 45%;
  }

}

@media only screen and (max-width: 850px) {
  .sidebar {
    width: 75%;
  }
}

@media only screen and (min-width: 727px) {
  .last {
    min-height: 100vh;
  }

  .contact-card {
    min-height: 200px;
  }
}

@media only screen and (max-width: 375px) {
  .last {
    min-height: 150vh;
  }
}

@media only screen and (max-width: 727px) {
  nav > .d-flex  {
    width: 100%;
  }

  nav.contact {
    height: 40vh !important;
  }

  .nav-text {
    padding-top: 100px;
  }
}

@media only screen and (min-width: 750px) {
  nav {
    height: 70vh;
  }
```

```
  .nav-text {
    padding-top: 200px;
  }
}

@media only screen and (min-width: 1200px) {
  nav {
    height: 100vh;
  }

  .nav-text {
    padding-top: 200px;
  }
}

/* charts */

.charts {
  display: grid;
  grid-template-columns: 1fr;
  grid-gap: 20px;
  width: 100%;
  padding: 20px;
  padding-top: 50;
}

.chart {
  background-color: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 7px 25px rgba(0, 0, 0, 0.08);
  width: 100%;
}

.chart h2 {
  margin-bottom: 5px;
  font-size: 20px;
  color: #666;
  text-align: center
}

@media (max-width:1115px) {
  .main {
    width: calc(100% - 60px);
    left: 60px;
  }
}

@media (max-width:880px) {
  .charts {
    grid-template-columns: 2fr;
```

```
    }
    .doughnut-chart {
        padding: 50px;
    }
    #doughnut {
        padding: 50px;
    }
}

@media (max-width:500px) {
    .doughnut-chart {
        padding: 10px;
    }
    #doughnut {
        padding: 0px;
    }
    .user {
        width: 40px;
        height: 40px;
    }
}
```

ANNEX III: Camera Calibration Codes

```
# builtins
import os,sys,time,traceback
from math import hypot

# must be installed using pip
# python3 -m pip install opencv-python
import numpy as np
import cv2

# local clayton libs
import frame_capture
import frame_draw

# camera values
camera_id = 0
camera_width = 1920
camera_height = 1080
camera_frame_rate = 30
#camera_fourcc = cv2.VideoWriter_fourcc(*"YUYV")
camera_fourcc = cv2.VideoWriter_fourcc(*"MJPG")

# auto measure mouse events
auto_percent = 0.2
auto_threshold = 127
auto_blur = 5

# normalization mouse events
norm_alpha = 0
norm_beta = 255
# you can make a config file "camruler_config.csv"
# this is a comma-separated file with one "item,value" pair per line
# you can also use a "=" separated pair like "item=value"
# you can use # to comment a line
# the items must be named like the default variables above

# read local config values
configfile = 'camruler_config.csv'
if os.path.isfile(configfile):
    with open(configfile) as f:
        for line in f:
            line = line.strip()
            if line and line[0] != '#' and (',' in line or '=' in line):
                if ',' in line:
                    item,value = [x.strip() for x in line.split(',',1)]
                elif '=' in line:
                    item,value = [x.strip() for x in line.split('=',1)]
                else:
                    continue
```

```
            if item in 'camera_id camera_width camera_height camera_frame_rate camera_fourcc
auto_percent auto_threshold auto_blur norm_alpha norm_beta'.split():
                try:
                    exec(f'{item}={value}')
                    print('CONFIG:',(item,value))
                except:
                    print('CONFIG ERROR:',(item,value))

# get camera id from argv[1]
# example "python3 camruler.py 2"
if len(sys.argv) > 1:
    camera_id = sys.argv[1]
    if camera_id.isdigit():
        camera_id = int(camera_id)

# camera thread setup
camera = frame_capture.Camera_Thread()
camera.camera_source = camera_id # SET THE CORRECT CAMERA NUMBER
camera.camera_width  = camera_width
camera.camera_height = camera_height
camera.camera_frame_rate = camera_frame_rate
camera.camera_fourcc = camera_fourcc

#1 start camera thread
camera.start()

# initial camera values (shortcuts for below)
width  = camera.camera_width
height = camera.camera_height
area = width*height
cx = int(width/2)
cy = int(height/2)
dm = hypot(cx,cy) # max pixel distance
frate  = camera.camera_frame_rate
print('CAMERA:',[camera.camera_source,width,height,area,frate])

draw = frame_draw.DRAW()
draw.width = width
draw.height = height

# distance units designator
unit_suffix = 'mm'

# calibrate every N pixels
pixel_base = 10

# maximum field of view from center to farthest edge
# should be measured in unit_suffix
cal_range = 72

# initial calibration values table {pixels:scale}
```

```
# this is based on the frame size and the cal_range
cal = dict([(x,cal_range/dm) for x in range(0,int(dm)+1,pixel_base)])

# calibration loop values
# inside of main loop below
cal_base = 5
cal_last = None

# calibration update
def cal_update(x,y,unit_distance):

    # basics
    pixel_distance = hypot(x,y)
    scale = abs(unit_distance/pixel_distance)
    target = baseround(abs(pixel_distance),pixel_base)

    # low-high values in distance
    low  = target*scale - (cal_base/2)
    high = target*scale + (cal_base/2)

    # get low start point in pixels
    start = target
    if unit_distance <= cal_base:
        start = 0
    else:
        while start*scale > low:
            start -= pixel_base

    # get high stop point in pixels
    stop = target
    if unit_distance >= baseround(cal_range,pixel_base):
        high = max(cal.keys())
    else:
        while stop*scale < high:
            stop += pixel_base

    # set scale
    for x in range(start,stop+1,pixel_base):
        cal[x] = scale
        print(f'CAL: {x} {scale}')

# read local calibration data
calfile = 'camruler_cal.csv'
if os.path.isfile(calfile):
    with open(calfile) as f:
        for line in f:
            line = line.strip()
            if line and line[0] in ('d',):
                axis,pixels,scale = [_.strip() for _ in line.split(',',2)]
                if axis == 'd':
                    print(f'LOAD: {pixels} {scale}')
```

```
            cal[int(pixels)] = float(scale)

# convert pixels to units
def conv(x,y):

    d = distance(0,0,x,y)

    scale = cal[baseround(d,pixel_base)]

    return x*scale,y*scale

# round to a given base
def baseround(x,base=1):
    return int(base * round(float(x)/base))

# distance formula 2D
def distance(x1,y1,x2,y2):
    return hypot(x1-x2,y1-y2)

# define display frame
framename = "CamRuler ~ ClaytonDarwin's Youtube Channel"
cv2.namedWindow(framename,flags=cv2.WINDOW_NORMAL|cv2.WINDOW_GUI_NORMA
L)

key_last = 0
key_flags = {'config':False, # c key
             'auto':False,   # a key
             'thresh':False, # t key
             'percent':False,# p key
             'norms':False,  # n key
             'rotate':False, # r key
             'lock':False,   #
             }

def key_flags_clear():

    global key_flags

    for key in list(key_flags.keys()):
        if key not in ('rotate',):
            key_flags[key] = False

def key_event(key):

    global key_last
    global key_flags
    global mouse_mark
    global cal_last

    # config mode
    if key == 99:
```

```
        if key_flags['config']:
            key_flags['config'] = False
        else:
            key_flags_clear()
            key_flags['config'] = True
            cal_last,mouse_mark = 0,None

    # normilization mode
    elif key == 110:
        if key_flags['norms']:
            key_flags['norms'] = False
        else:
            key_flags['thresh'] = False
            key_flags['percent'] = False
            key_flags['lock'] = False
            key_flags['norms'] = True
            mouse_mark = None

    # rotate
    elif key == 114:
        if key_flags['rotate']:
            key_flags['rotate'] = False
        else:
            key_flags['rotate'] = True

    # auto mode
    elif key == 97:
        if key_flags['auto']:
            key_flags['auto'] = False
        else:
            key_flags_clear()
            key_flags['auto'] = True
            mouse_mark = None

    # auto percent
    elif key == 112 and key_flags['auto']:
        key_flags['percent'] = not key_flags['percent']
        key_flags['thresh'] = False
        key_flags['lock'] = False

    # auto threshold
    elif key == 116 and key_flags['auto']:
        key_flags['thresh'] = not key_flags['thresh']
        key_flags['percent'] = False
        key_flags['lock'] = False

    # log
    print('key:',[key,chr(key)])
    key_last = key

# mouse events
```

```
mouse_raw  = (0,0) # pixels from top left
mouse_now  = (0,0) # pixels from center
mouse_mark = None  # last click (from center)

# mouse callback
def mouse_event(event,x,y,flags,parameters):

    #print(event,x,y,flags,parameters)

    # event =  0 = current location
    # event =  1 = left   down click
    # event =  2 = right  down click
    # event =  3 = middle down
    # event =  4 = left   up   click
    # event =  5 = right  up   click
    # event =  6 = middle up
    # event = 10 = middle scroll, flag negative|positive value = down|up

    # globals
    global mouse_raw
    global mouse_now
    global mouse_mark
    global key_last
    global auto_percent
    global auto_threshold
    global auto_blur
    global norm_alpha
    global norm_beta

    # update percent
    if key_flags['percent']:
        auto_percent = 5*(x/width)*(y/height)

    # update threshold
    elif key_flags['thresh']:
        auto_threshold = int(255*x/width)
        auto_blur = int(20*y/height) | 1 # insure it is odd and at least 1

    # update normalization
    elif key_flags['norms']:
        norm_alpha = int(64*x/width)
        norm_beta  = min(255,int(128+(128*y/height)))

    # update mouse location
    mouse_raw = (x,y)

    # offset from center
    # invert y to standard quadrants
    ox = x - cx
    oy = (y-cy)*-1
```

```
    # update mouse location
    mouse_raw = (x,y)
    if not key_flags['lock']:
        mouse_now = (ox,oy)

    # left click event
    if event == 1:

        if key_flags['config']:
            key_flags['lock'] = False
            mouse_mark = (ox,oy)

        elif key_flags['auto']:
            key_flags['lock'] = False
            mouse_mark = (ox,oy)

        if key_flags['percent']:
            key_flags['percent'] = False
            mouse_mark = (ox,oy)

        elif key_flags['thresh']:
            key_flags['thresh'] = False
            mouse_mark = (ox,oy)

        elif key_flags['norms']:
            key_flags['norms'] = False
            mouse_mark = (ox,oy)

        elif not key_flags['lock']:
            if mouse_mark:
                key_flags['lock'] = True
            else:
                mouse_mark = (ox,oy)
        else:
            key_flags['lock'] = False
            mouse_now = (ox,oy)
            mouse_mark = (ox,oy)

        key_last = 0

    # right click event
    elif event == 2:
        key_flags_clear()
        mouse_mark = None
        key_last = 0

# register mouse callback
cv2.setMouseCallback(framename,mouse_event)

# loop
while 1:
```

```
# get frame
frame0 = camera.next(wait=1)
if frame0 is None:
    time.sleep(0.1)
    continue

# normalize
cv2.normalize(frame0,frame0,norm_alpha,norm_beta,cv2.NORM_MINMAX)

# rotate 180
if key_flags['rotate']:
    frame0 = cv2.rotate(frame0,cv2.ROTATE_180)

# start top-left text block
text = []

# camera text
fps = camera.current_frame_rate
text.append(f'CAMERA: {camera_id} {width}x{height} {fps}FPS')

# mouse text
text.append('')
if not mouse_mark:
    text.append(f'LAST CLICK: NONE')
else:
    text.append(f'LAST CLICK: {mouse_mark} PIXELS')
text.append(f'CURRENT XY: {mouse_now} PIXELS')

if key_flags['norms']:

    # print
    text.append('')
    text.append(f'NORMILIZE MODE')
    text.append(f'ALPHA (min): {norm_alpha}')
    text.append(f'BETA (max): {norm_beta}')

if key_flags['config']:

    # quadrant crosshairs
    draw.crosshairs(frame0,5,weight=2,color='red',invert=True)

    # crosshairs aligned (rotated) to maximum distance
    draw.line(frame0,cx,cy, cx+cx, cy+cy,weight=1,color='red')
    draw.line(frame0,cx,cy, cx+cy, cy-cx,weight=1,color='red')
    draw.line(frame0,cx,cy,-cx+cx,-cy+cy,weight=1,color='red')
    draw.line(frame0,cx,cy, cx-cy, cy+cx,weight=1,color='red')

    # mouse cursor lines (parallel to aligned crosshairs)
    mx,my = mouse_raw
    draw.line(frame0,mx,my,mx+dm,my+(dm*( cy/cx)),weight=1,color='green')
```

```
        draw.line(frame0,mx,my,mx-dm,my-(dm*( cy/cx)),weight=1,color='green')
        draw.line(frame0,mx,my,mx+dm,my+(dm*(-cx/cy)),weight=1,color='green')
        draw.line(frame0,mx,my,mx-dm,my-(dm*(-cx/cy)),weight=1,color='green')

        # config text data
        text.append('')
        text.append(f'CONFIG MODE')

        # start cal
        if not cal_last:
            cal_last = cal_base
            caltext = f'CONFIG: Click on D = {cal_last}'

        # continue cal
        elif cal_last <= cal_range:
            if mouse_mark:
                cal_update(*mouse_mark,cal_last)
                cal_last += cal_base
            caltext = f'CONFIG: Click on D = {cal_last}'

        # done
        else:
            key_flags_clear()
            cal_last == None
            with open(calfile,'w') as f:
                data = list(cal.items())
                data.sort()
                for key,value in data:
                    f.write(f'd,{key},{value}\n')
                f.close()
            caltext = f'CONFIG: Complete.'

        # add caltext
        draw.add_text(frame0,caltext,(cx)+100,(cy)+30,color='red')

        # clear mouse
        mouse_mark = None

    elif key_flags['auto']:

        mouse_mark = None

        # auto text data
        text.append('')
        text.append(f'AUTO MODE')
        text.append(f'UNITS: {unit_suffix}')
        text.append(f'MIN PERCENT: {auto_percent:.2f}')
        text.append(f'THRESHOLD: {auto_threshold}')
        text.append(f'GAUSS BLUR: {auto_blur}')

        # gray frame
```

```
        frame1 = cv2.cvtColor(frame0,cv2.COLOR_BGR2GRAY)

        # blur frame
        frame1 = cv2.GaussianBlur(frame1,(auto_blur,auto_blur),0)

        # threshold frame n out of 255 (85 = 33%)
        frame1 = cv2.threshold(frame1,auto_threshold,255,cv2.THRESH_BINARY)[1]

        # invert
        frame1 = ~frame1

        # find contours on thresholded image
        contours,nada =
cv2.findContours(frame1,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)

        # small crosshairs (after getting frame1)
        draw.crosshairs(frame0,5,weight=2,color='green')

        # loop over the contours
        for c in contours:

            # contour data (from top left)
            x1,y1,w,h = cv2.boundingRect(c)
            x2,y2 = x1+w,y1+h
            x3,y3 = x1+(w/2),y1+(h/2)

            # percent area
            percent = 100*w*h/area

            # if the contour is too small, ignore it
            if percent < auto_percent:
                    continue

            # if the contour is too large, ignore it
            elif percent > 60:
                    continue

            # convert to center, then distance
            x1c,y1c = conv(x1-(cx),y1-(cy))
            x2c,y2c = conv(x2-(cx),y2-(cy))
            xlen = abs(x1c-x2c)
            ylen = abs(y1c-y2c)
            alen = 0
            if max(xlen,ylen) > 0 and min(xlen,ylen)/max(xlen,ylen) >= 0.95:
                alen = (xlen+ylen)/2
            carea = xlen*ylen

            # plot
            draw.rect(frame0,x1,y1,x2,y2,weight=2,color='red')

            # add dimensions
```

```
        draw.add_text(frame0,f'{xlen:.2f}',x1-((x1-x2)/2),min(y1,y2)-8,center=True,color='red')
        draw.add_text(frame0,f'Area: {carea:.2f}',x3,y2+8,center=True,top=True,color='red')
        if alen:
            draw.add_text(frame0,f'Avg: {alen:.2f}',x3,y2+34,center=True,top=True,color='green')
        if x1 < width-x2:
            draw.add_text(frame0,f'{ylen:.2f}',x2+4,(y1+y2)/2,middle=True,color='red')
        else:
            draw.add_text(frame0,f'{ylen:.2f}',x1-4,(y1+y2)/2,middle=True,right=True,color='red')

else:

    # small crosshairs
    draw.crosshairs(frame0,5,weight=2,color='green')

    # mouse cursor lines
    draw.vline(frame0,mouse_raw[0],weight=1,color='green')
    draw.hline(frame0,mouse_raw[1],weight=1,color='green')

    # draw
    if mouse_mark:

        # locations
        x1,y1 = mouse_mark
        x2,y2 = mouse_now

        # convert to distance
        x1c,y1c = conv(x1,y1)
        x2c,y2c = conv(x2,y2)
        xlen = abs(x1c-x2c)
        ylen = abs(y1c-y2c)
        llen = hypot(xlen,ylen)
        alen = 0
        if max(xlen,ylen) > 0 and min(xlen,ylen)/max(xlen,ylen) >= 0.95:
            alen = (xlen+ylen)/2
        carea = xlen*ylen

        # print distances
        text.append('')
        text.append(f'X LEN: {xlen:.2f}{unit_suffix}')
        text.append(f'Y LEN: {ylen:.2f}{unit_suffix}')
        text.append(f'L LEN: {llen:.2f}{unit_suffix}')

        # convert to plot locations
        x1 += cx
        x2 += cx
        y1 *= -1
        y2 *= -1
        y1 += cy
        y2 += cy
        x3 = x1+((x2-x1)/2)
        y3 = max(y1,y2)
```

```
        # line weight
        weight = 1
        if key_flags['lock']:
            weight = 2

        # plot
        draw.rect(frame0,x1,y1,x2,y2,weight=weight,color='red')
        draw.line(frame0,x1,y1,x2,y2,weight=weight,color='green')

        # add dimensions
        draw.add_text(frame0,f'{xlen:.2f}',x1-((x1-x2)/2),min(y1,y2)-8,center=True,color='red')
        draw.add_text(frame0,f'Area: {carea:.2f}',x3,y3+8,center=True,top=True,color='red')
        if alen:
            draw.add_text(frame0,f'Avg: {alen:.2f}',x3,y3+34,center=True,top=True,color='green')
        if x2 <= x1:
            draw.add_text(frame0,f'{ylen:.2f}',x1+4,(y1+y2)/2,middle=True,color='red')
            draw.add_text(frame0,f'{llen:.2f}',x2-4,y2-4,right=True,color='green')
        else:
            draw.add_text(frame0,f'{ylen:.2f}',x1-4,(y1+y2)/2,middle=True,right=True,color='red')
            draw.add_text(frame0,f'{llen:.2f}',x2+8,y2-4,color='green')

    # add usage key
    text.append('')
    text.append(f'Q = QUIT')
    text.append(f'R = ROTATE')
    text.append(f'N = NORMALIZE')
    text.append(f'A = AUTO-MODE')
    if key_flags['auto']:
        text.append(f'P = MIN-PERCENT')
        text.append(f'T = THRESHOLD')
        text.append(f'T = GAUSS BLUR')
    text.append(f'C = CONFIG-MODE')

    # draw top-left text block
    draw.add_text_top_left(frame0,text)

    # display
    cv2.imshow(framename,frame0)

    # key delay and action
    key = cv2.waitKey(1) & 0xFF

    # esc ==  27 == quit
    # q   == 113 == quit
    if key in (27,113):
        break

    # key data
    #elif key != 255:
    elif key not in (-1,255):
```

```
    key_event(key)

# close camera thread
camera.stop()

# close all windows
cv2.destroyAllWindows()

# done
exit()
```

ANNEX IV: User’s Manual

AQCLOUD

FOR YOUR SMART AQUAPONICS SYSTEM

User Manual
Version 1.0
July 3, 2023

Introduction

assist you in efficiently navigating and making the most of our product's features. We have taken great care to ensure that this manual includes detailed instructions and explanations to support your understanding.

Should you have any inquiries that are not addressed in this guide, please feel free to reach out to our Lead Programmer or email us at aqcloud_tup@gmail.com. We are here to provide further assistance and support.

Minimum Requirements

File size: 1.5GB

Computer Specs:

System type: 64-bit Operating System

OS: Windows 10 or similar

RAM: 4GB

CPU: Intel® Core™ i5-8250U CPU @ 1.60GHz 1.80GHz

Additional Drivers:

- RPI 4 Model B (RAM: 4GB)
- Web Camera FHD 1080P

Overview

Aquaponics is an innovative agricultural system replicating a natural ecosystem's dynamics by establishing a symbiotic relationship among water, aquatic organisms, bacteria, nutrient cycles, and plants. It emulates the interactions observed in waterways worldwide. By drawing inspiration from nature, aquaponics integrates these individual components in a bio-integrated manner. This entails utilizing fish waste as a nutrient source for bacteria, which converts it into an ideal fertilizer for plants. Consequently, the plants cleanse the water, which is then returned to the fish in a clean and safe state.

The Smart Aquaponic System, developed by AQCLOUD, is an environmentally sustainable solution designed to replicate and optimize the functionality of a natural ecosystem. It aims to stimulate local and sustainable production, contributing to the establishment of a circular economy. A key innovation of this project is the implementation of a web application that provides users with a data dashboard. This dashboard allows for interactive and graphical visualization of the collected inputs.

The team’s primary objective is to promote the utilization of aquaponics technology, which combines the principles of aquaculture and hydroponics. This integrated system facilitates the simultaneous growth of fish and vegetables in a controlled environment. Aquaponics represents a distinctive form of agriculture in which fish are raised in tanks while plants are cultivated without the use of soil. The nutrient-rich water resulting from fish cultivation serves as a natural fertilizer for the plants, while the plants assist in purifying the water for the fish.

AQCLOUD offers a smart web application that enables real-time monitoring and growth updates for lettuce and Nile tilapia. This application comprises several features, including a page displaying the status of Nile tilapia, another page presenting the status of individual lettuce heads, and a live video monitoring feature for observing the plants and fish as they develop.

Overall, the Smart Aquaponics System embodies the advantages of both aquaculture and hydroponics, providing a sustainable and efficient approach to agriculture. It minimizes resource consumption, eliminates the need for synthetic fertilizers, and maximizes productivity within a closed-loop system. By leveraging this technology, AQCLOUD aims to facilitate the establishment of environmentally conscious and self-sustaining food production systems

SIGN UP and LOG IN SET-UP

1. Click Sign up for the new account.
2. Fill out the information needed.
3. Click register.
4. After successfully registering, log in with your account.

New User? Sign up Here.

Existing User? Log in Here.

HOMEPAGE

The home page displays what the aquaponics system is and how it defines. Also, it shows the three icons below where you can click, and you will be routed to:

1. Fish monitoring icon where you will be directed to see the status of the fish.

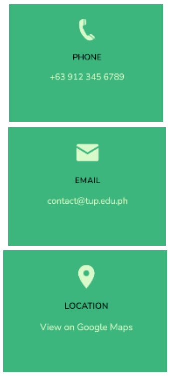
2. Plant monitoring icon where you will be directed to see the status of the plants.

3. Live monitoring icon where you will be directed to realtime monitoring of the plants and fishes.

ABOUT

The About section displays where the aquaponics system is deployed and tells about a brief history of its deployment's location.

CONTACT



1. Number to reach out for the aquaponics system

2. Mail address to reach out for the aquaponics system.

3. Where the aquaponics system is located.

NAVIGATION MENU 

1. Homepage button

2. About button

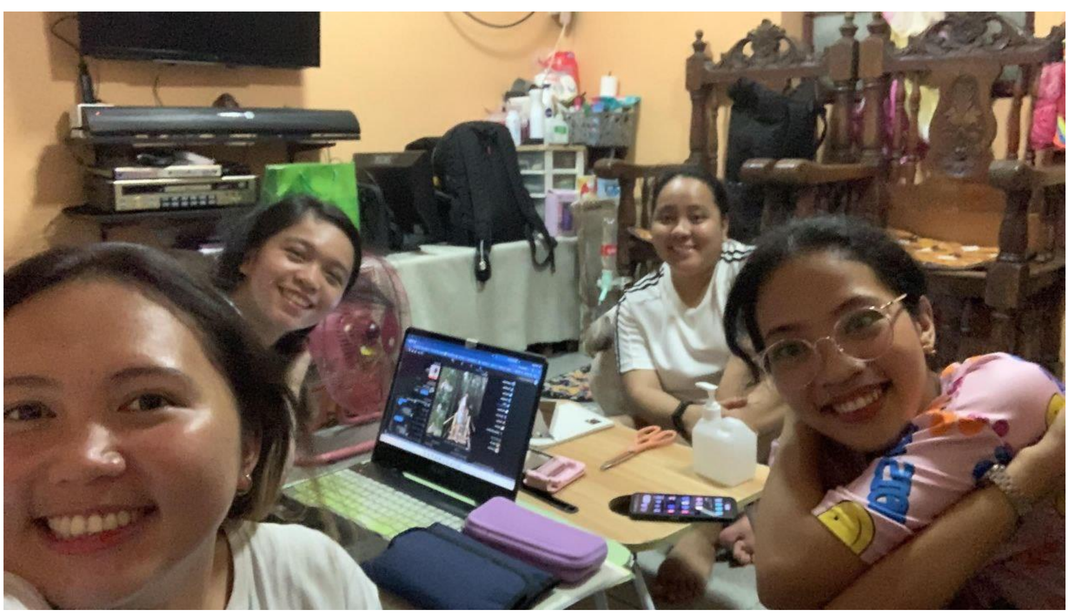
3. Contacts button

4. To navigate on the plant, fish and live monitoring of the smart aquaponics.

5. To sign out the account.

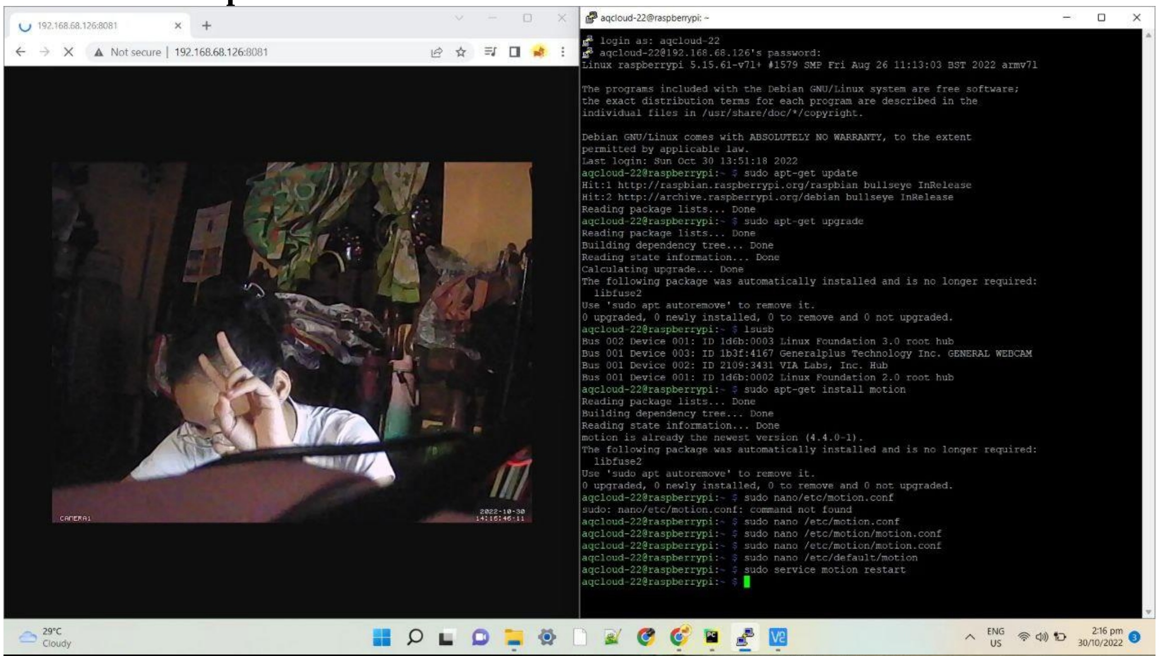
ANNEX V: Documentation

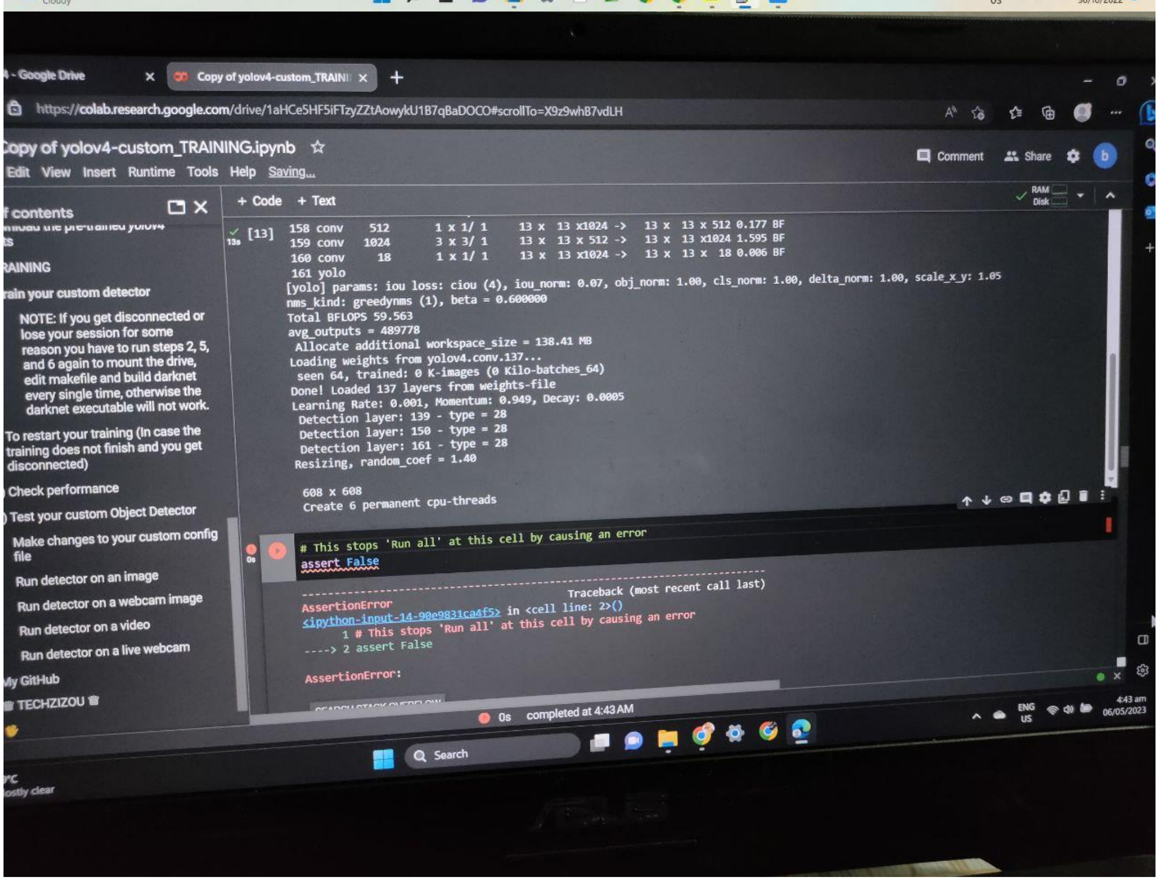
Face-to-Face and Online Meetings



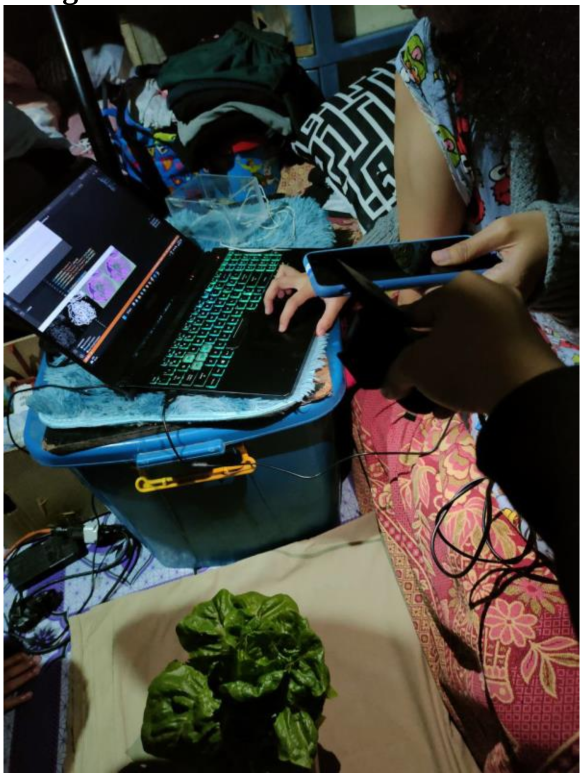
Consultations

Software Development



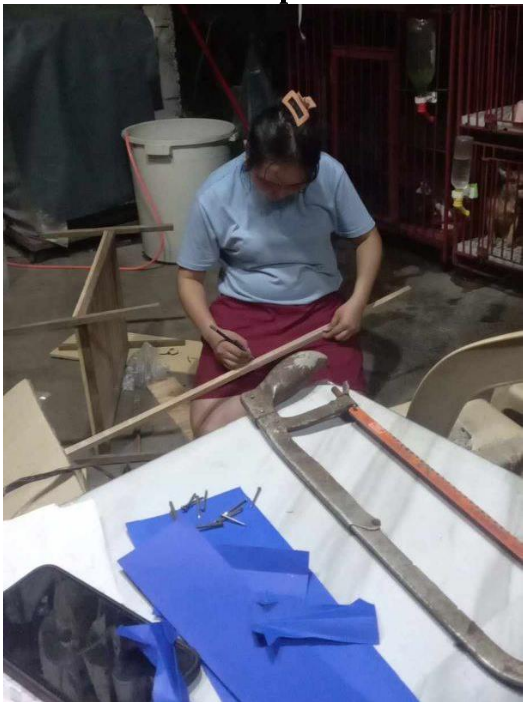


Machine Learning Testing

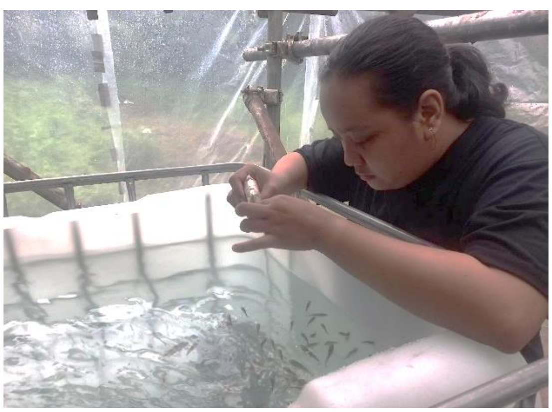


Camera Installation

APPRECIATE 2023 Preparation and Presentation

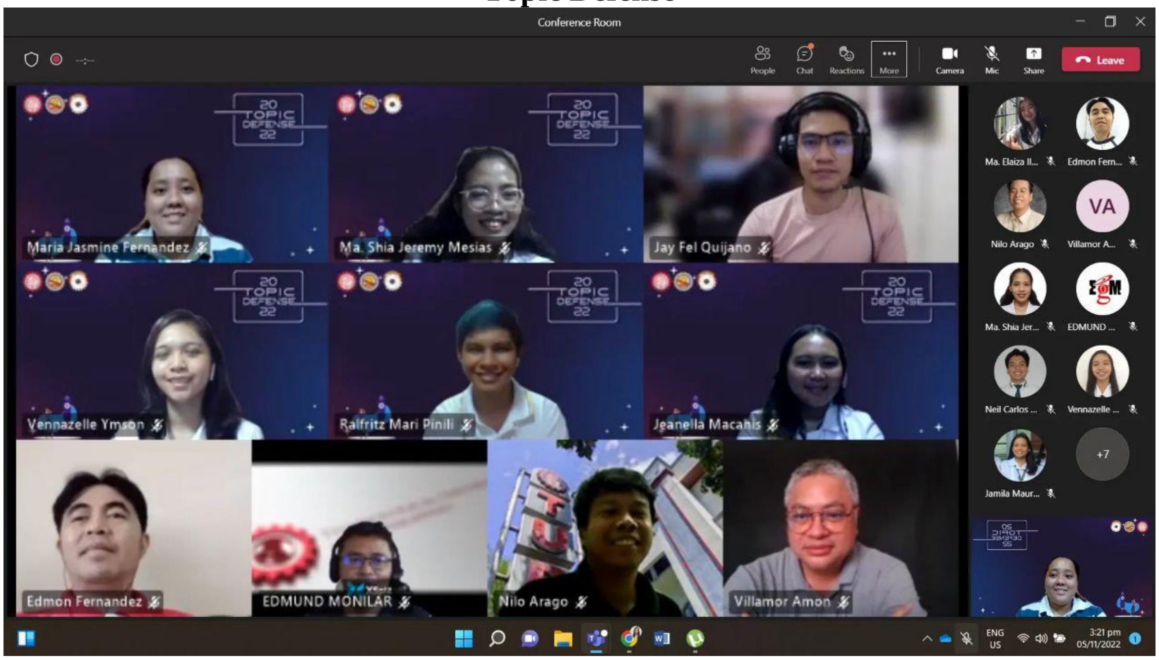


Deployment Visit and Data Gathering



ANNEX VI: Thesis Defense

Topic Defense



Title Defense

Progress Defense

Pre-Final Defense

APPRECIATE 2023

Final Defense

ANNEX VII: Student's Profile

MA. SHIA JEREMY MESIAS

R E S E A R C H E R

Analytical, organized and detail-oriented engineering student. Collaborative and team player student-leader with a track record in service and management. Throughout her academic journey, she has actively participated in seminars, workshops, trainings, certification exams, and other related activities, which have contributed to her acquisition of a diverse range of skills.

Contact

Address
Del Nacia Ville #4 Riverside Sauyo, Quezon City

Phone
+639619830835

E-mail
mesias.shia01@gmail,com

Skills

- Proficient in Filipino and English
- Comfortable in Writing and Public Speaking
- Proficient in MS Office
- Basic Knowledge in Python and MATLAB
- Basic knowledge with Machine Learning and IoT Devices.

Affiliations

Organization of Electronics Engineering Students
University Organization
Chairperson - Graduating Class
2022 - 2023
Senior Vice President
2021 - 2022
Member
2019 - 2022

Institute of Electronics Engineers of the Philippines Manila
Organization | Member
2019 - Present

Experience

2022 August — Tri-Power Sales and Electrical Services

Education

Tertiary

2019-2023 — Technological University of the Philippines – Manila

Secondary

2017-2019 — Dr. Carlos S. Lanting College

2013-2017 — Tandang Sora National High School

Primary

2005-2012 — Sauyo Elementary School

Awards/Certifications

(ISC)²

June 2023 — Cybersecurity and IT Security Candidate

Fortinet Training

May 2023 — Network Security Associate 1
Network Security Associate 2
Network Security Associate 3

Intellectual Property Office of the Philippines

July 2022 — IP Boost Camp Graduate

MARIA JASMINE FERNANDEZ

RESEARCHER

A forward-thinking and ambitious electronics engineering student set to graduate, equipped with a passion for creating and advancing state-of-the-art electronic systems. They consistently stay updated on the latest breakthroughs in electronics engineering, actively engaging in workshops, seminars, and online courses to broaden their expertise and capabilities.

Contact

Address
194 B. Malambing St., Brgy. 145, Bagong Barrio Caloocan City

Phone
+639777256876

E-mail
mariajasmine.fernandez@tup.edu.ph

Skills

- Proficient with MS Office
- Proficient with basic computer knowledge
- Basic knowledge in Python and MATLAB
- Good time management skills
- Good coordination skills

Affiliations

Organization of Electronics Engineering Students
University Organization | Member
2019 - 2023

Institute of Electronics Engineers of the Philippines Manila
Organization | Member
2019 - 2021

Experience

2022
August 15 - September 17

Supervised Industrial Training, Engineering Department
Now Corporation

Education

Tertiary

2019-2023 — Technological University of the Philippines – Manila

Secondary

2017-2019 — University of the East - Caloocan

2013-2017 — Salvador Araneta Memorial Institute

Primary

2005-2012 — Bagong Barrio Elementary School

Awards/Certifications

(ISC)²
June 2023 — Cybersecurity and IT Security Candidate

Fortinet Training
May 2023 — Network Security Associate 1
Network Security Associate 2
Network Security Associate 3

JEANELLA MACAHIS

R E S E A R C H E R

A passionate graduating electronics engineering student with basic knowledge

Contact

Address
Atay de Paz Street,
Electrical Road Pasay
CIty

Phone
09298696834

E-mail
jeanellamacahis@gmail.com

Experience

2019	Work Immersion, Airlink Internatinational Aviation -Avionics
2020	AJB insurance services - Typist Encoder
2021	Intern Tripower and Electrical Services
2022	Government Internship program PYAP internship

Skills

- Basic understanding in Arduino
- Strategic Thinking
- Craftmanship
- Communication skills
- Computer Literate

Education

Tertiary

2019-2023 Technological University of the Philippines – Manila

Secondary

2017-2019 Airlink International Aviation College

2013-2017 Parañaque National HighSchool-Baclaran

Primary

2005-2012 Baclaran ELementary School Unit II

Affiliations

Organization of Electronics Engineering Students
University Organization | Member
2019 - Present

Institute of Electronics Engineers of the Philippines Manila
Organization | Member
2019 - 2021

Awards/Certifications

(ISC)²
June 2023 Cybersecurity and IT Security Candidate

Fortinet Training
May 2023 Network Security Associate 1
Network Security Associate 2
Network Security Associate 3

RALFRITZ MARI PINILI

R E S E A R C H E R

Passionate Electronics Engineering student with a strong analytical aptitude and a keen interest in the field. Possessing basic programming skills, excellent communication abilities, and computer literacy. Eager to apply my knowledge and skills to contribute effectively in the realm of Electronics Engineering.

Contact

Address
10 T GOMEZ STREET SAN ROQUE NAVOTAS CITY

Phone
+639267302590

E-mail
ralfritzpinili13@gmail.com

Skills

- Basic knowledge in MATLAB
- Basic knowledge in programming
- Good time management skills
- Good communication skills
- Computer Literate

Affiliations

Organization of Electronics Engineering Students
University Organization | Member
2019 - Present

Institute of Electronics Engineers of the Philippines Manila
Organization | Member
2019 - 2021

Experience

2022 — Supervised Industrial Training, Tri-Power Sales and Electrical Services

2022 — Customer Care Representative, Concentrix

2023 — Customer Care Representative, Alorica

Education

Tertiary

2019-2023 — Technological University of the Philippines – Manila

Secondary

2017-2019 — San Jose Academy

2013-2017 — Governor Andres Pascual College

Primary

2005-2012 — Wawa Elementary School

Awards/Certifications

(ISC)²
June 2023 — Cybersecurity and IT Security Candidate

Fortinet Training
May 2023 — Network Security Associate 1
Network Security Associate 2
Network Security Associate 3

VENNAZELLE YMSON

RESEARCHER

An innovative and progressive electronics engineering student who possesses skills with a passion for designing and developing cutting-edge electronic systems. Have an unwavering dedication to staying up-to-date with the most recent advancements in the field and can skillfully apply this knowledge to create efficient and reliable solutions.

Contact

Address
1600 Int. 7 - A J. Fajardo
St. Sampaloc, Manila

Phone
+639498871824

E-mail
ymsonnazelle@gmail.com

Skills

- Proficient in basic computer knowledge
- Basic knowledge in Python and MATLAB
- Basic knowledge with Machine Learning
- Good time management skills
- Good coordination skills

Affiliations

Organization of Electronics Engineering Students
University Organization | Member
2019 - 2023

Institute of Electronics Engineers of the Philippines Manila
Organization | Member
2019 - 2021

Experience

2019
March 11 - March 15
Work Immersion,
Monitoring Division
Maritime Industry Authority

2022
August 15 - September 17
Supervised Industrial Training,
Engineering Department
Now Corporation

Education

Tertiary

2019-2023 Technological University of the Philippines – Manila

Secondary

2017-2019 La Consolacion College Manila

2013-2017 Ramon Magsaysay High School

Primary

2005-2012 Trinidad Tecson Elementary School

Awards/Certifications

(ISC)²
June 2023 Cybersecurity and IT Security Candidate

Fortinet Training
May 2023 Network Security Associate 1
Network Security Associate 2
Network Security Associate 3