PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

IBN KHALDOUN UNIVERSITY OF TIARET
FACULTY OF APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL ENGINEERING

FINAL YEAR DISSERTATION FOR THE MASTER'S DEGREE

Field: Science and Technology

Major: Electrical Engineering

Specialization: Electrical Control Systems

Research topic

Development of an Integrated IoT and Machine Learning System for the Management of a Photovoltaic-Powered Aquaponic Greenhouse

Prepared by:
KHIAL Ilyes
MEGHEZI Mohamed Oussama

Examination Committee:

Full Name	Academic Degree	Role
B. BELABBAS	MCA	President
A. TAHRI	MCB	Examiner 1
K. NEGADI	Pr	Examiner 2
M. MAASKRI	MCA	Supervisor
T. ALLAOUI	Pr	Co-Supervisor

2024/2025

بِســم الله الرَّحمَن الرَّحِيــم

We thank **Allah, the All-Powerful**, who gave us the courage to explore this new field and the strength to carry out this work, we also want to thank all those who appreciated and supported our efforts, especially our **family members** and **our Friends**, we couldn't have made it this far without them.

We sincerely thank **Dr. M. Maaskri** and **Pr. T. Allaoui**, our supervisors, for their patience, guidance, valuable advice, and continuous encouragement throughout our project. We are also grateful to them for providing us with all the essential and precious hardware we needed.

We also extend our sincere thanks to **the members of the Examination Committee** for the time they devoted in evaluating our work, as well as for their constructive and enriching comments. We are equally grateful to the members of the **L2GEGI laboratory** for their support, collaboration, and technical assistance throughout our research.

Finally, we would like to extend our thanks to **all the professors of the Department of Electrical Engineering** at the **University of Tiaret** for their support, as well as to everyone who contributed to the success of our project.

List of figures

List of Tables

Summary

Chapter I: General Information

Chapter II:System Hardware Overview

Chapter III:System Software Overview

Introduction

introduction

In the face of escalating global challenges such as population growth, climate change, water scarcity, and diminishing arable land, there is an urgent need to rethink and innovate the way we produce food. Traditional agricultural systems, though productive, are increasingly seen as unsustainable due to their heavy reliance on chemical inputs, excessive water consumption, and vulnerability to environmental fluctuations. As the demand for sustainable and efficient food production systems grows, aquaponics a symbiotic integration of aquaculture (fish farming) and hydroponics (soilless plant cultivation) emerges as a promising solution.

Aquaponics represents a closed-loop ecosystem where nutrient-rich water from fish tanks is circulated to nourish plants, which in turn purify the water through natural filtration. This environmentally friendly technique not only conserves water but also eliminates the need for chemical fertilizers and pesticides, making it ideal for urban and arid regions. However, managing aquaponic systems, especially on a larger scale, requires constant monitoring and precise control of several parameters including water pH, temperature, light, humidity, nutrient concentration, and fish health. Manual monitoring is labor-intensive, error-prone, and inefficient for continuous management.

To address these challenges, this project presents a Smart Aquaponics System enhanced with modern technologies such as Artificial Intelligence (AI), Internet of Things (IoT), Raspberry Pi microcontrollers, and Unmanned Aerial Vehicles (UAVs) or drones. The proposed system integrates various sensors to measure environmental conditions in real time and employs actuators (such as fans, heaters, pumps) controlled by a central Raspberry Pi unit. This unit processes incoming data and adjusts the system dynamically to maintain optimal growth conditions for both fish and plants.

A core innovation of this project is the use of AI-based disease detection, specifically targeting lettuce crops. By training a Convolutional Neural Network (CNN) model to classify leaf images into healthy, bacterial, or fungal categories, the system can identify early signs of disease, thereby preventing crop loss and improving productivity. This model is deployed using Python and TensorFlow, and is accessible through a user-friendly interface or even as a standalone executable software tool.

Additionally, the system incorporates a custom-designed drone to perform aerial surveillance of the aquaponic greenhouse. Equipped with GPS modules, cameras, and communication tools, the drone captures high-resolution images of the crops, which can be processed for health analysis or used for autonomous mission planning via tools like Betaflight, ArduPilot, and Mission Planner.

This multidisciplinary approach bridges several fields—electrical engineering, software development, artificial intelligence, and agricultural sciences—to demonstrate how smart farming can be both practical and scalable. The integration of open-source platforms, affordable components, and modular design principles ensures that the system is adaptable and cost-effective for both research and commercial deployment.

In summary, the Smart Aquaponics System developed in this project serves as a model for sustainable, data-driven, and autonomous agriculture. It showcases how intelligent monitoring and control, powered by AI and IoT, can transform traditional farming into a high-efficiency, low-impact solution fit for the modern world. The potential for expansion, customization, and real-world application makes this project a significant contribution to the future of precision agriculture.

Based on the above, we have structured our thesis into four main chapters as follows:

Chapter I: General Information

This chapter presents the general context of the project, outlining the problem statement, objectives, and the importance of smart systems in modern agriculture. It also provides an overview of aquaponics technology, AI, Drones, and Raspberry pi ,it also highlights the reasons for adopting it as a sustainable food production solution.

Chapter II: System Hardware Overview

This chapter describes the hardware components of the system, including sensors, actuators, the Raspberry Pi controller, and the custom-designed drone. It explains the role of each component in monitoring and regulating the agricultural environment.

Chapter III: System Software Overview

This chapter focuses on the software side of the system, including programming interfaces, AI algorithms used for plant disease detection, drone control tools, and data processing and analysis applications.

Chapter IV: Testing and Experimental Results

This chapter presents the results of the tests conducted on the smart system in real or semi-real environments. It includes performance analysis, disease detection efficiency, and the accuracy of environmental monitoring.

We conclude this work with a general summary of the accomplishments, followed by conclusions and perspectives for future development.

Bibliographical references:

1. **Goddek, S., Joyce, A., Wuertz, S., Körner, O., Bläser, I., & Jijakli, H. (2023).** ***Aquaponics and its role in sustainable food production*****. ScienceDirect.** **https://www.sciencedirect.com/science/article/pii/S0040162523003943**
2. **Love, D. C., Fry, J. P., Genello, L., Hill, E. S., Frederick, J. A., Li, X., & Semmens, K. (2015).** ***Commercial aquaponics production and profitability: Findings from an international survey*****. GALA University of Greenwich.** **https://gala.gre.ac.uk/id/eprint/23247**
3. **Palm, H. W., Knaus, U., Appelbaum, S., Goddek, S., Strauch, S. M., Vermeulen, T., & Jijakli, M. H. (2018).** ***Towards commercial aquaponics: A review of systems, technologies,andpracticalexperiences*****.Springer.** **https://link.springer.com/article/10.1007/s10499-018-0249-z**
4. **Rakocy, J. E., Bailey, D. S., Shultz, R. C., & Thoman, E. S. (2012).** ***Aquaponic systems for sustainable food production*****. In** ***Springer Handbook of Sustainable Development*****.** **https://link.springer.com/chapter/10.1007/978-981-97-5703-9_36**
5. **Sarker, I. H., Abdur, S. M., & Raihan, A. (2024).** ***AI-based monitoring and drone integration in smart agriculture. Artificial Intelligence Review.*** **https://link.springer.com/article/10.1007/s10462-0**
6. **Tyson, R. V., Treadwell, D. D., & Simonne, E. H. (2011).** ***Opportunities and challenges to sustainability in aquaponic systems*****.** ***HortTechnology, 21*****(1), 6–13.** **https://journals.ashs.org/horttech/view/journals/horttech/21/1/article-p6.xml**
7. **Cultivate Nation. (2024).** ***Raspberry Pi Smart Farming: Agritech Innovations and Applications*****.https://cultivatenation.com/raspberry-pi-smart-farming-agritech-innovations-projects-applications**
8. **University of Cambridge. (n.d.).** ***BUGALERT: Pest and Disease Monitoring in Greenhouses with Raspberry-Pi Network*****.** **https://www.gci.cam.ac.uk/global-challenges-research-cambridge/**

9. **Chollet, F. (2021). *Deep Learning with Python* (2nd ed.). Manning Publications. https://www.manning.com/books/deep-learning-with-python-second-edition**
10. ***TensorFlowdataprocessing,*https://www.tensorflow.org/tutorials/load_data/images**
11. **Betaflight.(2021).*Betaflight firmware* https://betaflight.dev**
12. **ArduPilot.(2023).*ArduPilotautopilotsoftware* https://ardupilot.org**
13. ***MissioPlannerArduPilotGCS*. https://ardupilot.org/planner**

Chapter I:

General Information

I.1. Introduction

Aquaponics is rapidly developing in response to the increasing need for sustainable food production, alongside the decreasing availability of freshwater and mineral reserves [1]. Initially starting with small-scale operations, aquaponics is now heading towards market adoption and attracting investments. Over the years, various methods and system designs have been developed, each focusing either on fish or plant production [2]. Public interest in aquaponics has significantly grown in recent years, aligning with the trend toward more integrated value chains, higher productivity, and a lower environmental impact compared to traditional production systems [3]. As a result, innovative business models are emerging, along with new clients and markets [2].

These small systems, which are also known as "Greenhouses," do not require complex surveillance systems, as a farmer alone can manage them. However, as systems grow in scale, surveillance, control, visualization, and protection systems become necessary to monitor everything [4]. After extensive research, the best options for long-range surveillance are the integration of AI based systems, modern monitoring techniques and a quadcopter (or several), also known as "drone(s)" [5].

In this chapter, we provide a precise definition of aquaponics, smart agriculture techniques (AI and Raspberry PI), and Quadcopters in order to assure a clear Framework for aquaponic agriculture, as This framework enables the identification of various technologies and methods, making it easier for authorities, clients, producers, and other stakeholders to understand their potential and limitations.

I.2. Aquaponics

I.2.1. Definition

Aquaponics is a combination of aquaculture (the farming of aquatic organisms) and hydroponics (the soil-free cultivation of plants). It refers to a system where nutrient-rich water from aquaculture serves as fertilizer for the plants. Microorganisms, particularly bacteria, play a crucial role as the third key component, facilitating the transformation of nutrients [1], the walls of aquaponics are made of either glass or plastic to make it easy for plants to absorb the light. the main factors that need to be considered in this closed-system are: Temperature, light, water (plus nutrients of the fish within), carbon dioxide’s level, and humidity [4].

The main advantage of aquaponics is assuring the availability of plants in every season regardless of weather conditions (rain, winds, . . . etc) [6].

I.2.2. Advantages of aquaponics

- Strength and durability;
- Functionality and ease of maintenance at small-scale operations;
- Energy savings with the integration of renewable sources;
- Low price of construction compared to benefits;

»» For extra information, Visit The "OKSTATE" site for more Aquaponics Ups and Downs [7]

I.2.3. Architectures of Aquaponics

Aquaponics systems function essentially as specialized greenhouses, here are the main architectures for a greenhouse:

- Lean-to greenhouse;
- Traditional greenhouse;
- Unequal-span greenhouse;
- Ridge and furrow greenhouse;
- Retractable roof greenhouse;
- Shade house;
- Cold frame;

I.2.3.1. Traditional greenhouse

A traditional greenhouse as shown in **"Figure I.1"**is a structure designed to protect tender or out-of-season plants from excessive cold or heat. In the 17th century, these greenhouses were ordinary brick or timber shelters with a normal proportion of window space and some means of heating. As glass became cheaper and more sophisticated heating methods became available [8].

Figure.I.1. Traditional greenhouse.

I.2.3.2. Retractable roof greenhouse

A retractable roof greenhouse **"Figure I.2"** is a type of greenhouse with a roof that can be opened or closed, allowing for adjustable exposure to sunlight and ventilation. It is typically used in gardening or farming to optimize plant growth by controlling environmental conditions like temperature and humidity [9].

Figure.I.2. Lean-to greenhouse.

I.2.3.3. Lean-to greenhouse

A lean-to greenhouse is a structure built against an existing building, such as a house or garage, utilizing one of its walls as one side of the greenhouse. This design typically features a single-sloped roof like **"Figure I.3"** and is ideal for small spaces, offering a controlled environment for plant growth [10].

Figure.I.3. Retractable roof greenhouse.

»» For extra information, Visit The site "archdaily" for more Greenhouse designs [11]

I.2.3.4. Advantages of Greenhouses

- Off-season production of vegetable and fruit crops;
- Maximize production and yield;
- Control of environmental conditions;
- Protection against pests and diseases;

I.2.3.5. Disadvantages of Greenhouses

- Climate control challenges;
- High initial Investments;
- High energy consumptions in case of using traditional sources;
- Complex maintenance at large-scale operations;

»» For extra information, Visit The site “Gardeningknowhow” for more Advantages and pitfalls of a Greenhouse [12]

I.2.4. Greenhouse monitoring

Effective greenhouse management requires continuous monitoring of temperature, humidity, light levels, and other requirements, The main Factors to consider are:

I.2.4.1. Temperature [13]

Temperature plays a crucial role in photosynthesis, with optimal temperatures generally speeding up the process if light and CO_2 are sufficient, each plant has its own optimal temperature, in this case we have two main Functions:

►***Heating***

The main purpose of heating in a greenhouse is to control and raise the temperature for optimal plant growth using a heat source “a radiator”. The absorbed heat from the air, soil, and heating sources is then re-radiated back into the environment.

►***Cooling***

Plants cool themselves through transpiration, losing about 50% of radiant energy this way. Cooling in greenhouses is typically achieved through ventilation (both passive and active), spraying water, or using heat exchangers. Ventilation is the most common and cost-effective method. The greenhouse in this project uses controlled roof and bottom openings with a fan to manage airflow, recent research focuses on using sustainable energy.

I.2.4.2. Light

Light is crucial for photosynthesis, where sunlight is turned into a substance that supports plant growth. Photosynthesis mainly occurs in red, blue, and violet light wavelengths, with green light being reflected, making plants appear green [14].

I.2.4.3. Humidity

Greenhouses typically have higher humidity levels than outdoor environments due to plant transpiration and limited air movement. Humidity is usually controlled through ventilation but can be adjusted with misting devices or heating [15].

I.2.4.4. Carbon dioxide

Carbon dioxide levels significantly affect plant growth, and greenhouses must be ventilated to ensure sufficient CO2 for plants [16].

»» For extra information, Visit The site "bacfertilizers" for more factors that affect plant growth and bloom [17]

I.2.5. Components of an aquaponics system

In an aquaponic system, fish, bacteria, and plants all work together like a well-organized machine. The fish provide waste, which is broken down by helpful bacteria in the biological filter into nutrients the plants can use. This process also keeps the water clean by removing harmful ammonia. The plants then absorb these nutrients straight from the water, acting as natural filters and helping to keep the system balanced and healthy for everyone involved. It's a harmonious cycle that benefits all parts of the ecosystem [18].

I.2.5.1. Fish species raised in an aquaponics system

Aquaponic systems often rely on hardy fish species like carp, perch, trout, and especially tilapia. Tilapia stands out because it's incredibly adaptable, it thrives in warm water (27°C-30°C) but can also handle extreme temperatures as low as 18°C and as high as 35°C. With the right conditions, it can live up to 10 years and grow to 5 kg, it also needs well-oxygenated water, ideally between 5 to 8 mg/L. Because of its resilience and fast growth, tilapia is one of the most popular choices for aquaponic farming [19].

I.2.5.2. Types of plants in an aquaponics system

Aquaponics systems can grow a wide range of plants, from everyday greens like lettuce and Swiss chard to flavorful herbs like basil and parsley. You'll also find popular vegetables such as tomatoes, beans, and cauliflower blooming in these ecosystems. Surprisingly, even root crops like beets and carrots do well, and some small fruit trees can also be successfully grown, making aquaponics a multipurpose and efficient farming method [20].

I.2.5.3. Types of bacteria in the aquaponic system

In an aquaponic system, bacteria play a crucial role in maintaining balance by managing nitrogen and phosphorus levels. Nitrogen exists in three key forms: ammonia, nitrites, and nitrates. Thanks to bacteria, ammonia from fish waste gets converted into nitrites and then into nitrates, an essential nutrient for plants. This process mimics the natural nitrogen cycle, keeping the water clean and ensuring healthy plant growth. The whole system is influenced by various water conditions, making bacteria the unseen heroes of aquaponics [21].

»» For extra information, Visit The site “gogreenaquaponics” for more Components of aquaponics systems [22]

I.2.6. The Operating Principle of Aquaponics

Aquaponics is all about teamwork between fish, plants, and bacteria as shown in **“Figure I.4”**. Fish live in a tank, producing waste that turns the water nutrient-rich. This water is sent to the grow bed, where helpful bacteria transform the waste into a form that plants can absorb. As the plants take in these nutrients to grow, their roots naturally clean and filter the water. The purified, oxygenated water then flows back to the fish tank, creating a continuous, self-sustaining cycle where both plants and fish thrive together [5].

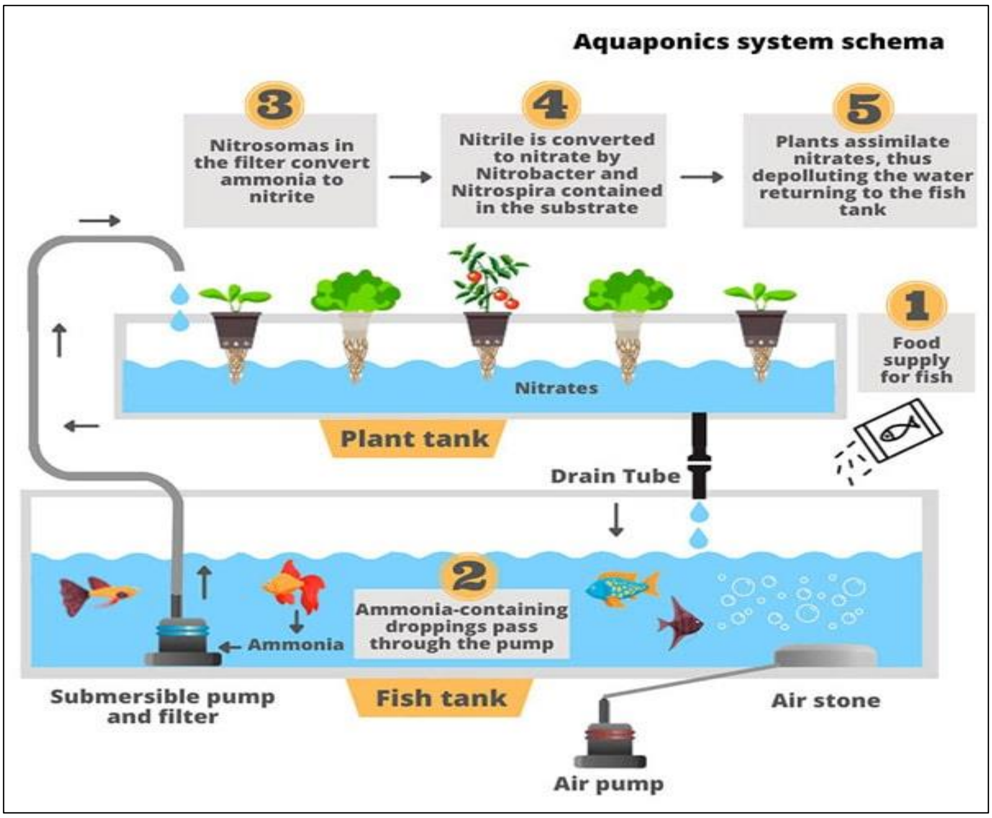

Figure.I.4. The principle of how an aquaponics system works. [23]

I.2.7. Introduction of Raspberry Pi in Farming [24][25]

In today's tech-driven world, even farming is getting smarter and at the heart of many of these changes is the tiny but powerful Raspberry Pi. This low-cost mini-computer is helping farmers, students, and innovators build smart systems to monitor crops, automate irrigation, all without needing expensive, high-end equipment.

Whether it's triggering a water pump when the soil gets too dry or using a simple camera setup to check plant health, Raspberry Pi brings practical, and easy control to the farm making advanced tech feel within reach, even in small-scale settings.

There are even full-scale open-source projects like FarmBot that use Raspberry Pi for automated, precision planting.

researchers at the University of Cambridge have shown how it can help farmers reduce crop diseases through real-time image analysis. It's a small device, but it's making a big difference in the field.

I.2.8. Surveillance in an Aquaponics system

Aquaponics is a sustainable agricultural system, Maintaining the right balance in this system requires continuous monitoring to ensure optimal water quality, plant growth, and fish health. Surveillance plays a vital role in detecting potential issues early, enabling farmers to make informed decisions and sustain productivity, in this case, we distinguish:

I.2.8.1. Small-scale surveillance ops

Small-scale aquaponics setups, such as those found in homes or small farms, typically rely on basic monitoring tools and manual observation. simple sensors can track parameters like water temperature, pH levels, and dissolved oxygen. Farmers often check these readings and make adjustments manually as needed to maintain balance in the system. While effective for smaller operations, this approach requires hands-on management and regular supervision [26].

I.2.8.2. Large-scale surveillance ops

In larger commercial aquaponics farms, manual monitoring becomes impractical due to the large size and complexity of the system. Instead, automated surveillance solutions, such as drones, AI and IoT technology, are used to oversee system health. the use of AI & drones equipped with high-resolution cameras to monitor plant growth, water conditions, and fish behavior. They can provide real-time data, enabling farmers to detect irregularities and take immediate corrective actions (or they can be taken automatically) [27].

I.2.9. Introduction of AI in Farming & Aquaponics surveillance

Artificial Intelligence is quietly transforming agriculture not by replacing farmers, but by giving them smarter tools to predict problems, optimize resources, and boost crop health. In aquaponics systems, where fish and plants grow together in a balanced ecosystem, AI helps maintain that balance by monitoring water quality, detecting plant illness, and automating feeding or lighting with real-time precision.

Instead of relying only on routine checks, farmers can now use AI to spot early signs of disease, adjust nutrient levels, and even Adjust temperatures all while reducing waste and improving yield. It’s like having a digital farming assistant that never rests.

From research on AI-based nutrient control systems to startups building fully autonomous aquaponics farms, the integration of AI is turning these eco-friendly systems into self-learning and self-regulating environments [28].

I.3. Surveillance & quadcopters

I.3.1. Introduction

Drones, or Unmanned Aerial Vehicles (UAVs), are aircraft systems that operate without an onboard human pilot. They can be controlled remotely or fly autonomously using pre-programmed flight plans or more complex dynamic automation systems. Equipped with various sensors and cameras, drones are utilized across multiple industries, including agriculture, surveillance, and environmental monitoring, mainly large-scale systems due to their capability to collect real-time data efficiently. In aquaponics systems, drones assist in monitoring water quality, fish health, and crop conditions, therefore enhancing operational efficiency and sustainability [29].

I.3.2. History of Drones

The concept of unmanned flight dates back to the 19th century, but modern drones emerged in the 20th century, primarily for military purposes. The first recorded use of UAVs occurred in 1849, when Austria used unmanned balloons carrying explosives to attack Venice, Later, in 1918, the United States developed the Kettering Bug, an early cruise missile prototype.

During World War II, drones were primarily used as target practice for fighter pilots. However, it was during the Cold War that UAVs gained significant importance for reconnaissance missions. The Vietnam War saw the introduction of the Ryan Model 147 Firebee, an early surveillance drone used by the U.S. military.

The 1990s marked a turning point with the development of the RQ-1 Predator, which played a crucial role in modern warfare, particularly in the Middle East, Today, drones such as the MQ-9

Reaper are used for both surveillance and targeted strikes, reducing the need for manned combat missions [30].

I.3.3. Types of Drones [31]

Drones come in various shapes and sizes, designed for different purposes. The most common types include:

I.3.3.1. Military Drones

Used for reconnaissance, surveillance, and combat (e.g., MQ-9 Reaper, Bayraktar TB2).

I.3.3.2. Commercial Drones

Used in agriculture, construction, and logistics (e.g., DJI Matrice, Amazon Prime Air drones).

I.3.3.3. Consumer Drones

Small quadcopters used for photography and recreational flying (e.g., DJI Mavic, Autel Robotics).

I.3.3.4. Industrial Drones

Employed in infrastructure inspections, energy sector monitoring, and emergency response.

I.3.3.5. Racing Drones

High-speed drones designed for competitive racing and sports events.

I.3.4. Applications of Drones [32]

Drones have transcended military use and are now integral to various industries:

I.3.4.1 Aerial Photography & Videography

Used in filmmaking, journalism, and real estate.

I.3.4.2 Agriculture

Monitor crops, spray pesticides, and optimize irrigation (Miller, 2020).

I.3.4.3 Disaster Relief & Search and Rescue

Locate survivors and assess damage in disaster-stricken areas.

I.3.4.4 Delivery Services:

Companies like **Amazon** and **Zipline** use drones for package and medical supply delivery.

I.3.4.5 Infrastructure Inspection

Monitor power lines, bridges, and pipelines to ensure safety and efficiency.

I.3.4.6 Law Enforcement & Security

Used for surveillance, traffic control, and emergency response.

I.3.4.7 Wildlife Conservation

Monitor endangered species and combat poaching.

I.3.4.8 Greenhouse Surveillance*:*

Large-scale greenhouses benefit from drone surveillance for monitoring plant health, irrigation, and pest control. Drones equipped with multispectral sensors and AI-powered analytics can detect plant diseases early, optimize resource distribution, and ensure sustainable farming practices, However, for small-scale greenhouses, the use of drones may not be necessary due to the manageable area size.

I.3.5. working Principles [32]

Drones are a marvel of modern engineering, combining aerodynamics, advanced electronics, and precision control systems. the key principles behind their operation are the following:

I.3.5.1. Aerodynamics and Lift

Drones generate lift through rapidly spinning propellers as shown in **"Figure I.5"**, which push air downward, creating an upward force that counteracts gravity. The design and angle of these propellers are crucial for efficient flight and stability. In multirotor drones, such as quadcopters, the coordinated variation in the speed of each rotor allows for precise control of movement and orientation [32].

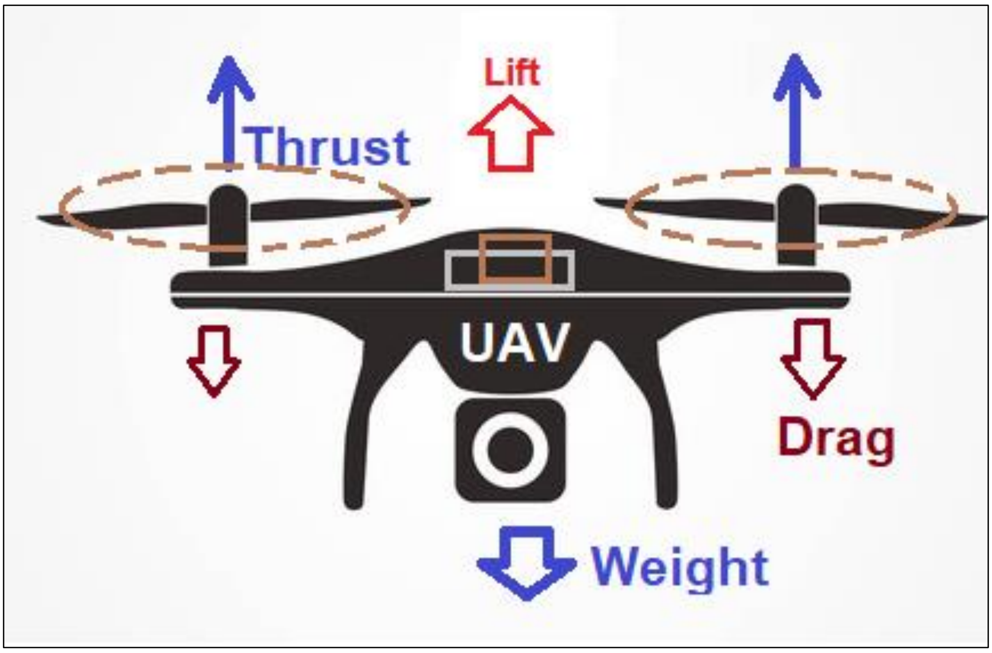

Figure.I.5. Aerodynamics and Lift demonstration. [33]

I.3.5.2. Propulsion System

The propulsion system of a drone consists of electric motors connected to the propellers. Electronic Speed Controllers (ESCs) regulate the power supplied to these motors, allowing for precise adjustments in propeller speed. By varying the speed of individual motors, the drone can maneuver in different directions, maintain stability, and perform complex flight patterns [32], as shown in **"Figure I.6"**.

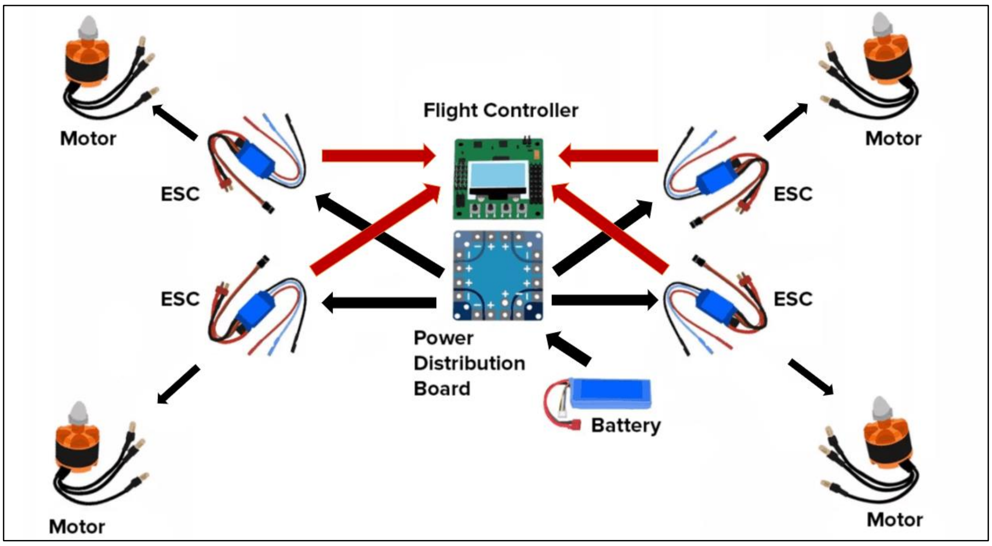

Figure.I.6. Drone power and control system architecture.[34]

I.3.5.3. Flight Control System [32]

At the heart of a drone sits the flight controller, which acts as the central processing unit. It receives input from various sensors, including gyroscopes, accelerometers and GPS modules, to monitor the drone's position, orientation, and speed. Based on this data, the flight controller adjusts motor speeds to maintain stability and execute pilot commands. These commands allow the pilot to do various actions and maneuvers which are shown in "**Figure I.7"** such as:

Elevation: This refers to the altitude of the drone, or how high it is off the ground

Yaw: Yaw is the rotation of the drone around its **vertical axis**, like turning your head left or right. When You adjust the yaw, the drone rotates horizontally

Roll: Roll involves tilting the drone **side to side**, which is controlled by the left or right movement of the joystick (on most controllers). It allows the drone to bank like an airplane

Pitch: Pitch refers to the **tilting forward or backward** of the drone, similar to tipping your head down or up. It's used to control forward or backward flight

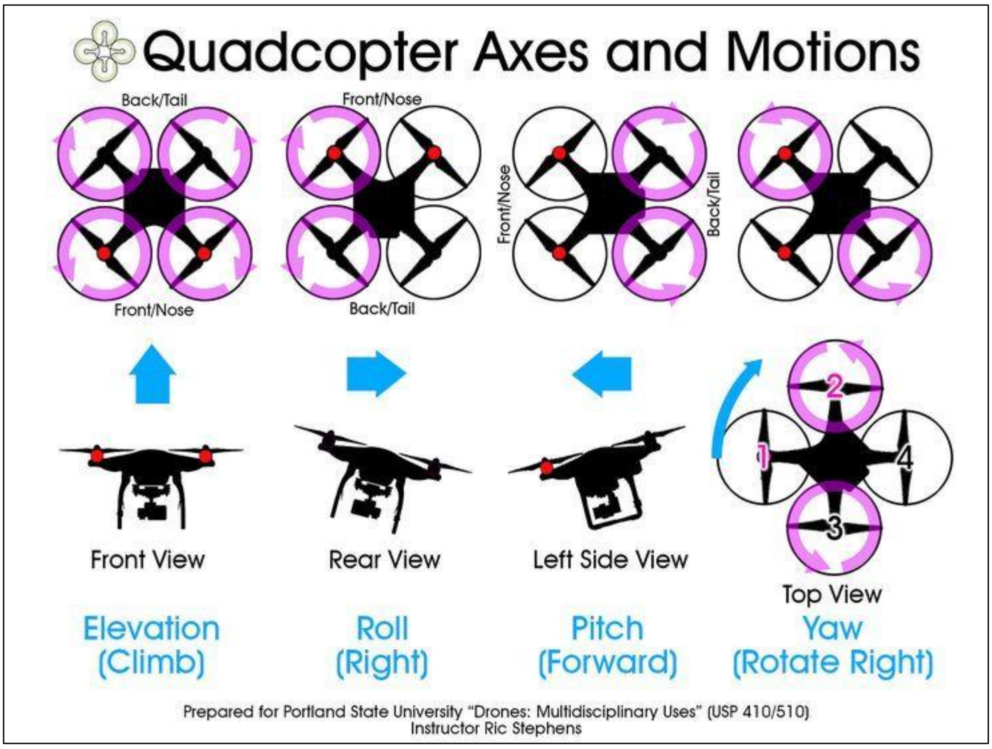

Figure.I.7. Quadcopter axes, motions and control mechanisms. [35]

I.3.5.4. Power Supply

Most drones are powered by lithium-polymer (Li-Po) batteries, known for their high energy density and lightweight properties. The capacity and voltage of these batteries determine the drone's flight time and performance. Efficient power management is essential, as it impacts the duration and safety of flight operations [32], check **"Figure I.8"** for a clearer vision.

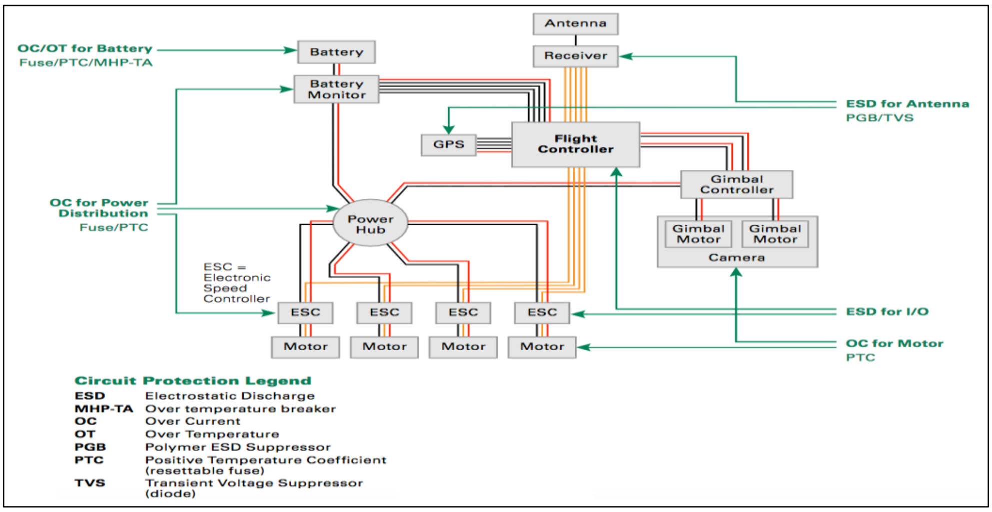

Figure.I.8. Drone power distribution and circuit protection diagram. [36]

I.3.5.5. Communication System

Drones are typically controlled remotely via radio signals as mentioned in **“Figure I.9”**. The remote controller sends commands to the drone's onboard receiver, which transfer the information to the flight controller. This setup allows pilots to maneuver the drone, adjust settings, and, in advanced models, receive real-time video feedback (FPV drones) [27].

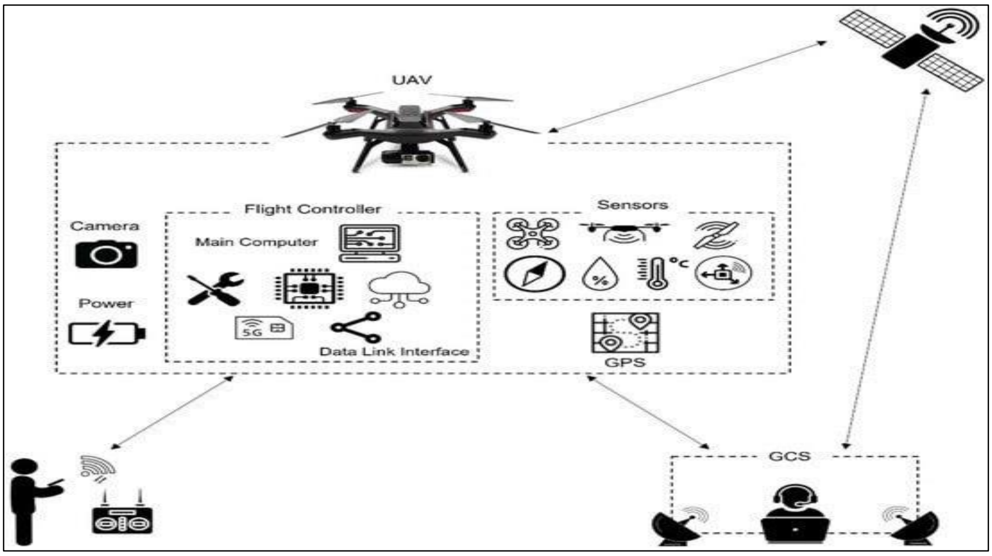

Figure.I.9. Technical diagram of a drone's communication system. [37]

I.3.5.6. Sensors and Navigation [32]

To navigate and maintain stability, drones rely on a suite of sensors:

- Inertial Measurement Unit (IMU): Combines accelerometers and gyroscopes to track the drone's orientation and movement;
- Magnetometer: Acts as a digital compass, providing heading information;
- Barometer: Measures atmospheric pressure to estimate altitude;
- GPS Module: Provides precise location data, enabling waypoint navigation and return-to-home functions. These sensors work in tandem to ensure accurate flight control and navigation;

In general, all the mentioned components and sensors work together to make a drone fully functional, **“Figure I.10”** shows the final set up for an operating drone

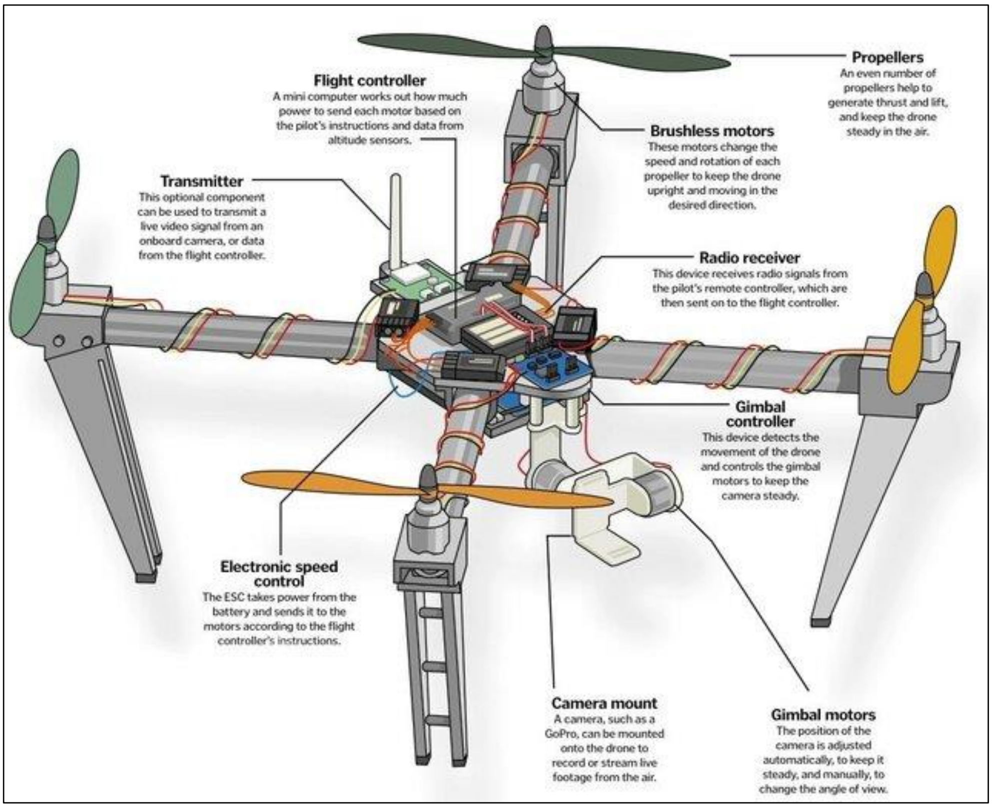

Figure.I.10. Technical diagram of a drone's full anatomy. [38]

I.4. Conclusion

The integration of drones to the side of Artificial Intelligence and Internet of Things in aquaponics surveillance & control marks a significant advancement in precision agriculture and sustainable food production. By utilizing Smart monitoring, aquaponics provides real-time data on water quality, plant health, and system efficiency, reducing manual labor and improving response times to environmental changes.

Automated surveillance through Raspberry Pi enhances the sustainability of aquaponics by optimizing resource use. Through precise monitoring, farmers can minimize water waste, ensure balanced nutrient distribution, and maintain ideal conditions for both fish and plants.

Despite the benefits, challenges such as high initial costs, regulatory restrictions, and technical expertise requirements must be addressed. Future advancements in drone autonomy, sensor miniaturization, and AI-driven analytics are expected to make this technology more accessible and cost-effective for aquaponics investors worldwide.

In conclusion, smart systems play a transformative role in modern aquaponics systems, providing a sustainable, efficient, and data-driven approach to food production. As technology continues to evolve, their integration will become increasingly indispensable, ensuring enhanced productivity and environmental conservation.

Chapter II:

System Hardware Overview

II.1. Introduction

In this chapter, we explore the heart of our smart aquaponics system, focusing on the hardware and design choices that drive its functionality. We dive into the different types of sensors and actuators used to maintain the perfect environment for both plants and fish. We also discuss how we upgraded our system's controller from the Raspberry Pi 4 to the more powerful Raspberry Pi 5 to meet growing demands. Finally, we present the design and components of a custom drone built to monitor the system from above, supporting AI-driven crop health analysis. Every decision was made with reliability, simplicity, and future expansion in mind.

II.2. Aquaponics System Overview

In this project, the aquaponics system is powered by a few important sensors and actuators to keep the plants and fish healthy and functioning.

II.2.1. Sensors

Aquaponics system relies on a few key sensors to monitor the environment and keep the plants and fish healthy. The main sensors we used are:

II.2.1.1. Temperature and Humidity Sensor DHT11

The DHT11 sensor **"Figure II.1"** is used to measure both the temperature and humidity levels inside the system. This sensor is critical for regulating the environment and ensuring that both the plants and fish are in optimal conditions. If the temperature or humidity strays too far from the ideal range, the system can automatically detect the issue [39].

Here's the Datasheet of the DHT11 sensor [39]:

- Ultra low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing.

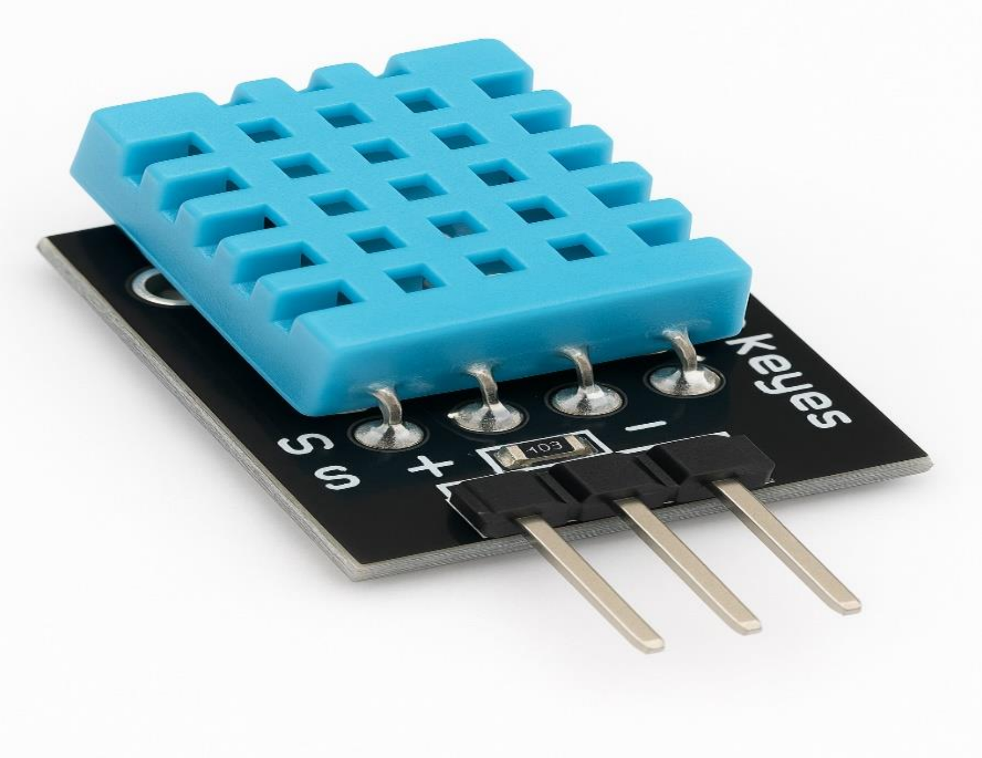

Figure.II.1.DHT11 sensor overview.[39]

II.2.1.2. Light Sensor

A light sensor as shown in **“Figure II.2”** is used to detect light levels in the environment [40]. This is particularly important for monitoring the plant’s exposure to light, which is essential for photosynthesis.

Here’s the Datasheet of the Light sensor Module [41]:

- ❖ Cadmium Sulfide (CdS) photoconductive cell
- ❖ Resistance decreases with increasing light intensity
- ❖ 1000 lux light resistance : 400 Ω
- ❖ 10 lux light resistance : 9 kΩ
- ❖ Rise time at 1000 lux : 2.8 ms
- ❖ Fall time at 1000 lux : 48 ms
- ❖ Operating temperature range: -60°C to +75°C
- ❖ Maximum voltage: 320V (AC or DC peak)

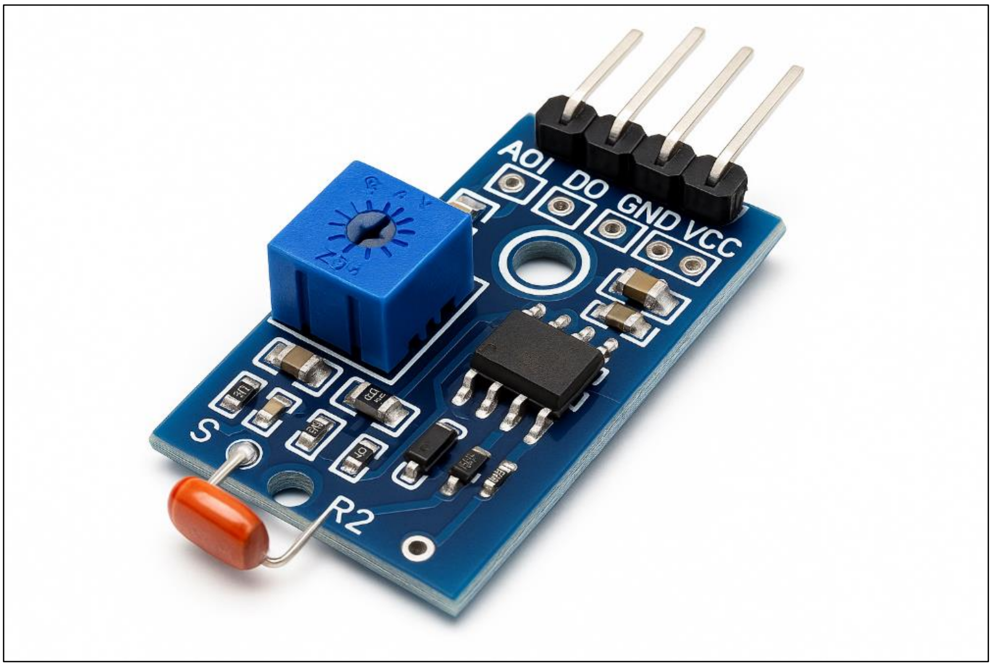

Figure.II.2. Light sensor module overview.[40]

II.2.1.3. PH Sensor

A **PH sensor "Figure II.3"** is used to monitor the acidity or alkalinity of the water in the aquaponics system. Maintaining the right pH balance is crucial for the health of both the plants and the fish [42].

Here's the Datasheet of the PH sensor [43]:

- ❖ Voltage supply: 3.3–5.5V (for compatibility with most microcontrollers)
- ❖ Analog output for easy integration into microcontroller-based systems (e.g., Arduino)
- ❖ Temperature compensation capabilities for accurate readings in varying temperatures
- ❖ Wide pH measurement range: typically, from 0 to 14 pH
- ❖ High-precision electrode for stable readings and long-term reliability
- ❖ Standard BNC connection or pH2.0 interface for easy connection to sensors
- ❖ Maximum operating temperature: ranges from 0°C to 130°C depending on the model
- ❖ Pressure resistance: capable of withstanding up to 690 kPa (100 psi) for industrial applications

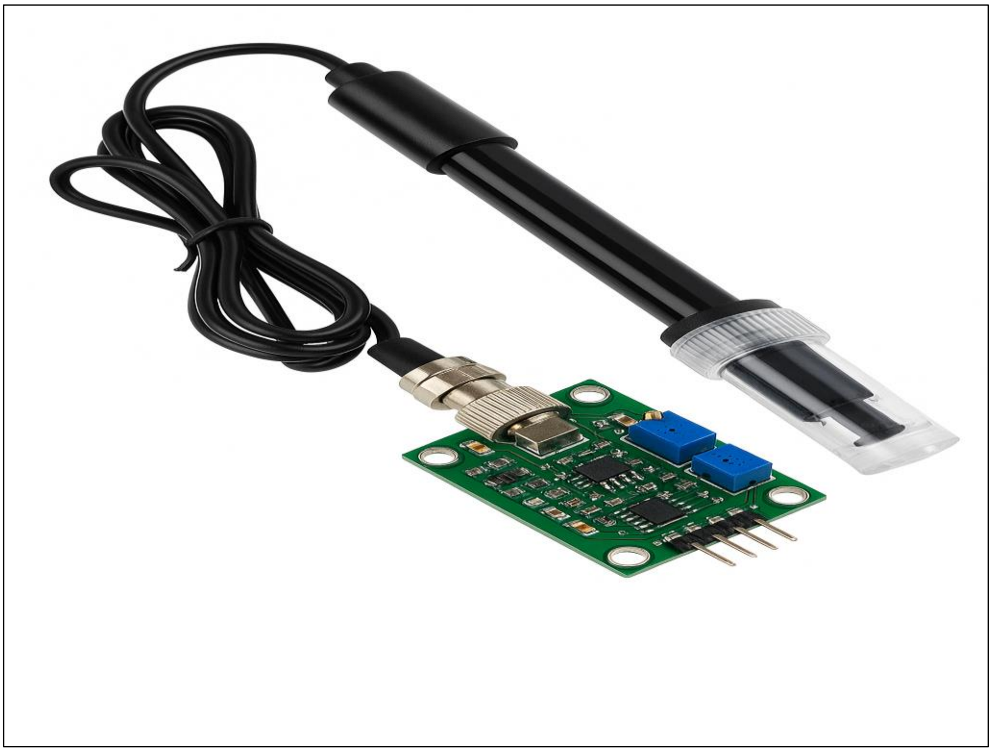

Figure.II.3. PH sensor overview.[44]

II.2.1.4. Water Level Sensor

The **water level sensor** ensures that the water levels in the system remain stable. This sensor detects if the water level is too high or low it takes action in order to adjust it. Maintaining the proper water level is critical for both the fish and plants, as it ensures the roots are submerged and the fish have enough space to thrive [45]

in summary, the sensors in the system are essential for maintaining the right conditions for both the plants and the fish. By constantly monitoring factors like temperature, humidity, light, pH, and water levels, these sensors ensure that the environment stays stable and healthy. Their real-time data helps the system make adjustments automatically, keeping everything in balance.

II.2.2. Actuators

In this part of the system, actuators are responsible for carrying out actions based on the information provided by the sensors. These components help control the environment and ensure that the conditions remain ideal for both the plants and the fish. The key actuators used in the system are:

II.2.2.1. Fan

The fan **"Figure II.4"** is used to regulate the temperature inside the system. When the temperature rises above a certain threshold, the fan receives an order to cool down the environment. This helps prevent overheating, which could harm both the plants and the fish. Proper temperature control is crucial in aquaponics systems to ensure healthy plant growth and fish survival [46].

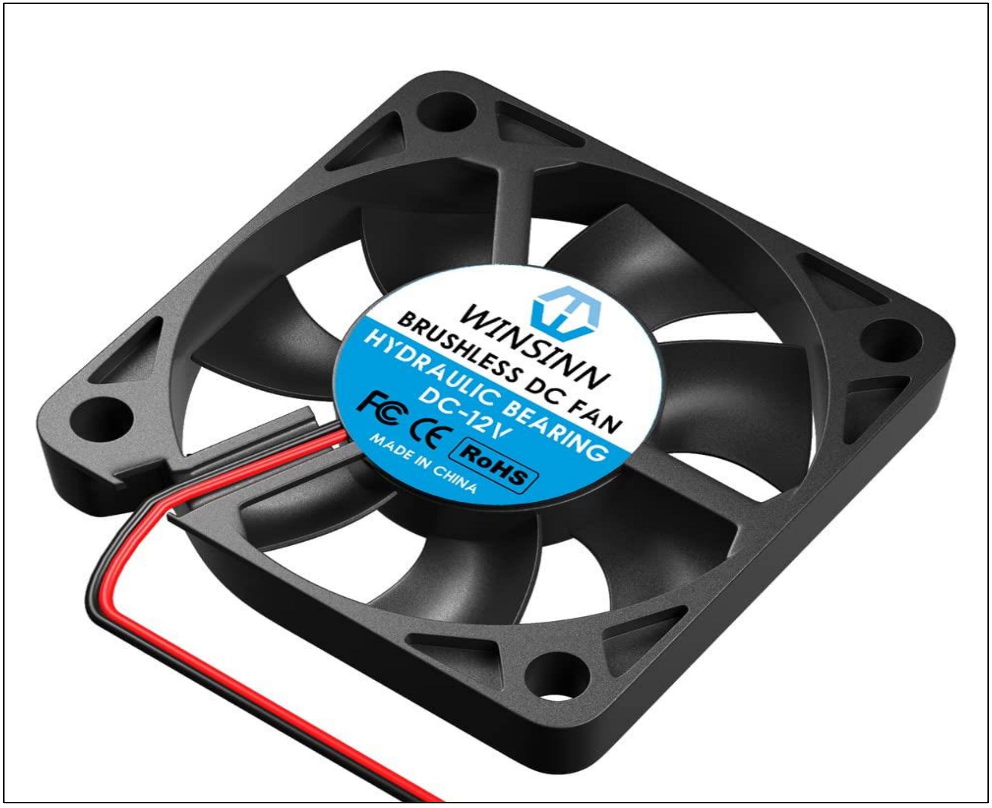

Figure.II.4. Cooling fan overview.

II.2.2.2. Heater

The is used to maintain a stable temperature, especially during colder conditions. When the temperature drops below a certain level, the heater is turned on to keep the system warm enough for the plants and fish to survive. Maintaining the right temperature is essential for the metabolic processes of both plants and fish in aquaponics systems [47]

II.2.2.3. Air pump

The **air pump** is used to regulate humidity levels in the system. When the humidity is low, the air pump is activated to release moisture into the air, helping to keep the environment suitable for the plants and fish. This is particularly useful for maintaining optimal growing conditions for the plants [48]

II.2.2.4. Water Pump (fill and drain Pump)

Water pumps control the flow of water throughout the system. The **filling pump** ensures that the system has enough water, while the **drain pump** removes excess water to prevent flooding or dehydration. Maintaining proper water levels is critical for both plant roots and fish health. The coordinated use of filling and drain pumps ensures that water levels remain stable and the plants have sufficient access to nutrients

To summarize, actuators play a crucial role in maintaining optimal environmental conditions for plant and fish growth. From regulating water pH levels to adjusting the temperature and light conditions, these devices ensure that the system operates smoothly. The key actuators, such as pumps, fans, lights, and aerators, work together to create a balanced ecosystem where both plants and aquatic life can thrive. By responding to sensor inputs, actuators enable automatic adjustments to environmental factors, reducing manual intervention and enhancing system efficiency.

The effective functioning of sensors and actuators relies heavily on the **controller**, which acts as the brain of the system. The controller is responsible for interpreting sensor data and sending commands to the actuators accordingly ensuring that each actuator performs its task based on real-time data collected from the sensors.

- ***Important Notes:***

1/-The communication between actuators and the controller Is established using relays that work as switches on/off (in our case AC: 10A/250V or DC: 10A/30V) **“FigureII.5”**

2/-The energy source for the aquaponics system is consisted of Two sets of 12V/1000mAH Car Batteries to satisfy Voltage and current needs “**FigureII.6”**

Figure.II.5. 10A/250V Relay Switch.

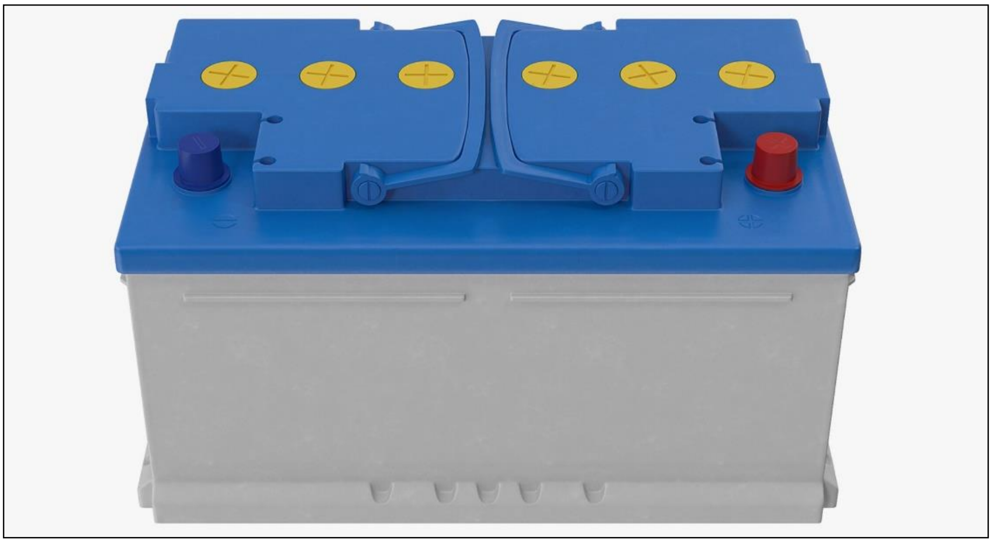

Figure.II.6. One set of 12V/1000mAh Car Battery.

II.2.3 microcontroller

Originally, we used the Raspberry Pi 4 Model B **"FigureII.7"**. It was a solid choice: it had enough processing power (with its quad-core CPU), enough RAM (up to 8GB versions), and good connectivity options (USB 3.0 ports, Wi-Fi, Bluetooth) to handle running the sensors, managing the actuators, and even dealing with AI-based disease detection. It was affordable, flexible, and small enough to fit neatly into the project setup.

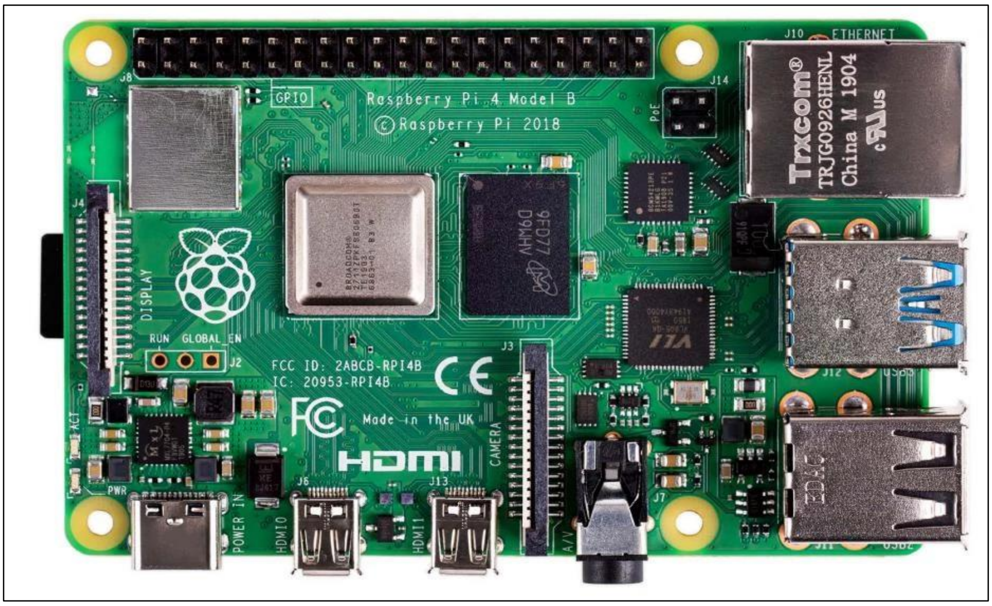

Figure.II.7. Raspberry pi 4 Model B [49]

But as the project grew more ambitious with heavier image processing, drone integration, and faster sensor communication needs, we realized the Pi 4 was starting to show its limits, so we decided to upgrade to the later version of the raspberry Pi series which is "Raspberry Pi 5"

II.2.3.1. Raspberry Pi 5: A Quick Overview

The Raspberry Pi 5 **“FigureII.8”** is the latest and most powerful version of the Raspberry Pi family, officially released by the Raspberry Pi Foundation in late 2023. It features a faster quad-core 64-bit ARM Cortex-A76 CPU running at 2.4 GHz, a new VideoCore VII GPU for better graphics, and much faster RAM and I/O speeds. Thanks to these improvements, the Pi 5 acts almost like a small desktop computer, making it perfect for heavier tasks such as AI, robotics, and complex smart systems.

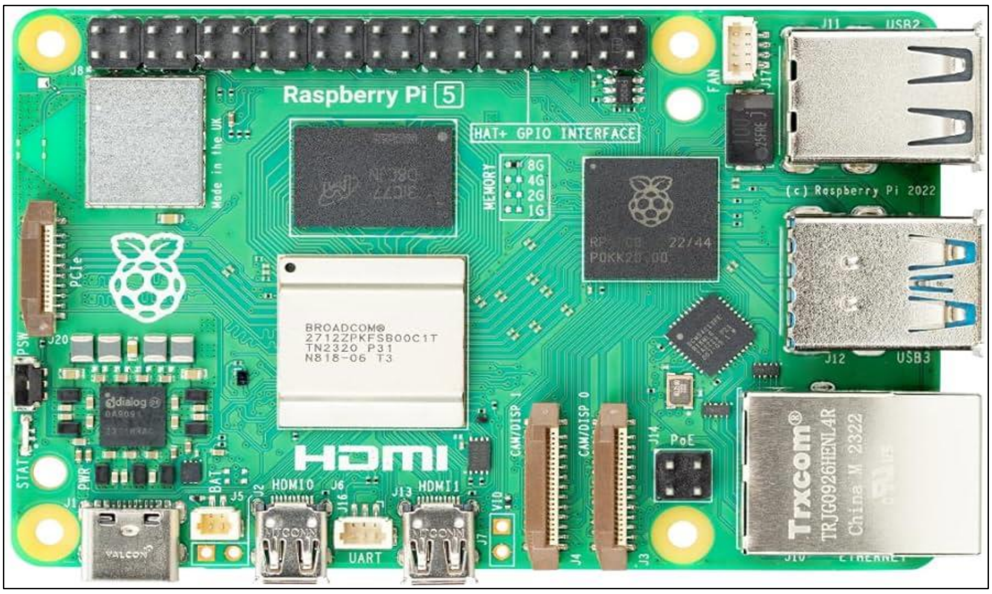

Figure.II.8. Raspberry pi 5. [50]

II.2.3.2. Raspberry Pi 4 vs Raspberry Pi 5

When selecting hardware for an aquaponics system, particularly one that integrates AI capabilities, the choice of microcontroller is pivotal. The Raspberry Pi series has been a popular choice due to its Flexibility and performance.

This comparison between the Raspberry Pi 4 and the newer Raspberry Pi 5 highlights the advancements in processing power, memory, connectivity, and additional features that may influence the decision for your project.

The purpose of this comparison is to highlight why the Raspberry Pi 5 is a better choice over the Raspberry Pi 4, showcasing its improvements in processing power, memory, and additional features that contribute to better performance for an aquaponics system with AI integration.

The comparison is presented in a table **“Table II.1”** that highlights the key differences and improvements of the Raspberry Pi 5 over the Raspberry Pi 4, focusing on aspects such as processing power, memory, and other critical features.[49] [50]

Feature	Raspberry Pi4	Raspberry Pi5
Release date	June 2019	October 2023
CPU	Quad-core Cortex-A72 (1.5 GHz)	Quad-core Cortex-A76 (2.4 GHz)
RAM options	2 GB, 4GB, 8GB LPDDR4	2 GB, 4GB, 8GB LPDDR4X
GPU	Videocore VI	Videocore VII
Storage	Micro SD card	Micro SD card + PCIe SSD
USB ports	2 × USB 2.0, 2 × USB 3.0	2 × USB 2.0, 2 × USB 3.0
Ethernet	GIGABIT Ethernet	True GIGABIT Ethernet
Wireless connectivity	Wi-Fi 802.11ac, Bluetooth 5.0	Wi-Fi 802.11ac, Bluetooth 5.0
Power supply	USB-C, 5V/3A	USB-C, 5V/5A
Video output	2 × Micro-HDMI (4K)	2 × Micro-HDMI (4K 60FPS)
PCIe support	No	Yes
Camera interface	2 × MIPI CSI	2 × MIPI CSI
General performance	Good for most projects	Better for AI tasks

Table.II.1.Comparison table Between Raspberry PI 4 and PI 5.

Note: According to the table, The Raspberry Pi 5 offers significant improvements in processing power, RAM, and PCIe support, making it a better choice for more demanding applications like AI tasks.

To conclude, the aquaponics system efficiently integrates fish and plant cultivation by maintaining balanced environmental conditions through the use of sensors and actuators. In order to enhance system speed, reliability, and accommodate future expansions, we upgraded from the Raspberry Pi 4 to the Raspberry Pi 5, which offers superior performance and improved hardware compatibility.

II.3. Custom Drone Overview

To enhance the monitoring and management of the aquaponics system, a custom drone was designed and built. The drone plays a key role in capturing aerial images and videos of the crops, providing valuable data for AI-based disease detection. This section presents the essential components used in the construction of the drone, along with their specific purposes and functions.

II.3.1. Main Components of a Drone:

II.3.1.1. Frame (Chassis)

The frame serves as the structural foundation of the drone, holding all components together in a secure and stable manner. It must be lightweight to enhance efficiency while maintaining durability to withstand crashes or rough landings. Frames are typically made from materials such as carbon fiber, aluminum, or plastic, each offering different levels of strength and weight. The design of the frame also plays a role in aerodynamics and stability, impacting how the drone moves through the air [51].

• For our project, we picked the **Mark 4.7-inch** frame, designed with a **295mm arm length** and a robust **5mm arm thickness "Figure II.9".**

Figure.II.9. Mark 4 7inch 295mm Arm thickness 5mm FPV racing drone.[52]

II.3.1.2. Flight & Electronic speed controllers (FC & ESC)

The flight controller is the brain of the drone, processing signals from the pilot and onboard sensors to ensure smooth and stable flight. It continuously adjusts motor speeds using the ESCs to keep the drone level and counteract external factors such as wind. The flight controller also enables different flight modes, such as altitude hold, GPS-assisted navigation, and autonomous flight [53].

• For our drone setup, we chose the Speedybee F405 Mini 35A, a compact yet powerful flight controller. With its integrated F4 processor and 35A ESC capability, this controller **"Figure II.10"** provides excellent responsiveness and reliability for controlling flight dynamics. Its small form factor allows for easy integration into our custom drone while delivering stable and smooth performance.

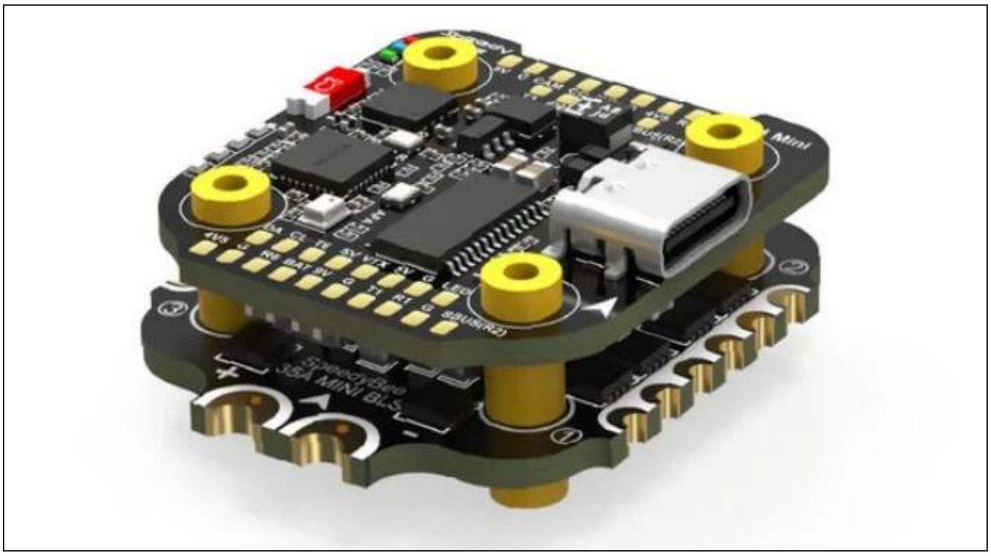

Figure.II.10. Speedybee F405 mini BLS 35A 20*20mm stack.[54]

II.3.1.3. Propulsion System

The propulsion system is responsible for generating the thrust needed for the drone to take off, hover, and maneuver in different directions. It consists of motors, propellers, and Electronic Speed Controllers (ESCs). The motors spin the propellers, which push air downward to create lift, while the ESCs regulate the power supply to each motor, allowing the drone to adjust speed and direction as needed. A well-balanced propulsion system ensures stable flight and efficient power usage [55].

• In our system, we picked the **RS2205 2300KV brushless motors**, known for their strong thrust and quick responsiveness. With their 2300KV rating and high spinning speed, we paired them with **HQPROP DP7X4X3 7040 3-blade PC propellers** to work hand-to-hand and ensure high efficiency and optimal aerodynamic performance **"Figure II.11"**.

Figure.II.11. RS2205 2300KV Brushless Motors [56] & HQPROP DP7X4X3 7040 3-BLADE PC propeller.[57]

II.3.1.4. Sensors

Drones rely on multiple sensors to gather data and make real-time flight adjustments. A gyroscope and accelerometer (both integrated in the internal system of our FC) detect changes in orientation and movement, helping the flight controller keep the drone stable. A GPS module provides precise location tracking, enabling autonomous navigation and return-to-home (RTH) functions. A barometer (also integrated in the internal system of our FC) measures altitude by detecting air pressure changes, allowing the drone to maintain a consistent height. Lastly, a magnetometer (compass integrated in the FC) helps the drone determine its direction, ensuring it stays oriented correctly during flight [58].

• In our case, since we need only a GPS Module, we picked the **Walksnail WS-M181 GPS (M10 GNSS module) "Figure II.12"**, known for its fast satellite connection, high positioning accuracy, and lightweight design which are ideal for drones.

Figure.II.12. Walksnail WS-M181 GPS (M10 GNSS module).[59]

II.3.1.5. Communication System

The **communication system** is a vital part of any drone because it enables the connection between the **pilot** (on the ground) and the **drone** (in the air). Without it, there would be no way to manually control the drone's movements or receive critical data during flight [60].

In a typical setup, the communication system includes:

- **Radio Transmitter (TX) :**

Handheld controller used by the pilot to send commands (move, turn, speed up, slow down, etc.)

►For our project with picked the Jumper 2.4GHz ELRS TX Module AION Nano T-Pro **"Figure II.13"**

Due to its low cost and practically long operating range

Figure.II.13. Jumper 2.4GHz ELRS TX Module AION Nano T-Pro.[61]

- **Radio Receiver (RX) :**

Mounted on the drone, it receives the signals from the transmitter and sends them to the Flight Controller (FC).

►For our project with picked the ELRS AION RX 2.4GHz Mini Receiver **"Figure II.14"** Due to its low cost and practically long operating range

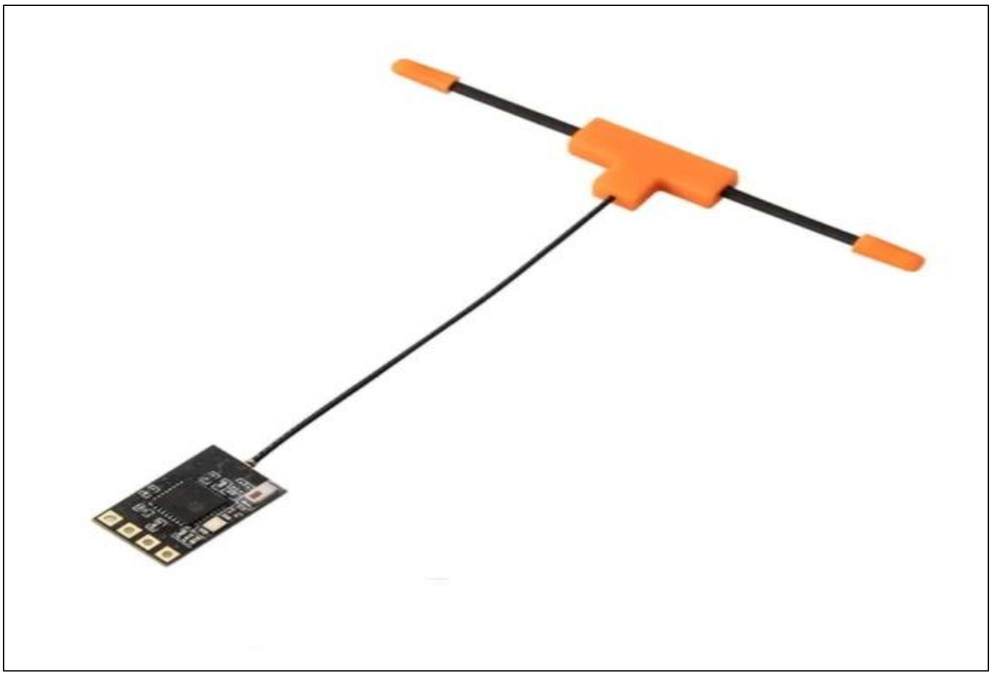

Figure.II.14. ELRS AION RX 2.4GHz Mini Receiver.[62]

The drone can operate Autonomously but we want the full manual control, now in order to control the drone manually, we need a remote controller that operates with our Transmitter and receiver [63]

►For our project with picked the Radiomaster Zorro ELRS remote controller **"Figure II.15"** because of its low price and compatibility with our equipment

Figure.II.15. Radiomaster Zorro ELRS Remote Controller.[64]

II.3.1.6. Battery & Power System

The battery supplies power to all drone components and satisfies energy needs for motors, flight controller, and sensors. Most drones use Lithium-Polymer (LiPo) batteries due to their high energy capacity, lightweight design, and high discharge ratio. The Power Distribution Board (PDB) ensures that power is evenly distributed to different components, preventing overloading or failures. Battery life directly impacts flight time, making energy efficiency a critical factor in drone design [65]

►For our project with picked the Gaoneng GNB 2600mAh 6S 22.2V 120C XT60 Battery **"Figure II.16"** due to its acceptable capacity and high discharge ratio

Figure.II.16. Gaoneng GNB 2600mAh 6S 22.2V 120C XT60 Battery.[66]

To make energy consumption even better, a capacitor **"Figure II.17"** is presented to smoothen voltage spikes and prevent ESC's overheating or destruction (the bigger the capacitors capacitance the smoother the energy drawn from the battery)

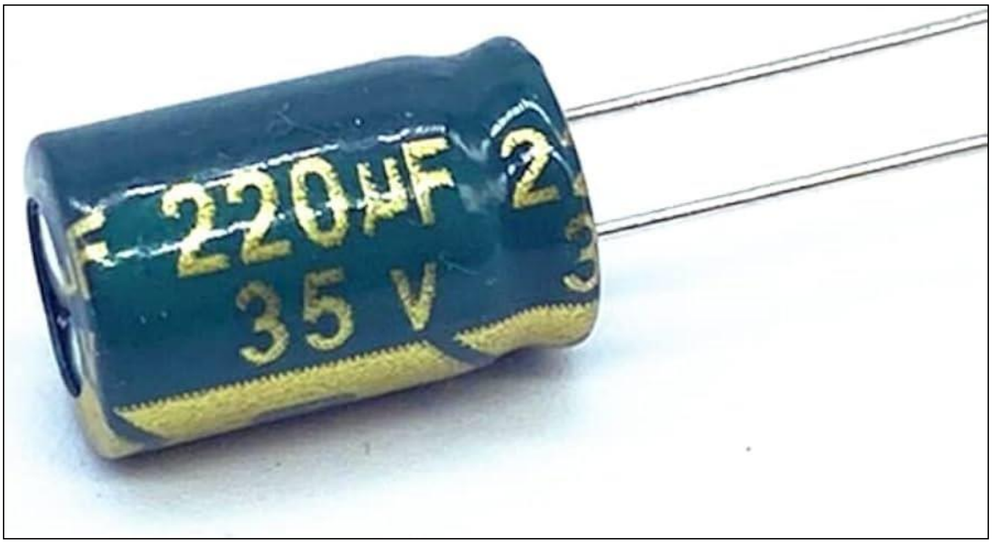

Figure.II. 17. Voltage Smoothing Capacitor 220uF/35V.

Note: the connection between the battery and the ESC needs a XT60 male to female plug adapter "Figure II.18"

Figure.II.18. XT60 male to female plug adapter.[67]

II.3.1.7. Camera & Gimbal

The drone is equipped with a camera to capture aerial footage, used for photography, videography, and mapping applications (or to capture crop's pictures in our case). However, camera movement can be affected by drone vibrations and wind. A gimbal stabilizes the camera, ensuring smooth and clear video recordings. Two-axis gimbals stabilize roll and pitch, while three-axis gimbals add stabilization for yaw, resulting in high-quality footage even in turbulent conditions [68].

►For our project, gimbal wasn't available, so we used a Runcam Thumb Pro W 4K V2 155° wide angle alongside 3D Printed TPU Mount **"Figure II.19"** to stabilize the footage and keep the camera in place as much as possible

Figure.II.19. Runcam Thumb Pro W 4K V2 155° wide angle [69] alongside 3D Printed TPU Mount.[70]

II.3. Conclusion

This chapter has provided a detailed look into the hardware backbone of our smart aquaponics system. From the selection of essential sensors and actuators to the decision to upgrade our controller, every component was chosen to create a reliable smart setup. The integration of a custom-built drone further expands the system's monitoring capabilities, paving the way for advanced AI applications in agriculture. By carefully balancing functionality, cost, and performance, we have built a system that is not only capable of maintaining an ideal environment for plants and fish but also ready to evolve with future technological advances. This foundation sets the stage for a smarter, more sustainable approach to modern farming.

Chapter III:

System Software Overview

III.1. Introduction

Software is what really brings our hardware to life, making the whole system smart and able to run by itself. In this chapter, we'll dive into how the Raspberry Pi collects data from sensors and controls the environment through actuators, how AI steps in to spot crop diseases early, and how everything including our drone work together as a team. Every script, every line of code, and every application used plays a key role in helping the aquaponics system run smoother, smarter, and with less hands-on effort.

III.2. Aquaponics System Overview

III.2.1. Introduction

Setting up the Raspberry PI 5 was a major step in turning our aquaponics system from a collection of sensors and hardware into a fully working smart farm. In this section, we'll walk through the basic setup of the PI, the software environment we built around it, and how it ties together all the sensors, actuators, and AI processing in a smooth and a reliable way.

To start with, we needed a stable and lightweight operating system that could handle multiple tasks without slowing down. For that reason, we installed Raspberry PI OS, the official operating system optimized for the hardware. It provided all the essential features we needed while still being light enough to keep the PI 5 running fast and efficiently.

III.2.2. Setting up Raspberry PI's OS

In order to set the Raspberry pi OS, we used the official Software provided by the Raspberry PI community which is "Raspberry PI imager", a light software used to download and boot the OS into the PI 5 [71], "**Figure III.1**" corresponds to Imager's interface:

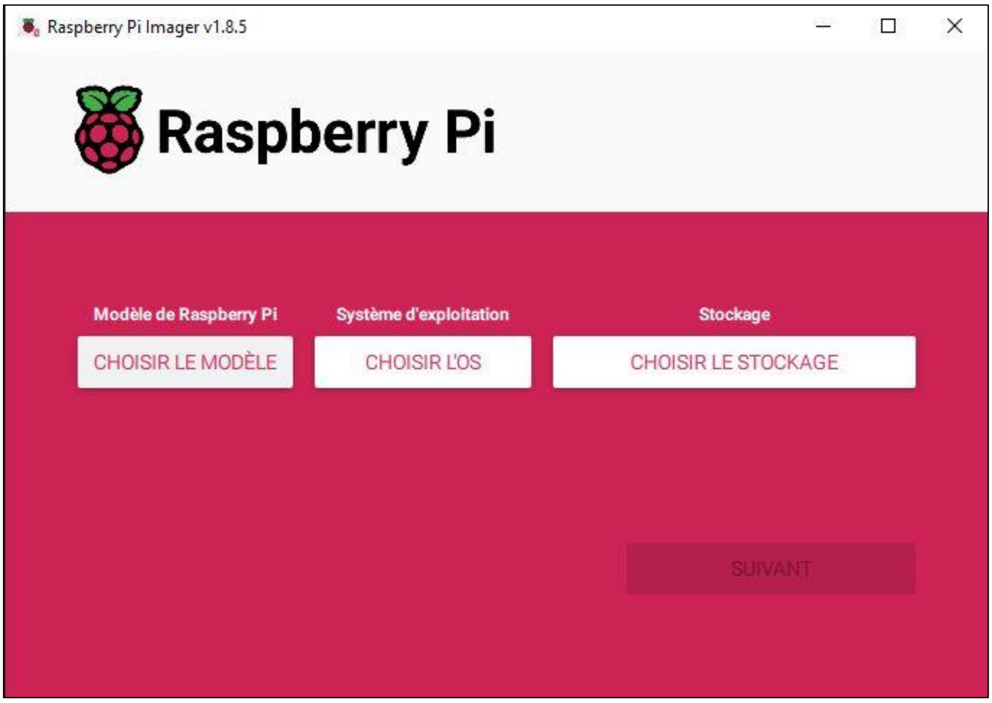

Figure.III.1. Raspberry PI imager interface.

III.2.2.1. Picking up the PI model

First thing to do is to pick the Raspberry PI model as shown in "**Figure III.2**":

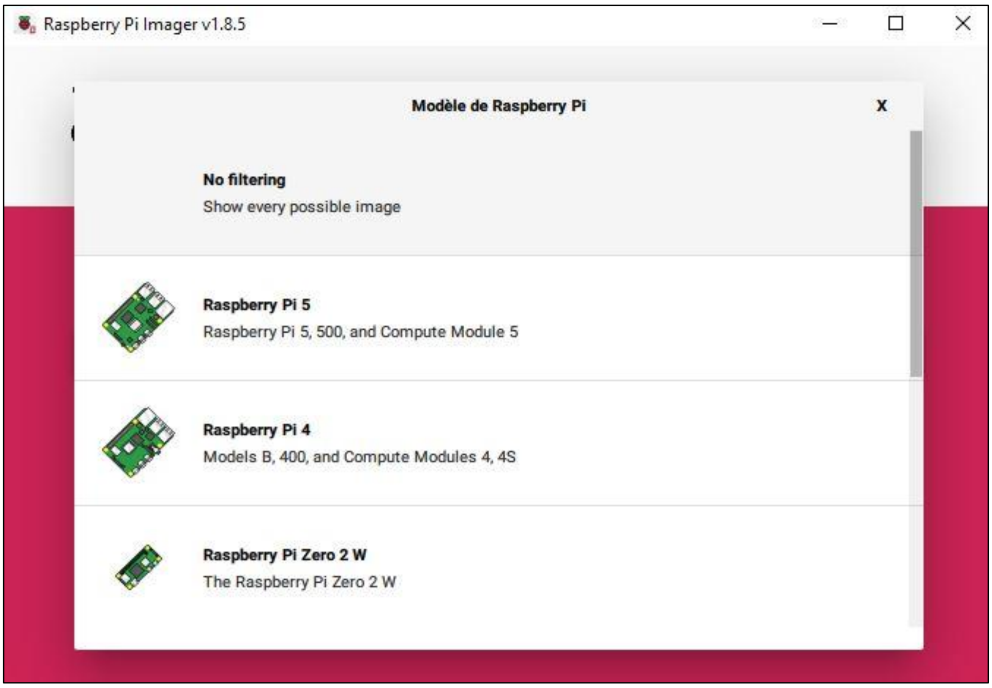

Figure.III.2. Model selection through Raspberry PI imager.

III.2.2.2. Picking up a suitable OS

Since we're using the PI 5, we select it and then head to the OS selection section, **"Figure III.3"** illustrates the process:

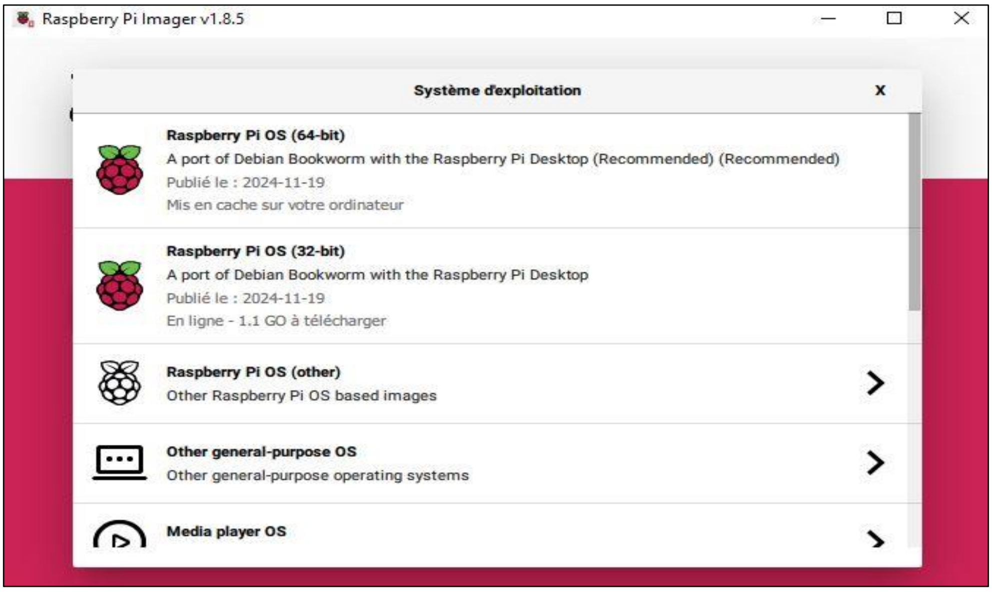

Figure.III.3. Operating system selection through Raspberry PI imager.

III.2.2.3. booting up the selected OS

Now for the last step, as shown in **"Figure III.4"**, pick a storage device (an empty 16GB+ SD Card /USB Disk) to boot our OS (64-bit) then click next to start downloading

Figure.III.4. Booting OS onto the storage through Raspberry PI imager.

III.2.3. Setting up VNC and SSH

After the Raspberry PI 5 was booted and ready to go, it normally would require a screen, a keyboard, and a mouse to operate directly. To avoid setting up extra hardware every time we wanted to work on it, we enabled VNC and SSH options.

VNC allowed us to remotely view and control the PI's full desktop environment from a laptop [71]

SSH gave us quick command-line access for running scripts or troubleshooting without needing a physical connection [71].

Here's how to do that; first attempt need a keyboard and a mouse alongside an HDMI monitor

III.2.3.1. Connecting pi5 to a network through wi-fi

in our case a phone's hotspot was enough as shown in **"Figure III.5"**:

Figure.III.5. Connecting Raspberry PI 5 to a local network.

III.2.3.2. Enabling VNC and SSH options

Through Raspberry pi configuration:

III.2.3.2.1 Enabling VNC option

Raspberry PI icon → Preferences → Raspberry PI Configuration → Interfaces → VNC → Enabled → OK [71]

the entire process is elaborated in **"Figure III.6"**

III.2.3.2.2 Enabling SSH option

Raspberry PI icon → Preferences → Raspberry PI Configuration → Interfaces → SSH → Enabled → OK [71]

The entire process is elaborated in **"Figure III.6"**

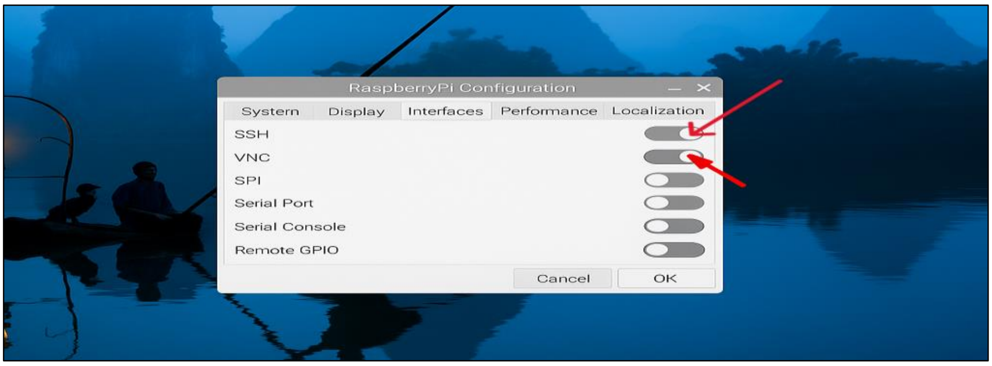

Figure.III.6. Enabling SSH & VNC for remote controlling.

III.2.3.3. Finding the address of the host

In PI's terminal, run the command:

>> **ifconfig** [71]

This will show you the address of the host, something like **192.168.X.X.**

check **"Figure III.7"** for a clearer vision:

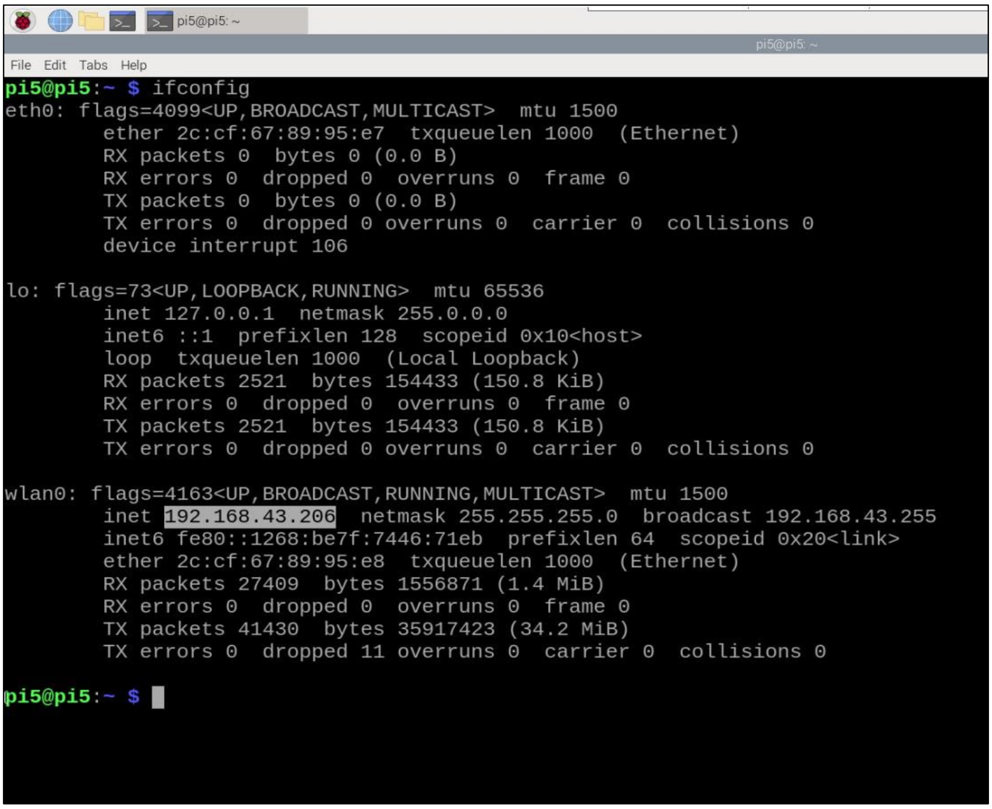

Figure.III. 7. Locating the address of the host.

III.2.3.4. Remote access & control

through the VNC viewer, we can access and/or control our PI remotely like shown in **"Figure III.8"** on condition they stay connected to the same network (if a new network is presented, a new address must be taken) [71]

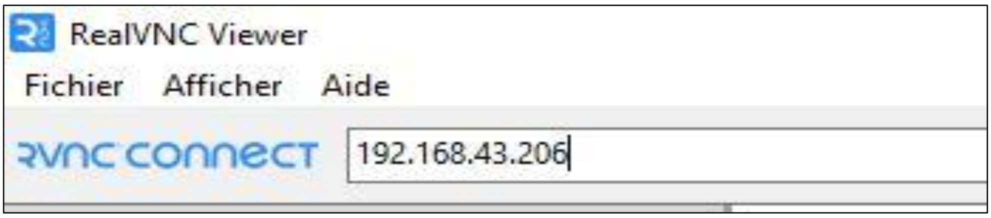

Figure.III.8. Connecting to PI 5 through VNC viewer.

III.2.4. Environment and libraries Installation

Now that the Raspberry Pi setup is all set, the focus moves to the code and the libraries installed. The aim is to make sure everything runs smoothly and efficiently, in order to assure that everything run smoothly, we must create a workshop which is also known as an **"environment"** [71], then head to the installation of libraries in use.

III.2.4.1. Environment setting process

First, we created a special environment to contain all needed libraries, through terminal, here's how:

a/. Environment creation step

In PI's Terminal, run the code: >> **python3 -m venv ai-env** [71]

#New environment named ai-env will be created

b/. Environment Activation/Access step

next, activate the created environment in order to start installing necessary libraries to, here's how:

in Terminal, run the code: >> **source ai-env/bin/activate**

#Activate the environment named ai-env

III.2.4.2. Libraries installation and checking step

last step is installing and then checking libraires; here's how:

a/. installing process

in Terminal, run the code: **>>pip install RPi.GPIO adafruit-circuitpython-dht psutil pyserial opencv-python picamera picamera2** [71]

#This is how to install all needed libraires for sensors and cameras processing

A progress bar will show the installation progress from 0% to 100% of each library

b/. checking process

in Terminal, run the code: >>**pip list**

#This will show a list of all installed libraries

III.2.5. Executing the system's code

now since all the libraries are in place, we can run the code with success, here's the final version of our aquaponics system's sensors and actuators code explained in a brief way using a simple diagram **"figureIII.9"** (excluding camera due to availability and budget constraints):

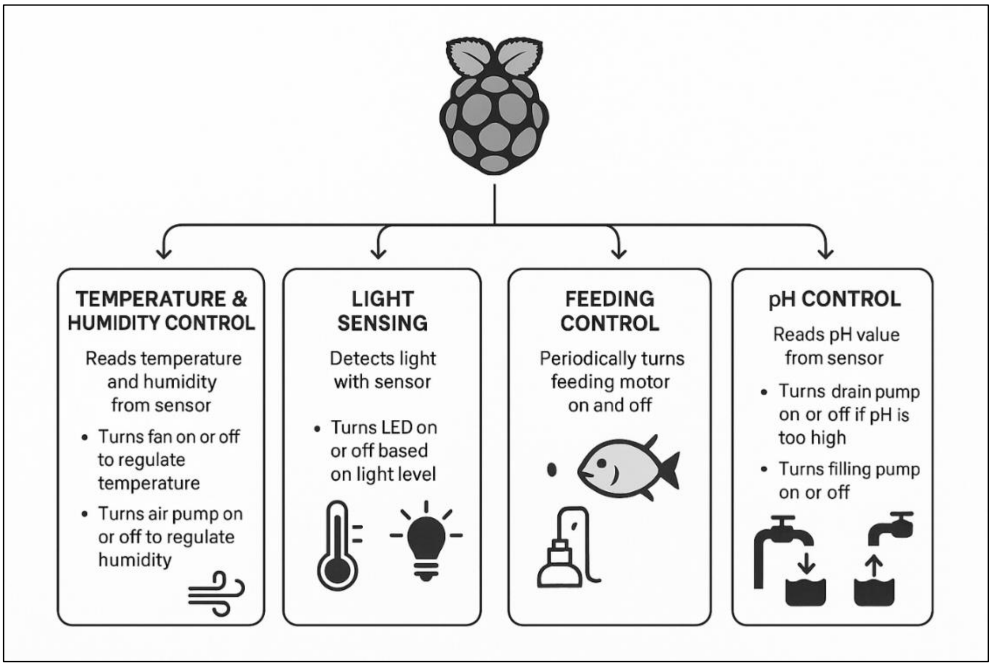

Figure.III.9. Sensors and Actuators Code demonstration.

To access and check the full code, check the Drive link for the provided Sensors and Actuators Final code [72]

now since the final code is ready, the system is good to go, it just needs all sensors and relays to be attached to their proper pins then run the code as shown in **“FigureIII.10”**:

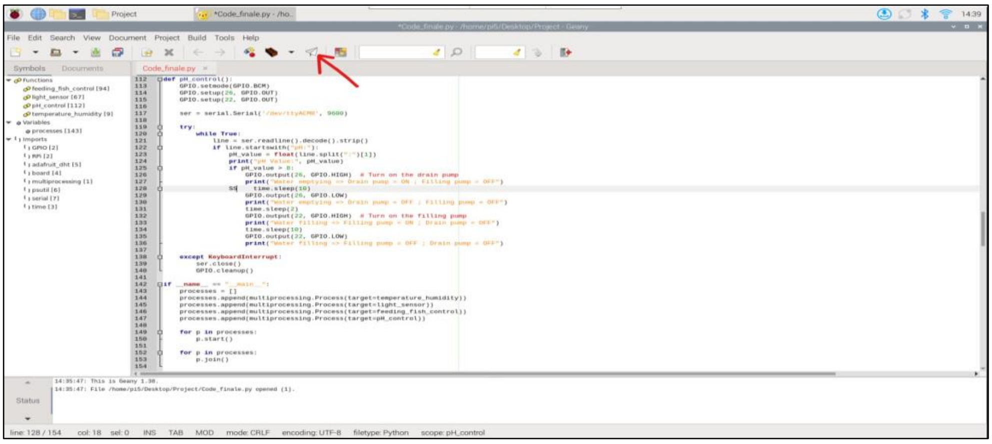

Figure.III. 10. Sensors and Actuators Code execution.

III.3. Disease’s Detection System Overview

III.3.1. Introduction

For this section, we dive into the software behind the smart lettuce disease detector. The core of the system is an AI model trained to analyze images of lettuce and classify them into categories such as bacterial, fungal, or healthy. Using machine learning algorithms like convolutional neural networks in our case (CNNs), the system processes images and provides accurate disease predictions. This Section will cover the steps involved in developing and implementing the AI model, as well as integrating it into our aquaponics system. Let's explore how the lettuce disease detector functions [73].

III.3.2. Neural Networks

III.3.2.1. Definition of a Neural Network

A neural network is a type of artificial intelligence model inspired by how the human brain works. Just like our brains use neurons to transmit signals and make decisions, neural networks use layers of artificial “neurons” to process data and learn patterns, Each neuron takes in one or more inputs, processes them, and sends an output to the next layer. These layers work together to find relationships in data, whether it’s recognizing a cat in a photo, understanding a voice command, or detecting plant disease from an image, In short, a neural network learns by example we feed it many examples during training, and over time, it becomes better at making predictions or decisions.

III.3.2.2. Main Types of Neural Networks

Neural networks come in many forms, each built for a specific kind of task, Here are the most common types mention in **“Figure III.11”**:

- **Feedforward Neural Networks (FNN):**
  This is the simplest type. Data flows in one direction from input to output like a one-way street. It’s good for basic classification tasks.
- **Convolutional Neural Networks (CNN):**
  These are experts at analyzing visual content images and videos, They use special layers called filters to detect features like edges, colors, and textures.
- **Recurrent Neural Networks (RNN):**
  These networks are designed for sequential data things that happen in order, like time-series data or spoken language. They can “remember” previous steps to improve predictions.
- **Radial Basis Function Networks (RBFN):**
  A bit more specialized, these networks use mathematical functions to classify data based on similarity to known examples.
- **Modular Neural Networks (MNN):**
  Instead of one big network, MNNs are made of several smaller ones, each handling a part of the problem, They’re great for solving complex or multi-task problems.

- **Generative Adversarial Networks (GANs):**
  These are creative networks! One part generates new data (like fake images), and another part tries to tell if it's real or fake, Together, they improve and can create surprisingly realistic content.

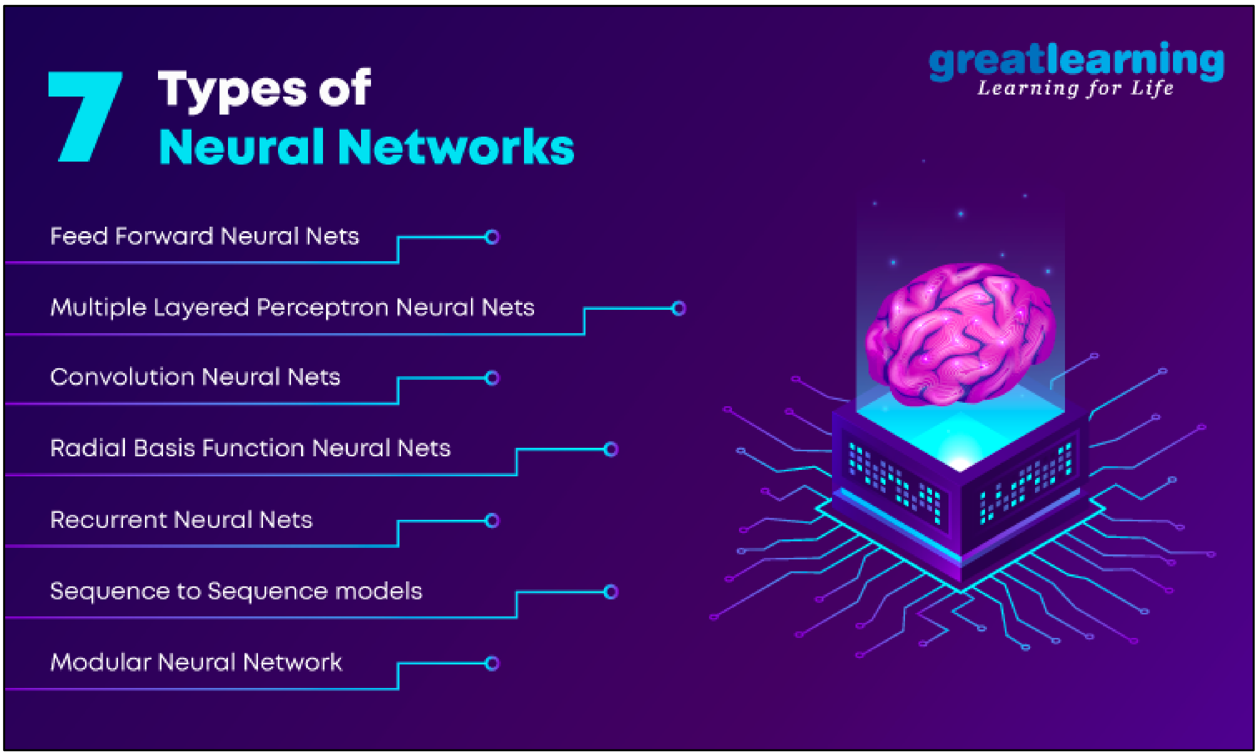

Figure.III. 11. Types of Neural Networks

Each of these architectures has its strengths. The key is choosing the right one based on the type of data and problem.

III.3.2.3. Convolutional Neural Networks (CNN)

For our project detecting lettuce diseases from images Convolutional Neural Networks (CNNs) were the best choice.

CNNs are made for visual tasks. They don't just look at the image as a whole, they break it down into small parts and analyze patterns like color changes, spots, edges, or textures. That's perfect for identifying symptoms of fungal or bacterial infections on lettuce leaves.

Another strength of CNNs is that they're very good at handling large image datasets while keeping performance high. Once trained, they can quickly and accurately classify new images, making them ideal for real-time or on-field use especially with Raspberry Pi and drone-captured images in a smart farming system.

So, in simple words: we chose CNN because it sees the way we do through patterns and details and that's exactly what disease detection in agriculture needs.

III.3.3. Programming language and interface

III.3.3.1. Programming language

To begin, it is essential to introduce the fundamentals of programming and the specific language used to develop our software. Our journey started with **"Python"** a simple yet powerful programming language that proved ideal for exploring the fields of artificial intelligence and automation. Python's clean and readable syntax makes it highly accessible to beginners. Its beginner-friendly nature is matched by its flexibility and strength, offering a vast ecosystem of libraries such as TensorFlow for AI, Flask and Django for web development, and Pandas for data analysis.

Python supports a wide range of applications, including building websites and web apps, developing AI and machine learning models, automating repetitive tasks like file organization or report generation, analyzing and visualizing complex datasets, and even creating games. Additionally, Python is cross-platform, running smoothly on Windows, macOS, Linux, and embedded systems like the Raspberry Pi. Whether we were handling image data, training models, or interfacing with hardware components, Python was the perfect fit for the technical demands of our project [74].

III.3.3.2. Programming interface

To work effectively with Python, we needed a user-friendly interface to manage our development environment, and that's where **"Anaconda Navigator"** came in. It was truly a lifesaver. Instead of struggling with complex command-line installations, Anaconda offered a clean, graphical interface that made it easy to manage Python versions, virtual environments, and packages. It also came bundled with powerful tools like Jupyter Notebook, Spyder, and access to the Conda environment manager [75], here's the anaconda homepage **"Figure III.12"** for more clear approach:

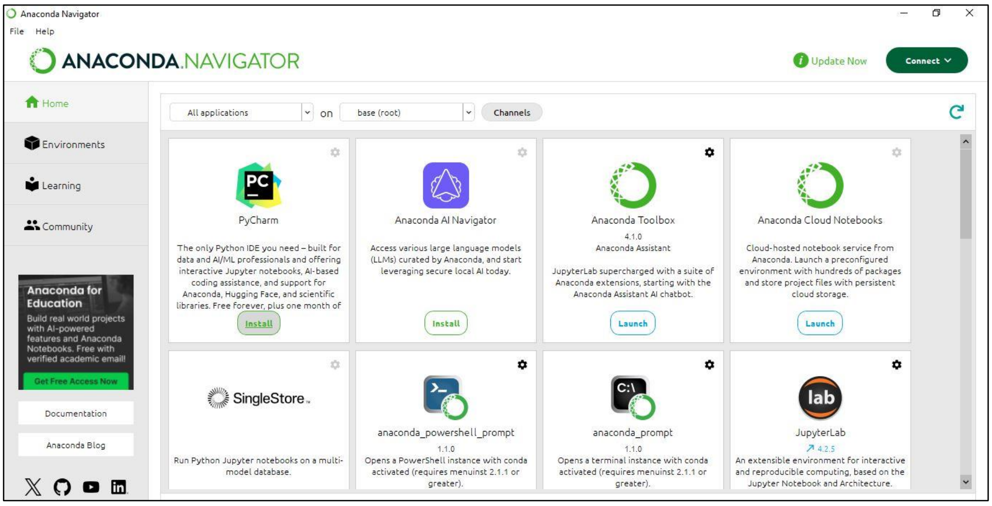

Figure.III. 12. Anaconda Navigator Homepage.

a/. Setting up working environment

Now since the image is clear now, we advance further for more clear vision, first we created an environment through the **"Environments"** Tab on anaconda **"Figure III.13"**

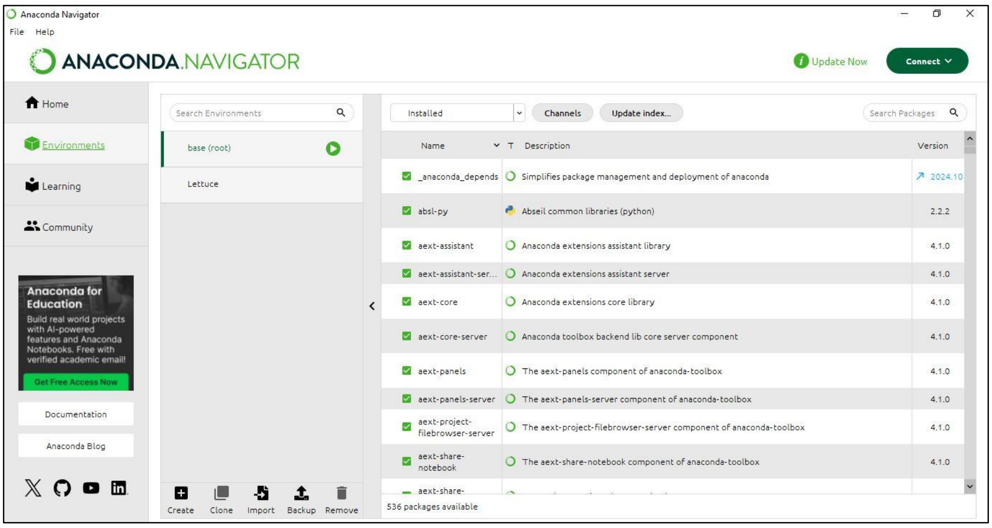

Figure.III. 13. Anaconda Navigator environments Section.

As you can see the **"Lettuce"** environment is already created, simply if you want to replicate the process, head to **"create"** tab below and choose your preferred paython version

b/. Installing needed libraries

Now, since the workshop is set, we need libraries to start out AI training, these libraries are the following:

- os → Helps manage file paths and directories (like accessing training images). [76]
- tensorflow → Main library for building, training, and saving the AI model using deep learning. [77]

- numpy → Handles arrays and numerical operations, especially for image data manipulation. [78]
- opencv-python → Used to capture, read, or process images (like resizing or converting color spaces). [79]
- Pillow → Another image library, used to open, convert, or display images (especially useful with TensorFlow). [80]
- matplotlib → Optional tool used to visualize training progress (like accuracy or loss curves).[81]

In the intention to install these libraires two approaches are used:

1/. Graphical Interface Method

→ Using Anaconda Navigator (usually called GUI-based installation)

2/. Command Line Method

→ Using the Anaconda Prompt (or Terminal)

Based on our experience, the terminal method is the most effective way possible (Debatable)

Here’s how to use the second method to install the libraires

a/. head to **“Anaconda prompt”** and activate the environment using >>**conda activate lettuce**

Then the status will change from (**base**) to (**your_environment_name**) like in **“FigureIII.14”**:

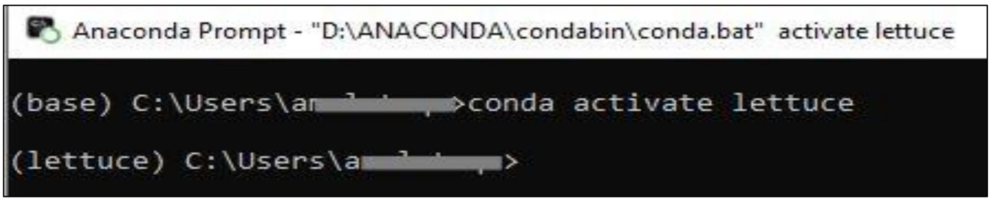

Figure.III. 14. Anaconda prompt overview.

b/. install the libraries mentioned earlier using the command >>**PIP install**

This will add all libraries to your environment (not to the base), **“Figure III.15”** shows the process

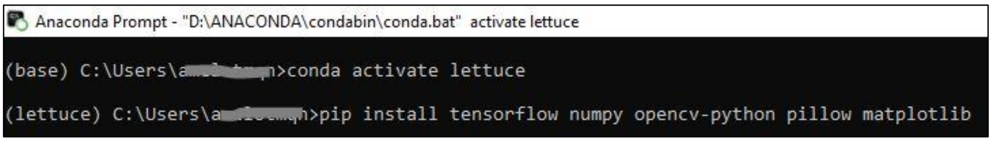

Figure.III. 15. Libraries installation through prompt.

c/. After execution, all libraries will install one at a time, you can check the installed libraries using the command >>**PIP list**

This will show all installed libraries as a form of a list including size and version of packages

III.3.4. Deep-Learning process

Now the environment is ready to run a deep-learning process, here's how that can be done:

III.3.4.1. Setting Up the Data

Our dataset was stored in two folders one for training and one for validation. These folders were organized with subfolders named after each class (*bacterial, fungal, healthy*); the role of each set is:

- ❖ Training set: This is the majority of the data, where the model learns from examples. [82]
- ❖ Validation set: Used to check the model's progress during training and ensure it's not overfitting (learning too much from the training data and performing poorly on new images) [82].

III.3.4.2. Loading and Organizing the Image Data [83]

Before feeding any images into our model, we had to make sure they were properly formatted. We used a tool provided by TensorFlow to load images during training. Instead of loading the whole dataset into memory, it reads the images in small batches. This is especially helpful when working with large datasets, we wrote a script to:

- ❖ **Rescale pixel values** from 0–255 (original range) to 0–1 (normalized range). This step is important because neural networks train better with smaller, normalized values.
- ❖ **Resize all images** to 224×224 pixels, which matched the input size expected by our model. If we skipped this step, images would have inconsistent shapes and the model wouldn't train correctly.
- ❖ **Batch the images** into groups of 32 at a time, batching helps reduce memory load and speeds up training by updating weights after every mini-group instead of every single image.
- ❖ **Auto-label the classes** using the folder names. Because we organized our dataset in separate folders for each category (bacterial, fungal, healthy), the algorithm automatically mapped labels to these classes.

This step not only gave the model clean, resized, and normalized images but also told it which image belonged to which category, this image preprocessing is what allowed our model to:

- Train faster and more efficient
- Avoid issues with incompatible image sizes
- Generalize better by working with standardized, clean input

III.3.4.3. Designing the Convolutional Neural Network (CNN)

With our images ready and organized, the next big step was to design the actual model that would learn how to detect lettuce diseases. We built a Convolutional Neural Network (CNN), a type of deep learning model that's really good at understanding images by spotting patterns like shapes, colors, and textures, The structure we created was made of layers stacked one after another, each doing something specific:

- ❖ **Conv2D Layers** → These are the pattern detectors. The first one scans the image using 32 filters (tiny grids), looking for edges and textures. Then we add two more Conv2D layers with 64 and 128 filters, which dive deeper into more complex patterns as the data passes through [83].
- ❖ **MaxPooling2D** → After each Conv2D layer, we used a MaxPooling layer. This reduces the size of the image data by keeping only the strongest features, which makes the model faster and less likely to overthink (or overfit). [83]
- ❖ **Flatten Layer** → Once all the patterns were detected, we flattened the data into a 1D layer, kind of like turning a grid into a straight line, so it can be passed into decision-making layers. [84]
- ❖ **Dense Layer** → This layer is fully connected and acts like a decision-maker. It has 128 neurons and uses activation to process the patterns and get closer to a prediction. [84]
- ❖ **Dropout Layer** → To help the model avoid memorizing the training images (overfitting), we added a dropout layer. It randomly turns off half of the neurons during training to assure better generalization. [84]
- ❖ **Output Layer** → Finally, we used a Dense layer with 3 neurons (because we have 3 categories: bacterial, fungal, and healthy), and applied a SoftMax activation function to turn the result into a clear percentage prediction for each class. [83] [84]

III.3.4.4. Compiling and Training the Model

once our model structure was designed, the next move was to prepare it for learning and then actually train it, First, we compiled the model, basically setting the rules for how it should learn. We used the **Adam optimizer** with a learning rate of 0.0001, which helps the model update itself smartly after each batch of images. For the loss function, we picked **categorical crossentropy** since we're dealing with three classes (healthy, bacterial, fungal), and we wanted the model to measure how far off its predictions were. We also added **accuracy** as a metric, to track how well it was doing as it learned, then came the training part. Using our clean and resized image data, we trained the model over **10** epochs (Due to hardware limitations we couldn't do more). During each epoch, the model looked at thousands of labeled images from the training folder, tried to guess the right class, then adjusted itself based on whether it was right or wrong. After each epoch, we let it check its progress on a smaller validation set to see if it was actually learning or just memorizing, this process helped the model slowly get better at spotting the visual patterns linked to each type of lettuce condition [84].

III.3.4.5. Saving the Trained Model for Future Use

Once the model completed training and showed reasonable accuracy, the next step was to save it so we can use it later for predictions on new images without retraining it every time.

We used the **model.save()** function to save the trained model as an **h5py** file. This file now contains everything the model learned, including the weights, optimizer settings, and model architecture. [84]

The training process should show you something like this:

Epoch ?/10

accuracy: 0.3470 - loss: 1.0928 - val_accuracy: 0.5609 - val_loss: 1.0898

Note: training speed depends on your GPU and CPU performances and also the amount of data used for process

In Our case, It Took approximately 5 hours to finish training data and creating the AI (CNN) model

Here’s the full training script explained briefly through a simple diagram **“Figure III.16”**:

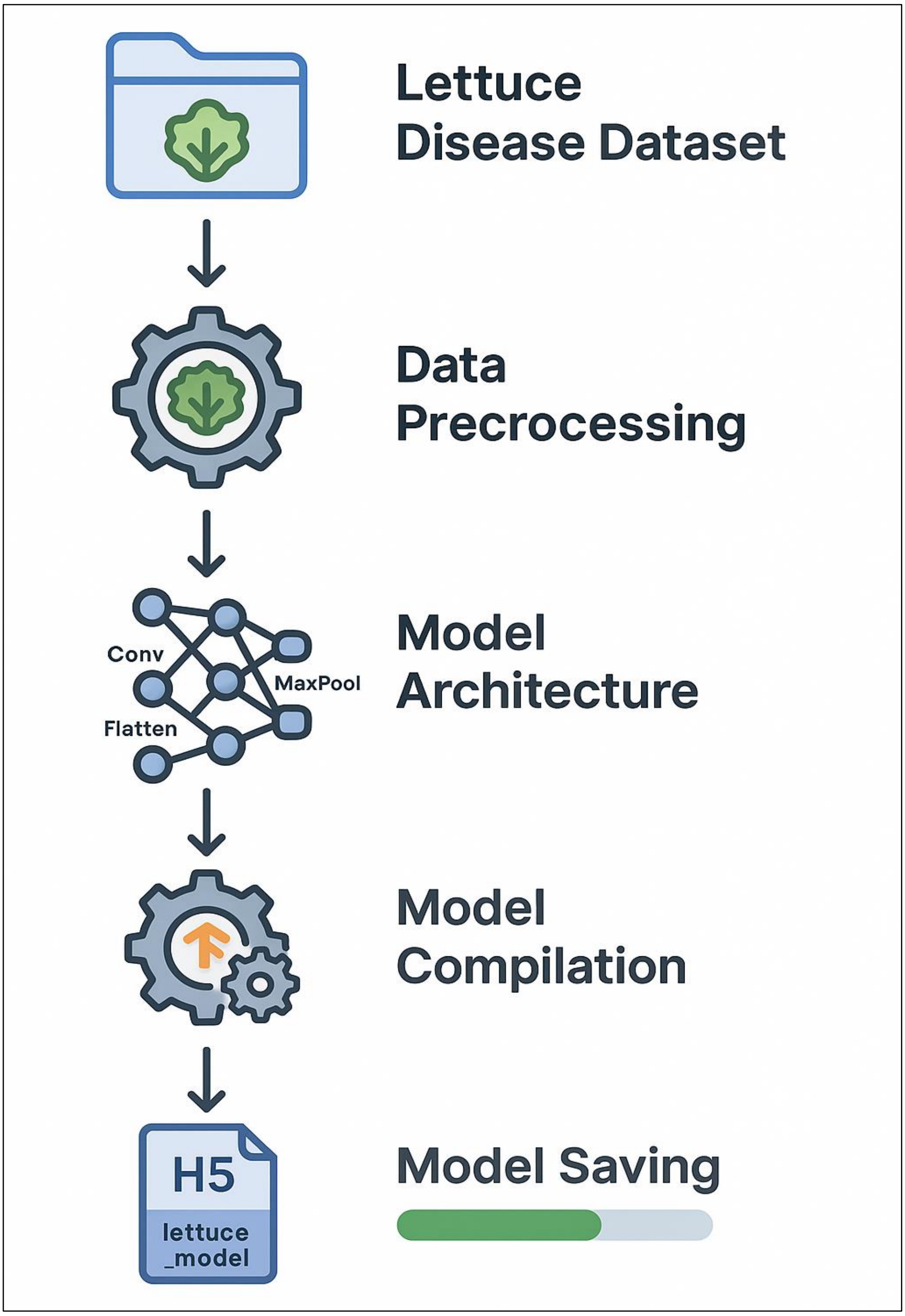

Figure.III. 16. Training script demonstration.

To access and check the full code, check the Drive link for the provided CNN training code [85]

III.3.4.6. Running the Smart Disease Detector

After training the model and saving it, we had two ways to run the lettuce disease detection system:

III.3.4.6.1. Running via Python on a Laptop (with Anaconda and all libraries)

The first method is the most direct one **"Figure III.17"**. We simply run the Python script (smart_predict.py) on our laptop using the Anaconda environment we had already set up (**"Lettuce"**). Since all required libraries (like TensorFlow, OpenCV, Pillow, etc.) were already installed in that environment, everything worked smoothly. This method is great for development and debugging.

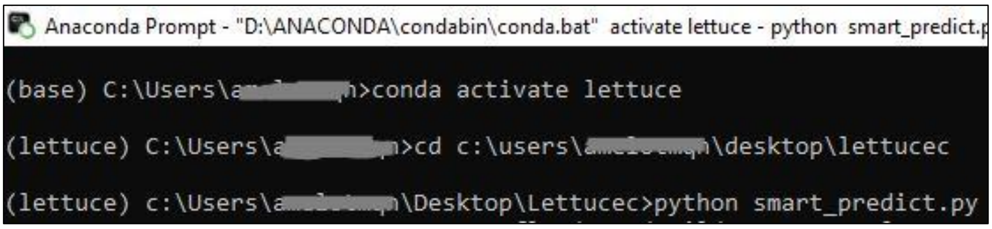

Figure.III. 17. Smart prediction software execution.

After executing the smart_predict.py file using prompt, this how the software looks like **"Figure III.18"**:

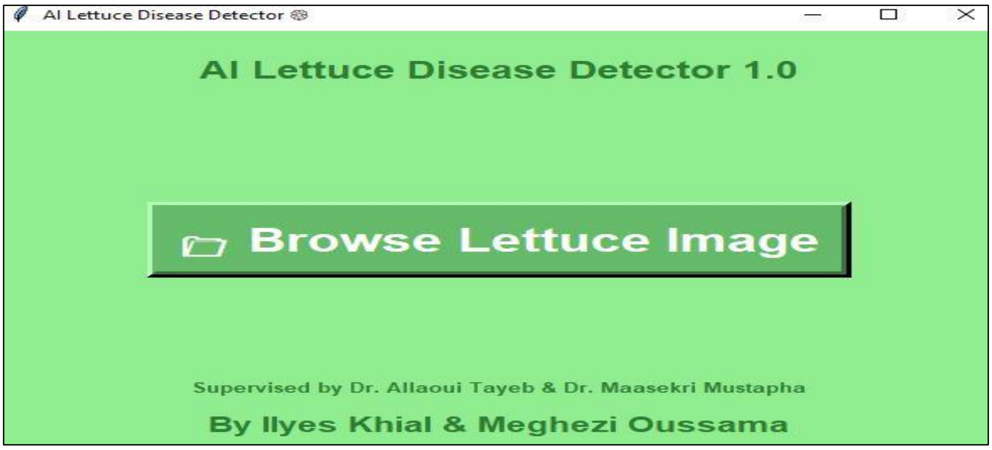

Figure.III. 18. Disease Detector Interface.

To use the disease detector, simply upload a lettuce picture and it will predict its class using the deep-learning model that was created earlier, AI based detector includes:

- Prediction confidence: with percentage %
- Class of prediction: bacterial; fungal; healthy
- Tips for care for each category

here's a demonstration on how it works **"Figures III.19/20/21"**:

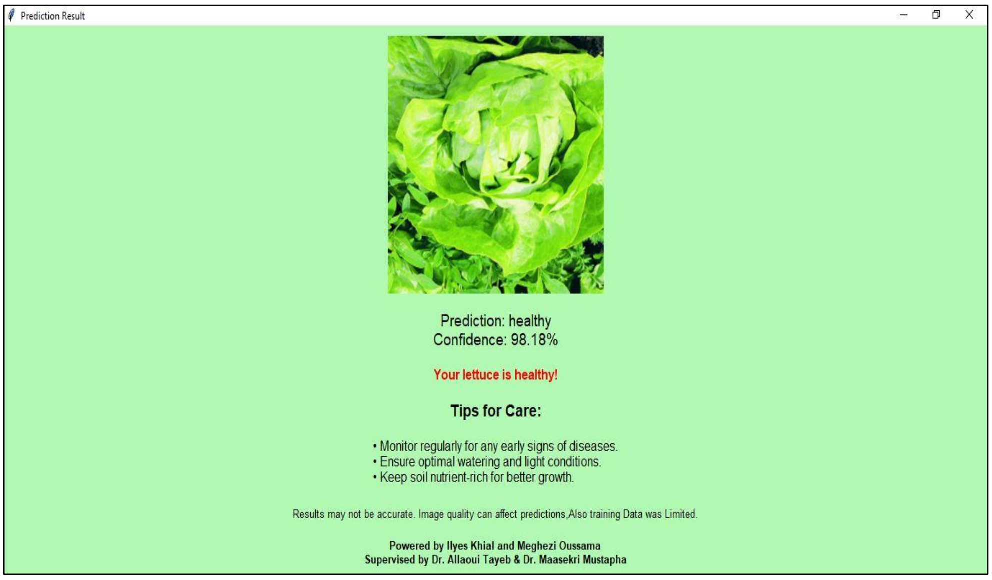

Figure.III. 19. Example of the AI disease detector 1

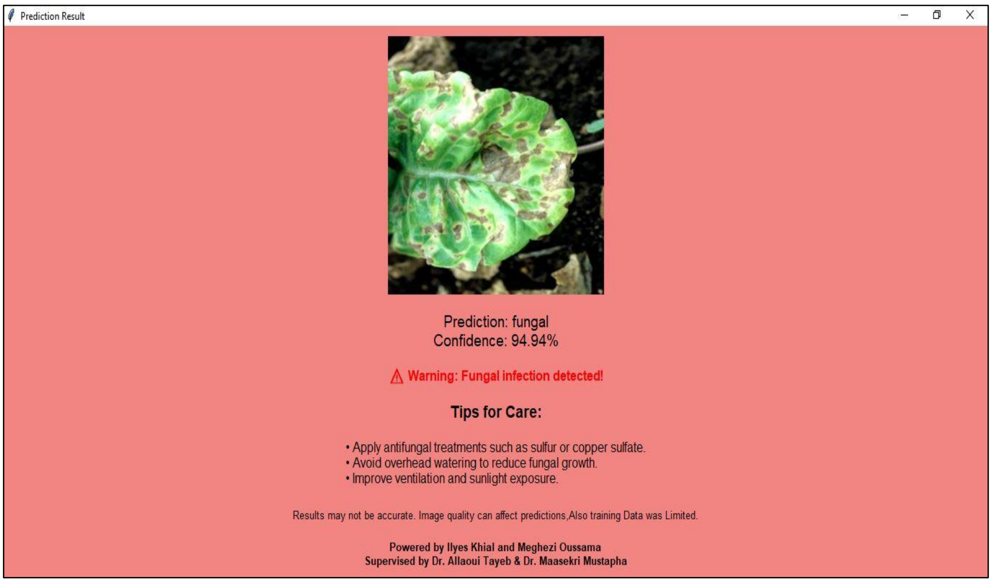

Figure.III. 200. Example of the AI disease detector 2.

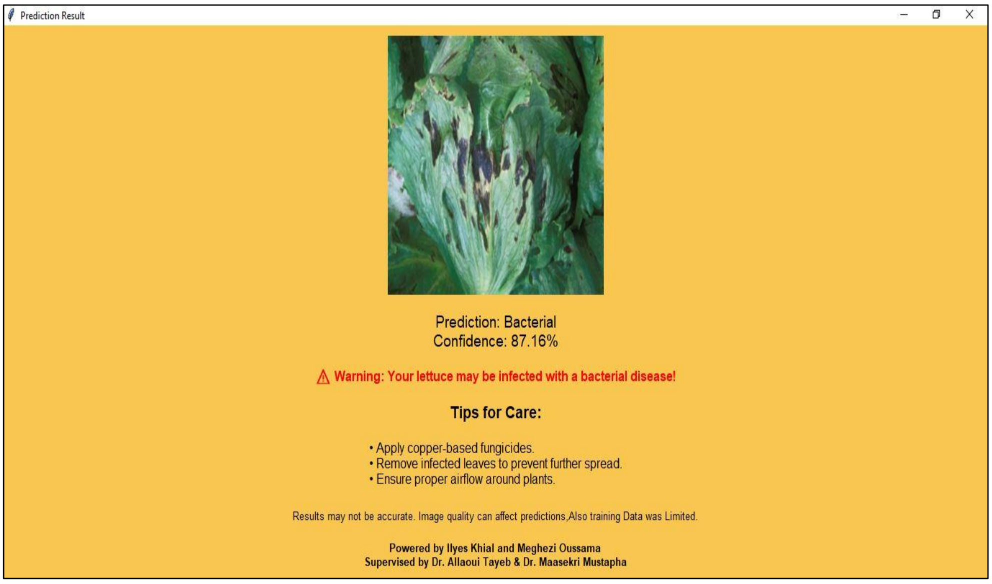

Figure.III. 21. Example of the AI disease detector 3

Note: the prediction may go wrong sometimes due to used images quality and due to the small dataset used for the training (approximately 5000 pictures)

To access and check the full code, check the Drive link for the provided AI Disease Detector Software [86]

III.3.4.6.2. Using a Standalone EXE File

For users who don't have Python or Anaconda installed, we also converted the project into a standalone **".exe"** file. This version bundles everything into one file, allowing the program to run independently on any Windows machine, here's how to create the standalone software, using "**PyInstaller**", which turns Python scripts into **".exe"** files [87].

First, we need to install **"Pyinstaller"** using the command:

»**Pip install pyinstaller**

When the package is installed, and before proceeding into the process, we need to organize files.

a/. file organization

Create a folder, inside it should contain these:

- smart_prediction.py ← Our main Python script
- lettuce_model.h5 ← The trained AI model
- lettuce.ico ← Icon for the **".exe"** file (Optional but recommended)

b/. Standalone software creation

Since the folder is ready to explore, we must first navigate into it using the command:

»**Cd C:\users\user_name_here\desktop\Folder**

And then use this specific command:

»**Pyinstaller --onefile --noconsole --icon=leaf.ico --add-data "lettuce_model.h5;." smart_prediction.py** [87]

This will bundle all files in the folder and create a standalone **".exe"** Version with a lettuce icon that runs without the need of anaconda navigator or python libraries on any other laptop (windows powered machine)

Once **"PyInstaller"** finishes bundling everything, it creates a **"dist"** folder as shown in **"Figure III.22"** in the same directory where your script is located. Inside that folder, you'll find your brand-new standalone application a « .exe » file, here is an overview for the folder:

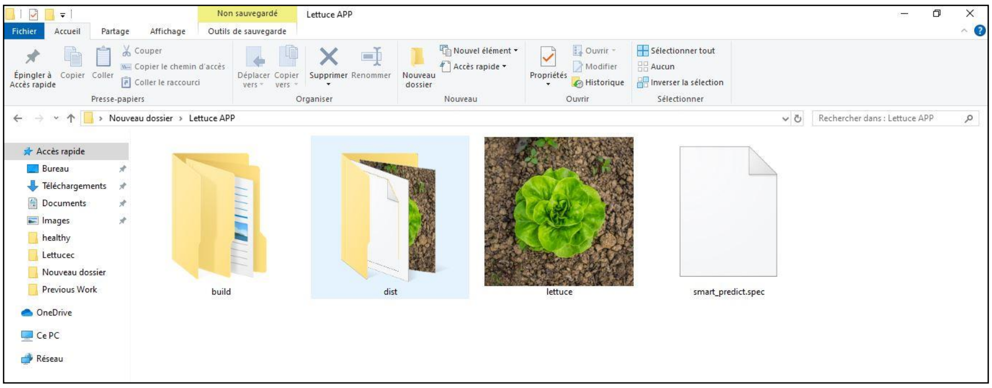

Figure.III. 22. Standalone file overview

Then we access the "dist" folder to run our standalone file **"Figure III.23"**:

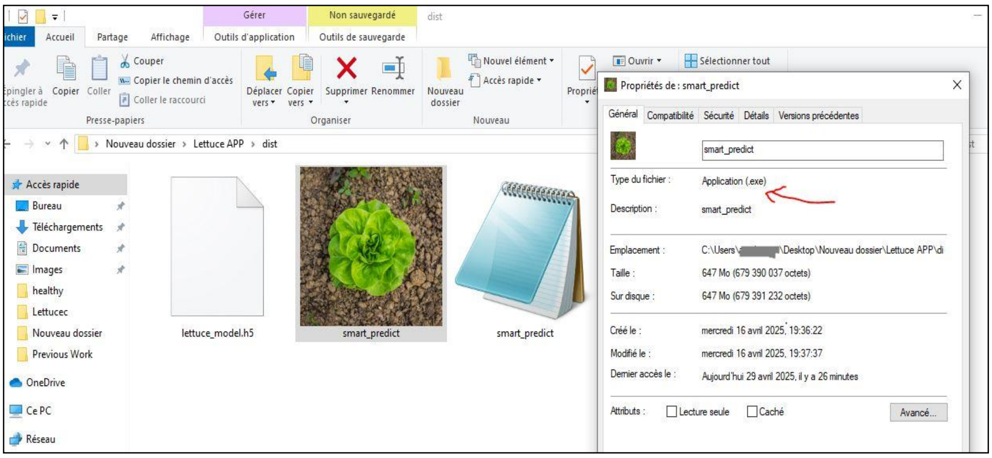

Figure.III. 23. Portable software version.

To summarize, the AI-based disease detector is a revolutionary way to predict and detect disease early in order to avoid crop loss due to spread infection between plants and also rise the profit by replacing sick plants to catch up with the rest of the healthy plants, and also to make farmers and aquaponics operators comfortable that an AI helper is assisting them to do their jobs and prevent any confusion or incorrect diagnosis

III.4. Drone Software Overview

III.4.1. Introduction

To ensure smooth and fully autonomous operation of our custom drone, we employ a combination of integrated software systems that manage flight control, hardware calibration, data capture, and communication with ground stations. In modern drone design, multiple software layers interact directly with critical components such as ESCs, motors, GPS modules, IMU sensors, and onboard cameras. For this project, the core platforms Betaflight, ArduPilot, and Mission Planner each serve a specific role in configuring flight controllers, calibrating sensors, and executing GPS-based autonomous missions. Together, they provide a robust software stack for reliable and intelligent drone operations.

III.4.2. Betaflight

III.4.2.1. Definition

Betaflight is an open-source flight controller firmware widely used in the FPV (First Person View) drone community. It offers real-time responsiveness, high configurability, and strong support for sensor calibration and motor tuning. Originally derived from Cleanflight, Betaflight is now a mature and high-performance system for optimizing drone behavior [87].

III.4.2.2. Components Controlled by Betaflight [88]

Betaflight directly interfaces with and regulates several critical flight components, including:

- ❖ **Electronic Speed Controllers (ESCs)**: It manages motor speed and direction for each propeller.
- ❖ **Gyroscope and Accelerometer**: For maintaining drone stability and detecting movement in 3D space.
- ❖ **Barometer**: For estimating altitude changes.
- ❖ **Receiver (RX)**: For channel input testing and fail-safe setup.
- ❖ **Motor Outputs**: Allows individual motor configuration and testing.

III.4.2.3. Role of the Software [88]

In this project, Betaflight is primarily used during the initial setup phase to:

- ❖ Calibrate and test ESCs and brushless motors.
- ❖ Align internal sensors (IMU, gyro, accelerometer).
- ❖ Validate channel mapping and receiver signals.
- ❖ Configure flight controller board orientation and port usage.
- ❖ Run live motor tests to ensure responsiveness before installing ArduPilot.

This stage ensures that all onboard components are properly connected and functional before flashing the drone with autonomous firmware, **"Figure III.24"** Represents the software's interface

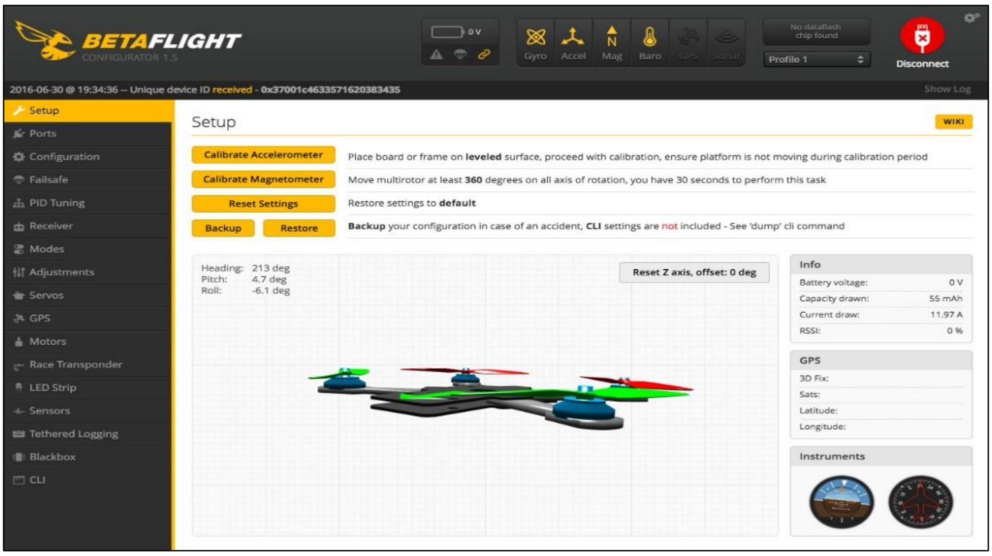

Figure.III. 24. Betaflight Configurator overview.[88]

Betaflight serves as the foundation for hardware-level calibration and flight controller verification. Once the system is confirmed to be operating correctly, we transition to ArduPilot and Mission Planner for full autonomous functionality.

III.4.3. ArduPilot and Mission Planner

III.4.3.1. Definition

"ArduPilot" is a versatile, open-source autopilot software platform that enables fully autonomous control of aerial, land, and marine vehicles. It supports a wide range of flight modes, sensor integrations, and safety features for both hobbyist and professional applications[90].

"Mission Planner" is a companion Ground Control Station (GCS) software for Windows, used to configure, monitor, and control ArduPilot-powered drones. It provides a graphical interface for mission planning, sensor calibration, parameter tuning, and real-time telemetry monitoring. **ArduPilot** is an open-source autopilot software suite used to control a variety of vehicles, including drones, airplanes, and rovers. It supports advanced features such as autonomous flight, waypoint navigation, automatic takeoff and landing, and real-time telemetry. ArduPilot is highly customizable, allowing us to adapt it for specific needs like GPS-based missions and sensor integrations [91].

III.4.3.2. Components Controlled by ArduPilot and Mission Planner [90][91]

After Betaflight setup, ArduPilot takes over as the main flight firmware, while Mission Planner functions as the external configuration and monitoring tool. Together, they manage:

- ➢ **GPS Module**: Used for autonomous navigation and location-based flight modes.

- ➤ **Compass**: For heading and direction control.
- ➤ **Barometer and IMU**: To maintain altitude and stabilize flight.
- ➤ **Failsafe Mechanisms**: Such as return-to-launch (RTL), auto-land, or hover on signal loss.
- ➤ **Telemetry Modules**: For live data transmission between drone and GCS.
- ➤ **Mission Waypoints**: Through drag-and-drop interface in Mission Planner.

III.4.3.3. Role of the Software

In our autonomous drone system:

III.4.3.3.1. ArduPilot [90]

Is flashed onto the flight controller after Betaflight calibration. It transforms the drone into a smart system capable of executing complex tasks without manual intervention, **"Figure III.25"** illustrates the Flash firmware file downloading process

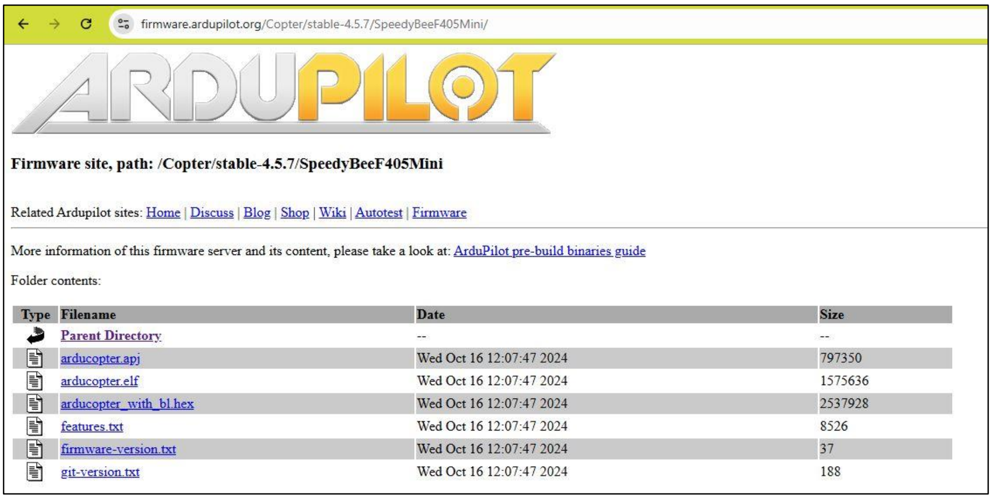

Figure.III. 25. Speedybee firmware through ArduPilot.

III.4.3.3.2. Mission Planner [91]

It is used to:

- ❖ Flash ArduPilot firmware.
- ❖ Calibrate sensors (compass, accelerometer, barometer).
- ❖ Upload waypoint missions and flight paths.
- ❖ Tune navigation and flight stability parameters.
- ❖ Monitor live telemetry (battery, GPS, altitude, signal strength).
- ❖ Set up automatic actions in response to specific events (failsafe triggers, flight mode changes).

Through Mission Planner's intuitive interface, the drone can be pre-programmed to fly over designated agricultural zones, capture aerial images, and return safely all without manual control, **"Figure III.26"** Demonstrates the Entire Function

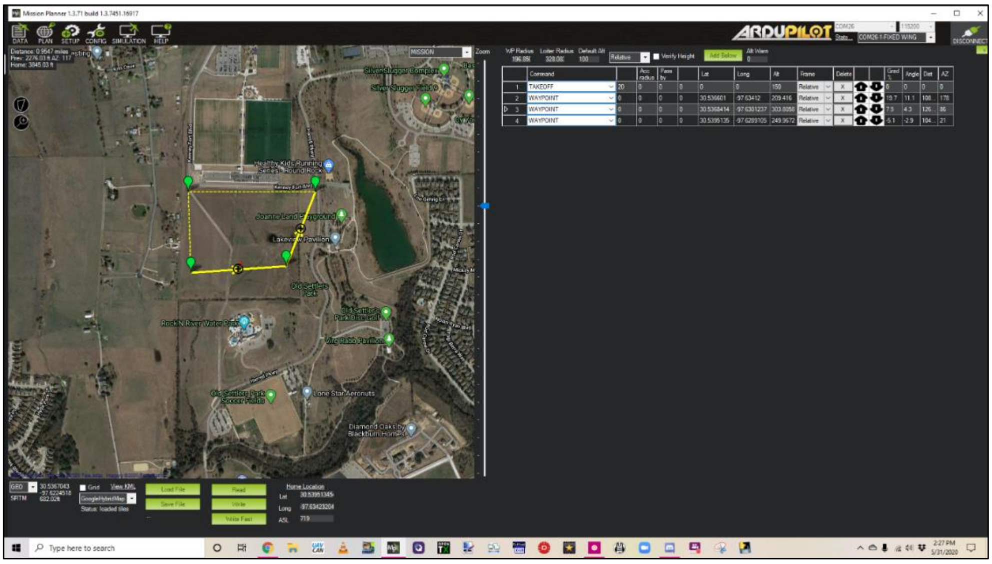

Figure.III. 26. Mission Planner Overview.

To summarize, With Betaflight handling the initial calibration and ArduPilot enabling advanced autonomous control, Mission Planner becomes our main interface for configuring, simulating, and deploying real-world missions. It offers full access to flight parameters, mission scripting, and fail-safe behaviors allowing us to test and optimize the drone's performance before actual deployment. Together, these tools form a complete software pipeline from calibration to autonomous execution, The process should be like this:

Betaflight
↓
[Calibrates ESCs, tests Motors, aligns Sensors]
↓
ArduPilot
↓
[Enables autonomous flight, GPS logic, failsafe]
↓
Mission Planner
↓
[Plans missions, configures parameters, runs simulations]

III.4.4. The Speedybee App

The Speedybee App is a lightweight, mobile-based alternative to traditional configuration tools. It connects wirelessly via Bluetooth or USB to supported flight controllers, making it extremely useful for quick sensor tuning, motor tests, and firmware management all without needing a PC [91], it can be used to:

- ❖ Calibrate and align sensors (accelerometer, gyro)
- ❖ Test Motors and ESCs
- ❖ Tweak PID and rate settings

The app provides full access to essential tuning tools, making it ideal for quick adjustments on-site.

Additionally, the Speedybee App **"Figure III.27"** offers a virtual joystick feature, which can be used to simulate basic RC input. This function can serve as a temporary substitute for a physical remote controller, allowing basic control of the drone for setup and testing purposes especially useful for ground tests or minimal movement scenarios [92].

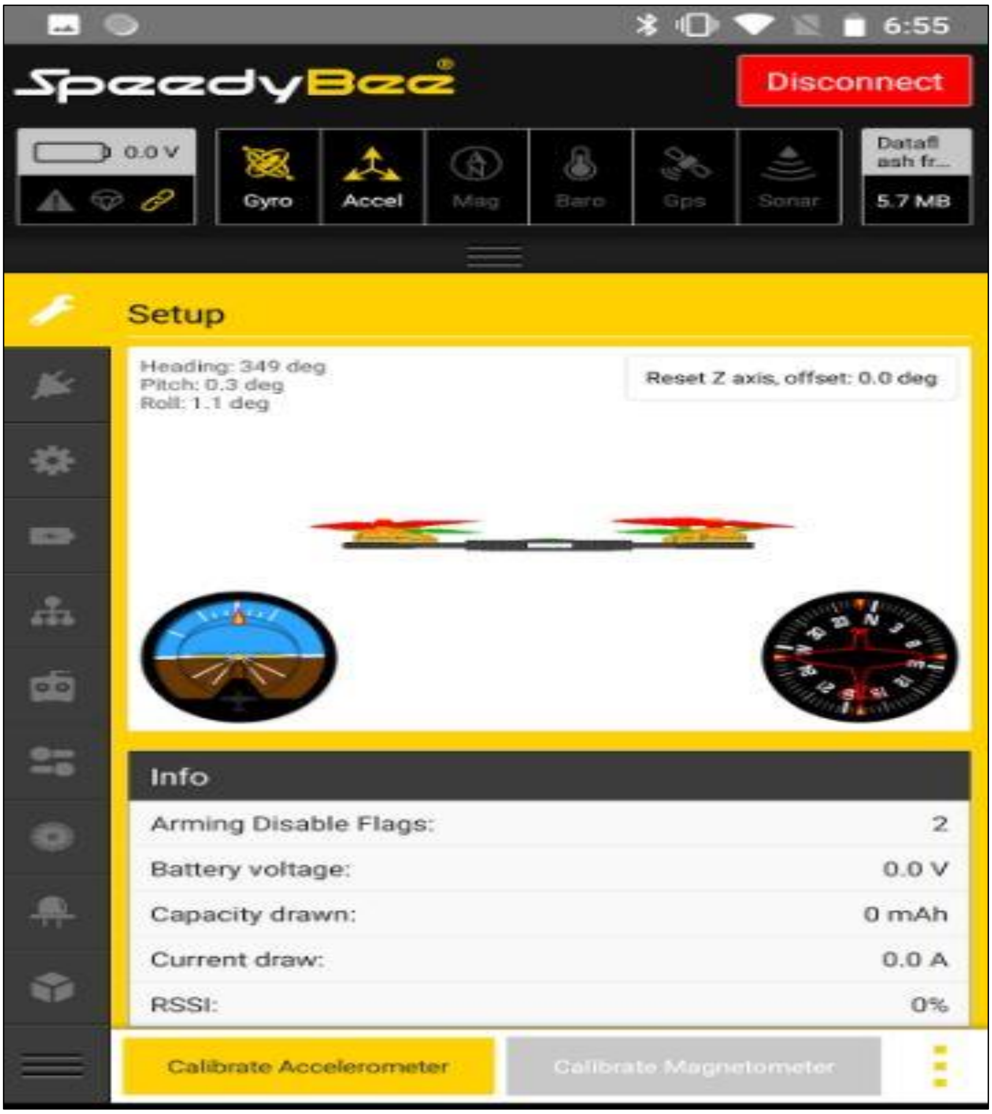

Figure.III. 27. Speedybee APP Overview.[92]

II.5. Conclusion

The development of our smart aquaponics system is a big step toward making farming more efficient and sustainable by combining technology in a smart way. At the heart of the system is the Raspberry Pi 5, which takes care of everything from reading sensor data to running the disease detection model and managing the devices that keep the system running, like pumps and fans. The AI disease detector, powered by TensorFlow, analyzes the health of lettuce plants in real-time, helping us quickly spot any issues like bacterial or fungal infections. Alongside this, the custom drone software captures aerial images of the crops, giving the AI model a closer look to make its predictions. This all works together to reduce the need for constant human involvement, offering farmers useful information that helps protect crops from disease and ensures they're thriving. It's a game-changer for anyone looking to streamline farming, and there's plenty of room to grow from improving the AI to integrating renewable energy for an even more sustainable future.

Chapter IV:

Testing and Experimental Results

IV.1. Introduction

In this chapter, we present the testing and evaluation of the various components of our smart aquaponics system. First, we assess the performance and accuracy of the sensors and actuators integrated into the aquaponics setup. Next, we evaluate the functionality and accuracy of the AI-powered disease detection system designed to identify lettuce diseases. Finally, since the drone has not been physically built, we demonstrate its design and operation through a 3D model, supported by performance data to illustrate its expected capabilities.

IV.2. AI Lettuce Disease Detector – Design and Evaluation

IV.2.1. AI Detector Design

The AI lettuce disease detector serves as a key intelligence layer within the smart aquaponics system. It is powered by a Convolutional Neural Network (CNN) model, developed using Python, TensorFlow, and Keras. The model was trained on a custom dataset of over 4500 labeled images categorized into three classes: *fungal*, *bacterial*, and *healthy* lettuce.

However, the training process faced several limitations in both data diversity and image quality. The dataset was primarily composed of *fungal* and *healthy* images, with relatively fewer high-quality examples of bacterial infections. This imbalance affected the model's ability to generalize well across all categories especially in accurately identifying bacterial diseases. Furthermore, some images lacked consistent lighting and clarity, reducing the overall training effectiveness.

Currently, the AI model is Used and developed on a laptop for testing purposes. Images are captured using a drone or mobile device and then manually transferred via USB or SD card to the laptop. Once received, the model processes each image and provides a disease prediction along with basic treatment recommendations. This semi-manual system enables ongoing testing and development while full automation is still under planning.

IV.2.2. Performance Evaluation

a/. Model Accuracy and Limitations

Despite the challenges, the model showed promising results under test conditions:

- **Training accuracy :** 96-98%
- **Validation accuracy :** 89-94%
- **Average upload time per image:** 1-2 seconds (on laptop CPU)

However, the uneven distribution of training data had a noticeable impact. The model demonstrated high confidence and accuracy when predicting *fungal* and *healthy* lettuce but showed lower reliability in identifying *bacterial* diseases due to the limited number of examples and variability in image quality.

b/. Image Testing Workflow

The current image testing procedure follows these steps:

1. **Capture**: Lettuce images are taken via drone or mobile camera.
2. **Transfer**: Images are moved to the laptop using USB or SD card.
3. **Detection**: The AI script (smart_prediction.py) analyzes the image and classifies it.
4. **Result**: The prediction is displayed, along with basic crop care tips.

This setup, while not yet real-time, provides a clear performance evaluation under realistic field conditions.

The Drive link Provides a Set of examples for the three classes (10 samples each),it also contains the CNN model and python script that were used to build the detector [93]

For those who can't access the detector through anaconda navigator, here's a standalone software that can run on any windows machine without a python interface, check this drive link and look for "Dist folder" [94]

IV.2.3. Future Plans for Lettuce AI Detector

In the next development stage, the goal is to enhance and deploy the AI lettuce disease detector as a fully autonomous tool within the smart aquaponics system. While the current setup relies on manual image transfers and laptop-based processing, future improvements aim to streamline both input and analysis workflows.

Planned upgrades include :

- Integrating the model into the Raspberry Pi 5 for on-site, real-time image detection
- Expanding and balancing the dataset, particularly by collecting more bacterial infection images
- Improving image quality standards to boost detection accuracy under varying lighting and crop conditions
- Automating image input via camera modules or wireless drone/web transfers instead of USB or SD cards
- Linking detection results to dashboard logs or automated alerts for responsive crop management

Once implemented, these improvements will elevate the lettuce AI detector from a semi-manual testing tool to a reliable, real-time field diagnostics system—supporting healthier crop management in a smart, scalable way.

IV.3. Sensors and Actuators Tests

For this section, we visualized and analyzed environmental data from the DHT11 (temperature and humidity sensor) and a light sensor using the Pi5 framework. The processed data also triggers LEDs (Play the Role of actuators due to hardware limitations) based on threshold logic.

The script Used in the Testing process is illustrated in **“Figure IV.1”**:

```
# -*- coding: utf-8 -*-
import time
import board
import adafruit_dht
import lgpio

# === GPIO Pin Definitions ===
LED_HIGH = 17    # GPIO17 (Pin 11) SPA Heater LED
LED_LOW = 27     # GPIO27 (Pin 13) SPA Cooler LED
LED_PUMP = 25    # GPIO25 (Pin 22) SPA Air Pump LED (changed from 23)
LDR_DO_PIN = 22 # GPIO22 (Pin 15) SPA LDR Digital Output
LED_LDR = 24     # GPIO24 (Pin 18) SPA LDR Status LED

# Open GPIO chip
h = lgpio.gpiochip_open(0)

# === Setup Pins ===
lgpio.gpio_claim_output(h, LED_HIGH, 0)
lgpio.gpio_claim_output(h, LED_LOW, 0)
lgpio.gpio_claim_output(h, LED_PUMP, 0)
lgpio.gpio_claim_output(h, LED_LDR, 0)
lgpio.gpio_claim_input(h, LDR_DO_PIN)

# Initialize DHT11
sensor = adafruit_dht.DHT11(board.D23)

try:
    while True:
        try:
            temperature = sensor.temperature
            humidity = sensor.humidity

      if temperature is not None and humidity is not None:
        print(f"??? Temp: {temperature} | ?? Humidity: {humidity}%")

         # Temperature logic
         if temperature >= 32:
             lgpio.gpio_write(h, LED_HIGH, 1)
             lgpio.gpio_write(h, LED_LOW, 0)
          else:
             lgpio.gpio_write(h, LED_HIGH, 0)
             lgpio.gpio_write(h, LED_LOW, 1)

          # Humidity logic
         if humidity >= 60:
             lgpio.gpio_write(h, LED_PUMP, 1)
          else:
             lgpio.gpio_write(h, LED_PUMP, 0)

          else:
        print("?? Failed to read sensor data.")
             lgpio.gpio_write(h, LED_HIGH, 0)
             lgpio.gpio_write(h, LED_LOW, 0)
             lgpio.gpio_write(h, LED_PUMP, 0)

          # LDR logic
         if lgpio.gpio_read(h, LDR_DO_PIN) == 0:
        print("?? Light ON")
             lgpio.gpio_write(h, LED_LDR, 0)  # LED off when bright
          else:
        print("?? Light OFF / Darkness")
             lgpio.gpio_write(h, LED_LDR, 1)  # LED on when dark

            time.sleep(2)

        except RuntimeError as e:
        print("Sensor error:", e.args[0])
            time.sleep(2)

except KeyboardInterrupt:
  nnnn  print("?? Stopped by user.")

finally:
    # Turn off all LEDs and clean up
    for pin in [LED_HIGH, LED_LOW, LED_PUMP, LED_LDR]:
        lgpio.gpio_write(h, pin, 0)

    lgpio.gpiochip_close(h)
    sensor.exit()
```

Figure.IV.1. Sensors and Actuators Tests script.

IV.3.1. Hardware used

- Raspberry Pi5
- DHT11 as a temperature and humidity sensor
- Photoresistor (LDR) as a light intensity sensor
- Four LEDs as output actuators **“Figure IV.2”**
  - Red : mimics the heater
  - Bleu : mimics the cooler
  - Green or yellow : mimics lights
  - Bleu : mimics the air pump

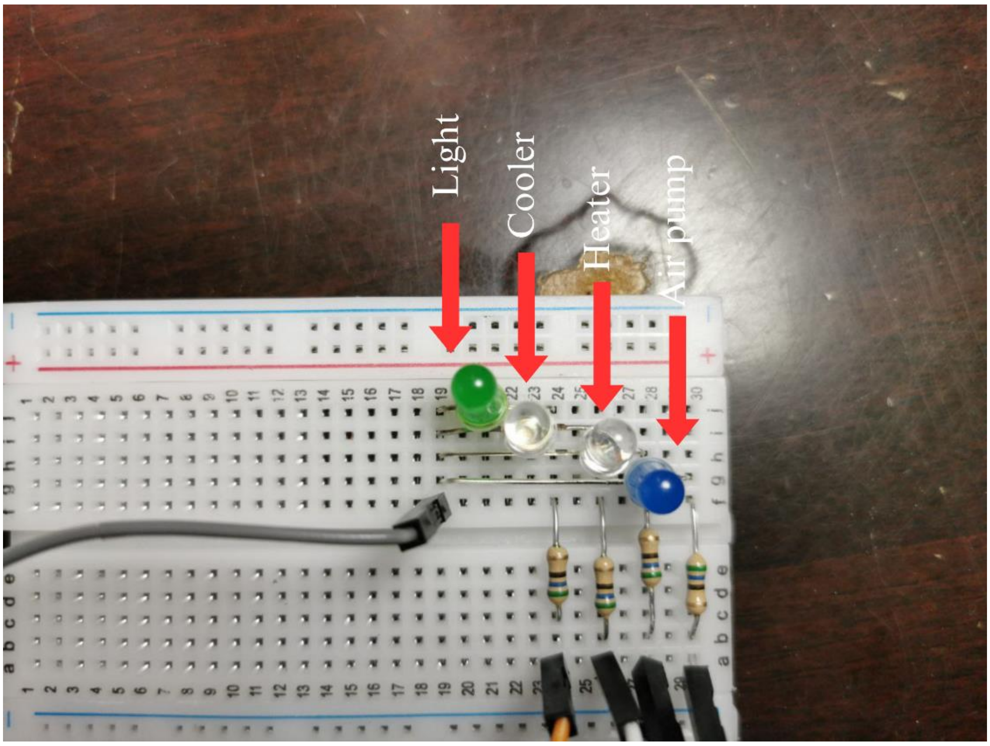

Figure.IV.2. LEDs used to mimic actuators.

IV.3.1.1. Temperature and humidity Tests:

We used the DHT11 sensor, this sensor provides digital data output, which was read and processed through the microcontroller, and then visualized in real-time using the Pi5 environment. This setup allowed for clear and dynamic monitoring of environmental changes, **“Figure IV.3/4/5/6”** demonstrate the entire process.

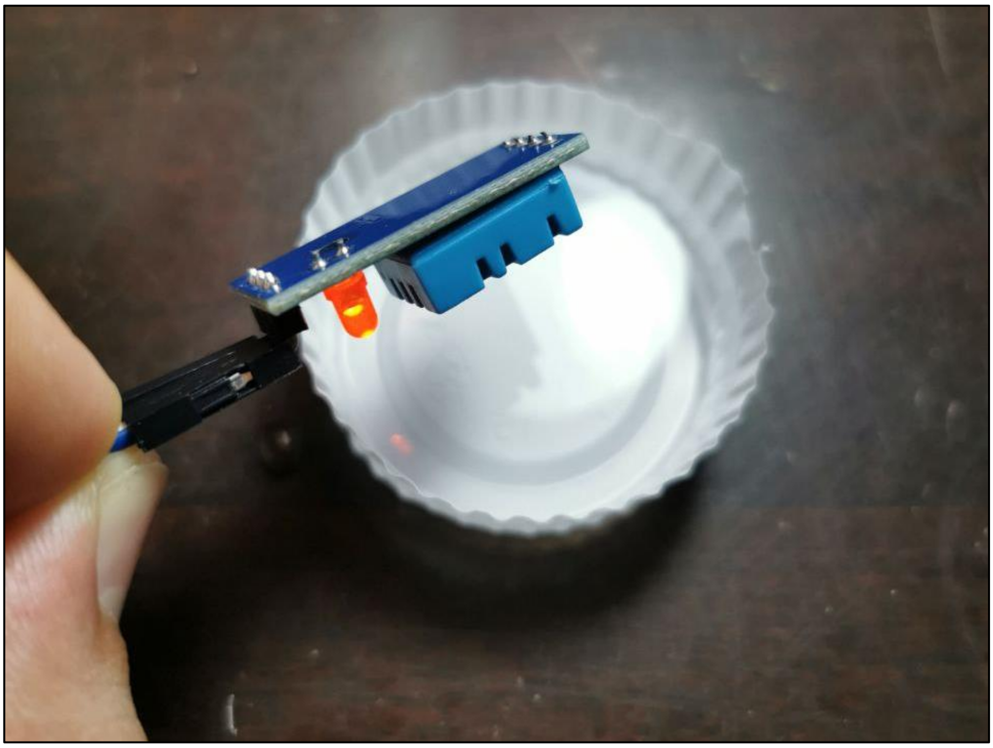

Figure.IV.3. Measurement of Temperature and humidity near a cold surface.

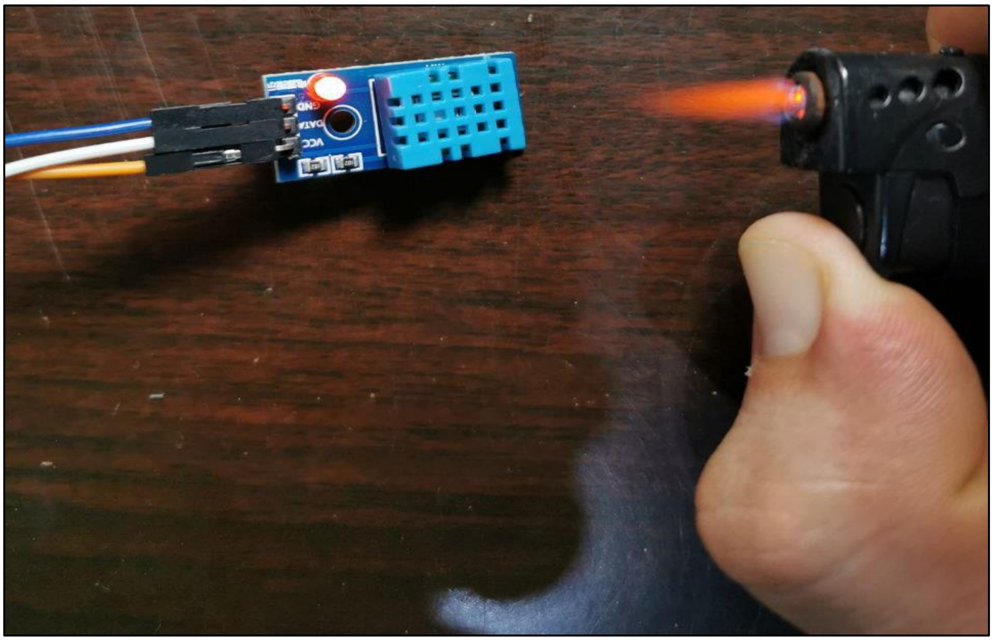

Figure.IV.4. Measurement of Temperature and humidity near a heat source.

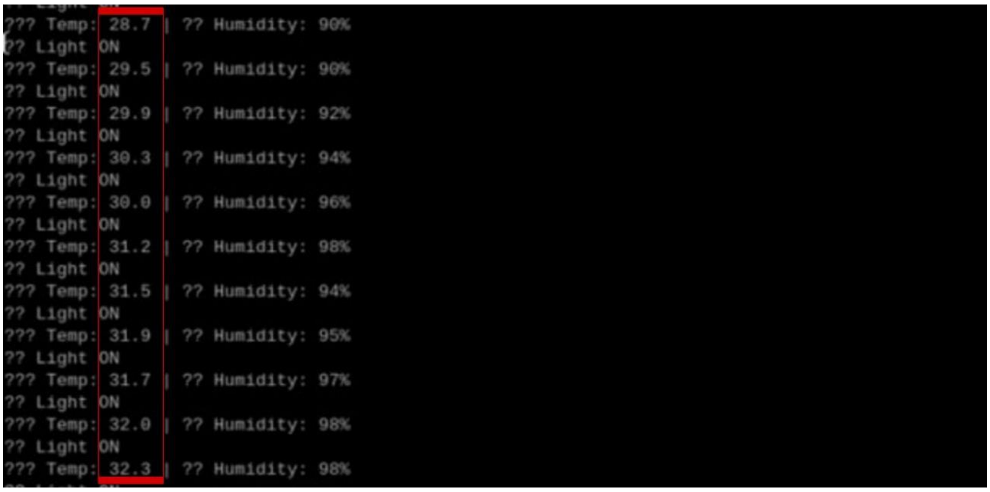

Figure.IV.5. Temperature readings from DHT11 through PI5 prompt.

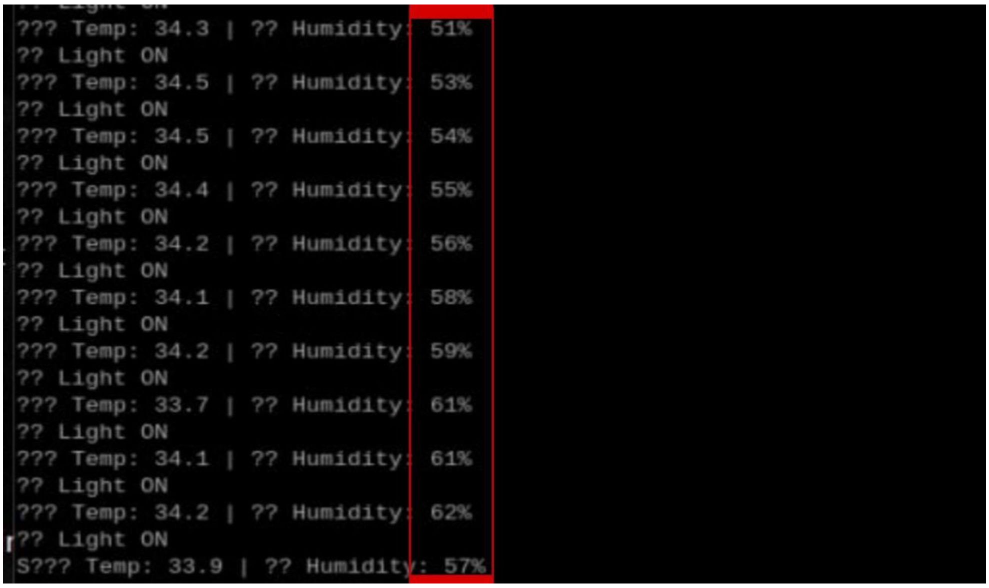

Figure.IV.6. Humidity Readings from DHT11 through PI5 prompt.

IV.3.1.2. Light Intensity Tests

To measure ambient light levels, we used a photoresistor (LDR – Light Dependent Resistor) as the primary light sensor. The LDR's resistance changes based on the amount of light it receives decreasing in bright environments and increasing in darkness. This analog signal was converted to digital using an Analog-to-Digital Converter (ADC) and processed by the microcontroller **"Figure IV.7/8"** show the entire operation.

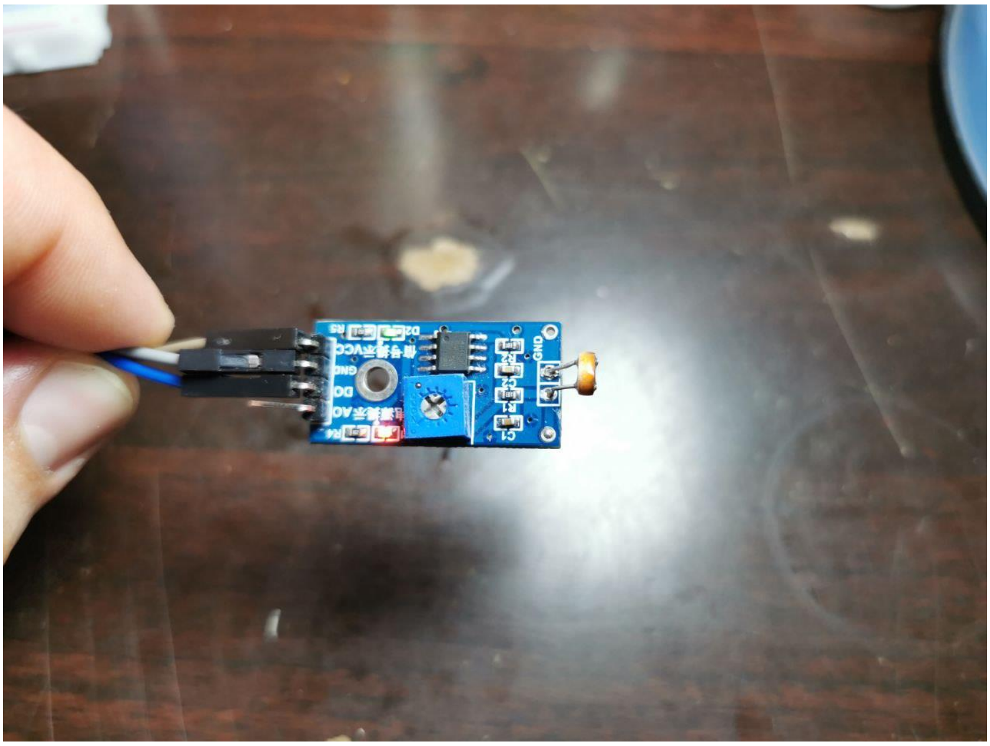

Figure.IV.7. Light presence test.

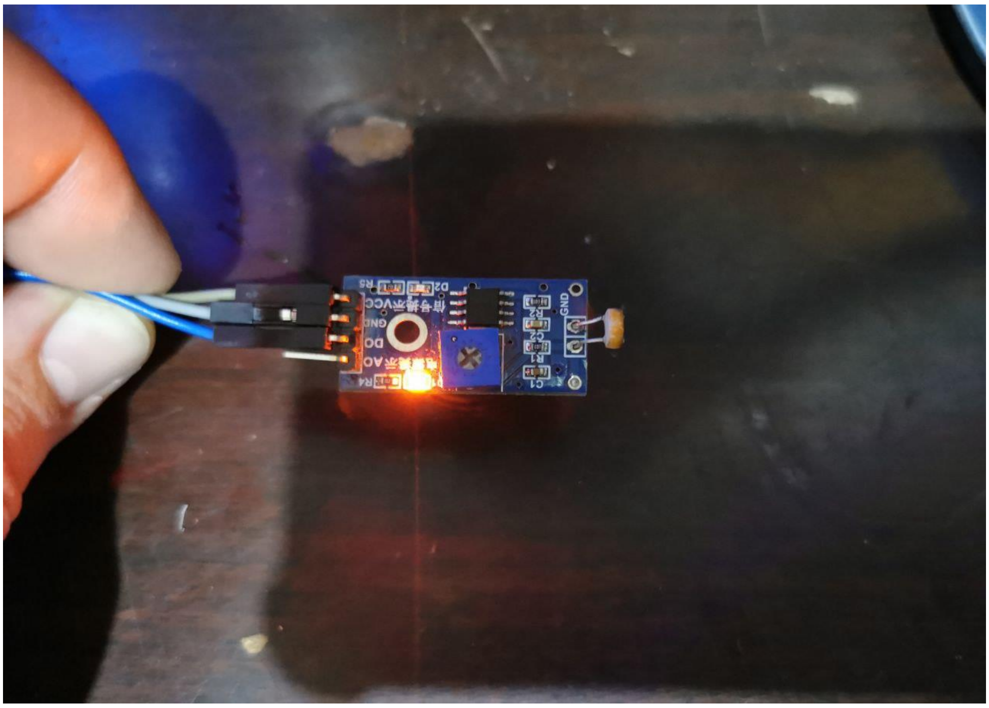

Figure.IV.8. Light absence test.

The real-time light intensity values were visualized using the Pi5 environment, which allowed for dynamic display and easier interpretation of changes in lighting conditions **"Figure IV.9"**.

Figure.IV.9. Light Readings from the LDR through PI5 prompt.

IV.3.1.3. Actuators (LEDs) Tests

The experiments were conducted indoors under various ambient conditions to test the sensors. To represent the system's reaction to specific thresholds, **Four LEDs** were used as:

- **Red LED (represents The Heater):** Turns ON when the **temperature drops below 27°C**, simulating a heating response **"Figure IV.10"**.
- **Blue LED 1 (represents The Fan):** Turns ON when the **temperature exceeds 29°C**, simulating a cooling response **"Figure IV.11"**.
- **Blue LED 2 (represents The Air pump):** Turns ON when the **humidity exceeds 60%**, simulating a ventilation response **"Figure IV.12"** .
- **Green LED (represents the lights):** Turns ON when it's dark (or when no lights are detected), simulating an automatic lighting response **"Figure IV.13"** .

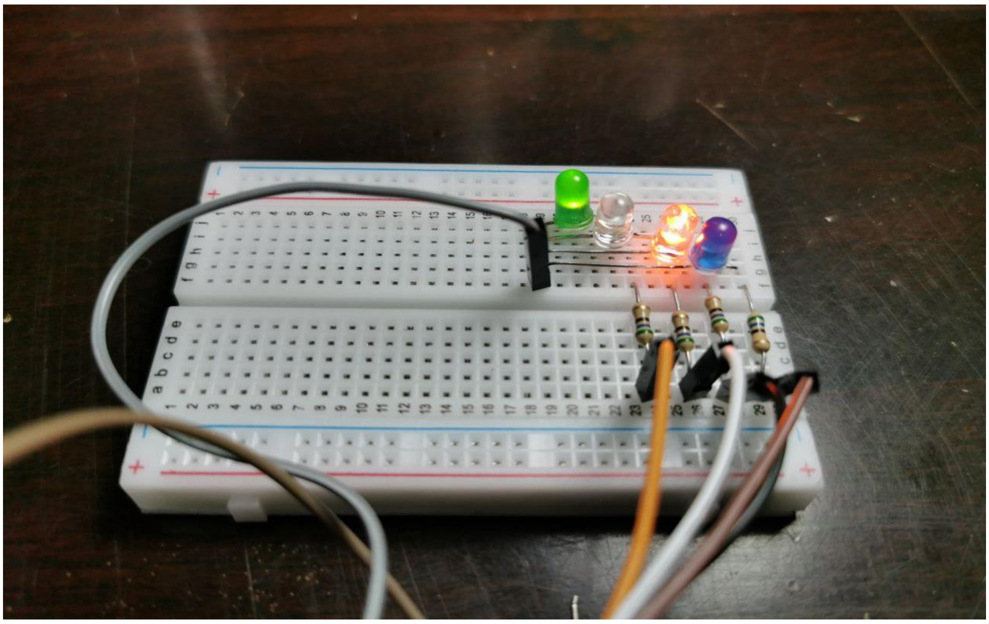

Figure.IV.10. Heating response simulation.

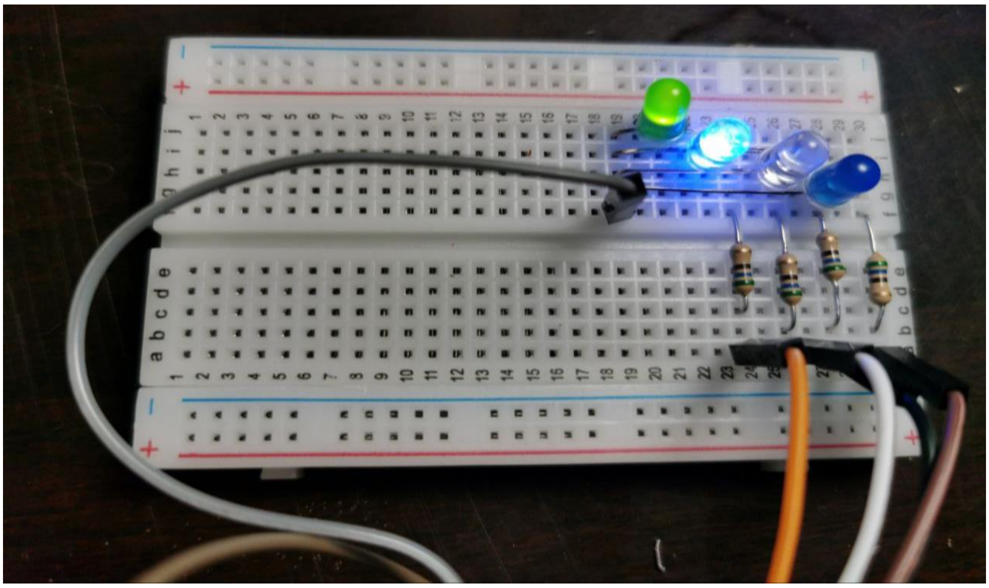

Figure.IV.11. Cooling response simulation.

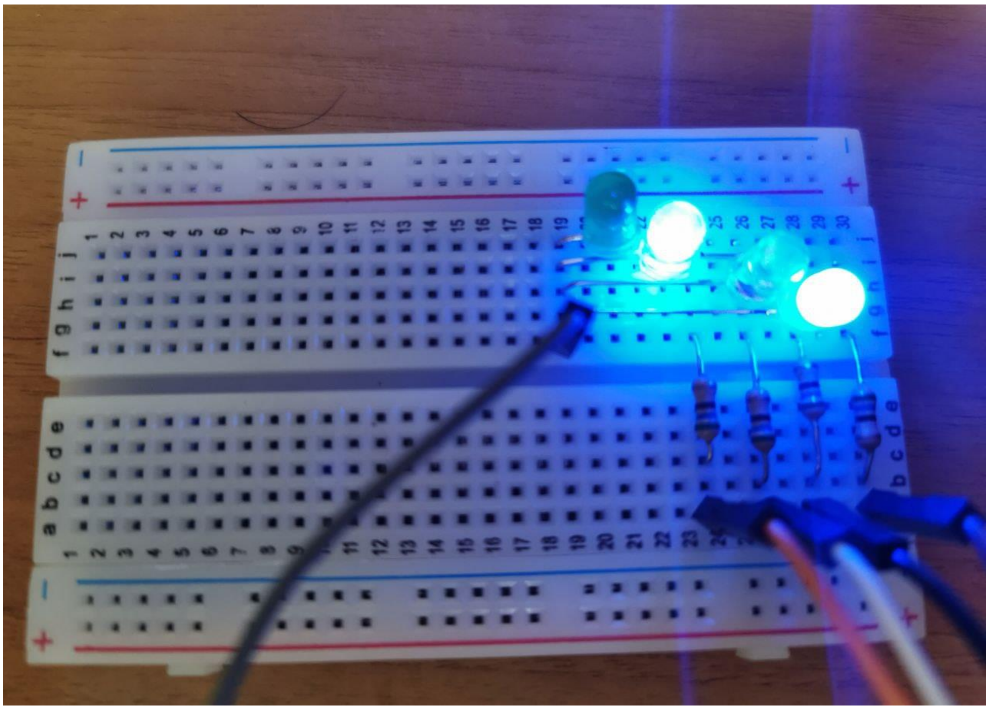

Figure.IV.12. Ventilation response simulation.

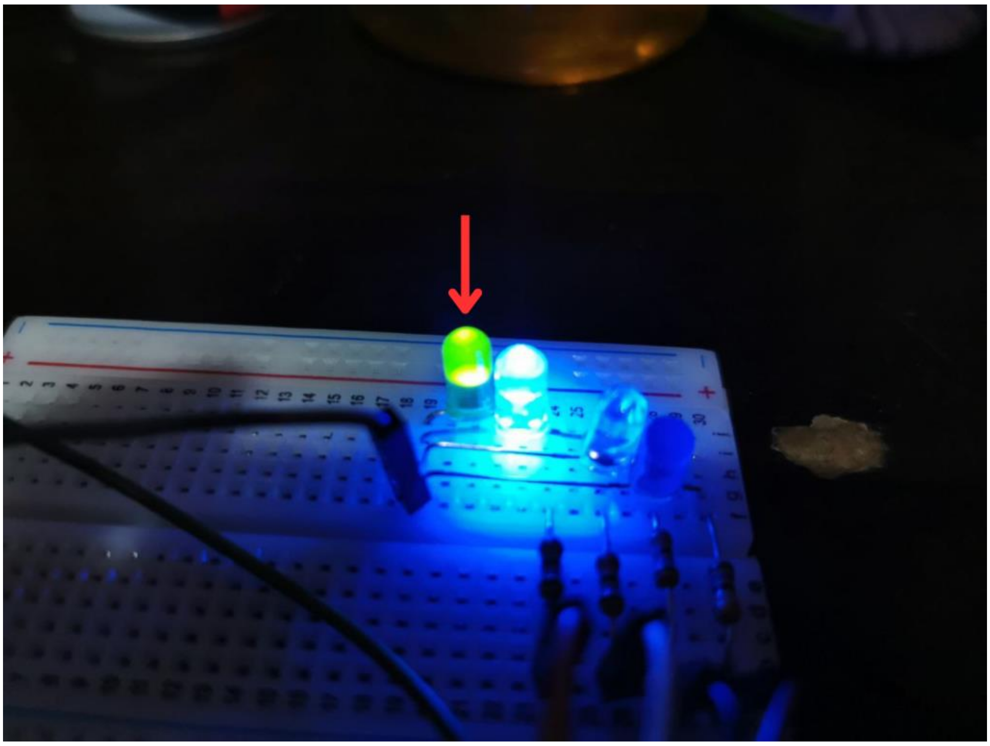

Figure.IV.13. Lightning response simulation.

IV.3.2. Future Plans for Sensors and actuators

The current setup includes a working DHT11 sensor for monitoring temperature and humidity, as well as an LDR sensor for measuring light intensity. These sensors have been successfully integrated and tested. The pH sensor, however, has not yet been tested and remains a key component for future development, given its importance in maintaining balanced water conditions in the aquaponics system.

Future improvements will focus on calibrating and integrating the pH sensor into the system. Once functional, it is expected to work alongside automated dosing mechanisms that can correct pH imbalances in real time. Additional enhancements may include the introduction of water level sensors to support better control over irrigation cycles and prevent overflow or dry conditions.

On the actuator side, improvements are expected in making control systems more responsive to live data. This includes optimizing the behaviour of components such as the fish feeder, water pumps, aeration systems, and lighting shifting from basic timed operations to sensor-driven automation. These upgrades aim to increase system autonomy, efficiency, and long-term stability.

IV.4. Drone Design and Performance Evaluation

IV.4.1. Drone Design

Due to constraints in resources and time, the physical drone has not yet been constructed. However, to ensure the full evaluation, a detailed 3D model closely representing the actual design has been developed. This model includes all critical components such as the frame, motors, flight controller, sensors, and battery. Using this virtual representation, we analyze the expected performance, including flight time, range, and power consumption, providing valuable insights into the drone's operational capabilities within the smart aquaponics system.

A/. The 3D designed model **"Figure IV.14"** through **"Sketchfap"** [95]

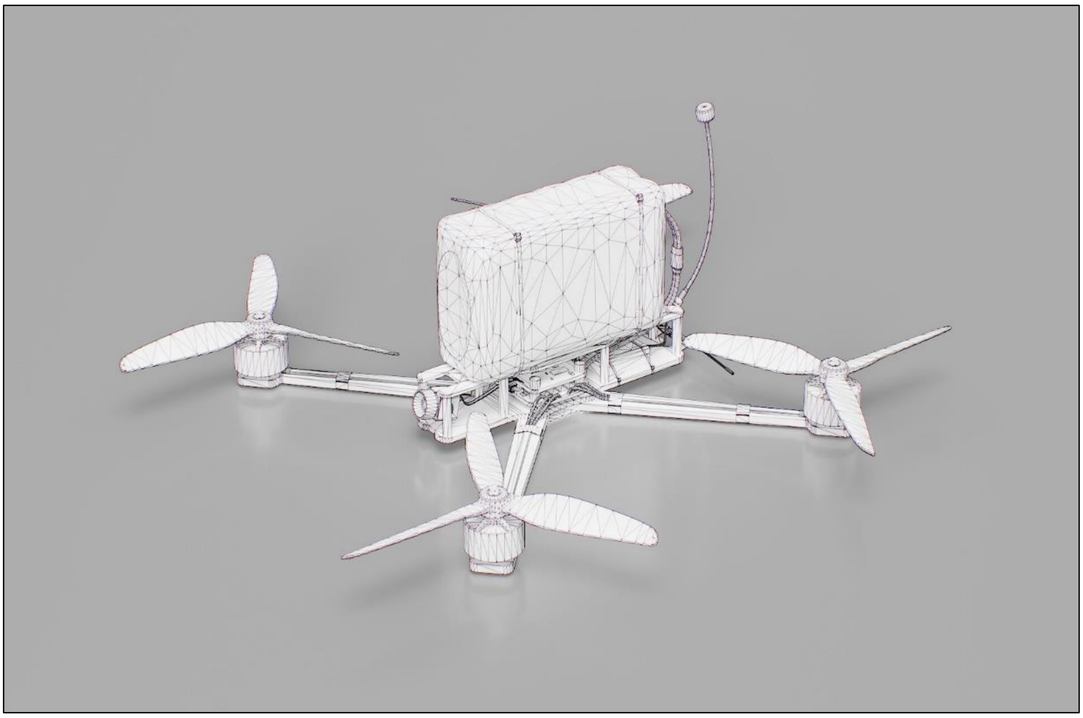

Figure.IV.14. 3D drone Model.

After rendering and resizing, this is the final result is elaborated in **"Figure IV.15"** below**:**

Figure.IV.15. Rendered 3D drone Model.

The model represents most of the parts (this is the closest we could approach to reality):

Frame, Flight Controller + ESC, Motors + Propellers, RX radio receiver, Battery + XT60 connector

This model reflects how the custom drone would've mostly looked like if it was built

IV.4.2. Performance Evaluation

IV.4.2.1. Estimated Flight Time

The flight time of the drone depends primarily on the battery capacity and the average current draw during flight. The key components are motors, flight controller, and sensors, here's the main data to estimate flight time:

- **Battery capacity (C) :** 2600 mAh = 2.6 Ah
- **Battery voltage (V):** 6S LiPo battery ≈ 22.2 V (3.7 V nominal per cell × 6 cells)
- **Average current draw (I):** Estimated from motor and electronics consumption at 50% throttle

Assuming each motor draws about 5 A at 50% throttle (typical for RS2205 2300KV with 7-inch propellers)

Note: 50% throttle is enough to lift the drone, here's how much speed motors can reach in this case:

$$\boldsymbol{motor\ speed = 2300KV \times 22V \times 50\% \ = 25300RPM}$$ [96]

The total motor current is:

$$\boldsymbol{motors\ current = 4 \times 5A = 20A}$$

Adding approximately 2 A for the flight controller, GPS, and other electronics:

$$\boldsymbol{total\ current = 20A + 2A = 22A}$$

Then, the estimated flight time [97] (in hours) is:

$$\boldsymbol{flight\ time = \frac{C}{Itotal} = \frac{2.6}{22} \approx 0.118hours = 7.1minutes}$$

To summarize, based on the estimated total current draw of approximately 22 A at 50% throttle, (which can provide a speed of the drone's battery capacity of 2.6 Ah supports an expected flight time of about **7 minutes**. This duration is sufficient for capturing detailed aerial imagery of the lettuce fields within the smart aquaponics system. Although actual flight time may vary depending on environmental conditions and payload, this estimation provides a realistic baseline for planning operational flights and scheduling data collection tasks.

IV.4.2.2. Estimated Flight Range [98]

The operational range of the drone depends on its flight time and average cruising speed. Since the drone must complete a round trip from the control station to the target area and back, the maximum effective range is half of the total distance it can cover during a full battery cycle.

Assuming an average cruising speed v of 30 km/h and the previously estimated flight time t of approximately 0.118 hours (7.1 minutes), the total distance dtotal the drone can travel is calculated as:

$$\boldsymbol{total\ flight\ distance = v \times t = 30 \times 0.118 = 3.54\ km}$$

Since the drone needs to return to the control point safely, the maximum operational radius drange is half of this total distance:

$$\boldsymbol{flight\ range = \frac{total\ flight\ distance}{2} = \frac{3.54}{2} = 1.77\ km}$$

To conclude, the drone's estimated range is around 1.77 km, however real-world conditions can reduce that distance. Since it uses a 2.4 GHz signal to communicate with the ground station, things like trees, buildings, hills, or even bad weather can weaken the connection. Because of that, it's safer to leave a buffer and not fly the drone to its absolute limit this helps ensure the signal stays strong and the drone can return safely.

Even with that in mind, the range is more than enough to film and monitor a lettuce field within the smart aquaponics system. It gives good coverage for collecting aerial data and checking crop health from above.

IV.4.2.3. Estimated Weight and lifting capacity

Accurate weight estimation is crucial for understanding the drone's flight performance and available lifting margin. Below is a breakdown of the drone build with no payload **"Table IV.1"**:

Component	**Estimated weight (g)**
Frame (Mark 4.7")	**150-180**
Speedybee F405 Mini stack	**25-30**
RS2205 2300Kv Motors × 4	**120-140**
HQProp 7×4×3 Props × 4	**20-24**
Walksnail GPS	**10-12**
AION Nano RX	**5-8**
Runcam thumb Pro 4K	**20-25**
Wiring and harness	**20-30**
Capasitor (220µF/35V)	**3-5**
GNB 2600mAh 6S LiPo	**250-270**

Table.IV. 1. Estimated weight for each component used in drone's building process.

Total Estimated Weight (No Payload): 623–724 grams, then the Average Operational Weight: 670 grams, Since we know that Each RS2205 2300KV motor can provide 700–800g of maximum thrust, at 50% throttle, the estimated per-motor thrust is around 350–400g [99].

$$\textbf{Total Thrust at 50\%} = \textbf{4 motors} \times \textbf{375g (avg)} = \textbf{1,500g}$$

In this case, we get:

$$\textbf{Remaining Lift Capacity} = \textbf{1,500g (thrust)} - \textbf{670g (drone weight)}$$
$$= \textbf{830g excess thrust}$$

This indicates that we can safely carry up to 700–750g of payload at 50% throttle before exceeding efficient flight margins.

IV.4.3. Future Plans for Drone Development

Although the drone has not yet been physically built, we see it as a key part of our smart aquaponics system, and we have several ideas for how to improve and expand its capabilities in the future.

One of the first goals is to actually build the drone using the components we've selected. Once assembled, we'll be able to test how it performs in real conditions and compare the real flight time, range, and lifting capacity to the values we estimated in this chapter.

Looking ahead, we plan to add a First-Person View (FPV) system, which would allow us to see exactly what the drone sees while flying. This would be especially helpful when manually inspecting parts of the field or navigating through tight areas. It also makes flying more engaging and gives us better control during missions.

Another exciting idea is to add a small spraying system, so the drone can be used for more than just monitoring. For example, if our AI model detects signs of disease in the lettuce, the drone could help apply organic treatments or natural pesticides in specific areas. Since the drone has a good amount of extra lift capacity, it can safely carry a lightweight spray bottle and pump without losing stability.

To improve energy efficiency, we're also thinking about integrating a small solar panel. While it wouldn't power the motors, it could help power up smaller electronics like the FPV camera or sensors.

We also hope to make the drone more independent over time by adding GPS-based autopilot features. This would allow it to follow a preplanned route, collect images automatically, and return to the base when it's done. With some obstacle sensors, we could also make it safer and smarter during flights.

Lastly, we plan to set up failsafe mechanisms such as auto-return-to-home in case the signal drops or the battery runs low. This way, the drone can safely come back without human input, reducing the risk of losing it.

Overall, our vision is to turn this drone into a reliable tool that does more than just fly, it can crop, help with treatment, and support the overall goal of a smarter and more sustainable farming system.

IV.5. Economic and Technical approach

IV.5.1. Small-scale system budget estimation [100]

This project represents a small-scale operational aquaponics prototype, built to demonstrate the real-world applicability of smart agriculture methods using low-cost, energy-efficient components. It integrates:

- Environmental monitoring sensors (DHT11, Light sensor, pH sensor)
- Environmental actuators (Fan, Heater, Pumps, Air pump)
- A Raspberry Pi 5 controller for automated AI control
- A custom-built quadcopter drone for aerial monitoring and image capturing
- A Convolutional Neural Network (CNN)-based disease prediction model trained to detect lettuce diseases.

In this case, we can estimate the total cost for the small-scale aquaponics prototype using financial information through Trusted sources online **"Table IV.2"**

Component category	**Item (Model/Type)**	**Estimated cost (USD)**
Microcontroller	Raspberry Pi5 (8GB) + SD card + Power supply	130-160
Sensor	DHT11, Light, PH, Water level	50-100
Actuators	Fan, Heater, water pumps (2), Air pump, Relays	90-170
Power supply	2 × 12 V/1000mAh car batteries	60-70
Drone frame	Mark 4 (7-inches FPV frame)	45-65
Flight controller	Speedybee F405 Mini stack (FC +ESC 35A)	65-90
Motors + Props	RS2205 2300Kv × 4 + HQProps 7040	55-80
GPS	Walksnail WS-M181 GNSS module	35-60
TX/RX + controller	Jumper AION Nano TX + AION RX + Radiomaster Zorro	100-130
Battery & Adapter	GNB 6S 2600mAh LiPo + Capacitor + XT60 plug	70-100
Camera & Mount	Runcam Thumb pro W 4K + TPU Mount	85-105
Assembly tools and supplementary Parts	Wires, Jumper cables, electrical tape, plastic straps, screw drivers, pliers…..etc	60-80
Total estimated cost		**905-1100**

Table.IV.2. Small scale system cost estimation.

Note: our aquaponics prototype was developed on a no budget using manual assembly, reused components, and open-source software tools.

IV.5.2. Future Large-Scale Expansion Estimation [101]

To scale this smart aquaponics system from a functional prototype to a commercial large scale fully automated farm, several additional investments and improvements must be considered **"Table IV.4"**. here's the list of future upgrades :

Upgrade / Feature	**Estimated Cost (USD)**
Advanced environmental sensors (Total Dissolved Solids, Dissolved Oxygen … ETC)	250~300
Solar-powered energy system (panels + battery bank)	600~800
High-resolution FPV drone camera	600~1000
Drone auto-docking station with charging	400~500
Cloud/server-based AI + IoT integration	300~450
Waterproof housing and commercial setup	800~1000
Reinforced structural frame and tanks	700~850
Web-based dashboard and remote control	200~250
Training data expansion (image collection & labeling)	250~300
Total Additional Large-Scale Upgrades	**4500~6000**

Table.IV.3. Large scale commercial system expansion cost estimation.

Compared to the prototype, the large-scale system introduces:

- ❖ 24/7 autonomous monitoring via web dashboard and real-time alerts.
- ❖ Drone-based disease mapping across wider crop areas using FPV imaging.
- ❖ Improved AI model accuracy due to larger datasets and cloud-based model retraining.
- ❖ Longer system lifespan using solar energy and weatherproof materials.
- ❖ Reduced manual labor and optimized resource consumption.
- ❖ Predictive analytics to forecast crop performance and detect disease.

IV.5.3. Lettuce Crop Economical Return [102]

The economic return of a smart aquaponics system depends largely on the crop yield, market demand, and operational efficiency. Lettuce, being a high-turnover, fast-growing crop with consistent demand, is ideal for this type of system.

In a controlled aquaponics environment, each lettuce plant can mature in 30 to 40 days, enabling up to 9 harvest cycles per year. Using vertical stacking and optimized water/nutrient management, crop density and growth rates improve significantly compared to traditional soil farming.

On average, one healthy lettuce plant sells for \$0.50 to \$1.00 USD, depending on quality and market. Organic and hydroponically grown produce often fetches higher prices in urban and health-conscious markets; since large-scale systems occupy roughly 120 square meters, with vertical farming racks and automated environmental control, we can harvest :

- 800 to 1200 lettuce plants per cycle (Each plant requires 0.1–0.15 square meters of grow bed space)
- 9 cycles per year
- 7200 to 11000 heads of Lettuce annually

Assuming a conservative average sale price of $1.5 per plant, the potential annual revenue is $11000– $16500 PER YEAR

IV.6. Conclusion

This study explores the potential of building a smart aquaponics system using low-cost components and modern technologies like AI, drones, and automation. While the prototype was not physically implemented, the design, cost estimation, and system planning were all based on real components, and current market prices.

Technically, the system integrates essential environmental sensors, a Raspberry Pi 5 controller, an AI model for early disease detection, and a drone for crop surveillance. Each element contributes to a smart and responsive growing environment, capable of supporting high-efficiency lettuce farming.

Economically, a scaled-up version covering 120 square meters could produce 7,000 to 11,000 lettuce plants annually, at a market price of around $1.50 per plant, the system could generate $11,000 to $16,500 per year in revenue making it an affordable and scalable solution for sustainable agriculture.

Although this project remains theoretical for now, the research demonstrates that with the right resources and support, smart aquaponics can become a viable option for real-world deployment especially in areas facing land, water, or labor limitations.

Conclusion

Conclusion

This project set out to explore how technology could be used to rethink and improve traditional agricultural practices through the development of a smart aquaponics system. At its core, the project aimed to merge multiple technologies—artificial intelligence, embedded systems, IoT sensors, and drone-based vision into one cohesive and sustainable farming solution. The result is a working prototype that successfully demonstrates the feasibility and value of intelligent automation in agriculture.

Aquaponics itself represents a sustainable approach to food production, combining aquaculture (raising fish) and hydroponics (growing plants in water) in a mutually beneficial cycle. This project harnessed that natural synergy and enhanced it with modern engineering, making it smarter, more precise, and capable of operating with minimal human intervention. The integration of real-time monitoring tools, such as temperature, humidity, and light sensors, enabled the system to make intelligent decisions through a Raspberry Pi, automating core tasks like lighting, feeding, and water circulation.

One of the major achievements of this project is the successful implementation of an AI-powered lettuce disease detector. Using a convolutional neural network (CNN) model trained on image data, the system can distinguish between healthy, fungal-infected, and bacterial-infected lettuce leaves with high accuracy. This feature demonstrates the powerful role that artificial intelligence can play in managing crop health early and efficiently, especially in environments where manual inspection is impractical or resource-limited.

In addition to software and data, attention was also given to system hardware and physical design. The modular architecture and use of affordable, widely available components make the system not only replicable but also adaptable to different scales and environments. From a simple academic prototype, it has the potential to evolve into a robust solution for small farms, greenhouses, and even urban agriculture setups.

The real strength of this work lies in its interdisciplinary nature. Electrical engineering, control systems, embedded programming, artificial intelligence, agriculture, and environmental science were all combined to solve a modern challenge: how to grow healthy food efficiently, with minimal waste, and with smarter tools. Each discipline contributed to making the system more intelligent, responsive, and reliable.

Equally important, this project serves as a stepping stone for future exploration and research. It introduces not just a working system, but also a framework one that can be expanded and improved in many directions. From completing the drone integration to achieving full real-time control and remote accessibility, the possibilities for evolution are both meaningful and exciting.

In conclusion, this memoire presents more than a technical achievement it represents a vision for the future of agriculture: one that is data driven, environmentally conscious, and powered by smart automation. It proves that with the right tools and multidisciplinary thinking, even small systems can make a big impact. As global concerns around food security, climate resilience, and sustainable development grow, projects like this show how engineering and innovation can and must play a role in shaping tomorrow’s world.

Future Work and Perspectives

While the current system is functional and promising, several avenues remain open for enhancement to improve autonomy, intelligence, and usability. One of the most anticipated developments is the physical construction of the drone, which is expected to significantly increase the system's surveillance and operational range. Once built and tested, it will be upgraded with features such as a First-Person View (FPV) camera, GPS-guided autopilot, obstacle detection, and possibly a lightweight spraying mechanism for targeted treatment of diseased crops. Integrating solar panels to power auxiliary electronics like sensors or cameras is also under consideration to improve energy efficiency. In parallel, the AI-powered lettuce disease detector will undergo major enhancements, including full deployment on the Raspberry Pi 5 for real-time field diagnostics, expansion of the dataset especially for underrepresented classes such as bacterial infections and automated image input via drone or web-based transfers. The detection results could be linked to live dashboards or automated alert systems, creating a closed-loop feedback mechanism for smart crop management. Additionally, the sensor and actuator framework will be expanded with a calibrated pH sensor, water level monitoring, and smarter, sensor-driven control of pumps, lighting, and feeding systems to replace basic timer operations with real-time decision-making. To ensure greater accessibility and oversight, a user-friendly web interface is planned, allowing remote monitoring and control of the system via smartphones or PCs. Altogether, these future upgrades are aimed at transforming the prototype into a fully autonomous, data-driven agricultural assistant that supports sustainable farming with minimal human input.

Bibliography

[1]. Goddek, S. et al. (2023). Aquaponics and its role in sustainable food production. ScienceDirect.

https://www.sciencedirect.com/science/article/pii/S0040162523003943

Accessed on: March 15, 2025, 11:30 AM

[2]. Love, D. C. et al. (2015). Commercial aquaponics production and profitability: Findings from an international survey. GALA University of Greenwich.
https://gala.gre.ac.uk/id/eprint/23247

Accessed on: March 15, 2025, 11:38 AM

[3]. Palm, H. W. et al. (2018). Towards commercial aquaponics: a review of systems, technologies, and practical experiences. Springer.

https://link.springer.com/article/10.1007/s10499-018-0249-z

Accessed on: March 15, 2025, 12:45 AM

[4]. Rakocy, J. E. et al. (2012). Aquaponic systems for sustainable food production. Springer Handbook of Sustainable Development.
https://link.springer.com/chapter/10.1007/978-981-97-5703-9_36

Accessed on: March 15, 2025, 1:30 PM

[5]. Sarker, I. H. et al. (2024). AI-based monitoring and drone integration in smart agriculture. Artificial Intelligence Review.

https://link.springer.com/article/10.1007/s10462-024-11003-x

Accessed on: March 15, 2025, 1:37 PM

[6]. Tyson, R. V., Treadwell, D. D., & Simonne, E. H. (2011). Opportunities and challenges to sustainability in aquaponic systems. HortTechnology, 21(1), 6–13.
https://journals.ashs.org/horttech/view/journals/horttech/21/1/article-p6.xml

Accessed on: March 15, 2025, 2:15 PM

[7]. https://extension.okstate.edu/fact-sheets/aquaponics.html

Accessed on: March 15, 2025, 2:19 PM

[8]. Britannica. Greenhouse effect.

https://www.britannica.com/science/greenhouse-effect

Accessed on: March 15, 2025, 2:32 PM

[9]. University of Massachusetts. Retractable roof greenhouses and shadehouses. https://www.umass.edu/agriculture-food-environment/greenhouse-floriculture

Accessed on: March 15, 2025, 2:41 PM

[10]. Urban Farm Online. The lean-to greenhouses and their benefits. https://www.urbanfarmonline.com/the-lean-to-greenhouses-and-their-benefits

Accessed on: March 16, 2025, 9:01 AM

[11]. https://www.archdaily.com/979425/how-can-greenhouse-design-change-architecture

Accessed on: March 16, 2025, 9:17 AM

[12]. https://www.gardeningknowhow.com/gardening-pros-cons/pros-and-cons-of-growing-in-a-greenhouse

Accessed on: March 16, 2025, 9:26 AM

[13]. Greenhouse Grower. Insights on retractable roof greenhouses. https://www.greenhousegrower.com/technology/structures/insights-on-retractable-roof-greenhouses

Accessed on: March 16, 2025, 9:35 AM

[14]. Gardening Know How. How much light do seedlings need? https://www.gardeningknowhow.com/garden-how-to/seeds/how-much-light-do-seedlings-need

Accessed on: March 16, 2025, 9:48 AM

[15]. Greenhouse Grower. Great plants for 2018 that stand up to heat and humidity. https://www.greenhousegrower.com/crops/great-plants-for-2018-that-stand-up-to-heat-and-humidity

Accessed on: March 16, 2025, 9:59 AM

[16]. Greenhouse Grower. Monitoring carbon dioxide in the greenhouse. https://www.greenhousegrower.com/production/monitoring-carbon-dioxide-in-the-greenhouse

Accessed on: March 16, 2025, 10:11 AM

[17]. https://www.bacfertilizers.com/knowledge-centre/blog/3778-which-factors-affect-plant-growth-and-bloom

Accessed on: March 16, 2025, 2:06 PM

[18]. Go Green Aquaponics. Basics of aquaponics. https://gogreenaquaponics.com/blogs/news/basics-of-aquaponics

Accessed on: March 16, 2025, 2:16 PM

[19]. Agri Farming. Tilapia fish farming in aquaponics: A full guide. https://www.agrifarming.in/tilapia-fish-farming-in-aquaponics-a-full-guide

Accessed on: March 16, 2025, 2:20 PM

[20]. Farming Aquaponics. Best aquaponics plants.

https://farmingaquaponics.com/best-aquaponics-plants

Accessed on: March 16, 2025, 2:23 PM

[21]. Go Green Aquaponics. The importance of nitrifying bacteria in aquaponics systems. https://gogreenaquaponics.com/blogs/news/the-importance-of-nitrifying-bacteria

Accessed on: March 16, 2025, 2:30 PM

[22]. https://gogreenaquaponics.com/blogs/news/basics-of-aquaponics

Accessed on: March 16, 2025, 2:37 PM

[23]. Human Being. (n.d.). Smart Aquaponics System.

https://humanbeing.my/smart-aquaponics-system/

Accessed on: March 16, 2025, 3:10 PM

[24]. Cultivate Nation. (2024). Raspberry Pi Smart Farming: Agritech Innovations and Applications.

https://cultivatenation.com/raspberry-pi-smart-farming-agritech-innovations-projects-applications/

Accessed on: March 16, 2025, 3:17 PM

[25]. University of Cambridge. (n.d.). BUGALERT: Pest and Disease Monitoring in Greenhouses with Raspberry-Pi Network.

https://www.gci.cam.ac.uk/global-challenges-research-cambridge/ongoing-research-projects/bugalert-pest-and-disease-monitoring

Accessed on: March 16, 2025, 3:25 PM

[26]. Go Green Aquaponics. (2025). How to Choose the Right Aquaponics System Monitoring Tools.

https://gogreenaquaponics.com/blogs/news/choosing-the-right-aquaponics-system

Accessed on: March 16, 2025, 3:30 PM

[27]. Channa, A. A., Munir, K., Hansen, M., & Tariq, M. F. (2022). Water IoT Monitoring System for Aquaponics Health and Fishery Applications. Sensors, 22(19), 7679. https://www.mdpi.com/1424-8220/22/19/7679

Accessed on: March 16, 2025, 3:47 PM

[28]. Editorial Staff. (2019, April 4). The need for battery safety systems for drones [Figure – Circuit diagram]. Power Electronics News

https://www.powerelectronicsnews.com/the-need-for-battery-safety-systems-for-drones/

Accessed on: March 16, 2025, 3:52 PM

[29]. Khan, M. U., Zia, M. Y. I., & Baig, M. F. (2022). Drone and controller detection and localization: Trends and challenges [Figure – Communication system diagram]. ResearchGate.
https://www.researchgate.net/publication/366156655_Drone_and_Controller_Detection

Accessed on: March 16, 2025, 4:02 PM

[30]. D. Accardo, "Normativa droni," Ingenio Web, Nov. 19, 2021. [Figure – Yaw, pitch, and roll diagram]. [Online]

https://www.ingenio-web.it/pdfs/academy-accardo-normativa-droni.pdf

Accessed on: March 16, 2025, 4:06 PM

[31]. Rammohan, Y. S., Prashanth, U., & Vinay, K. R. (2022). Design and development of a robotic drone: A drover [Figure – Assembly of rover]. International Journal for Research in Applied Science & Engineering Technology (IJRASET)

https://www.ijraset.com/research-paper/design-and-development-of-a-robotic-drone

Accessed on: March 16, 2025, 4:24 PM

[32]. Salah, A., Makhloufi, A., & Benaicha, S. (2024). Watch the Skies: A Study on Drone Attack Vectors, Forensic Approaches, and Persisting Security Challenges [Figure - Anatomy of a Drone]. ResearchGate.
https://www.researchgate.net/publication/382290450_Watch_the_Skies_Study_on_Drone

Accessed on: March 16, 2025, 4:36 PM

[33]. Authors Unknown. (n.d.). Ground Target Tracking UAV [Figure 4 – Forces applied]. ResearchGate.

https://www.researchgate.net/publication/368955861_Ground_Target_Tracking_UAV/figures?lo=1

Accessed on: March 16, 2025, 4:40 PM

[34]. Koç, M. T. Drone Technologies and Applications. Academia.edu. https://www.academia.edu/105646007/Drone_Technologies_and_Applications.

Accessed on: March 16, 2025, 5:00 PM

[35]. Wikipedia contributors. History of Unmanned Aerial Vehicles. Wikipedia, the free encyclopaedia.

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

Accessed on: March 16, 2025, 5:14 PM

[36]. Aero Corner. Types of Drones: Everything You Need to Know. https://aerocorner.com/blog/types-of-drones.

Accessed on: March 16, 2025, 5:21 PM

[37]. HusFarm. (2023). The Role of Drones in Facilitating Sustainable Aquaculture Practices.: https://husfarm.com/article/the-role-of-drones-in-facilitating-sustainable-aquaculture-practices

Accessed on: March 16, 2025, 5:30 PM

[38]. HusFarm. (2023). AI in Aquaponics: Optimizing Water Use and Fish Feeding with Robotics. https://husfarm.com/article/ai-in-aquaponics-optimizing

Accessed on: March 16, 2025, 5:47 PM

[39]. https://learn.adafruit.com/dht/overview
Accessed on: 02 April 2025, 10:15 AM

[40]. https://how2electronics.com/how-to-use-ldr-sensor-module-with-arduino/
Accessed on: 02 April 2025, 11:42 AM

[41]. https://components101.com/sites/default/files/component_datasheet/LDR%20Datasheet.pdf

Accessed on: 02 April 2025, 12:00 PM

[42]. https://gogreenaquaponics.com/blogs/news/the-importance-of-ph-in-aquaponics

Accessed on: 02 April 2025, 2:30 PM

[43]. https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0161-V2_Web.pdf

Accessed on: 02 April 2025, 3:15 PM

[44]. https://ar.aliexpress.com/item/1005006366896348.html

Accessed on: 02 April 2025, 4:00 PM

[45]. https://www.mdpi.com/2079-9292/13/5/820

Accessed on: 02 April 2025, 4:45 PM

[46]. https://friendlyaquaponics.com/how-to-manage-temperature-and-humidity-in-your-aquaponics-greenhouse/

Accessed on: 03 April 2025, 9:30 AM

[47]. https://www.thermal-engineering.org/temperature-control-in-aquaponics-systems

Accessed on: 03 April 2025, 10:45 AM

[48]. https://gogreenaquaponics.com/blogs/news/what-is-the-best-humidity-for-aquaponics-systems

Accessed on: 03 April 2025, 11:20 AM

[49]. https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

Accessed on: 03 April 2025, 1:15 PM

[50]. https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf

Accessed on: 03 April 2025, 2:30 PM

[51]. https://www.flyeye.io/drone-technology-materials-design/

Accessed on: 03 April 2025, 3:45 PM

[52]. https://www.ebay.com/itm/167244264228

Accessed on: 03 April 2025, 4:00 PM

[53]. https://www.grepow.com/blog/what-is-a-drone-flight-controller.html

Accessed on: 04 April 2025, 9:15 AM

[54]. https://www.speeddrones.nl/speedybee-f405-mini-bls-35a-20x-20-stack.html

Accessed on: 04 April 2025, 10:30 AM

[55]. https://www.dronezon.com/learn-about-drones-quadcopters/how-drone-motors-esc-propulsion-systems-work/

Accessed on: 04 April 2025, 11:45 AM

[56]. https://www.amazon.com/Readytosky-RS2205-2300KV-Brushless-Multicopter/dp/B088NGCZ64

Accessed on: 04 April 2025, 1:00 PM

[57]. https://www.hqprop.com/hq-durable-prop-7x4x3-light-grey-2cw2ccw-poly-carbonate-popo-p0133.html

Accessed on: 04 April 2025, 2:15 PM

[58]. https://www.rfwireless-world.com/terminology/other-wireless/understanding-drone-sensors

Accessed on: 04 April 2025, 3:30 PM

[59]. https://caddxfpv.com/products/walksnail-ws-m181

Accessed on: 05 April 2025, 9:45 AM

[60]. https://www.drone-operator.com/how-do-drones-communicate-unraveling-the-mystery/

Accessed on: 05 April 2025, 11:00 AM

[61]. https://www.amazon.com/Jumper-Module-Compatible-Control-Airplane/dp/B09TDKBD7G

Accessed on: 05 April 2025, 12:15 PM

[62]. https://www.jumper-b2b.com/new-jumper-r1-v2-mini-d16-receiver-compatible-with-frsky-d16-xm-protocol-rxsr-sbus-jumper-t-pro-t-lite-t18-t16-t12-t8sg-p0101.html

Accessed on: 05 April 2025, 1:30 PM

[63]. https://www.droneblog.com/drone-controller/

Accessed on: 05 April 2025, 2:45 PM

[64]. https://www.radiomasterrc.com/products/zorro-radio-controller

Accessed on: 05 April 2025, 4:00 PM

[65]. https://www.droneassemble.com/power-distribution-boards-how-to-choose-the-right-one/

Accessed on: 06 April 2025, 9:15 AM

[66]. https://www.amazon.com/Gaoneng-2600mAh-250-330mm-Brushless-Helicopter/dp/B0BK87SQ5H?th=1

Accessed on: 06 April 2025, 10:30 AM

[67]. https://www.amazon.com/Laisomeke-Adapter-Connector-Battery

Accessed on: 06 April 2025, 11:45 AM

[68]. https://www.droneblog.com/drone-gimbal/

Accessed on: 06 April 2025, 1:00 PM

[69]. https://ar.aliexpress.com/item/1005006212852158.html?gatewayAdapt=glo2ara

Accessed on: 06 April 2025, 2:15 PM

[70]. https://ar.aliexpress.com/item/1005008291165771.html

Accessed on: 06 April 2025, 3:30 PM

[71]. Raspberry Pi Foundation. "Raspberry Pi Imager." ***Raspberry Pi Documentation*****. Raspberry Pi Foundation**

https://www.raspberrypi.org/documentation/

Accessed on: 8 April 2025, 09:13 AM

[72]. https://drive.google.com/file/d/1CZclQ6_L7eZov8n396nMGiXxtNvogZq3/view

Accessed on: 8 April 2025, 09:23 AM

[73]. Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.
TensorFlow. "Image classification." *TensorFlow Core Documentation*
https://www.tensorflow.org/tutorials/images/classification

Accessed on: 8 April 2025, 09:38 AM

[74]. Python Software Foundation, *"Welcome to Python.org"*,

https://www.python.org

Accessed on: 8 April 2025, 10:00 AM

[75]. Anaconda, Inc. (n.d.). *Anaconda Navigator Overview*. Retrieved May 12, 2025,
https://www.anaconda.com/products/distribution

Accessed on: 8 April 2025, 10:09 AM

[76]. Python Software Foundation. (n.d.). *os — Miscellaneous operating system interfaces*
https://docs.python.org/3/library/os.html

Accessed on: 8 April 2025, 10:18 AM

[77]. TensorFlow. (n.d.). *TensorFlow: An end-to-end open-source machine learning platform*.

https://www.tensorflow.org

Accessed on: 8 April 2025, 10:26 AM

[78]. NumPy Developers. (n.d.). *NumPy: The fundamental package for scientific computing with Python*.

https://numpy.org

Accessed on: 8 April 2025, 10:32 AM

[79]. OpenCV.org. (n.d.). *OpenCV-Python Tutorials*.

https://docs.opencv.org

Accessed on: 8 April 2025, 10:37 AM

[80]. Pillow Developers. (n.d.). *Pillow (PIL Fork) Documentation*.

https://pillow.readthedocs.io

Accessed on: 8 April 2025, 11:00 AM

[81]. Matplotlib Developers. (n.d.). *Matplotlib: Visualization with Python*.

https://matplotlib.org

Accessed on: 8 April 2025, 11:13 AM

[82]. Brownlee, J. (2019). *How to organize data for image classification*.

https://machinelearningmastery.com/how-to-organize-data-for-machine-learning/

Accessed on: 8 April 2025, 01:13 PM

[83]. TensorFlow Team. (2024). *Image data preprocessing with* image_dataset_from_directory. TensorFlow Documentation

https://www.tensorflow.org/tutorials/load_data/images

Accessed on: 8 April 2025, 01:19 PM

[84]. Chollet, F. (2021). *Deep Learning with Python* (2nd ed.). Manning Publications.

https://www.manning.com/books/deep-learning-with-python-second-edition

Accessed on: 8 April 2025, 01:27 PM

[85]. https://drive.google.com/file/d/1s3t11oG_Vm0JtN7sKYdX0HUEdjDYHvTq/view

Accessed on: 8 April 2025, 01:35 PM

[86]. https://drive.google.com/file/d/1q8-eViTAGQwVduu5An_RfU2UrA5OwDic/view?usp=sharing

Accessed on: 8 April 2025, 01:39 PM

[87]. Python Software Foundation. (2021). *PyInstaller documentation*.

https://pyinstaller.readthedocs.io

Accessed on: 8 April 2025, 01:47 PM

[88]. Betaflight. (2021). *Betaflight firmware*.

https://betaflight.dev

Accessed on: 8 April 2025, 01:58 PM

[89]. https://flyingsquirrel.ch/how-to-switch-inflight-between-betaflight-osd-profiles/

[90]. ArduPilot. (2023). *ArduPilot autopilot software*.

https://ardupilot.org

Accessed on: 8 April 2025, 02:32 PM
[91]. Mission Planner. (2023). *Mission Planner - ArduPilot GCS*.

https://ardupilot.org/planner/

Accessed on: 8 April 2025, 02:48 PM

[92]. Speedybee. (2023). *Speedybee App - Mobile drone configuration tool*
https://www.speedybee.com

Accessed on: 8 April 2025, 03:00 PM

[93]. Drive link with a Set of examples for the three classes (10 samples each) and the CNN model and python script that were used to build the detector

https://drive.google.com/drive/folders/1LiwmxYPqFB1WdkNUUxP150LVWjt64FC8

Accessed on: 8 April 2025, 03:10 PM

[94]. Drive link with a standalone "EXE" version of the AI driven disease detector

https://drive.google.com/drive/folders/1ZORR3xK-gIHj_Q6z7g-s1sJ27HU?usp=sharing

Accessed on: 8 April 2025, 03:15 PM

[95]. 3D Drone model design

https://www.cgtrader.com/3d-models/fpv-drone

Accessed on: 25 may 2025, 09:00 PM

[96]. RS2205S 2300KV Datasheet

https://oscarliang.com/emax-rs2205s-2300kv-motors/

Accessed on: 25 may 2025, 09:18 PM

[97]. Flight time estimation formulas

https://www.flitetest.com/articles/how-to-estimate-the-flight-time-for-a-battery

Accessed on: 25 may 2025, 10:00 PM

[98]. Flight range and performances estimation formulas

https://news.quadpartpicker.com/how-to-estimate-and-calculate-drone-flight-characteristics/

Accessed on: 25 may 2025, 10:30 PM

[99]. Thrust and lifting capacity estimation

https://craycle.com/product/rs2205-2300kv-brushless-motor-ccw/

Accessed on: 27 may 2025, 09:00 PM

[100]. Small scale aquaponics system average cost

https://bootstrapbee.com/aquaponics/aquaponics-average-cost-and-profit-per-square-foot

Accessed on: 27 may 2025, 09:20 PM

[101]. Large scale aquaponics system average cost

https://www.sciencedirect.com/science/article/abs/pii/S0959652622023514

Accessed on: 27 may 2025, 09:39 PM

[102]. Lettuce prices and annual revenue estimations per market price

https://aquaponics.com/aquaponic-systems/commercial-systems/

Accessed on : 27 may 2025, 11 :00 PM

الملخص:

يعرض هذا المشروع نظامًا ذكيًا للأكوابونيك يجمع بين الاستزراع السمكي والزراعة بدون تربة، باستخدام تقنيات حديثة مثل الذكاء الاصطناعي، وإنترنت الأشياء، وراسبيري باي، والطائرات بدون طيار. يعتمد النظام على حساسات ومشغلات لضبط الظروف البيئية، ويستخدم نموذجًا للذكاء الاصطناعي قائمًا على الشبكات العصبية الالتفافية (CNN) لكشف أمراض الخس. كما يعتمد هذا النظام على طائرة بدون طيار مخصصة للمراقبة الجوية. يمثل هذا النظام المتكامل خطوة فعالة نحو زراعة ذكية ومستدامة تعمل بشكل آلي.

الكلمات المفتاحية: الزراعة الذكية، الأكوابونيك، الذكاء الاصطناعي، إنترنت الأشياء، راسبيري باي، الطائرات بدون طيار، كشف الأمراض، الشبكات العصبية الالتفافية، المراقبة البيئية، الزراعة المستدامة.

Abstract

This project introduces a Smart Aquaponics System that combines aquaculture and hydroponics into a unified smart farming approach, leveraging modern technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), drones, and embedded systems like Raspberry Pi. The system employs sensors and actuators to maintain optimal conditions for both fish and plants, while an AI model based on Convolutional Neural Networks (CNNs) detects diseases in lettuce crops. A custom-built drone supports aerial monitoring by capturing high-resolution images to assess plant health. This integrated solution promotes sustainable, automated, and resource-efficient agriculture.

Keywords: Smart agriculture, aquaponics, AI, IoT, Raspberry Pi, drones, disease detection, CNNs, environmental monitoring, sustainable farming.

Résumé :

Ce projet présente un système aquaponique intelligent, combinant aquaculture et hydroponie avec des technologies modernes telles que l’intelligence artificielle (AI), l’Internet des objets (IoT), les microcontrôleurs Raspberry Pi et les drones. Des capteurs et des actionneurs assurent des conditions optimales pour les poissons et les plantes, tandis qu’un modèle d’intelligence artificielle basé sur des réseaux de neurones convolutifs (CNN) détecte les maladies affectant les laitues. Un drone personnalisé améliore la surveillance aérienne grâce à des images haute résolution. Cette solution intégrée favorise une agriculture durable, automatisée et performante.

Mots-clés : Agriculture intelligente, aquaponie, AI, IoT, Raspberry Pi, drones, détection des maladies, CNNs, surveillance environnementale, agriculture durable.