RoBoat: An Automated Aquatic Vehicle for Collecting Solid Water Waste Using Raspberry Pi 4 Microcontroller with Computer Vision for Real-time Waste Detection and Monitoring Via Web Page Application

A Project Study Presented to the Faculty of Electronics Engineering Department College of Engineering Technological University of the Philippines

In Partial Fulfillment of the Course Requirements for the Degree of

Bachelor of Science in Electronics Engineering

Submitted by:

ARCIETE, Raiken E.

HICARO, Doenelle John Carl

LABASAN, John Claude R.

MEJILLA, Janellyn P.

NUÑEZ, Vincent Carlo C.

Advisers:

Engr. Jessica S. Velasco

Engr. Alvin Gazo

2023

APPROVAL SHEET

This project entitled "**RoBoat: An Automated Aquatic Vehicle for Collecting Solid Water Waste Using Raspberry Pi 4 Microcontroller with Computer Vision for Real-time Waste Detection and Monitoring Via Web Page Application**", has been prepared and submitted by the following proponents:

Hicaro, Doenelle John Carl | Mejilla, Janellyn P.

Arciete, Raiken E. | Nuñez, Vincent Carlo C.

Labasan, John Claude R

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

ENGR. JESSICA S. VELASCO
Project Adviser

ENGR. ALVIN GAZO
Project Adviser

ENGR. AUGUST C. THIO-AC
Panel Member

ENGR. JOMER V. CATIPON
Panel Member

ENGR. MARK P. MELEGRITO
Panel Chair

Accepted and approved in partial fulfilment of the requirements for the degree of **Bachelor of Science in Electronics Engineering.**

ENGR. TIMOTHY M. AMADO
Head, ECE Department

ENGR. NILO M. ARAGO
Dean, College of Engineering

ABSTRACT

This study aims to address the growing concern of marine pollution caused by solid waste in bodies of water by developing an automated aquatic waste vehicle to collect such waste. The goal was to reduce the amount of waste in the seas and protect the health and safety of water species and the local community. To achieve this, the study employed computer vision and the use of YOLOv5 programming to develop an algorithm that directed the vehicle's movement. The algorithm used a camera and sensors to detect the presence of waste and obstacles in the water, allowing the vehicle to navigate through the water and collect solid waste. The vehicle was developed using stainless steel for the structure, making it durable and resistant to water damage. The researchers used an Arduino Mega to control the vehicle's movement and a Raspberry Pi 4B for image processing. A conveyor belt and a trash bin were installed in the prototype for efficient and practical garbage collection. The results of the study showed that the automated aquatic waste vehicle is a promising solution to address the growing problem of marine pollution caused by solid waste in bodies of water. The proposed solution can reduce the amount of waste in the seas, thereby protecting the ecosystem and improving the quality of life for aquatic life and the local community.

ACKNOWLEDGEMENT

The researchers would like to express their earnest appreciation and gratitude to everyone who contributed to making this project successful. The assistance, advice, and support that they received have been vital throughout the entire process. First and foremost, they would like to express their heartfelt gratitude to Engr. Jessica S. Velasco and Engr. Alvin Gazo, their project advisers, for their consistent support, helpful advice, and constant encouragement. Their experience and guidance were critical in establishing the project's direction and ensuring its successful completion. They also like to thank the panelists, Engr. Jomer V. Catipon, Engr. Mark P. Melegrito, Engr. August C. Thio-ac and Engr. Glenn C. Virrey, for their insightful comments, constructive criticism, and recommendations. They are also grateful for the assistance and resources provided by the staff and people of Barangay San Antonio in Cavite City. Their assistance and technical support were critical to the project's data gathering. Finally, they want to thank their family and friends for their constant support, understanding, and encouragement during this effort. With their assistance and valuable insights, this project was possible. They appreciate their time, feedback, and willingness to help to make this project possible.

Table of Contents

CHAPTER 3: METHODOLOGY

LIST OF FIGURES

LIST OF TABLES

Chapter 1

The Problem and Its Background

This chapter includes the introduction, background of the study, research gap, research objectives, significance of the study, scope and limitation, and the definition of terms used.

1.1 Introduction

Around 71% of the Earth's surface consists of water and serves as a habitat for aquatic organisms. Water is necessary for the human population, as well as other animals, plants, and microbes, to have an adequate food supply and a productive environment. However, modernization and coastal development have resulted in the destruction of natural habitats, particularly for marine animals such as fish.

Pollution has become a significant environmental issue due to human disregard and carelessness. Limiting or mitigating all forms of pollution due to their harmful effects on humans and the environment is crucial. The Philippines, with its extensive coastline and rich marine biodiversity, faces challenges in marine conservation due to coastal development, pollution, overfishing, and destructive fishing practices. The country has a significant plastic pollution problem, and a large amount of plastic ends up in the ocean every year. This is especially true for coastal areas like Manila Bay.

Water pollution arises from various sources, including natural, artificial, and domestic contamination. Domestic sources, such as sewage and laundry waste, contribute significantly to water pollution. Waste generation, including water waste, remains an ongoing environmental concern that poses challenges for resolution. With population growth, there is an increase in waste production and environmental contamination. The

Philippines is a major contributor to marine plastic pollution, generating significant amounts of waste daily, including plastic waste that piles up on land and clogs coastlines, ultimately spilling into the sea.

The influence of marine litter, particularly plastics, on marine life and ecosystems is a critical problem investigated by the scientific community. The seafloor acts as a sink for marine litter, where human pressure and environmental conditions affect the type, density, and accumulation rates of litter. Debris can accumulate on the seafloor, particularly near populated regions and submarine canyons, with diverse effects on benthic ecosystems and fauna.

Microplastics also contribute to water pollution, with trillions of microplastics present in the world's largest upper oceans. These tiny particles harm animals and ecosystems, as they are accessible to many organisms and can be ingested. Each microplastic particle has a unique makeup due to its production, use, and disintegration.

1.2 Background of the Study

In a study by Ruangpayoongsak et al. (2017), the Floating Trash Scooper Robot (FWSR) was developed to replace human labor in floating waste collection. The robot features two waste scooper designs and incorporates an adjustable conveyor belt and driving speeds to optimize waste collection. Using a scoop net, the FWSR significantly increases the amount of waste collected.

Another study by Akib et al. (2019) presents a robot prototype for waterborne waste collection. The robot utilizes two propellers driven by DC motors for maneuvering, a mobile application-based Bluetooth control system, and a robotic hand for efficient trash collection. The prototype also incorporates a conveyor belt and sensor systems to prevent overloading. The robot's lightweight and water-resistant design, using polyvinyl chloride (PVC) board, allows it to operate autonomously for up to four hours, collecting up to 10 kg of trash in a 3000 square centimeter area.

Ang et al. (2013) introduce the Automated Waste Sorter (AWS) and Mobile Robot Waste Deliver System (MRWDS), a garbage collector robot designed to collect debris from multiple containers. The MRWDS is controlled using a button that provides radio frequency signals to guide the robot to the designated bins. The robot navigates predetermined paths, collects the waste bins, and empties the accumulated garbage into a larger bin on its back. The waste is then fed onto a conveyor belt and sorted accordingly using a sensor array.

Zavare et al. (2017) propose a Smart Garbage Bin solution to optimize trash management. The system utilizes GSM and GPS technologies to provide advance notification when the bin is nearing full capacity and to relay its location to the appropriate authorities. This helps in timely cleaning and efficient waste collection, reducing tours by garbage trucks and lowering fuel emissions.

Wahyutama et al. (2022) developed an intelligent trash bin that detects and classifies recyclables using a webcam and real-time object detection. The system employs the You Only Look Once (YOLO) algorithm on a Raspberry Pi, allowing the bin lid to rotate and reveal the appropriate compartment for disposal based on the classification result. The IoT hardware, including an Arduino Uno, enables capacity monitoring through ultrasonic sensors and captures the bin's location using GPS.

Bairachtaris et al. (2010) focus on designing an electric drive system for an electric boat. They utilize highly efficient electric machines and power electronic converters to achieve maneuverability and reduce noise and pollution. The study describes using a low-voltage permanent magnet DC motor and an interleaved four-channel DC/DC converter to control the propulsion motor's RPM. The experiments demonstrate the efficiency and suitability of the proposed solution for driving low voltage-high current DC motors.

1.3 Research Gap

Intelligent aquatic cleaning vehicles that successfully clean different bodies of water are currently unavailable in the Philippines. An ideal aquatic cleaning vehicle is an automated accumulator capable of identifying objects floating in the water while also improving the vehicle's overall capability to be deployed on seas or rivers. The Internet of Things also makes good use in developing the ideal aquatic cleaning vehicle. Using computer vision to image process floating objects in the water could help create a successful self-sustaining robot. To ensure the vehicle is automated, an algorithm can be created to program it to move freely without human intervention.

1.4 Research Objectives

1.4.1 General Objective

This Research aims to create an Automated Aquatic Waste Vehicle that collects solid waste in bodies of water to help reduce the amount of waste in seas. It also aims to protect the health and safety of water species as well as the local community.

1.4.2 Specific Objectives

1. To develop an automated aquatic plastic waste collector vehicle with Raspberry Pi 4B, Arduino Mega, and sensors.
2. To create an aquatic vehicle body structure that can withstand the weight of the garbage gathered.
3. To implement image processing techniques and algorithms for detecting and collecting plastic waste.
4. To develop a Web Application for tracking and monitoring the Aquatic vehicle.
5. To test and validate the developed device using ISO standards.

1.5 Significance of the Study

This study primarily focuses on developing an aquatic vehicle to collect solid waste on the water's surface. Collecting waste in the sea significantly affects the environment and community safety. Waste can be reduced, water pollution can be minimized, and marine animals can have better habitats and will directly increase in numbers. In addition, minimizing water pollution will prevent the spread of waterborne diseases and make the place safer for the community. Moreover, using different techniques to develop a waste collector can contribute to the existing knowledge by solving the problem of solid waste in water with a different approach.

The study is inclined toward water security, specifically in the Tubig ay Buhayin at Ingatan (TUBIG) program of the Harmonized National R&D Agenda (HNRDA). Moreover, the study contributes to the Sustainable Development Goal (SDG) Number 14 to conserve and sustainably use the oceans, sea, and marine resources. One part of sustainable development goal number 14 is to prevent and significantly reduce all types of marine pollution, particularly from land-based activities, such as marine debris and nutrient pollution, by 2025.

Using the aquatic robot to collect plastic waste on the sea's surface will significantly help sustain a better environment and lessen water pollution in different areas. Plastics and marine pollution are spreading all over the sea and oceans resulting in ocean warming, acidification, and fishery collapse, putting our oceans under severe threats. Collecting plastic wastes using the aquatic robot equipped with image processing and a global positioning system (GPS) tracker will help clean the oceans and provide a better and more sustainable daily living for over 3 billion people worldwide.

1.6 Scope and Limitations

The study aims to develop an automated aquatic vehicle that efficiently detects solid waste on the surface of bodies of water. Philippine seaports and coastal areas are ideal places to deploy the robot, with piles of water waste present around these areas. It uses a DC motor to move a conveyor belt responsible for waste collection, and the waste collected is placed in a waste bin at the back of the robot. The Raspberry Pi 4B microcontroller will automate the system and image processing of the developed algorithm. A camera connected to the boat generates images that will be processed using Deep Learning to determine solid waste on the water's surface. An Arduino Mega is used for the Internet of Things. Sensors are connected to the microcontroller, such as a weight sensor and GSM sensor, which will notify the user if the robot has collected its maximum waste. A GPS sensor is also connected to the boat to locate its location, and an ultrasonic sensor is used to detect large objects on its path.

However, the aquatic vehicle has its limitations. The robot will collect any objects it recognizes as garbage in the water but avoid objects registered as obstacles. It only collects waste floating in the water, and the limit in the amount of plastic waste collected is 50 kg. It will roam at a maximum distance of 900 meters from the shore. The boat does not operate on renewable energy, like solar energy. It is powered by three lead-acid batteries.

1.7 Definition of Terms

Arduino Mega - an advanced iteration of the Arduino Uno microcontroller board, specifically created to offer enhanced capabilities and greater processing power for handling intricate projects. It builds upon the foundation of the Arduino Uno, aiming to provide expanded functionalities and improved performance for more complex undertakings.

Conveyor Belt - A mechanical apparatus that transports goods or materials from one place to another. It consists of a belt that is driven by pulleys and forms a continuous loop around them.

GPS module - a device that captures signals transmitted by GPS satellites and provides location and time information to other devices. The Global Positioning System (GPS) is a constellation of over 30 navigational satellites orbiting the Earth.

GSM module - enables the linkage between electrical devices and the GSM network. The GSM network refers to the Global System for Mobile Communications, a widely used standard for cellular communication networks.

Lead-acid Battery - A rechargeable battery powered by a combination of lead and sulfuric acid is utilized. The battery operates through a controlled chemical reaction facilitated by immersing lead in sulfuric acid. This reaction enables the battery to produce electrical energy. Subsequently, the process is reversed to recharge the battery.

Marine Litter - a persistent form of waste, which includes solid substances that have been discarded, disposed of, or abandoned within the maritime and coastal environment. Such waste tends to have a long-lasting presence in these areas, posing potential environmental and ecological risks.

Microplastics - Microscopic plastic debris in the environment due to consumer and industrial waste disposal and breakdown. These are defined as plastics with a diameter of less than five millimeters (0.2 inches).

Raspberry Pi 4b - a highly capable and flexible single-board computer. With enhanced performance, expanded connectivity options, and increased expandability, it serves as a versatile platform suitable for a broad spectrum of projects and applications.

You Only Look Once (YOLO) Algorithm - a regression-based method that can simultaneously predict classes and bounding boxes for an entire image with a single algorithmic execution. It falls under the category of Deep Learning algorithms and is specifically designed to process and analyze images.

Chapter 2

Review of Related Literature and Studies

The literature related to this study discusses and focuses on the plastic waste in the water and its collection. The related literature and studies will help further understand the study's goal to develop an automated aquatic vehicle that collects solid waste on water surfaces.

2.1 Waste Collector/Management

The Automatic waste management system is a step forward in automating the manual collection and detection of waste. The newly created system is integrated using four subsystems. The Smart Trash System (STS), Local Base Station (LBS), Smart Vehicle System (SVS), Smart Monitoring and Control RFID, and the Internet are used to control the prototype in which it would pioneer solid waste work processes for data collection, monitoring, and management. This waste management proposal is more efficient and time-saving than the currently employed method in which concerned municipal employees have to manually look for the filled waste bins across different spots in an area/street.

According to Bashir et al. (2013), solid waste management has emerged as a prominent environmental concern, disrupting the ecological balance and posing health risks to society. The detection, monitoring, and management of waste have become urgent issues in our time. The conventional practice of manually monitoring waste in bins is a laborious and time-consuming task that is not compatible with modern technologies and demands significant human effort, time, and financial resources. This paper presents a novel approach to waste management by proposing an automated system. The proposed system aims to automate the monitoring and management of solid waste collection processes. The employed technologies are well-suited to ensure efficient and accurate monitoring and management of the solid waste collection process, promoting a greener environment.

Domestic waste disposal into rivers or lakes causes a drop in water quality, significantly impacting water pollution in the long run. Several countries developed semi-automated and fully automated garbage collection craft equipped with the new system and kept up with technological advancements. Developing an automated vehicle is important for reducing garbage and the manpower required to handle the cleaning process. The objective of this project is to develop a Water Trash Collector (WTC) model with excellent stability, utilizing high-grade stainless steel for its fabrication. The catamaran design is identified as the most suitable hull structure for effectively collecting surface-level garbage in water bodies.

Furthermore, Shamsuddin et al. (2020) indicated in this study that the electrical goal is to develop a remotely controlled WTC using a SkyFly controller and a battery as the primary power source. Based on the USV element data, a 3D catamaran model is sketched in Siemen NX10 software. The design incorporates essential components such as a trash bin collector and a waterwheel to facilitate collection and movement of the Water Trash Collector (WTC). To construct the WTC frame, square hollow mild stainless steel is chosen as it offers several advantages including cost-effectiveness, high corrosion resistance, durability, and longevity. During testing on the lake, the WTC demonstrated satisfactory performance without any leakage on both hulls after the attachment of shafts and propellers. The WTC is operated and propelled through the motors, which are controlled by a dedicated controller.

Three key considerations in designing aquatic robots are cost-effectiveness, reliability, and durability. Somal et al. (2020) presented a research paper outlining the design of a robot specifically engineered for collecting floating waste on water surfaces. The project's objective is to optimize efficiency in terms of time, energy, and overall process speed. To address the cleaning task's requirements, the robot's layout was designed with a car-like mechanism to ensure stability, maneuverability, and efficient waste collection. The use of a plastic pipe container was found to be the most suitable choice, meeting the necessary structural stability criteria. A motor-driven conveyor belt was developed to facilitate waste collection and deposit it into a plastic box attached to the platform. By establishing a centralized system, multiple robots can be controlled and supervised. The prototype also integrates an image processing algorithm. Testing of the robot prototype demonstrated its ability to collect waste and return it to a designated waypoint. The robot has a maximum trash load capacity of 5 kg. Successful testing was conducted on the designed conveyor's capability. It was observed that both the robot's driving speed and the conveyor belt's speed influenced waste collection, with optimal performance achieved at a driving speed of 0.38 m/s and a belt speed of 0.5 m/s. The maximum number of bottles collected per minute was recorded as 1.71 kg/minute, showcasing a 75% faster collection rate compared to human labor using a scoop net. Consequently, the Ocean Surface Trash Collector has proven its potential to replace human effort in water surface cleaning tasks.

The global threat of environmental pollution necessitates the implementation of effective solutions to address air, soil, and water pollution. Pollution control measures, particularly innovative ones, are crucial in combating this issue. Given that water constitutes a significant portion of the Earth's surface, with limited freshwater resources available for human consumption, it becomes imperative to address water pollution cases. Existing pollution control initiatives include manual and machine-based cleaning, which require continuous human supervision and can pose risks to human safety. Therefore, the development of an autonomous robot capable of cleaning water bodies holds immense potential in pollution control efforts. This paper discusses the design and analysis of a river-cleaning robot that can perform tasks such as collecting floating waste and conducting underwater inspections. The robot consists of a frame, cylindrical hull, thrusters, and wide waste disposal arms. Additionally, the paper presents the determination of hydrodynamic coefficients using the ANSYS Fluent solver, hydrodynamic modeling, static structural analysis, hull buckling analysis, and prototype development (Nair et al., 2019).

Akib et al. (2019) developed an affordable floating trash collection robot that can be operated remotely in various water bodies such as canals, ponds, rivers, and oceans. The exponential rise in plastic usage globally has led to severe ecological disruptions and health risks, including cancer, congenital disabilities, and immune system problems. The robot prototype comprises two propellers driven by DC motors, allowing for omnidirectional movement, a Bluetooth-based mobile application for remote control, and a robotic hand for trash collection. A conveyor belt system guides the collected trash to a collection box, equipped with a sensor system to prevent overloading. The manufacturing and maintenance costs of the robot are minimized.

Table 1. Summary of Water Collector/Management Related Studies

Author	Year	Title	Findings	Relationship to the study
Bashir et al.	2013	Concept, Design, and Implementation of Automatic Waste Management System	The researchers concluded that an automatic waste management system is a significant advancement in the manual collection and detection of waste.	Their smart trash system can be innovated to adjust for our goal of an automated aquatic waste collector.
Shamsuddin et al.	2020	Development of Water Trash Collector	The researchers produced a Catamaran-type design of water trash collected which has the highest stability and can carry the trash	Their design of a Water Trash Collector can be our support in making our prototype because the researchers developed a low-cost model.
Somal et al.	2020	Ocean Surface Trash Collector	The study demonstrates how the conveyor belt speed and the robot's driving speed affect how well the robot can gather waste. The optimal velocity for the driving speed is recorded as 0.38 m/s, while the ideal belt speed is measured at 0.5 m/s.	Their model has a conveyor belt for the collection of trash on the surface of the ocean. We can use their study as a guide in creating the conveyor system of our prototype.
Nair et al.	2019	Design And Fabrication of River Cleaning Robot	The design of their robot aims to also be suitable for inspecting and collect waste underwater.	Their paper has detailed information about their proposed water waste cleaning robot.
Akib et al.	2019	Unmanned Floating Waste Collecting Robot	The robot's prototype has two propellers attached to two DC motors that allow it to move forward, backward, left, and right. It also has a robotic hand that can easily pick up rubbish and a Bluetooth control system that is based on mobile applications.	Their paper has information about creating a mobile application to monitor the robot. Their idea of using a Bluetooth control system to control their prototype at a distance can also be our guide in creating a self-sustaining robot.
Pathak et al.	2018	Garbage Level Monitoring System using Raspberry Pi	The researcher proposes a system that could collect waste without any human intervention. In his paper, he inserted real-time fill level information to monitor the garbage being collected and inform the operation before the instances of overflowing occur.	We saw a potential in this paper to be our guide in creating a garbage level monitoring system for the collected water waste of our conveyor belt.

With a trash collection capacity of up to 10 kg, the prototype can clean an area of approximately 3000 square centimeters while consuming only 45 watts of battery power. It can operate continuously for four hours before requiring recharging. To ensure suitability for aquatic environments, the robot's design incorporates a streamlined, lightweight, water-resistant body made of Polyvinyl chloride (PVC) board.

A floating tube ensures buoyancy, while the streamlined construction aids in propulsion through water currents. The conveyor belt features a roughened rubber surface selected for its water-repelling properties. All components, including the conveyor belt, are constructed using water-resistant materials that do not add excessive weight. The trash collector includes etched spaces for additional garbage deposition, and the water from the conveyor system can freely return to the water body without adding to the weight of the collected trash. The propeller is laser-cut and polished, and the robot utilizes the breastshot water wheel principle.

2.2 Conveyor Belt

Noordin, A. (2013) developed a lightweight conveyor system for easy transportation, specifically designed for handling light loads. The conveyor bed utilizes a leather belt due to its high frictional properties with PVC pipes. Powered by a 12 V DC motor, the conveyor eliminates the need for additional power sources. The load vehicle is driven by a 6 V DC motor operating at a low speed of 10 rpm to provide sufficient torque. Each wheel is equipped with six DC motors, generating a one kgf/cm torque to achieve a load capacity of at least 3 kg.

Furthermore, the research emphasizes the advantages of belt conveyors, such as their ability to carry heavy loads, accommodate long conveying distances, feature a straightforward design, require minimal maintenance, and exhibit high operational dependability. Commonly in foundry shops, conveyor belt systems play a vital role in material transport, facilitating the supply and distribution of molding sand, molds, and waste removal. The study concludes that the designed system is suitable for practical implementation and discusses future developments to enhance usability, adaptability to different pipe sizes, and exploration of alternative solutions such as pneumatic systems and leadscrews.

Amarnath et al. (2019) introduced an innovative solution called the automatic waste segregator (AWS) to address the demand for an affordable and user-friendly domestic waste segregation system. The AWS effectively sorts garbage into three primary categories: metallic, organic, and plastic, streamlining the waste management process by facilitating direct routing for appropriate processing. The system incorporates ultrasonic sensors in all waste bins to monitor the level of waste accumulation. Once the waste reaches a certain threshold, the sensors trigger an alert sent to a microcontroller. Leveraging GSM technology, the microcontroller promptly sends an SMS notification to the garbage collection truck driver, ensuring timely and efficient waste collection.

Efficiency in an advanced manufacturing setting can be enhanced by appropriately designing mechanisms from the onset of the manufacturing process. Salawu et al. (2020) aimed to model and simulate the design parameters of a conveyor system responsible for transporting parts between production stages. The movement of materials on the conveyor belt necessitates two different types of power: maximum power for driving the electric motor and minimum power for driving the pulley. Through a model-based design and simulation approach, optimal efficiency of the conveyor belt during operation was achieved.

Table 2. Summary of Conveyor Belt Related Studies

Author	Year	Title	Findings	Relationship to the study
Noordin, A.	2013	Design and Selecting the Proper Conveyor-Belt	With the aid of standard model calculations, this paper demonstrates how to design a conveyor system that incorporates belt speed, width, motor selection, specification, shaft diameter, pulley, and gearbox selection.	Since we will be using conveyor belts to transfer the collected plastic to its bin, the knowledge gained from this study is essential to developing our next prototype. This document will help us build a conveyor system that will work for the prototype.
Amarnath, M et al.	2019	Design and Analysis of Floating Waste Cleaning Machine	A straightforward device incorporating a bucket conveyor has been developed for the purpose of solid waste removal. The construction of the machine is completed, and Autodesk Fusion 360 software is employed to calculate its load capacity, efficiency, as well as the stress and strain exerted on the machine frame. As the wheel navigates through obstacles, it experiences stress and strain.	We can use the indicated materials in this paper to trial and error the best conveyor system.
Salawu, G et al.	2020	Modelling and Simulation of a Conveyor Belt System for Optimal Productivity	A conventional model was constructed, assuming specific parameters, to examine the impact of varying loads below and above the rated speed of an electric motor. Equations were derived to describe the operational characteristics of a manufacturing system for conveyor belts, utilizing designated notations.	The paper that Salawu and co-researchers focus on is creating a conveyor belt system that will increase the productivity of the prototype that will use; that is why this paper could be used as a reference in creating an ideal conveyor belt for our waste collector.

2.3 Image processing

Image processing is a branch of artificial intelligence in which a huge number of photos are analyzed to extract information. Image processing algorithms are commonly used to detect objects. Based on the detection, an alert will be sent to the authority via the app to take appropriate measures. The image processing algorithm detects human action on the coast using OpenCV and deep learning models to classify the action (polluting water bodies/not polluting water bodies) Pajankar, A. (2017).

According to the research conducted by Tang et al. (2016), the findings demonstrate that hashing techniques exhibit resilience against common digital operations applied to photos, such as rotation at any angle. These techniques also possess desirable discriminative capacity and sensitivity to changes in visual information. In comparative analyses using receiver operating characteristic (ROC) curves with other popular hashing algorithms, the developed hashing method outperforms comparable methods in terms of robustness and discrimination in classification.

Effective waste management is crucial for local governments in Senegal, as they are responsible for managing waste disposal sites, monitoring landfills, and preventing illegal dumping. In this project, a model was developed to detect waste deposits. The model incorporates various tools, including remote sensing and Convolutional Neural Networks (CNN), a Deep Learning technology for object detection. Due to its sophisticated and abstract nature, CNN is considered an exemplary architecture for image processing and object detection, making it a recommended choice for achieving these objectives. In general, the combination of deep learning and remote sensing has broad applications in addressing environmental issues, flooding, real estate, and other related challenges (Youme et al., 2016).

The HASH function is commonly used to generate fixed-length outputs from variable-length input data, enhancing data processing efficiency. It finds application in various real-world scenarios, such as image processing, information retrieval, and cryptography, where it efficiently processes variable-length data. Image hashing, a technique utilized in applications such as image retrieval, image authentication, digital watermarking, image copy detection, image indexing, image quality assessment, and multimedia forensics, effectively extracts content-based compact representations from input images.

The project presented by Rovetta et al. (2009) introduces a unique application that combines distributed sensor technologies and geographic information systems for municipal solid waste monitoring. The application was successfully field-tested and evaluated in Shanghai's Pudong district, China. The project focuses on establishing a network of garbage bins equipped with sensors connected to a data management system. Regular rubbish bins were fitted with sensors, and the study outlines the design, implementation, and validation techniques employed. Data gathering and evaluation were conducted to assess the feasibility of the final device. Waste container data, collected and processed by purpose-designed prototypes, were transmitted to a central monitoring server through a GPRS connection. The data management and monitoring modules were integrated into an existing application used by local governments. A field test program was carried out in the Pudong area, evaluating the real-time data flow from network nodes (containers) and optimization functions, including collection vehicle routing and scheduling. Significant results were obtained in waste weight and volume estimations, which were subsequently utilized for optimizing collection truck routing and evaluating material density.

Accurate and efficient treatment of domestic waste is a crucial aspect of urban management, as it directly impacts long-term societal growth. Previous research on waste picture classification has predominantly focused on single-category waste recognition, necessitating the fulfillment of real-world waste categorization requirements. This study presents the development of a YOLO-WASTE multi-label waste classification model based on transfer learning. The model demonstrates the ability to recognize and classify various waste types rapidly. To enhance the learning efficiency of the model, a multi-label waste image dataset is created, with each image containing multiple waste items or waste categories. The YOLO-WASTE model achieves an mAP (mean average precision) value of 92.23 percent, and the average time for picture detection is 0.424 seconds, outperforming other image classification methods. The proposed YOLO-WASTE model provides valuable insights into challenging waste identification tasks, contributing to the advancement of trash management for sustainable urban development (Zhang, Q. et al., 2022).

Table 3. Summary of Image Processing Related Studies

Author	Year	Title	Findings	Relationship to the study
Pajankar, A.	2017	Raspberry Pi Image Processing Programming	This book focuses on the different uses of the microprocessor Raspberry Pi in image processing. The author teaches the reader to acquire images with various types of the camera sensor and introduces us to Pillow, Tkinter, SciPy toolkit, NumPy ndarrays, and matplotlib.	The book that Pajankar made is our guide in learning how to use Raspberry Pi in image processing, and we want to apply this knowledge in creating a robot that can distinguish water trash from obstacles.
Tang et al.	2016	Robust Image Hashing with Ring Partition and Invariant Vector Distance	According to the researchers, the resilient Image Hashing has strong rotation robustness and achieves optimal discriminating. Its primary contribution is the fact that vector distance is invariant to commonly used digital operations, which allows us to compress our hash while maintaining it discriminatively.	We want to learn image hashing to maximize the ability of our robot in solid water waste identification.
Youme et al.	2021	Deep Learning and Remote Sensing: Detection of Dumping Waste Using UAV	The researchers created a model that can detect garbage deposits. The model was also built using a number of valuable technologies, including remote sensing and CNN, a Deep Learning technology for object detection. Because of its abstraction and sophistication, the researchers conclude that CNN is a good architecture for image processing and object detection.	The paper contains information on deep learning, namely Convolutional Neural Network (CNN), which we will employ for image processing techniques for the detection and classification of recyclables into proper categories.

Rovetta et al.	2009	Early Detection and Evaluation of Waste Through Sensorized Containers for A Collection Monitoring Application	The researchers concluded that the project's third key goal, the classification of collected garbage, has resulted in a high level of accuracy in waste volume and density estimation. Future research and development will focus on trash classification using sensor fusion approaches.	Their study presented a container for garbage collecting that would monitor waste volume and density estimation, which contains a feature of our prototype, which would use sensors to monitor the plastic waste collected.
Zhang et al.	2022	A multi-label waste detection model based on Transfer learning	The YOLO-WASTE model's mAP value is 92.23 percent, and the average time to detect a picture is 0.424 seconds, indicating that its classification performance is substantially better than that of other image classification methods. The proposed YOLO-WASTE model can help promote efficient trash management for sustainable urban development by providing fresh insights into difficult waste identification.	The21ommunich paper of Zhang et. Al. revealed important details on the usage of the YOLO algorithm for garbage detection and classification. This research will assist us in improving our image categorization approach, which will primarily focus on plastic waste. Their model also encourages efficient waste management and urban development, which is something we hope to achieve with our research.

2.4 YOLO Algorithm

Jiang et al. (2021) conducted a comprehensive review of the You Only Look Once (YOLO) algorithm, which has gained significant popularity and widespread usage for object detection tasks. The study provides an overview of the original YOLO algorithm introduced by Redmon et al. in 2015, as well as subsequent versions and variations, including YOLO V2, YOLO V3, YOLO V4, YOLO V5, and YOLO-LITE.

Roy et al. (2021) proposed a modified model based on the improved You Only Look Once version 4 (YOLOv4) algorithm. The modifications in the network architecture aim to enhance detection accuracy and speed. The inclusion of DenseNet in the backbone optimizes feature transfer and reuse. In contrast, introducing two new residual blocks in the backbone and neck improves feature extraction and reduces computational costs. The Spatial Pyramid Pooling (SPP) enhances the receptive field, and a modified Path Aggregation Network (PANet) preserves fine-grained localized information and improves feature fusion. Using the Hard-Swish function as the primary activation further enhances the model's accuracy by facilitating better nonlinear feature extraction.

In aquaculture, real-time detection and monitoring of feed pellet consumption is crucial for developing efficient feeding methods that minimize waste and contamination. Hu et al. (2021) proposed an improved version of the You Only Look Once (YOLO)-V4 network for detecting uneaten feed pellets in aquaculture. The modifications include

1. replacing the feature map responsible for large-scale information,
2. utilizing a finer-grained YOLO feature map, and
3. Modifying the residual connection mode with the inclusion of DenseNet in CSPDarknets to enhance feature reuse and network efficiency.

The network complexity is reduced through a de-redundancy operation while maintaining detection accuracy. Experiments conducted in a real fish farm demonstrated better detection accuracy than the original YOLO-V4 network, with an average precision increase of 27.21 percent from 65.40 percent to 92.61 percent (at an intersection over a union of 0.5). Moreover, computation requirements were reduced by approximately 30 percent. Consequently, the upgraded YOLO-V4 network successfully detects underwater feed pellets and can be applied in real-world aquaculture settings.

The COVID-19 pandemic has significantly increased household waste due to food delivery packaging, exacerbating its environmental impact. In their study, Liu et al. focused on designing and implementing Faster-RCNN, SSD, and YoloV4 models for detecting and classifying municipal waste. The dataset specifically searched for two types of waste: plastic and aluminum cans, which constitute a substantial portion of household waste. The study compared the loss value accuracy for waste classification detection among the three models: Faster-RCNN, SSD, and YoloV4.

The YOLOv4 model was selected during the research for one-stage detection, while the Faster R-CNN model is employed for two-stage detection. The Faster-RCNN model integrates the entire target detection process with a CNN structure, distinct from existing two-stage methods. It utilizes the Select Search method to generate detection frames, significantly enhancing generation speed through Region Proposal Networks (RPN). The SSD model exhibited the best results, achieving an average precision of 99.99 percent, 97.65 percent for plastics, and an mAP value of 99.78 percent (Liu et al., S.-H, 2021).

The sorting of plastic waste involves categorizing it into various types. Padalkar et al. (2021) conducted a study on object detection and scaling models for plastic trash sorting, utilizing the WaDaBa dataset to detect four different types of plastics. The Scaled-Yolov4 and EfficientDet, object detection and scaling models, were compared in this research. The results showed that Scaled-Yolov4-CSP outperformed the state-of-the-art Colour-Histogram-based Canny-Edge-Gaussian Filter by 21 percent. The study aimed to assess the accuracy and precision of object detection and scaling models in sorting plastic waste. Scaled-Yolov4-CSP outperformed Scaled-Yolov4-p7, EfficientDet-d0, and EfficientDet-d7x in accuracy, mAP, training time, inference time, and model size (420MB). Feature pyramid networks surpassed the state-of-the-art Canny edge Gaussian filter in object detection and scaling by 21 percent. This study demonstrates the practical application of the Object Detection and Scaling Model (ODSM) Scaled-Yolov4-CSP in plastic waste sorting (Padalkar et al., 2021).

In their study, Wahyutama et al. (2022) developed an intelligent trash bin that utilizes a camera and the YOLO algorithm for real-time object detection on a Raspberry Pi. The system is designed to segregate and collect recyclable waste by rotating the trash bin lid, revealing the appropriate compartment for disposal. The YOLO model achieved an accuracy of 91 percent in an ideal computing environment and 75 percent when deployed on the Raspberry Pi. The monitoring of the trash bin is facilitated by Internet of Things (IoT) hardware, including a GPS for tracking coordinates and an ultrasonic sensing device for measuring capacity, both controlled by an Arduino Uno. The capacity and position information collected by the Arduino Uno is sent to the ESP8266 Wi-Fi module and uploaded to the Firebase Database. A mobile application created using MIT App Inventor retrieves the data from the Firebase Database and displays the real-time trash bin capacity as a monitoring feature. The YOLOv4-tiny model, trained with 1000 images over 9000 epochs, achieved 75 percent accuracy. The accuracy measurement increased to 91 percent when the full-size YOLOv4 model was trained with fewer epochs on a full-size personal computer. Despite a slight decrease in accuracy when deployed on the Raspberry Pi, the system demonstrated satisfactory performance, achieving over 90 percent accuracy in real-life scenarios.

In another study by Saputra, the issue of waste sorting is addressed using deep learning algorithms for object recognition, specifically YOLOv4 and YOLOv4-tiny with Darknet-53. The dataset comprised of 3870 waste images classified into glass, metal, paper, and plastic categories. Both models were tested with images, videos, and webcams as inputs. Experiments were conducted with different hyperparameters, including subdivision values and mosaic data augmentation, for the YOLOv4-tiny model. The results showed that YOLOv4 outperformed YOLOv4-tiny in object detection, while YOLOv4-tiny exhibited faster computation speed. The best results achieved by the YOLOv4 model were mAP of 89.59 percent, precision of 0.76 percent, recall of 0.90 percent, F1-score of 0.82 percent, and Average IoU of 64.01 percent. The best results for YOLOv4-tiny were mAP of 81.84 percent, precision of 0.59 percent, recall of 0.83 percent, F1-score of 0.69 percent, and Average IoU of 48.35 percent. The study also demonstrated that smaller subdivision values and mosaic data augmentation yielded better results.

In a study by Yang, machine vision and deep learning were explored to identify and categorize recyclable goods. Traditional machine learning algorithms such as SIFT and SVM could only identify one type of item in a photograph. In contrast, deep learning allows for detecting multiple elements based on rich feature information. The yolov4 method was analyzed in this study for categorizing various types of goods in a restaurant dataset, and it yielded satisfactory experimental results compared to the other techniques. The accuracy of the yolov4 algorithm on the test set was 77.78 percent, which was 15 percent higher than the SSD algorithm, and it achieved a faster detection speed of 20 frames per second compared to the fast RCNN technique. These findings highlight the suitability of the yolov4 algorithm for complex scenarios.

Table 4. Summary of YOLO Algorithm Related Studies

Author	Year	Title	Findings	Relationship to the study
Jiang et al.	2021	A Review of Yolo Algorithm Developments	The researchers provided a brief overview of the You Only Look Once (YOLO) algorithms and their subsequent advanced versions, differences, and similarities. This paper adds to the body of YOLO and other object detection literature. This paper could concentrate on comparing implementations, such as scenario analysis, for future research.	The study of Jiang et al. has provided insightful information about the versions of the You Only Look Once (YOLO) algorithm and between YOLO and Convolutional Neural Networks (CNN), which we will be using for the real-time object detection that detects and classifies recyclables into proper categories.
Roy et al.	2021	A Fast Accurate Fine-grain Object Detection Model Based on YOLOv4 Deep Neural Network	The researchers provided a framework for high-performance real-time fine-grain object detection. The suggested model is built on an improved version of the You Only Look Once (YOLOv4) algorithm. The researchers propose several modifications to optimize both detection accuracy and speed of the real-time object detection model based on the YOLOv4, which are then validated in various complex detection tasks in noisy environments using traditional performance measures for detecting objects.	The study of Roy et al. has introduced an algorithm for object detection, a YOLOv4 algorithm modified to optimize detection speed and accuracy. The YOLOv4 model performance is verified by detecting objects under various challenging environments. This research can help us to provide an efficient method for accurate fine-grain detection of different object classes in complex scenarios like waste identification.
Hu et al.	2021	Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network	The researchers present an enhanced YOLO-V4 network for the detection of uneaten feed pellets in aquaculture. The proposed model outperforms the original YOLO-V4 network, achieving a significantly higher detection accuracy and average precision. Specifically, the average precision improves from	The study by Hu et al. has proposed a YOLOv4 algorithm that can effectively detect underwater feed pellets and can be used in an actual aquaculture environment. As a result, this research will aid us in the optimization design that was carried out to improve the detection accuracy of the YOLOv4 algorithm.

			65.40 percent to 92.61 percent when considering an intersection over union of 0.5, representing a notable increase of 27.21 percent. Additionally, the computational load is reduced by approximately 30 percent. These findings demonstrate that the improved YOLO-V4 network is effective in detecting underwater feed pellets and has practical applications in real-world aquaculture settings.	
Liu et al.	2021	Municipal waste classification system design based on Faster-RCNN and YoloV4 mixed model 1	The researchers created and implemented YOLOv4 models for detecting and classifying municipal waste. The YOLOv4 algorithm was chosen in one stage for a comparative analysis of the proposed models. Plastics have a precision average of 97.65% and a mAP value of 99.78%.	Liu et al. used a data set that investigates two types of plastics, which account for a large proportion of household waste and the different types of aluminum cans. Their data set is relevant to our goal of collecting waste, specifically plastics and microplastics, which the robot will collect.
Padalkar et al.	2021	An Object Detection and Scaling Model for Plastic Waste Sorting	The researcher's goal is to evaluate the accuracy and precision of the Object Detection and Scaling Model in sorting plastic waste. This study demonstrates how the ODSM Scaled-Yolov4-CSP can be used to sort plastic waste.	Padalkar et al. created and implemented a plastic waste sorting architecture in their research. This study can be used as a reference as their WaDaBa dataset was passed to object detection and scaling models to create an inference to classify and detect plastic object types.
Saputra, A.P.	2021	Waste Object Detection and Classification using Deep Learning Algorithm: YOLOv4 and YOLOv4-tiny	The researchers proposed a deep learning algorithm for object detection using YOLOv4 and YOLOv4-tiny with Darknet-53. The dataset includes 3870 waste images divided into four categories: glass, metal, paper, and plastic. Each model uses three different	Saputra, A.P creates a waste classification model and real-time image detection that can display the bounding box and prediction probability of objects in each image. This research will guide our study in the YOLOv4 algorithm that we will be using to object detection through image input, video

			inputs during the testing stage: images, videos, and webcams. Experiments with hyperparameters on subdivision values and mosaic data augmentation were also conducted in the YOLOv4-tiny model. This study also shows that models with smaller subdivision values and that use a mosaic perform best.	input, and real-time using a Raspberry Pi with a webcam that will classify the information.
Yang, Y.	(n.d.)	Waste Classification Based on Yolov4	The researchers proposed a deep learning-based method for classifying restaurant recycling. The experimental platform is built with yolov4, SSD, and fast RCNN. According to the experimental results, yolov4 has the best detection effect, with a detection accuracy of 77.78 percent and a detection speed of 39.3 FPS.	Yang, Y. proposed a classification method for restaurant recycling based on deep learning. Yolov4, SSD, and fast RCNN were used to construct experimental platforms. With a detection accuracy of 77.78 percent and a detection speed of 39.3 FPS, yolov4 has the best detection effect, according to the experimental results. The YOLOv4 algorithm has the best detection effect among the other tested algorithms, so the detection accuracy of this research will serve as our basis for determining the percentage rate.

2.5 Boat Motor

In their report, Dandabathula et al. (2021) discuss Aquayaan, an affordable robotic watercraft with precise maneuverability and an intuitive user interface. The system utilizes the L298N Motor Driver Module, a high-power motor driver module suitable for driving DC and stepper motors. This module supplies up to 5 volts to connected circuits, making it ideal for controlling motors in IoT-based applications, such as microcontrollers, switches, and relays. Aquayaan can be enhanced with robotic arms and sensors to collect water samples and monitor pollution levels in inland water bodies. By eliminating direct human contact with potentially hazardous water conditions, the system alleviates the physical and labor-intensive nature of environmental monitoring. It possesses the ability to navigate long distances on water without losing connection, and its geographic tracking can be visualized in a WebGIS environment. The watercraft can be operated remotely through an internet connection. The report details the design aspects of the hull and frame, the configuration of the rudder and propeller, the implementation of electronics and motors,29ommunication interfaces, and software integration. Aquayaan's cost-effective approach is aimed at facilitating various field surveys in inland water bodies, including rivers, reservoirs, lakes, backwaters, tanks, and ponds, among others, enabling the collection of water samples and measurement of various parameters using additional sensors.

The Aqua smart boat incorporates an automatic fishing technique and utilizes DTMF technology, enabling wireless mobilization and control. Remote control fishing, which involves a remote-control boat, is employed in this project. The DC motors in the Aqua smart boat are driven by the L293D driver, allowing the boat to move in various directions. Brushed DC motors are utilized for their advantages, including low initial cost, high reliability, and simple speed control. The smart aqua boat can be conveniently operated by users using their everyday mobile phones, particularly in lakes or ponds suitable for fishing. Equipped with a motor connected to the fishing line, the boat features a sensor covered by bait at the end of the string. When the boat enters the water body, and a fish takes the bait, the sensor is triggered, drawing the fish into the boat (Pearson T. et al., 2019).

Table 5. Summary of Boat Motor Related Studies

Author	Year	Title	Findings	Relationship to the study
Dandabathula, G et. al	2010	Design and Development of Aquayaan: An IoT based Robotic Boat for Inland Water Surveys	The researchers present the design and development of Aquayaan, a prototype robotic boat capable of mounting various sensor(s) and performing field surveys in inland waters. Aquayaan can be controlled remotely via an Internet connection, allowing it to travel long distances in water without losing contact, and its geo-track can be tracked in the WebGIS environment.	L298N is used to drive the DC motor, and the DC motor, in turn, drives the propeller module, which allows the boat to move forward and backward. This Aquayaan study contains information that can be used to benefit our study because they can communicate via the Internet and can maneuver even when there is no line of sight.
Satheesh, J et al.	2020	Wireless Communication based Water Surface Cleaning Boat	The High Torque Mini 12V, DC Gear Motor of 600 rpm was used in the study by Satheesh J et al. for the conveyor belt. The conveyor belt speed was chosen to be average, thus this motor. In this case, an L298N-based Motor Driver Module is used to drive two DC motors, which is common in high-power motor scenarios. It consists of an H-bridge that aids in the forward and reverse movement of the boat by allowing easy and separate control of two motors of up to 2A each in both directions.	In this study, the potential of using DC motors on the conveyor system was considered because the researchers used an advanced conveyor system and conveyor materials to increase the efficiency of garbage collection. This project made use of three DC motors. One is used to control the conveyor's motion, while the other two are used to control the boat's motion. The weight of the motor (0.1 Kg) was also kept low enough so that it did not add to the boat's overall weight.
Pearson, T. et al.	2019	Design and Implementation of Smart Boat Technique	The aqua smart boat can be operated in a lake or pond that is ideal for fishing using a mobile phone that the users carry with them daily. The boat has a motor linked to the fishing string, with a sensor at the end that is covered by bait (fish food). The sensor activates when the boat enters the lake/pond and a fish catches the bait, and the fish is brought into the boat.	In this project, two DC motors are connected to the aqua smart boat's propellers, which are used to move the boat in various directions. The benefits of a brushed DC motor include low initial cost, high reliability, and simple speed control. The L293D driver is used for DC motors.

Satheesh et al. conducted research to address the adverse effects of floating debris, notably the pollution caused by uncollected wastes like plastics and plant weeds. Their project focuses on designing a waste collection boat capable of scavenging water bodies and removing floating waste. The conveyor belt of the boat is powered by a High Torque Mini 12V DC Gear Motor with a speed of 600 rpm. The motor chosen for the conveyor belt operates at an average speed to effectively handle waste collection. An L298N-based Motor Driver Module is employed for driving the two DC motors, which is commonly used in high-power motor applications. This module incorporates an H-bridge that facilitates the forward and reverse movement of the boat, providing easy and independent control of two motors, each capable of handling up to 2A of current in both directions.

CHAPTER 3
METHODOLOGY

The researchers will focus on developing and programming the automated solid water waste vehicle. The researchers will first gather the materials needed in the development of the study, next is to calibrate the sensors and modules to be used, implement YOLOv4 for the image processing of the vehicle and lastly, create the webpage for the GPS tracking system.

3.1 Research Design

3.1.1 Developmental Research

Developmental Research is a research design used by the proponents since the study focuses on developing innovation of a system or technology that aims to decrease the water pollution in seas utilizing microcontrollers, sensors, and machine learning technology. Moreover, it is a way to establish new procedures, techniques, and tools based on a methodological problem analysis. Thus, the researchers will employ developmental Research.

The study's structure is represented by the Input-Process-Output (IPO) diagram, providing an overview of the research. The development of the system involves the integration of both software and hardware components. The construction of the robot includes the utilization of a raspberry pi microcontroller and programming it using Python programming language. Image processing will be facilitated by an A4tech HD camera, while the ultrasonic sensor will measure the distance to the detected waste. Additional sensors will enable the DC motor to move the vehicle upon waste detection through the conveyor belt, directing the waste to the garbage storage at the back. Once the sensors and motors are attached, the aquatic vehicle will be capable of collecting plastic waste and storing it in a garbage bin. The GPS tracker will be employed to track the robot's location, and the GSM module will notify the user when the garbage bin reaches its full capacity, prompting the user to retrieve the trash.

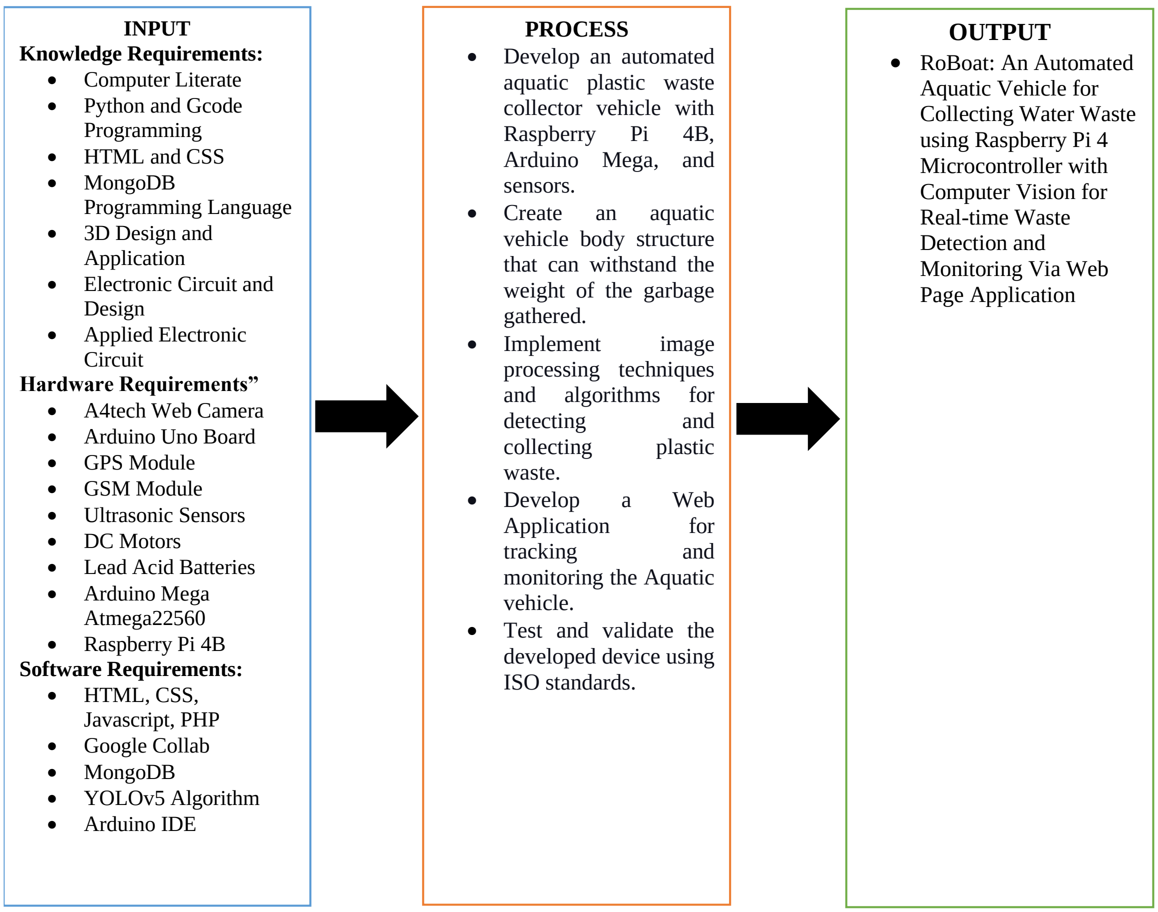


Figure 1. Input-Process-Output of the Project

3.2 Research Process Flow

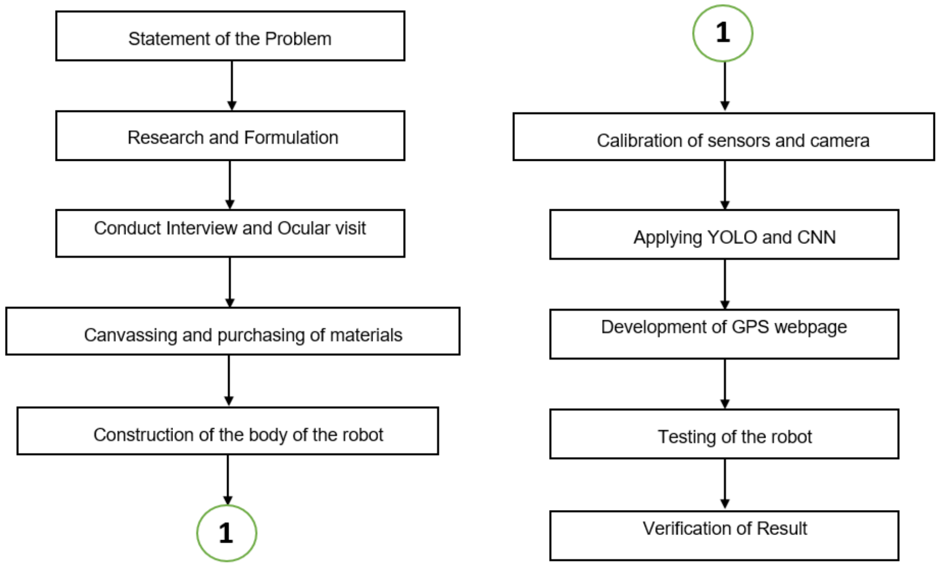


Figure 2. Research Process Flow

This section presents the major steps that the research study will take to complete the researcher's thesis. It also serves as a foundation for indicating the study's project milestones.

3.3 Development of an Automated Solid Water Waste Collector Vehicle

This section discusses the design and building of the Automated Solid Water Waste Collector Vehicle.

3.3.1 Materials and Equipment

The aquatic vehicle will be composed of stainless steel for the structure, DC motors and four air blowers for the movements, a garbage bin to store the waste collected, and three Lead-acid batteries responsible for the robot's power source. For the automation of the vehicle, a Raspberry Pi 4B microcontroller will be used along with the PK-910H 1080p Full-HD Webcam for the image processing algorithm. The researcher utilizes an Arduino Mega to control the boat's movement, along with sensors to regulate the automation. These sensors include a GPS module responsible for locating and retrieving, a GSM module for notifying the user that the garbage storage has already reached its maximum capacity, a weight sensor to detect the total trash collected, and relays. An Arduino Nano is also connected to the Arduino Mega for the controller of the boat.

Table 6. Automated Aquatic Vehicle Components

Material	Quantity	Item and Specifications
DC Motor	4	**Model Name: XD37GB-520YSY** • Voltage: 12 V • Rated speed: 45 RPM at 12 V • Current rating: 0.68 A • Rated torque: 9 kg.cm • Power: 5W • Weight: 210 g
Battery	3	**Model Name: Lead-Acid** • Voltage: 12.8V • Ampere: 65 AH/20HR
Raspberry Pi	1	**Model Name: Raspberry Pi 4B** • 4GB LPDDR4-3200 SDRAM
Camera	1	**Model Name: PK-910H 1080p Full-HD Webcam** • Image Capture (16:9 W): 2.0 MP, 3 MP, 6 MP, 15 MP • Video Capture (16:9 W): 360p, 480p, 720p, 1080p • Frame Rate (max): 1080p @ 30fps
SD Card	1	• Digital Storage Capacity: 32 GB • Flash memory type: SD • Memory Storage Capacity: 4 GB
GSM Module	1	**Model Name: SIM900A GSM module** • Power Input: 3.4V to 4.5V • Operating Frequency: EGSM900 and DCS1800 • Transmitting Power Range: 2V for EGSM900 and 1W for DCS1800
GPS Module	1	**Model Name: Neo6Mv2**

		• Operating voltage: 3.3V to 5VDC. • Dimension: 36 x 26 mm. • Weight: 22g
Ultrasonic Sensor	1	**Model Name: HC-SR04** • Power Supply: DC 5V • Ranging Distance: 2cm – 400cm/4m • Dimension: 45mm x 20mm x 15mm
Relay	4	**Model Name: Relay Module SPDT** • 250VAC 10A, 28VDC 10A • Indicator LEDs • 12V control voltage • Module size: 55mm x 16mm
Conveyor Belt	1	• Material: Stainless steel • Belt Width: Customized • Belt Thickness: 8.5mm to 9mm ,10.5mm, 12mm • Tensile Strength: 8.5mm to 9mm ,10.5mm ,400/4 (12mm) 630/4 • Size: Customized • Features: Customized • Number Of Plies: 4 Ply • Rubber Material Grade: M24 • Brand: Continental Belting
Marine Air Blower	4	• Voltage: 12V DC, Flow capacity: 270 CFM (Cubic Feet per minute) • Constructed with white ABS plastic and includes molded mounting brackets.

Table 7. Waste Storage

Material	Quantity	Item
Weight Sensor	1	• Capacity - 40-50kg
Garbage Bin	1	• Square shape Stainless steel build

3.3.2 Sensor Calibration

Sensors are electronic devices that detect changes in their operating environment. Undesirable output values can arise due to unexpected alterations in the working conditions of sensors. Sensor calibration plays a crucial role in enhancing sensor performance and accuracy. The researchers will calibrate the waste-detection sensors to ensure precise readings of various parameters associated with waste detection. By calibrating the sensors, the proponents aim to mitigate any discrepancies and ensure reliable and accurate data collection in their study.

3.3.3 Designing the Aquatic Vehicle

The aquatic vehicle has a modern, slick design and is scalable, allowing it to clean larger and smaller bodies of water effectively. The RoBoat's body consists of stainless-steel sheets that can keep all the equipment and materials intact and secure. Due to its metal body, the automated aquatic waste collector is designed to handle waves in different bodies of water.

The RoBoat consists of four air blowers, which are the primary source of movement for the automated aquatic waste collector. The 12V Rechargeable Lead acid battery is the most efficient and compatible power source for an automatic aquatic waste vehicle.

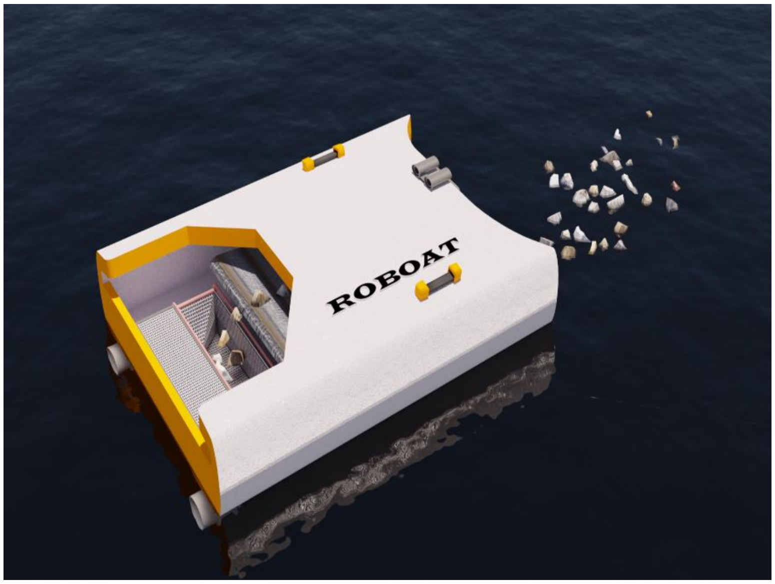


Figure 3. 3D Model of the RoBoat

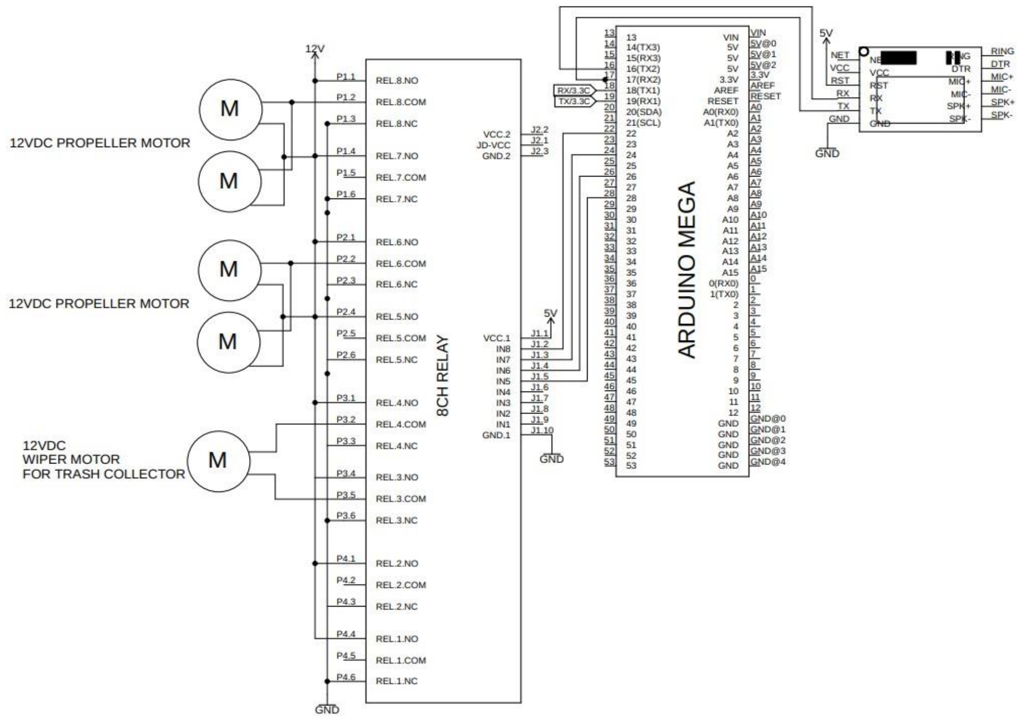


Figure 4. Schematic Diagram

The schematic diagram above illustrates the connection of various components for the movement of the boat. It showcases the interplay between lead-acid batteries, marine-grade air blowers, Arduino Mega microcontroller, and relays. The batteries provide the necessary power supply, while the four air blowers, controlled by the Arduino Mega, are for the smooth movement of the boat. The relays act as switches, facilitating the coordinated movement of the boat by controlling the propulsion system, steering mechanism, and other relevant functionalities. This comprehensive and well-designed connection scheme ensures the seamless operation and maneuverability of the boat, making it a reliable and efficient system for its intended purpose.

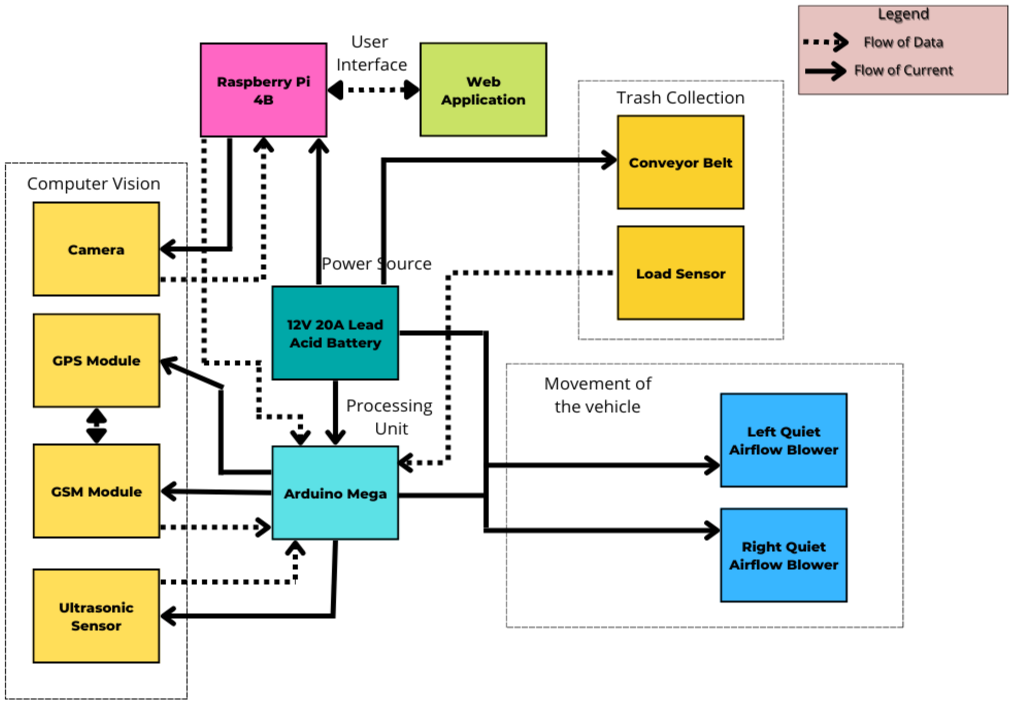


Figure 5. Block Diagram

The design of the system must be streamlined, lightweight, and durable with improved overall capabilities in order to be deployed with ease on any body of water. The system is an automated aquatic vehicle for garbage collection. Stainless steel was selected for the vehicle's body while keeping all factors in mind. The vehicle is propelled by marine grade air blowers and DC motors. The vehicle also has a roughened conveyor belt that was chosen for its ability to repel water. The robot's whole body, including the conveyor belt, is built of materials that are water-resistant. To facilitate the drainage of excess water from the conveyor system, the vehicle is designed with specialized grooves that direct the water back into the water body. This feature ensures that the weight of the garbage inside the container is not increased by water accumulation. To counteract the effects of water currents and prevent the motor from involuntary rotation, high-torque motors are employed.

The automation of the vehicle will be achieved through the utilization of a raspberry pi microcontroller and an A4tech Webcam. These components will work in tandem with an image processing method to automate the vehicle's operations and enhance its functionality.

The three 12V Lead Acid Battery is the most effective, long-lasting, and appropriate power source for the automated aquatic vehicle. Additionally, it works with many different models and doesn't discharge when not in use.

3.4 Implementation of Image Processing Algorithms for Solid Water Waste Detection and Collection

This section discusses various methodologies for developing appropriate Image processing Algorithms and the respective software that will be used in programming the robot.

3.4.1 Algorithm Testing

The proponents have decided to utilize the YOLOv5 Algorithm for real-time image detection of plastic waste. YOLOv5 was chosen due to its advanced features and improved performance in object detection tasks. YOLOv5 is a popular single-stage object detector that outperforms its predecessors regarding speed and accuracy. Compared to YOLOv4, YOLOv5 offers several advancements and optimizations. It introduces streamlined architecture and incorporates new techniques to enhance object detection. YOLOv5 utilizes a different backbone, neck, and head configuration compared to YOLOv4. The backbone of YOLOv5 employs a modified version of the EfficientNet architecture, which efficiently balances model size and performance. This backbone architecture allows for more efficient and effective feature extraction from input images.

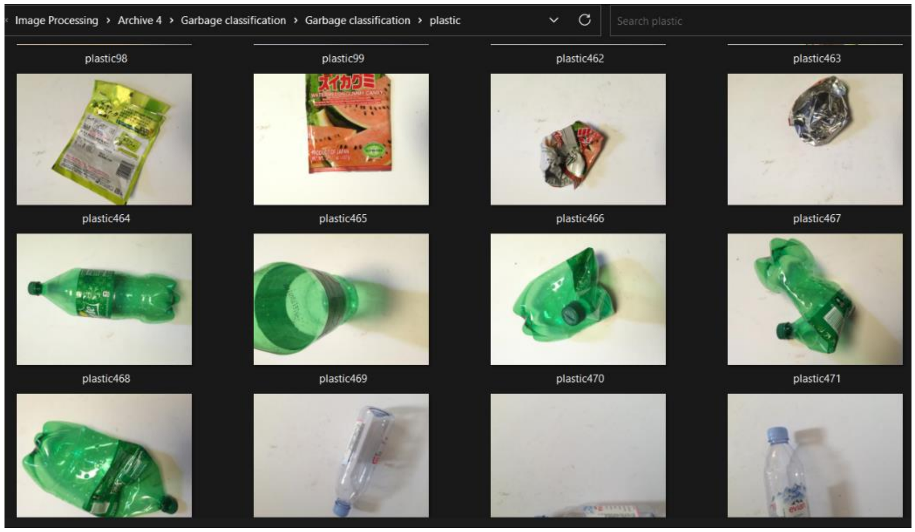


Figure 6. Sample image for Machine learning Models

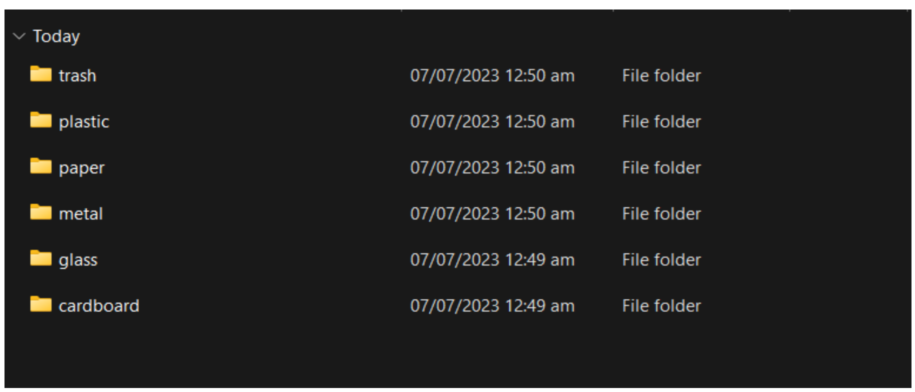


Figure 7**.** Collected Images for Data Set

The researchers diligently searched for relevant datasets on the internet to obtain a comprehensive collection of images for training machine learning models. In their efforts, they successfully amassed a total of 2527 images to be trained from Kaggle, Google Datasets and proponents taking actual pictures. To proceed with training machine learning using YOLOv5 in Python, a series of crucial steps were undertaken. Initially, a meticulous process of labeling the collected images with objects of interest, specifically plastic waste, was conducted. Subsequently, the dataset was divided into distinct subsets, namely the training and validation sets, to facilitate effective model evaluation. Employing a deep learning framework like PyTorch, the YOLOv5 model was implemented. To enhance the diversity of the training data, the images underwent preprocessing techniques, including resizing and augmentation. The model was trained by inputting the labeled images into the YOLOv5 network and optimizing its parameters through backpropagation and gradient descent algorithms. A total of 2527 images were used, 1768 for training, 431 for testing and 328 for validation therefore the dataset distribution is approximately 69.9% training datasets, 17.0% testing datasets, and 13% validation datasets. Throughout the training process, metrics such as loss and accuracy were continuously monitored. Following the completion of training, the model's performance was rigorously assessed by evaluating its predictions on the validation set, enabling the researchers to make necessary adjustments and refine the model as required.

3.4.3 Programming of the Robot

The Aquatic vehicle's programming was accomplished using the Python programming language specifically for the Raspberry Pi, as well as the Arduino IDE for the Arduino Mega microcontroller. To upload the programs onto the microcontrollers, a USB cable will be connected from the laptop to both the Raspberry Pi 4B and Arduino Mega. Once the entire system is installed and powered on, the main processing unit, the Raspberry Pi, executes the program and initiates the ultrasonic sensor for operation.

3.4.3 Device Flowchart

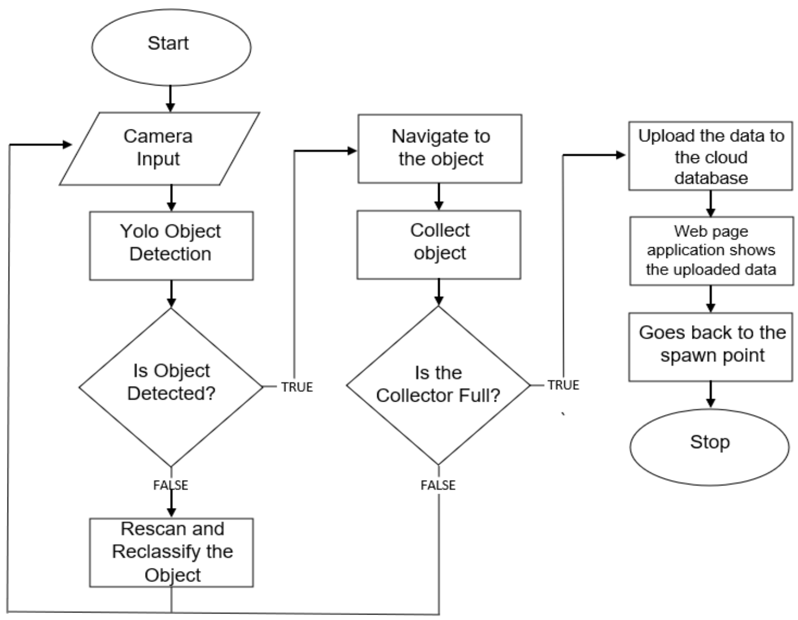


Figure 8. Device Flowchart

The flowchart begins with the webcam retrieving the video image, which is then fed to the Raspberry Pi, where YOLO object detection is used. When the camera sees an object, it triggers YOLO to detect and classify the object. After the object has been classified, the result is sent to Raspberry Pi, where it rotates the motor to a certain angle according to the classification result. If YOLO cannot classify the object or results in an error, it will automatically rescan the object to trigger YOLO again until it can classify it. An ultrasonic sensor was attached to the vehicle to determine its distance from an obstacle. The weight sensor measures the waste, and the data collected by the weight sensor is uploaded to the cloud database. A web application retrieves the uploaded data to monitor the trash bin capacity in real-time.

3.5 Development of Aquatic Vehicle Monitoring and Notification System

This section discusses the various methodologies for developing the website to track and monitor the aquatic vehicle and a notification system that notifies the users when the aquatic vehicle reaches its capacity and back at the spawn point.

3.5.1 Web Development

The study developed a Web Page for monitoring the movement and the location of the Aquatic Vehicle. Google cloud platform manages the infrastructure and APIs required for the application's real-time communication layer. The waste collector or waste storage has a weight sensor connected that records the weight. When the weight hits 50 kg, the robot notifies the user. Global Positioning System, or GPS, is the most used technology for tracking and routinely monitoring vehicles. The goal of the tracking system is to follow and keep track of the aquatic vehicle so that the user may view its current location.

Figure 9. HTML, TailwindCSS, ReactJS and MongoDB Atlas Logos

MongoDB Atlas is a cloud database that manages the complexities of deploying, managing, and healing your cloud deployments.

It is connected to the Raspberry Pi 4B, wherein it sends the collected data to the MongoDB Atlas Cloud Platform, where it shows collected real-time data on the website that also saves and record the data simultaneously. A web interface is accessible via a mobile device or a personal computer for easy monitoring.

To create the webpage, the researchers will use HTML and design using TailwindCSS and ReactJS for the Frontend. It was integrated with Raspberry Pi 4. The HTML was uploaded with the domain roboat-tupm.online, bought, and hosted in Hostinger. In this way, it was efficient as it can be open anywhere as long as there's an internet connection.

3.5.2 Program Flowchart

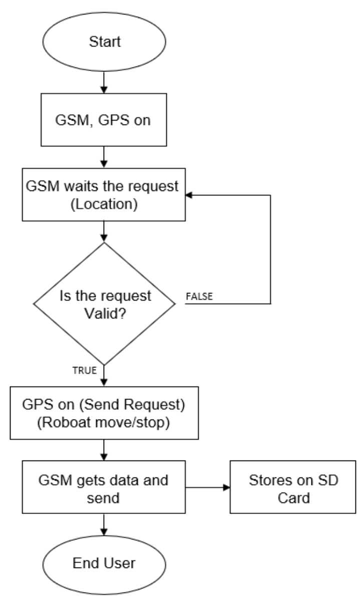


Figure 10. Program flowchart

The data in the Web Page Application will be based on the Raspberry Pi 4 data. The information gathered will be displayed on the Web Page. The user can view the Automated Aquatic Vehicle's current location and track its movement. The Web Page also displays the current weight of the collected garbage and notifies the user if the collector reaches its maximum capacity.

3.6 Validation of User and Experts Acceptance of the Software and Field Testing of the Automated Aquatic Waste Collector Vehicle

This section discusses the prepared methodologies for expert consultation and testing of the prototype.

3.6.1 Data Collection

The researchers will manually collect the data based on how many liters/kg of waste the aquatic vehicle can collect in a day. To assess whether the aquatic vehicle can reach the maximum amount of garbage that is set at 50 kg of waste, the researchers will monitor and continue data gathering for a week.

3.6.2 Prototype Testing

The prototype will be tested initially in a swimming pool to evaluate how well it could detect and collect garbage. To aid in the detection and collecting of the prototype, plastic would be used as a sample object in the pool. The researchers would evaluate and confirm whether the prototype was functioning properly and if there were additional changes needed to its functionality. Following the prototype's initial testing in the pool, the researchers will deploy it in the Navotas fish port and conduct more testing to see if it will perform satisfactorily there.

3.6.3 ISO Standard Validation

To determine the prototype's functionality and application, the proponents will gather feedback on the performance of RoBoat: An Automated Aquatic

Vehicle for Collecting Solid Water Waste using Raspberry Pi 4 Microcontroller with Computer Vision for Real-time Waste Detection and Monitoring Via Web Page Application. The evaluation will be based on many variables in accordance with ISO 25010 to determine the quality of the software and a technical evaluation, as well as ISO 10218 to demonstrate the study's contribution to the development of industrial robots.

3.6.4 Target User and Professional Feedback

The researchers will analyze and interpret the results obtained from the initial testing of the prototype for the autonomous solid water waste collector boat. To evaluate its performance, we will design a questionnaire survey based on the Likert scale. The survey will be administered to the local government units involved in the prototype deployment. The Likert scale will use a 5-point rating system, where a rating of 5 indicates the highest level of satisfaction or effectiveness. This survey will provide valuable feedback from the stakeholders, allowing us to assess the system's performance and make informed decisions for future enhancements.

Table 8. Questionnaire for target users and professionals

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 – 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 – 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 – 5

4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 – 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 – 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 – 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 – 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 – 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 – 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 – 5
Comments and Suggestions:

3.7 Statistical Analysis

This study about An Automated Aquatic Vehicle for Collecting Solid Water Waste used Arithmetic Mean for statistical analysis. The proponents can use the mean to summarize and analyze the responses provided by the users. By incorporating the statistical analysis of calculating the mean, the proponents can quantify and analyze the target user feedback, enabling the proponents to make informed decisions based on the aggregated responses. They designed a questionnaire survey using the Likert scale to evaluate the system's performance. The survey was distributed to the local government units involved or professionals that helped deploy the prototype, aiming for 30 respondents. Using a 5-point rating system on the Likert scale, ranging from 1 to 5, with 5 indicating the highest level of satisfaction or effectiveness, aiming to gather valuable insights into the stakeholders' perceptions and experiences.

Formula:

$$\mu = (\Sigma x) / n$$

Where:

μ represents the mean,

Σx denotes the sum of all the values in the dataset, and

n represents the number of values in the dataset.

3.8 Project Work Plan (Gantt Chart)

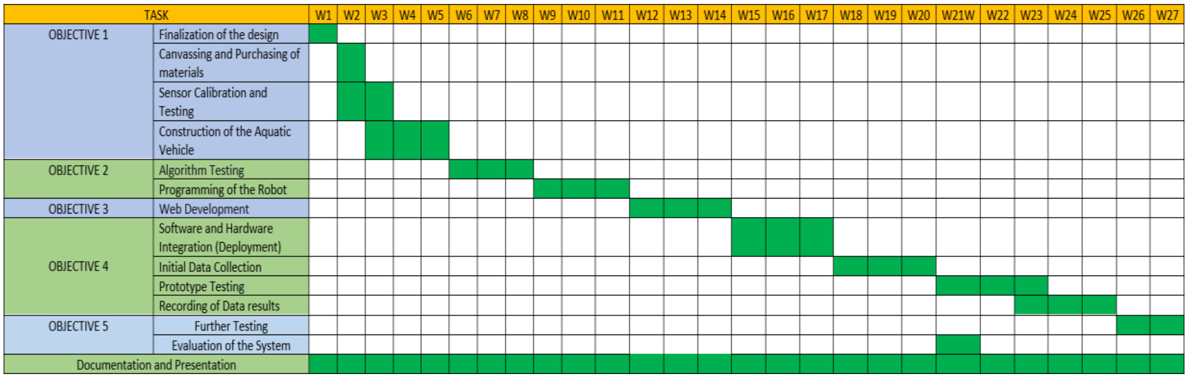


Figure 11. Gantt Chart

The entire project is estimated to be completed within a timeframe of seven months. To ensure effective project management, the researchers have divided the project development process into weekly segments, with each week dedicated to specific tasks that are projected to be finished. The project aims to create an automated solid water waste collector, encompassing five main objectives. The first objective focuses on the boat's design and construction, considering factors such as size, stability, and maneuverability. The second objective centers around image processing for garbage detection. This involves developing algorithms and utilizing appropriate technologies to accurately identify solid waste in the water. The researchers leverage computer vision techniques to enhance the system's ability to detect and differentiate between various types of garbage. The third objective involves web development to enable possible real-time tracking of the boat's progress. By creating a web-based interface, users can remotely monitor the automated waste collector's location, activities, and performance. The fourth objective focuses on the deployment and data gathering phase. Once the system is constructed and functional, the researchers tested its performance in real-world conditions, collecting data on its effectiveness in waste collection, navigation, and durability. This data will be crucial for evaluating the system's efficiency and making any necessary improvements. Lastly, the fifth objective centers on evaluating the system as a whole. The researchers conducted comprehensive assessments to determine the system's overall performance, reliability, and user-friendliness. Feedback from stakeholders and end-users was be gathered and analyzed to gauge the system's success in addressing the problem of water waste collection.

Throughout the seven-month duration, the researchers closely monitored progress, adjust when need, and maintain effective communication to ensure that the project stays on track and achieves its objectives.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter interprets the acquired data and analyzes the results based on the tests that were conducted and performed.

4.1 RoBoat Technical Description

RoBoat is an autonomous aquatic vehicle designed to collect solid waste from water surfaces. The vehicle utilizes a Raspberry Pi 4 microcontroller and computer vision technology to enable real-time waste detection and monitoring through a web-based application. The control of RoBoat's movement is managed by an Arduino Mega microcontroller programmed with C++, while the Raspberry Pi 4, programmed with Python, handles image detection and automation tasks. A dedicated web page serves as the interface for tracking the RoBoat's location, monitoring battery percentage, and displaying the current weight of the collected plastic waste.

The RoBoat employs the YOLOv5 algorithm, a state-of-the-art image processing technique, to identify solid waste present on the water's surface. Once the waste is detected, the RoBoat autonomously moves towards it, activating the conveyor belt mechanism for efficient collection and deposition into the onboard waste storage bin. To enhance its functionality, the RoBoat incorporates various sensors, including a load sensor to measure the weight of collected waste and an ultrasonic sensor to detect nearby objects or obstacles.

The collected waste data from the HX711 module is transmitted to the web page, where users can view the total weight of waste collected by the RoBoat in real-time. Additionally, the web page is equipped with a GSM module to send notifications when the waste storage reaches full capacity. The GPS tracking device embedded within the RoBoat provides precise location information, which is also displayed on the web page. This innovative design and integration of hardware components and software algorithms make the RoBoat an efficient and technologically advanced solution for addressing plastic waste pollution in water bodies.

4.2 Project Structural Organization

4.2.1 Parts of the System

The system consists of Raspberry Pi 4 and Arduino Mega microcontrollers, Relays, Wiper Motor, Battery, Transceiver, Sensors, GSM, GPS and HX711 Modules. The Raspberry Pi 4 serves as the object detection and automation microcontroller. In the meantime, the Arduino Mega is responsible for the RoBoats Movement and Conveyor Belt. The web page, on the other hand, serves as both monitoring and notification for the user.

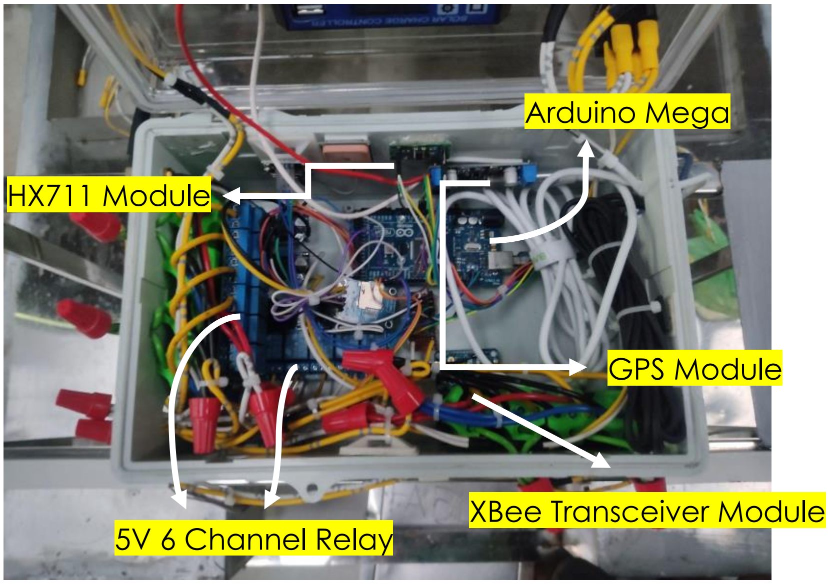


Figure 12. Electronic Components and Wirings

The boat's structure and conveyor belt are made entirely of stainless steel. A wiper motor is in charge of the conveyor belt's rotation, while the air blower is in charge of movement. Both are controlled by the Arduino Mega and are powered by four 12V lead acid batteries. A step-down relay is also used to reduce the voltage to the required 5V for the Arduino Mega. Meanwhile, there is a 30A 12V channel automotive relay for the Air blower's propeller to reduce current consumption.

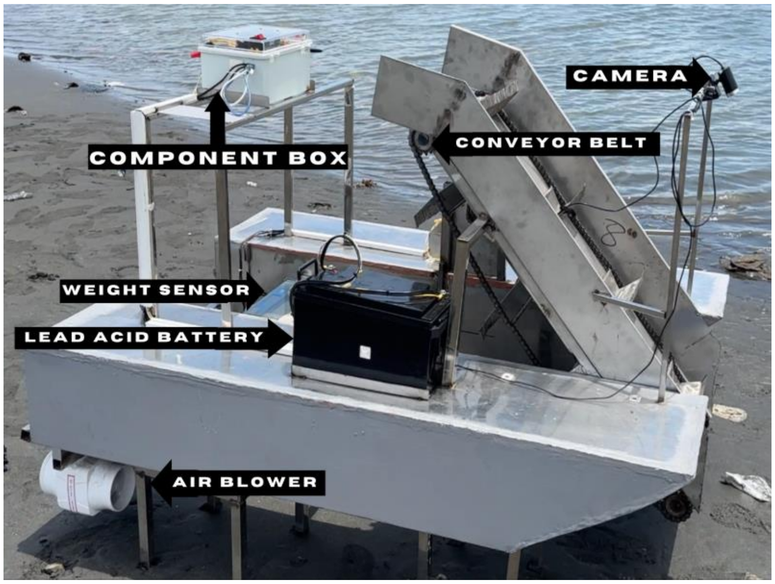


Figure 13. Actual RoBoat Design

Furthermore, a backup manual controller is included as a failsafe measure in case of any automation system failures. The manual controller consists of a rechargeable 3.7V Li-Po battery that powers the system. To ensure compatibility with the RoBoat's requirements, a booster is utilized to increase the output voltage to 5V. Wireless communication between the manual controller and the RoBoat is facilitated by an Xbee S2C transceiver with an antenna. The movement and activation of the conveyor belt are controlled by an Arduino Nano Microcontroller, operated through a joystick for user-friendly control. The inclusion of these sensors and the backup manual controller system enhances the reliability and flexibility of the RoBoat, allowing for effective waste collection even in case of automation system failures or manual intervention requirements.

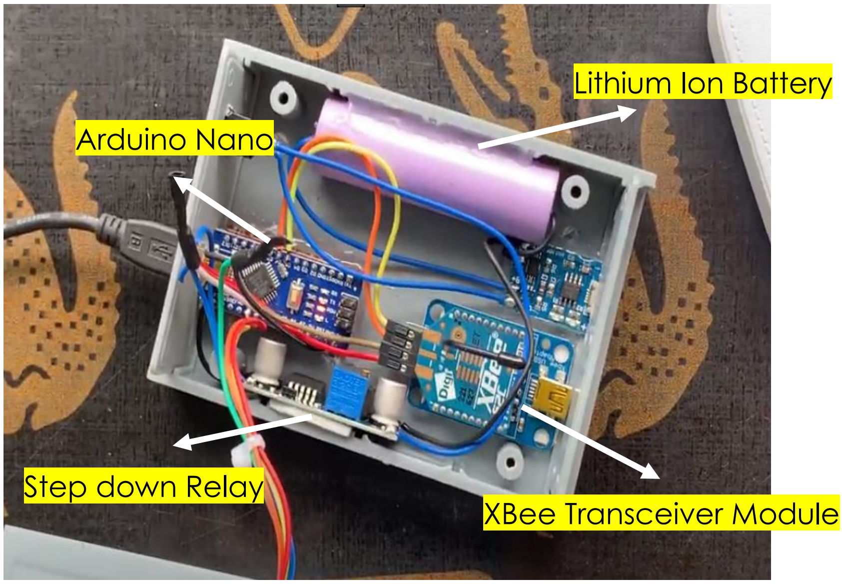


Figure 14. Backup Controller for the RoBoat

4.3 Image Processing Result

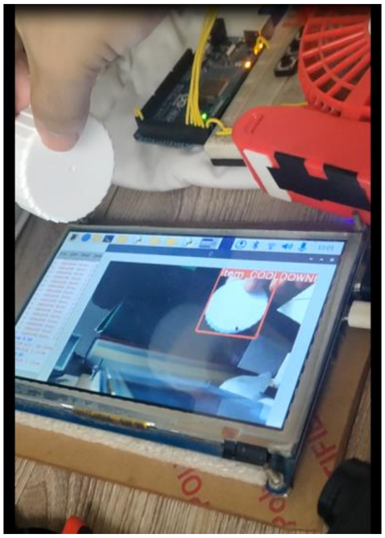


Figure 15. Image Processing Testing

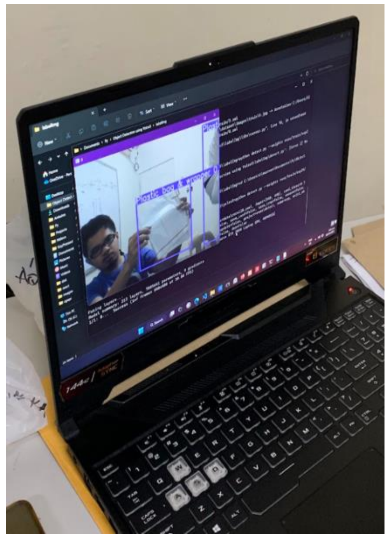


Figure 16. Test Results of Image Processing

The figures shown above demonstrate the test results from the detection of objects like plastic bags and wrappers. The researchers used YOLOv5 to detect objects in an accurate and efficient manner. This innovative technology enables the system to identify and classify various objects of interest, including plastic bags and wrappers accurately and reliably. The test results provide vital insights into the YOLOv5 algorithm's performance and efficacy in detecting and categorizing these specific objects.

4.4 Web Application and Tracking

The RoBoat was successfully able to integrate the Raspberry Pi to the Website. It is shown in the figure below the exact location of the RoBoat which in Haydn Road in Cavite City where the proponents successfully able to integrate it into the website. It is also shown the last Data received from the RoBoat to the mongoDB cloud database, the current time and date, the weight of the total waste collected in kilograms (kg) and if the RoBoat is online and offline.

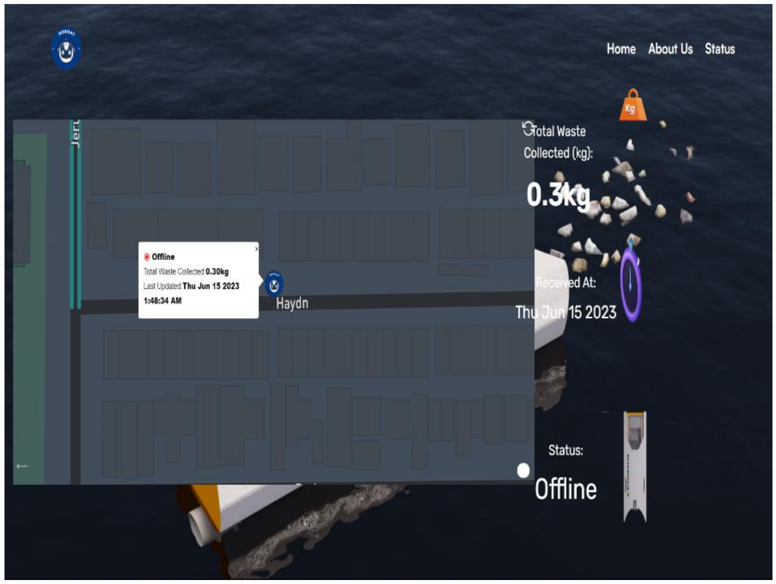


Figure 17. Status page of RoBoat website

4.5 Final Testing

This section of the study presents the analysis and interpretation of the gathered result and initial testing of the prototype:

4.5.1 Deployment Plan

Figure 18. Coast in Cavite City

Initially, the prototype is scheduled for deployment at the Navotas Fish Port in collaboration with the Technological University of the Philippines (TUP) and the Philippine Fisheries Development Authority (PFDA) of the Navotas Fish Port Complex (NFPC). The project proponents are currently handling the necessary paperwork for the partnership and eagerly anticipate deployment once the prototype is completed. Building on the recommendations from the panelists, the proponents also intend to extend the deployment to other areas, such as Manila Bay, the Pasig River, and nearby provinces of Metro Manila renowned for their beaches, including Cavite and Bulacan. This expansion aims to further assess the feasibility of the prototype as an effective solid water waste collector and to promote its adoption within the relevant local government units.

4.5.2 Actual Testing

Testing of the prototype

(A) (B)

Figure 19 A & B. Photos taken on the day of deployment.

The proponents conducted testing of the autonomous solid water waste collector boat to assess its effectiveness in collecting solid waste from water bodies. The team measured and recorded the weight of garbage collected per hour, providing valuable data on the boat's waste collection efficiency. These tests allowed the proponents to evaluate the boat's performance and make necessary adjustments to optimize its waste collection capabilities.

4.5.3 Waste Collected Analysis

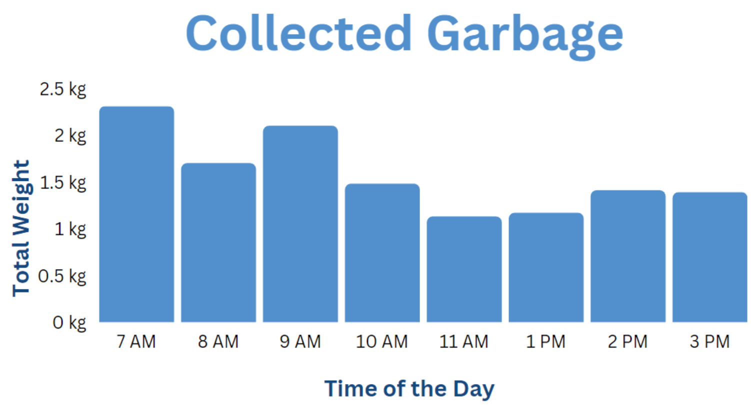


Figure 20. The average waste collected from deployment site per hour.

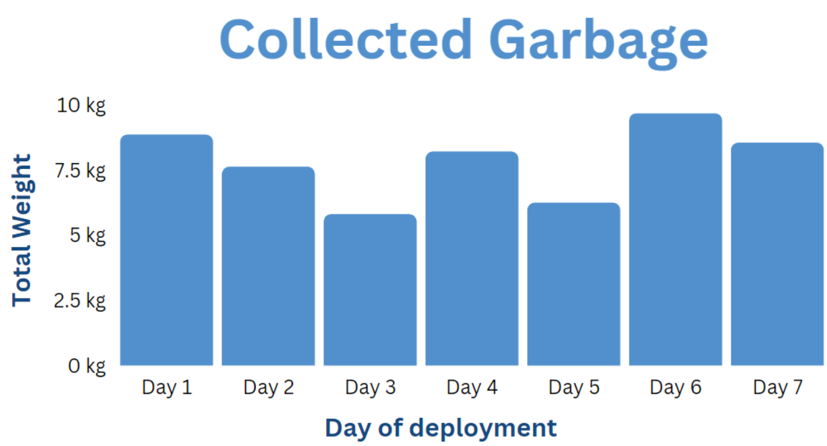


Figure 21. The total weight of collected solid waste per day of deployment.

The researchers conducted a comprehensive deployment in Cavite City for a duration of seven days, focusing on the collection of solid water waste in the sea. This deployment aimed to assess the effectiveness and efficiency of the RoBoat in a real-world marine environment. The researchers closely monitored and recorded the performance of the RoBoat, diligently tracking the amount of solid water waste collected each day. This extensive deployment provided valuable insights into the RoBoat's capabilities, highlighting its ability to effectively contribute to the mitigation of plastic pollution in marine ecosystems.

The analysis of the collected data reveals an interesting observation in the graph, where the amount of garbage collected by the RoBoat appears to be significantly higher during the morning hours. One possible explanation for this pattern could be the influence of high tide during the morning. High tides often bring in an increased amount of debris and waste materials onto the water's surface, resulting in a higher concentration of garbage available for collection by the RoBoat. The higher collection rate during the morning aligns with the natural tidal patterns, indicating that the RoBoat's operations effectively target periods of increased waste accumulation.

4.5.4 Descriptive Statistics of Questionnaire Results

Table 9. Descriptive Statistics

	Valid	Missing	Mean	Std. Deviation	Minimum	Maximum
Question 1	30	0	4.133	0.819	2.000	5.000
Question 2	30	0	4.533	0.507	4.000	5.000
Question 3	30	0	4.600	0.498	4.000	5.000
Question 4	30	0	4.633	0.556	3.000	5.000
Question 5	30	0	4.567	0.626	3.000	5.000
Question 6	30	0	4.567	0.626	3.000	5.000
Question 7	30	0	4.700	0.535	3.000	5.000
Question 8	30	0	4.633	0.615	3.000	5.000
Question 9	30	0	4.700	0.466	4.000	5.000
Question 10	30	0	4.700	0.596	3.000	5.000

Average mean: **4.576**

The researchers ran a survey with 30 selected respondents to collect evaluations and opinions on the project. The criteria for selecting assured that the participants were either relevant to the study or had a direct engagement in the project. The researchers analyzed the descriptive statistics after gathering the responses. The findings yielded a positive mean rating with an average mean of 4.576, indicating that respondents had generally positive feelings towards the initiative. The descriptive statistics table revealed the central tendency and variability of the evaluations, allowing the researchers to measure general satisfaction and find any notable trends or patterns in the data.

CHAPTER 5

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

This chapter presents the comprehensive summary of the findings, elucidates the conclusions derived from the obtained results and offers recommendations for enhancing the study.

5.1 Summary of Findings

The researchers successfully developed a prototype vehicle using Raspberry Pi 4B, Arduino Mega, and sensors. The vehicle was designed to navigate water bodies and collect solid water waste, specifically plastic waste.

The proponents were able to design and build a sturdy body structure for the aquatic vehicle that could withstand the weight of the gathered garbage. The structure was engineered to ensure durability and buoyancy while accommodating the collected waste.

The researchers integrated computer vision and image processing techniques into the system. Using advanced algorithms, the system could identify and detect plastic waste in real time. This feature enhanced the efficiency of waste collection by enabling the vehicle to identify and target plastic waste autonomously.

A web application was created to provide real-time aquatic vehicle tracking and monitoring. The application allowed users to remotely view the vehicle's location, monitor the amount of waste collected, and access visual data captured by the onboard cameras.

The developed device was subjected to rigorous testing procedures following ISO standards—the testing phase aimed to evaluate the automated aquatic vehicle's performance, reliability, and accuracy. The testing results demonstrated that the device met the required standards and exhibited satisfactory waste collection and detection performance.

5.2 Conclusion

The conclusions drawn are based on the evaluation and analysis of the test results discussed in the preceding chapter.

The study was successfully created a robust and sturdy aquatic vehicle body structure capable of handling the weight of collected waste which is crucial for the overall effectiveness and longevity of the system. Making the vehicle automated by using Microcontrollers equipped with sensors and image processing is a significant step towards addressing the problem of solid waste in water bodies. By leveraging advanced technology, such as the Raspberry Pi platform, camera system, and sensors, the vehicle is equipped with the necessary tools for efficient waste collection and real-time monitoring.

The design and construction of the vehicle's body structure have been carefully implemented to ensure its durability and ability to withstand the harsh conditions typically encountered in water environments. The structure is made of stainless steel for robust construction, allowing the boat to operate efficiently and effectively even in challenging circumstances. Additionally, the boat has a solid waste collection capacity of up to 50 kilograms, effectively gathering and managing substantial amounts of waste.

Implementing image processing techniques and algorithms for detecting and collecting solid waste is a significant achievement in automating the waste collection process. The boat is utilizing image processing algorithms which means the vehicle can accurately identify and locate solid waste in water bodies, enabling targeted and efficient collection efforts.

The created web application for tracking and monitoring the aquatic vehicle provides an intuitive and user-friendly interface for users to access real-time data. This web application allows users to track the vehicle's location, monitor the progress of waste collection, and make data-driven decisions for optimizing waste management strategies.

The rigorous testing and validation of the developed aquatic waste collector device using ISO standards ensures its reliability, performance, and adherence to quality standards. By conducting thorough testing and validation procedures, any potential flaws or shortcomings in the device can be identified and addressed, ensuring that the final product meets the necessary quality standards for effective waste collection and management.

5.3 Recommendations

To further improve the study, the following recommendations are suggested:

Consider further enhancing the automation capabilities of the aquatic plastic waste collector vehicle. Explore advanced machine learning and artificial intelligence techniques to improve waste detection and collection efficiency. This could involve training the system to recognize different types of waste and adapt its collection methods accordingly.

Continuously refine the body structure of the aquatic vehicle to ensure it can withstand even more significant amounts of garbage. Conduct stress tests and simulations to identify potential weak points and reinforce those areas. Explore using lightweight and durable materials to improve buoyancy and increase the vehicle's load capacity.

Investigate and implement more advanced image processing techniques and algorithms to enhance the detection and collection of plastic waste. The proponents recommend the newer versions of the YOLO Algorithm for more precise Image Processing. Consider incorporating deep learning models for improved object recognition and classification. Additionally, explore techniques to differentiate between biodegradable and non-biodegradable materials for more effective waste sorting.

Expand the capabilities of the Web Application for tracking and monitoring the aquatic vehicle. Integrate features like real-time video streaming from onboard cameras, data analytics for waste collection trends, and user feedback mechanisms. Enhance the user interface and user experience to make it more intuitive and user-friendly.

Finally, continue performing tests on the device using international standards, such as ISO, to guarantee its performance, dependability, and safety. Consider conducting field trials in various water environments to assess the device's effectiveness and adaptability to different conditions. Continuously improve the device based on feedback from testing and validation processes.

References

Akib, A., Tasnim, F., Biswas, D., Hashem, M. B., Rahman, K., Bhattacharjee, A., & Fattah, S. A. (2019). Unmanned Floating Waste Collecting Robot. IEEE Region 10 Annual International Conference, Proceedings/TENCON, 2019-Octob, 2645–2650. https://doi.org/10.1109/TENCON.2019.8929537

Alrumman, S. A., El-kott, A. F., & Keshk, S. M. A. S. (2016). Water Pollution: Source and Treatment Water Pollution: Source & Treatment. Am J Environ Eng, 6(June), 88–98. https://doi.org/10.5923/j.ajee.20160603.02

Amarnath, M., Prasannakumar, M., Nandhakumar, S., Naveen, M., & Kannan, P. (2019). Design and Analysis of Floating Waste Cleaning Machine. 9(May), 4679–4683.

Ang, F. G., Gabriel, M. K. A. R., Sy, J., Tan, J. J. O., & Abad, A. C. (2013). Automated Waste Sorter with Mobile Robot Delivery Waste. Presented at the Research Congress 2013 De La Salle University Manila March 7-9, 2013, April 1–7.

Bairachtaris, N., Petroulas, V., Safacas, A., Tatakis, E., & Tsotoulidis, S. (2010). Design and construction of a four-channel interleaved buck dc/dc converter for an electric boat application. International Conference on Electrical Systems for Aircraft, Railway and Ship Propulsion, ESARS 2010, March 2016. https://doi.org/10.1109/ESARS.2010.5665239

Bashir, A., & Banday, S. A. (2013). Concept, Design, and Implementation of Automatic Waste Management System. International Journal on Recent and Innovation Trends in Computing and Communication, July, 604 – 609.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. April. http://arxiv.org/abs/2004.10934

Dandabathula, G., Remote, N., Centre, S., Sitiraju, S. R., Raj, U., Jha, C. S., Remote, N., & Centre, S. (2021). Design and Development of Aquayaan: An IoT based Robotic Boat for Inland Water Surveys. January. https://doi.org/10.13140/RG.2.2.32247.14247

Hu, X., Liu, Y., Zhao, Z., Liu, J., Yang, X., Sun, C., Chen, S., Li, B., & Zhou, C. (2021). Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network. Computers and Electronics in Agriculture, 185(March), 106135. https://doi.org/10.1016/j.compag.2021.106135

Isobe, A., Uchida, K., Tokai, T., & Iwasaki, S. (2015). East Asian seas: A hot spot of pelagic microplastics. Marine Pollution Bulletin, 101(2), 618–623. https://doi.org/10.1016/j.marpolbul.2015.10.042

Jambeck, J. R. et al, Hoegh-Guldberg, O., Cai, R., Poloczanska, E., Brewer, P., Sundby, S., Hilmi, K., Fabry, V., & Jung, S. (2015). Plastic waste inputs from land into the ocean. Science, September 2014, 1655–1734.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2021). A Review of Yolo Algorithm Developments. Procedia Computer Science, 199, 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135

Liu, G., & Lee, S.-H. (2021). Municipal waste classification system design based on Faster-RCNN and YoloV4 mixed model 1. International Journal of Advanced

Culture Technology, 9(3), 305–314. https://doi.org/10.17703/IJACT.2021.9.3.305

Nair, S. S., Sudheer, A. P., & Joy, M. L. (2019). Design and fabrication of river cleaning robot. ACM International Conference Proceeding Series. https://doi.org/10.1145/3352593.3352663

Noordin, A. (2013). Design and Selecting the Proper Conveyor-Belt. Ijaect, IV(II), 43–49. https://www.academia.edu/35699052/DESIGN_AND_SELECTING_THE_PROPER_CONVEYOR_pdf?bulkDownload=thisPaper-topRelated-sameAuthor-citingThis-citedByThis-secondOrderCitations&from=cover_page

Padalkar, A., Pathak, P., & Stynes, P. (n.d.). CAIP 2021. Google Books. https://books.google.com.ph/books?id=q69cEAAAQBAJ&lpg=PA18&ots=2LxhAtxsNZ&dq=yolov4%20object%20detection%20waste&lr&pg=PA18#v=onepage&q=yolov4%20object%20detection%20waste&f=false

Pajankar, A. (2017). Raspberry Pi Image Processing Programming. Raspberry Pi Image Processing Programming. https://doi.org/10.1007/978-1-4842-2731-2

Pathak, O., Nalawade, A., Kaundanya, C., & Parode, S. (2018). Garbage Level Monitoring System using Raspberry Pi. 9518–9522. https://doi.org/10.15680/IJIRSET.2018.0709037

Pearson, T., Pranitha, S., Sheshank, K., Goud, M. S., & ManiNithin, L. (2019). Design and implementation of smart boat technique. International Journal of Scientific and Research Publications (IJSRP), 9(4). https://doi.org/10.29322/ijsrp.9.04.2019.p8846

Pedà, C., Longo, F., Berti, C., Laface, F., De Domenico, F., Consoli, P., Battaglia, P., Greco, S., & Romeo, T. (2022). The waste collector: information from a pilot study on the interaction between the common octopus (Octopus vulgaris, Cuvier, 1797) and marine litter in bottom traps fishing and first evidence of plastic ingestion. Marine Pollution Bulletin, 174(September 2021). https://doi.org/10.1016/j.marpolbul.2021.113185

Philippines. SEA circular. (2020, October 27). Retrieved June 14, 2022, from https://www.sea-circular.org/country/philippines/?fbclid=IwAR1Ly_fyKB7K-2ZET7dp-lnhrvwPVIot2uoQCkSWcDXPYuoBQHPw0b_bM4g

Pimentel, D., Berger, B., Filiberto, D., Newton, M., Wolfe, B., Karabinakis, E., Clark, S., Poon, E., Abbett, E., & Nandagopal, S. (2007). Water resources: Agricultural and environmental issues. Food, Energy, and Society, Third Edition, 54(10), 183–200. https://doi.org/10.1201/9781420046687

Richey, R.C., Klein, J.D. (2005). Developmental research methods: Creating knowledge from instructional design and development practice. J. Comput. High. Educ. 16, 23–38. https://doi.org/10.1007/BF02961473

Rovetta, A., Xiumin, F., Vicentini, F., Minghua, Z., Giusti, A., & Qichang, H. (2009). Early detection and evaluation of waste through sensorized containers for a collection monitoring application. Waste Management, 29(12), 2939–2949. https://doi.org/10.1016/j.wasman.2009.08.016

Roy, A. M., Bose, R., & Bhaduri, J. (2021). A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. http://arxiv.org/abs/2111.00298

Ruangpayoongsak, N., Sumroengrit, J., & Leanglum, M. (2017). A floating waste scooper robot on water surface. International Conference on Control, Automation and Systems,017-Octob (Iccas),1543–1548. https://doi.org/10.23919/ICCAS.2017.8204234

Salawu, G., Bright, G., & Onunka, C. (2020). Modelling and Simulation of a Conveyor Belt System for Optimal Productivity. International Journal of Mechanical Engineering and Technology (Ijmet), 11(1), 115–121. https://doi.org/10.34218/ijmet.11.1.2020.012

Saputra, A. P. (2021). Waste Object Detection and Classification using Deep Learning Algorithm: YOLOv4 and YOLOv4-tiny. Turkish Journal of Computer and Mathematics Education, 12(14), 1666–1677.

Satheesh, J., Nair, A. P., Devipriya, M., Chithra, A., Mahesh, G., & Jayasree, P. R. (2020). Wireless Communication based Water Surface Cleaning Boat. Proceedings of the 4th International Conference on Trends in Electronics and Informatics, ICOEI 2020, Icoei, 716–720. https://doi.org/10.1109/ICOEI48184.2020.9142960

Shamsuddin, P. N. F. M., Mansor, M. A., Hadi, A., Abidin, N. Z., & Ibrahim, R. (2020). Article Development of Water Trash Collector. ACCESS Online Journal JAIERAJournal-Journal of Advanced Industrial Electronics Research and Applications, 1(January), 1–05.

Somal, S., Phadke, G., Gaikwad, P., Gavade, S., & Mane, M. (2020). Ocean Surface Trash Collector. ADBU Journal of Electrical and Electronics Engineering (AJEEE) |, 4(1), 23–27. www.tinyurl.com/ajeee-adbu

Tang, Z., Zhang, X., Li, X., & Zhang, S. (2016). Robust Image Hashing with Ring Partition and Invariant Vector Distance. IEEE Transactions on Information Forensics and Security, 11(1), 200–214. https://doi.org/10.1109/TIFS.2015.2485163

Wahyutama, A. B., & Hwang, M. (2022). YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins. Electronics (Switzerland), 11(9). https://doi.org/10.3390/electronics11091323

Yang, Y. (n.d.). Waste Classification Based On Yolov4. In JOURNAL OF SIMULATION (Vol. 9, Issue 6).

Youme, O., Bayet, T., Dembele, J. M., & Cambier, C. (2021). Deep Learning and Remote Sensing: Detection of Dumping Waste Using UAV. Procedia Computer Science, 185, 361–369. https://doi.org/10.1016/j.procs.2021.05.037

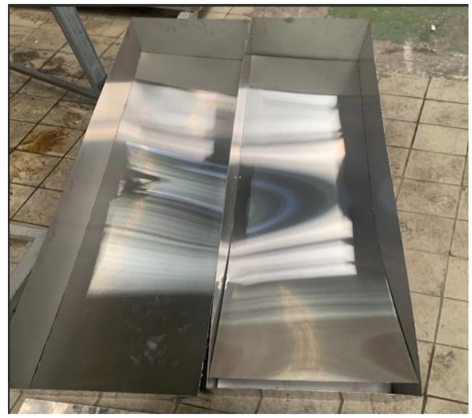
Zavare, S., Parashare, R., Patil, S., Rathod, P., & Babanne, V. (2017). Smart City Waste Management System Using GSM. International Journal of Engineering Science and Computing, 5(3), 74–78. http://ijesc.org/

Zhang, Q., Yang, Q., Zhang, X., Wei, W., Bao, Q., Su, J., & Liu, X. (2022). A multi-label waste detection model based on Transfer learning. Resources, Conservation and Recycling, 181, 106235. https://doi.org/10.1016/j.resconrec.2022.106235

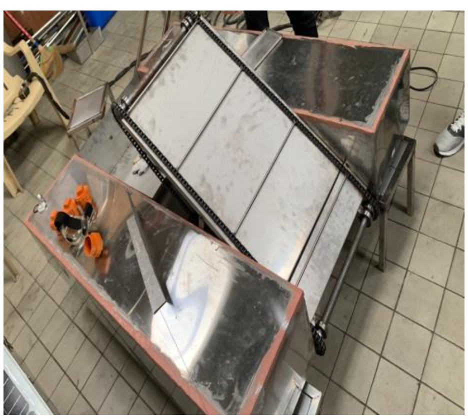
APPENDIX A

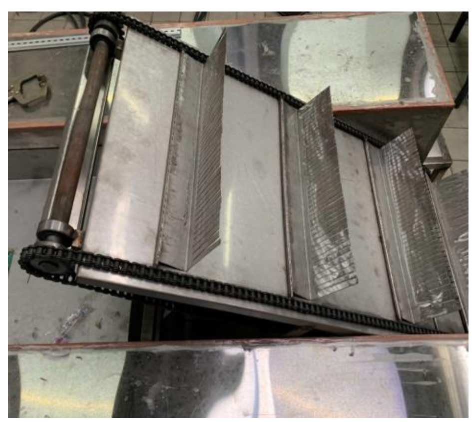
DOCUMENTATION

I. Preparation of Materials

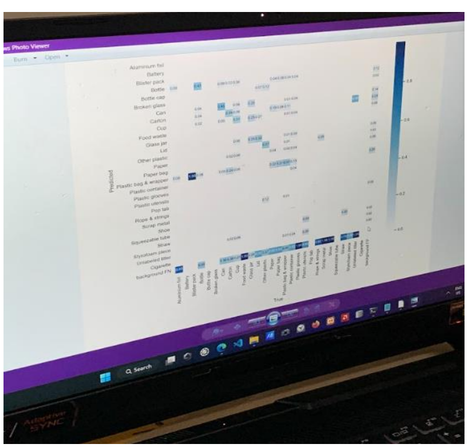


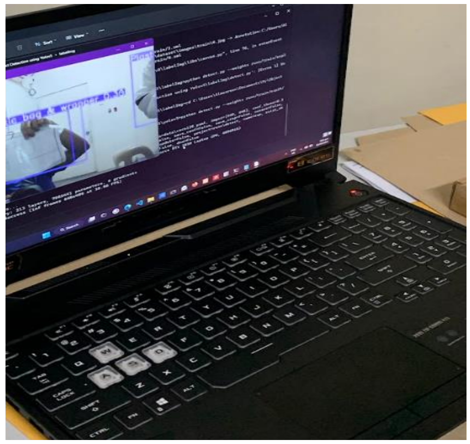
II. Fabrication of Boat Structure





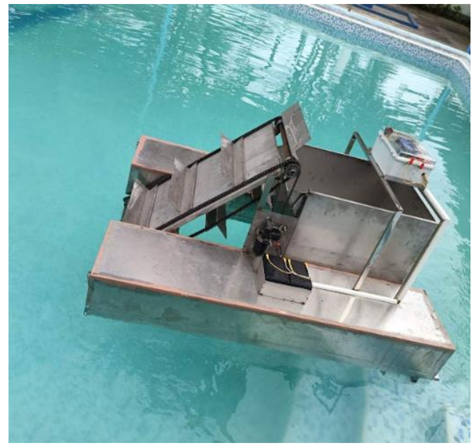
III. Software Integration



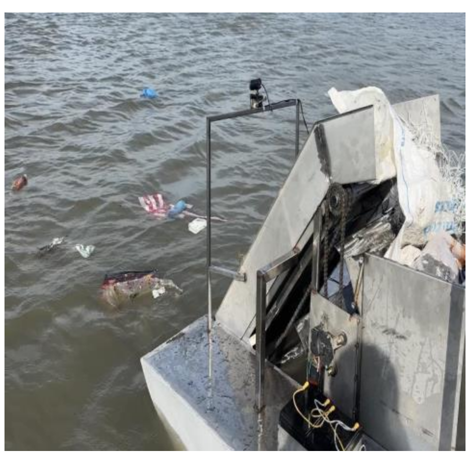
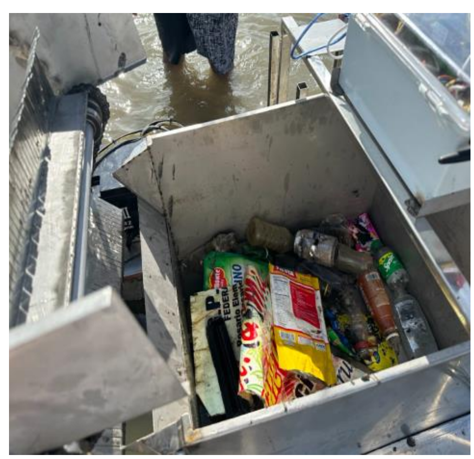


IV. Deployment area

V. Pre-testing

VI. Final Testing

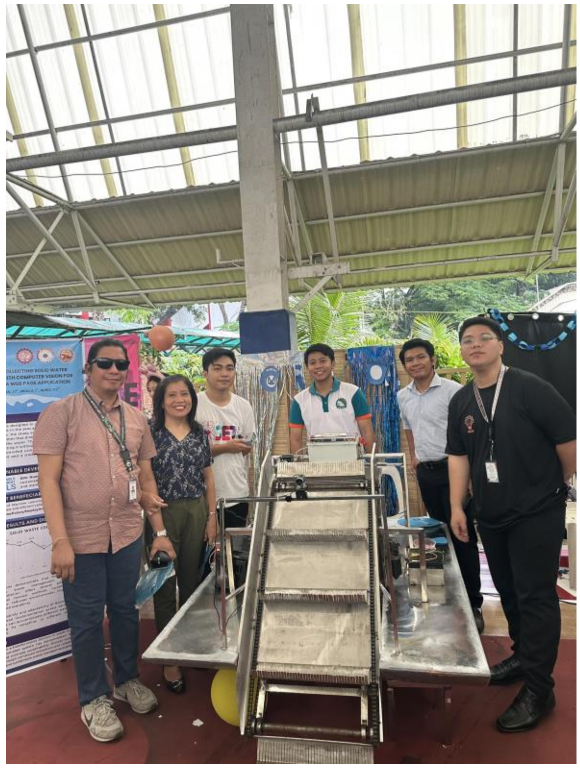






VII. Project Evaluation

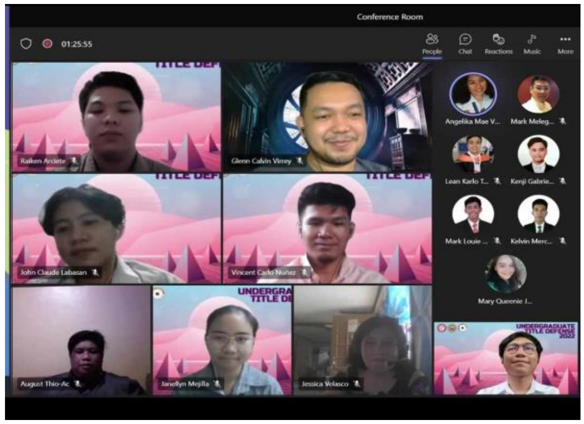

 

VIII. Appreciate




GRADUATING
CLASS

IX. Defense

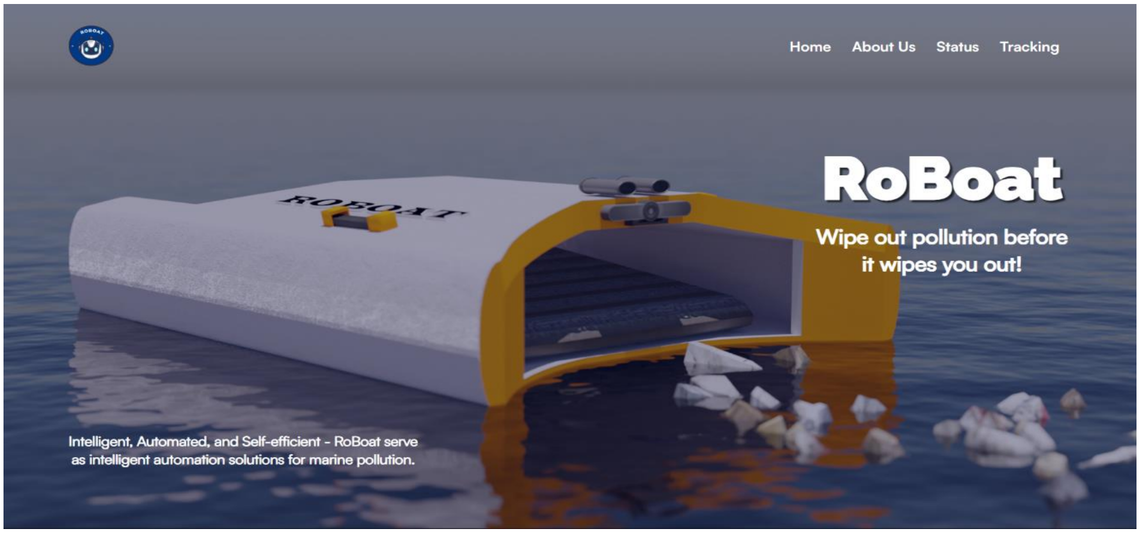

Conference Room
01:25:55
People
Chat
Reactions
Music
More
Raiken Arciete
Glenn Calvin Virrey
John Claude Labasan
Vincent Carlo Nuñez
August Thio-Ac
Janellyn Mejilla
Jessica Velasco
Angelika Mae V...
Mark Meleg...
Lean Karlo T...
Kenji Gabrie...
Mark Louie ...
Kelvin Merc...
Mary Queenie J...
UNDERGRADUATE TITLE DEFENSE 2022

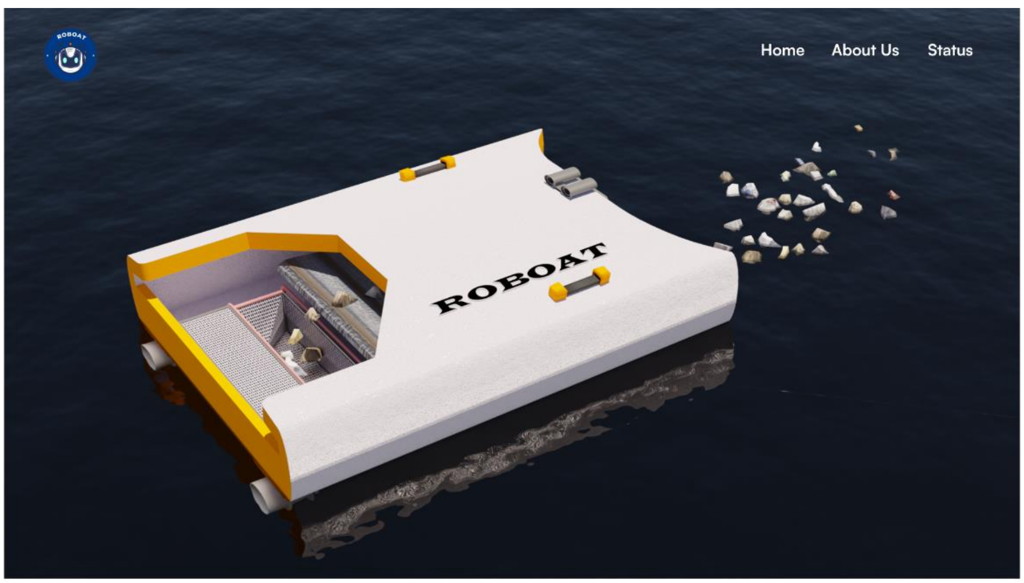

03:56
LABASAN


MEJILLA

APPENDIX B

ROBOAT WEBPAGE



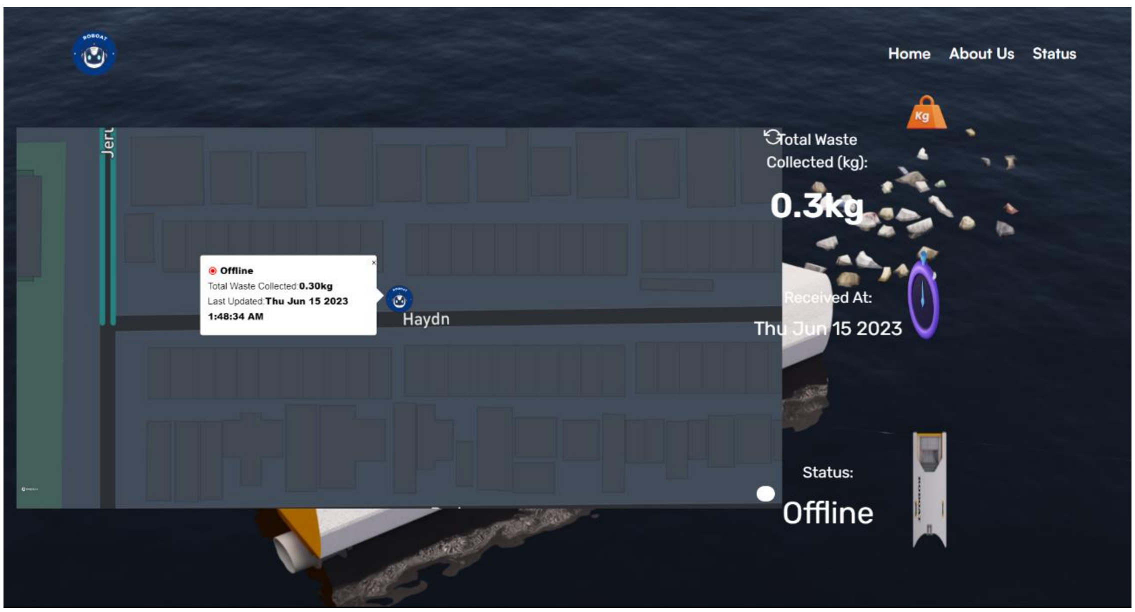


About Us

We made RoBoat because we care about preserving the environment.

Our objective is to create a way to make bodies of water clean from solid waste. Our objective is to design and build autonomous, energy-efficient water vehicles to ensure the cleanliness of our oceans, seas, and rivers. Building boats that can function safely and autonomously in various sea settings is a difficult task, but we managed to create one for our goal. Also, we believe that this is the next step in the development of tools that will make the prevention of water pollution safer, less time-consuming, more affordable, and require less human interaction.


The Creators
RAIKEN ARCIETE
Hardware Developer
DOENELLE JOHN HICARO
Project Leader and UX Developer
JOHN CLAUDE LABASAN
Lead Predictive Model Developer
JANELLYN MEJILLA
Lead Program Developer
VINCENT CARLO C. NUÑEZ
UI/UX Designer & Developer
and Hardware Developer
©2022-2023 RoBoat


Home
About Us
Status
Offline
Total Waste Collected 0.30kg
Last Updated Thu Jun 15 2023
1:48:34 AM
Haydn
Total Waste
Collected (kg):
0.3kg
Received At:
Thu Jun 15 2023
Status:
Offline

APPENDIX C

PROGRAM CODES

Arduino

```
#include <NewPing.h>2

NewPing Sonar1(8, 9, 400);

float d1;

float distanceThreshold = 30; //30cm

float WeightFull = 5; //30kg



int debug = 0;

#include <SoftwareSerial.h>



// SoftwareSerial Serial3(11, 10); //(RX, TX)

#include <TinyGPS.h>

float flat = 0.0;

float flon = 0.0;

TinyGPS gps;



unsigned long age;

static void smartdelay(unsigned long ms);
```

```
#include "HX711.h"

HX711 cell(A0, A1);

float weight = 0;


// #include <SoftwareSerial.h>

// SoftwareSerial Serial1(11, 10); // TX Pin, RX Pin


#define Button (2)


#define LeftForward_Pin (30)

#define LeftBackward_Pin (32)

#define RightForward_Pin (26)

#define RightBackward_Pin (28)


#define Conveyor_Pin (24)


#include <millisDelay.h>

millisDelay Timer1;
```

```
millisDelay CheckWeightTimer;



#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4);



long RotationInterval = 60000;

long CheckWeightInterval = 10000;



void setup() {

  Serial.begin(9600);

  Serial1.begin(9600);

  Serial3.begin(9600);

  // Serial3.listen();

  Serial2.begin(115200);

  delay(1000);

  Serial1.print('Z');
```

```
    //Buttons
    pinMode(Button, INPUT_PULLUP);

    //Relays
    pinMode(LeftForward_Pin, OUTPUT); digitalWrite(LeftForward_Pin, HIGH);
    pinMode(LeftBackward_Pin, OUTPUT); digitalWrite(LeftBackward_Pin, HIGH);
    pinMode(RightForward_Pin, OUTPUT); digitalWrite(RightForward_Pin, HIGH);
    pinMode(RightBackward_Pin, OUTPUT); digitalWrite(RightBackward_Pin, HIGH);
    pinMode(Conveyor_Pin, OUTPUT); digitalWrite(Conveyor_Pin, HIGH);

    // lcd.init();
    // lcd.backlight();

    Serial2.print("AT\r"); delay(1000);
    Serial2.print("ATE0"); delay(1000);
    Serial2.print("AT+CMEE=2\r"); delay(1000);
    Serial2.print("AT+CMGF=1\r"); delay(1000); //Because we want to send the SMS in
text mode
```

```
    Serial2.print("AT+CMGDA=\"DEL ALL\"\r"); //Because we want to send the SMS in
text mode


    cell.set_scale(22000);
    cell.tare(); delay(1000);
    // while (1) {
    //     weight = cell.get_units();
    //     Serial.println(weight);
    //     delay(1000);
    // }
    CheckWeightTimer.start(CheckWeightInterval);


    // GetGPS();
    // while (flat == 1000) {GetGPS(); delay(1000);}
    // DBG("Successful");
}
int Full = 0;
char receivedChar;
```

```
void loop() {

  if (Serial1.available()) {

        receivedChar = Serial1.read();

        Serial.println(receivedChar);

           if (receivedChar == 'F') {

             Forward();

             while (1) {if (Serial1.available()) {break;}}

             Stop();

           }

        else if (receivedChar == 'B') {

             Backward();

             while (1) {if (Serial1.available()) {break;}}

             Stop();

          }

        else if (receivedChar == 'R') {

             RotateClockwise();

             while (1) {if (Serial1.available()) {break;}}

             Stop();
```

```
        }
      else if (receivedChar == 'L') {
          RotateCC();
          while (1) {if (Serial1.available()) {break;}}
          Stop();
        }
      else if (receivedChar == 'C') {
          digitalWrite(Conveyor_Pin, LOW);
          while (1) {if (Serial1.available()) {break;}}
          Stop();
        }
    }


  if (Serial.available()) {
    receivedChar = Serial.read();


       if (receivedChar == '1') {
          d1 = Sonar1.ping_cm(); delay(30);
```

```
            DBG(String(d1));
            if ((d1 < distanceThreshold) && (d1 > 0)) {
              DBG("Obstacle");
              }
            else {
              DBG("Forward"); digitalWrite(Conveyor_Pin, LOW);
              Forward(); delay(5000); Stop();
              }
            }
    else if (receivedChar == '2') {Serial.println(); Backward(); delay(5000); Stop();}
    else if (receivedChar == '3') {RotateClockwise(); delay(6000); Stop();}
    else if (receivedChar == '4') {


    }
  }
  receivedChar = ' ';
  if (CheckWeightTimer.justFinished()) {
    GetWeight();
```

```
    if ((weight > WeightFull) && (Full == 0)) {
        Full = 1;
        DBG("Sending");
        SendMSG2("09914043637","Bin is full");
    }
    else if (weight < WeightFull - 0.01) {
        Full = 0;
    }
    DBG(String(weight));
    CheckWeightTimer.start(CheckWeightInterval);
    GetGPS();
    d1 = Sonar1.ping_cm(); delay(30);
    Serial.print("!");
    Serial.print(d1); Serial.print(",");
    Serial.print(weight); Serial.print(",");
    Serial.print(flat,7); Serial.print(",");
    Serial.print(flon,7); Serial.println("@");
}
```

```
}

void Forward() {
  digitalWrite(LeftForward_Pin, LOW);
  digitalWrite(RightForward_Pin, LOW);
  digitalWrite(LeftBackward_Pin, HIGH);
  digitalWrite(RightBackward_Pin, HIGH);
}

void Backward() {
  digitalWrite(LeftForward_Pin, HIGH);
  digitalWrite(RightForward_Pin, HIGH);
  digitalWrite(LeftBackward_Pin, LOW);
  digitalWrite(RightBackward_Pin, LOW);
}

void RotateClockwise() {
  digitalWrite(LeftForward_Pin, LOW);
```

```
    digitalWrite(RightForward_Pin, HIGH);

    digitalWrite(LeftBackward_Pin, HIGH);

    digitalWrite(RightBackward_Pin, LOW);

}



void RotateCC() {

    digitalWrite(LeftForward_Pin, HIGH);

    digitalWrite(RightForward_Pin, LOW);

    digitalWrite(LeftBackward_Pin, LOW);

    digitalWrite(RightBackward_Pin, HIGH);

}



void Stop() {

    digitalWrite(LeftForward_Pin, HIGH);

    digitalWrite(RightForward_Pin, HIGH);

    digitalWrite(LeftBackward_Pin, HIGH);

    digitalWrite(RightBackward_Pin, HIGH);

    digitalWrite(Conveyor_Pin, HIGH);
```

```
}

void ToggleLH(long delayings, int relayName) {
    digitalWrite(relayName, LOW); delay(delayings);
    digitalWrite(relayName, HIGH);
}

void ToggleHL(long delayings, int relayName) {
    digitalWrite(relayName, HIGH); delay(delayings);
    digitalWrite(relayName, LOW);
}

void RelayLOWwithLimit(int relayName, int MSName) {
    if (digitalRead(MSName) == 1){
        digitalWrite(relayName, LOW); delay(50);
        while(1) {if (digitalRead(MSName) == 0) {break;}}
        digitalWrite(relayName, HIGH);
    }
```

```
}

void LCDprint(int x, int y, String z) {
  lcd.setCursor(x,y); lcd.print(z);
}

void LCDClear() {
  LCDprint(0,0,"          ");
  LCDprint(0,1,"          ");
  LCDprint(0,2,"          ");
  LCDprint(0,3,"          ");
}

void DBG(String str) {
  if (debug) {Serial.println(str);}
}

void SendMSG2(String PhoneNumber, String Msg) {
```

```
    Serial2.print("AT+CMGF=1\r"); delay(500); //Because we want to send the SMS in text mode
    Serial2.print("AT+CMGS=\"" + PhoneNumber + "\"\r"); //to be sent to the number specified.
    delay(1000);
    Serial2.print(Msg);
    Serial2.println();

    delay(1000);
    Serial2.write(0x1A);  //Equivalent to sending Ctrl+Z
    delay(8000);
}

static void smartdelay(unsigned long ms){
  unsigned long start = millis();
//   Serial3.listen();
  Serial3.flush();
  do{
    while (Serial3.available())
```

```
      gps.encode(Serial3.read());
  } while (millis() - start < ms);
}

void GetGPS(){
  smartdelay(1000);
  gps.f_get_position(&flat, &flon, &age);
  Serial.println();
}

void GetWeight() {
    weight = cell.get_units();
    if (weight < 0.02) {weight = 0;}
}
```

Python Codes

```
import argparse
import os
```

```
import sys

from pathlib import Path

import time

import requests



import torch

import torch.backends.cudnn as cudnn

import serial

SerialData = serial.Serial("/dev/ttyACM0", 9600, timeout = 0.2)

# SerialData = serial.Serial("com6", 9600, timeout = 0.2)

# SerialData.setDTR(False)

# time.sleep(1)

# SerialData.flushInput()

# SerialData.setDTR(True)

# time.sleep(2)

# SerialData.write(bytes("2", "utf-8"))

# reading = SerialData.readline().decode("utf-8")

# reading = SerialData.readline().decode("utf-8")
```

```
#   if reading == 'A':
#       pass


class MyTimer:
    def __init__(self):
        self.StartTime = 0
        self.TargetTime = 0
        self.Force = 0


    def start(self, TargetTime):
        self.Force = 0
        self.StartTime = time.time()
        self.TargetTime = TargetTime


    def justFinished(self):
        # print(self.Force,time.time()-self.StartTime, self.TargetTime)
        if self.Force == 1:
            return False
```

```
        if (time.time()-self.StartTime > self.TargetTime):
            self.Force = 1
            return True
        else:
            return False
    def elapsed(self):
        return time.time()-self.StartTime

    def remaining(self):
        return self.TargetTime - (time.time()-self.StartTime)

    def stop(self):
        self.Force = 1

Timer1 = MyTimer()
Timer1.justFinished()
Timer1.start(5)
```

FILE = Path(__file__).resolve()

ROOT = FILE.parents[0] # YOLOv5 root directory

if str(ROOT) not in sys.path:

    sys.path.append(str(ROOT)) # add ROOT to PATH

ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative

from models.common import DetectMultiBackend

from utils.dataloaders import IMG_FORMATS, VID_FORMATS, LoadImages, LoadStreams

from utils.general import (LOGGER, check_file, check_img_size, check_imshow, check_requirements, colorstr, cv2,

increment_path, non_max_suppression, print_args, scale_coords, strip_optimizer, xyxy2xywh)

from utils.plots import Annotator, colors, save_one_box

from utils.torch_utils import select_device, time_sync

print(torch.version.cuda)

@torch.no_grad()

```
def run(
        weights=ROOT / 'yolov5s.pt',  # model.pt path(s)
        source=ROOT / 'data/images',  # file/dir/URL/glob, 0 for webcam
        data=ROOT / 'data/coco128.yaml',  # dataset.yaml path
        imgsz=(640, 640),  # inference size (height, width)
        conf_thres=0.25,  # confidence threshold
        iou_thres=0.45,  # NMS IOU threshold
        max_det=1000,  # maximum detections per image
        device='',  # cuda device, i.e. 0 or 0,1,2,3 or cpu
        view_img=False,  # show results
        save_txt=False,  # save results to *.txt
        save_conf=False,  # save confidences in --save-txt labels
        save_crop=False,  # save cropped prediction boxes
        nosave=False,  # do not save images/videos
        classes=None,  # filter by class: --class 0, or --class 0 2 3
        agnostic_nms=False,  # class-agnostic NMS
        augment=False,  # augmented inference
        visualize=False,  # visualize features
```

```
        update=False,  # update all models
        project=ROOT / 'runs/detect',  # save results to project/name
        name='exp',  # save results to project/name
        exist_ok=False,  # existing project/name ok, do not increment
        line_thickness=3,  # bounding box thickness (pixels)
        hide_labels=False,  # hide labels
        hide_conf=False,  # hide confidences
        half=False,  # use FP16 half-precision inference
        dnn=False,  # use OpenCV DNN for ONNX inference
):
    source = str(source)
    save_img = not nosave and not source.endswith('.txt')  # save inference images
    is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)
    is_url = source.lower().startswith(('rtsp://', 'rtmp://', 'http://', 'https://'))
    webcam = source.isnumeric() or source.endswith('.txt') or (is_url and not is_file)
    if is_url and is_file:
        source = check_file(source)  # download
```

```
    # Directories
    save_dir = increment_path(Path(project) / name, exist_ok=exist_ok)  # increment run
    (save_dir / 'labels' if save_txt else save_dir).mkdir(parents=True, exist_ok=True)  #
make dir

    # Load model
    device = select_device(device)
    model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data, fp16=half)
    stride, names, pt = model.stride, model.names, model.pt
    imgsz = check_img_size(imgsz, s=stride)  # check image size

    # Dataloader
    if webcam:
        view_img = check_imshow()
        cudnn.benchmark = True  # set True to speed up constant image size inference
        dataset = LoadStreams(source, img_size=imgsz, stride=stride, auto=pt)
        bs = len(dataset)  # batch_size
    else:
```

```
    dataset = LoadImages(source, img_size=imgsz, stride=stride, auto=pt)
    bs = 1  # batch_size
vid_path, vid_writer = [None] * bs, [None] * bs

# Run inference
model.warmup(imgsz=(1 if pt else bs, 3, *imgsz))  # warmup
seen, windows, dt = 0, [], [0.0, 0.0, 0.0]
for path, im, im0s, vid_cap, s in dataset:
    t1 = time_sync()
    im = torch.from_numpy(im).to(device)
    im = im.half() if model.fp16 else im.float()  # uint8 to fp16/32
    im /= 255  # 0 - 255 to 0.0 - 1.0
    if len(im.shape) == 3:
        im = im[None]  # expand for batch dim
    t2 = time_sync()
    dt[0] += t2 - t1
```

```
        # Inference
        visualize = increment_path(save_dir / Path(path).stem, mkdir=True) if visualize else False
        pred = model(im, augment=augment, visualize=visualize)
        t3 = time_sync()
        dt[1] += t3 - t2

        # NMS
        pred = non_max_suppression(pred, conf_thres, iou_thres, classes, agnostic_nms, max_det=max_det)
        dt[2] += time_sync() - t3

        # Second-stage classifier (optional)
        # pred = utils.general.apply_classifier(pred, classifier_model, im, im0s)

        # Process predictions
        for i, det in enumerate(pred):  # per image
            seen += 1
            if webcam:  # batch_size >= 1
```

```
    p, im0, frame = path[i], im0s[i].copy(), dataset.count
    s += f'{i}: '
else:
    p, im0, frame = path, im0s.copy(), getattr(dataset, 'frame', 0)


p = Path(p)  # to Path
save_path = str(save_dir / p.name)  # im.jpg
txt_path = str(save_dir / 'labels' / p.stem) + ('' if dataset.mode == 'image' else
f'_{frame}')  # im.txt
s += '%gx%g ' % im.shape[2:]  # print string
gn = torch.tensor(im0.shape)[[1, 0, 1, 0]]  # normalization gain whwh
imc = im0.copy() if save_crop else im0  # for save_crop
annotator = Annotator(im0, line_width=line_thickness, example=str(names))
label = ""
reading = SerialData.read().decode("utf-8")
AccStr = ""
Acc = 0
if reading == '!':
```

```
        # print("!")
        while True:
            reading = SerialData.read().decode("utf-8")
            if reading == "":
                # print("blank")
                pass
            elif reading == '@':
                # print("end")
                llli = AccStr.split(",")
                obstacle = llli[0]
                weight = llli[1]
                lat = llli[2]
                long = llli[3]
                postHttpLink ="https://2mgpttbt63.execute-api.ap-northeast-
1.amazonaws.com/dev/roboat?obstacle="+str(obstacle)+"&weight="+str(weight)+"&lon
g="+str(long)+"&lat="+str(lat)+"&status=Ongoing"
                httplink =
"http://www.domain.com/store.php?d="+str(obstacle)+",weigth="+str(weight)+",long="+
str(long)+",lat="+str(lat)
```

```
            try:

                response = requests.get(url=httplink, timeout=10)

                post_response = requests.post(url=postHttpLink, timeout=10)

                print(httplink,response.text)

                print(postHttpLink,post_response.text)

            except:

                print(httplink,"Error")

            break

        else:

            # print(reading)

            AccStr = AccStr + reading



if len(det):

    # Rescale boxes from img_size to im0 size

    det[:, :4] = scale_coords(im.shape[2:], det[:, :4], im0.shape).round()



    # Print results

    for c in det[:, -1].unique():
```

```
        n = (det[:, -1] == c).sum()  # detections per class
        s += f"{n} {names[int(c)]}{'s' * (n > 1)}, "  # add to string

    # Write results

    for *xyxy, conf, cls in reversed(det):
        if save_txt:  # Write to file
            xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-1).tolist()
# normalized xywh
            line = (cls, *xywh, conf) if save_conf else (cls, *xywh)  # label format
            with open(f'{txt_path}.txt', 'a') as f:
                f.write(('%g ' * len(line)).rstrip() % line + '\n')

        if save_img or save_crop or view_img:  # Add bbox to image
            c = int(cls)  # integer class
            label = None if hide_labels else (names[c] if hide_conf else f'{names[c]}
{conf:.2f}')

            name = f'{names[c]}'
```

```
                    pred = float(f'{conf:.2f}')

                    print(name,pred)

                    if pred > 0.8:

                        if Timer1.justFinished():

                            SerialData.write(bytes("1", "utf-8"))

                            Timer1.start(10)

                            annotator.box_label(xyxy, name + " FORWARD", color=colors(c,
True))

                            LOGGER.info("SYSTEM WILL GO FORWARD. (Send forward
signal to Arduino)")

                            im0 = annotator.result()

                        else:

                            annotator.box_label(xyxy, name + "
COOLDOWN"+str(int(Timer1.remaining())), color=colors(c, True))

                    else:

                        annotator.box_label(xyxy, name + " <confidence("+str(pred)+")",
color=colors(c, True))

                if save_crop:
```

```
                save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] /
f'{p.stem}.jpg', BGR=True)

            if view_img:

                if p not in windows:

                    windows.append(p)

                    cv2.namedWindow(str(p), cv2.WINDOW_NORMAL |
cv2.WINDOW_KEEPRATIO)  # allow window resize (Linux)

                    cv2.resizeWindow(str(p), im0.shape[1], im0.shape[0])

                cv2.imshow(str(p), im0)

                cv2.waitKey(1)  # 1 millisecond



            # Save results (image with detections)

            if save_img:

                if dataset.mode == 'image':

                    cv2.imwrite(save_path, im0)

                else:  # 'video' or 'stream'

                    if vid_path[i] != save_path:  # new video

                        vid_path[i] = save_path

                        if isinstance(vid_writer[i], cv2.VideoWriter):
```

```
                vid_writer[i].release()  # release previous video writer
            if vid_cap:  # video
                fps = vid_cap.get(cv2.CAP_PROP_FPS)
                w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
                h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
            else:  # stream
                fps, w, h = 30, im0.shape[1], im0.shape[0]
            save_path = str(Path(save_path).with_suffix('.mp4'))  # force *.mp4 suffix
on results videos
            vid_writer[i] = cv2.VideoWriter(save_path,
cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))
        cv2.imwrite("final.jpg", im0)
        time.sleep(1)


    # Print time (inference-only)
    LOGGER.info(f'{s}Done. ({t3 - t2:.3f}s)')


  # Print results
  t = tuple(x / seen * 1E3 for x in dt)  # speeds per image
```

```
    LOGGER.info(f'Speed: %.1fms pre-process, %.1fms inference, %.1fms NMS per
image at shape {(1, 3, *imgsz)}' % t)
    if save_txt or save_img:
        s = f"\n{len(list(save_dir.glob('labels/*.txt')))} labels saved to {save_dir / 'labels'}"
if save_txt else ''
        LOGGER.info(f"Results saved to {colorstr('bold', save_dir)}{s}")
    if update:
        strip_optimizer(weights)  # update model (to fix SourceChangeWarning)


def parse_opt():
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', nargs='+', type=str, default=ROOT /
r'runs/train/exp18/weights/best.pt', help='model path(s)')
    parser.add_argument('--source', type=str, default=0, help='file/dir/URL/glob, 0 for
webcam')
    parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml',
help='(optional) dataset.yaml path')
    parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640],
help='inference size h,w')
```

parser.add_argument('--conf-thres', type=float, default=0.7, help='confidence threshold')

parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')

parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per image')

parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')

parser.add_argument('--view-img', action='store_true', help='show results')

parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')

parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt labels')

parser.add_argument('--save-crop', action='store_true', help='save cropped prediction boxes')

parser.add_argument('--nosave', action='store_true', help='do not save images/videos')

parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --classes 0 2 3')

parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')

parser.add_argument('--augment', action='store_true', help='augmented inference')

parser.add_argument('--visualize', action='store_true', help='visualize features')

parser.add_argument('--update', action='store_true', help='update all models')

```
    parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to project/name')
    parser.add_argument('--name', default='exp', help='save results to project/name')
    parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do not increment')
    parser.add_argument('--line-thickness', default=3, type=int, help='bounding box thickness (pixels)')
    parser.add_argument('--hide-labels', default=False, action='store_true', help='hide labels')
    parser.add_argument('--hide-conf', default=False, action='store_true', help='hide confidences')
    parser.add_argument('--half', action='store_true', help='use FP16 half-precision inference')
    parser.add_argument('--dnn', action='store_true', help='use OpenCV DNN for ONNX inference')
    opt = parser.parse_args()
    opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1  # expand
    print_args(vars(opt))
    return opt
def main(opt):
```

```python
    check_requirements(exclude=('tensorboard', 'thop'))
    run(**vars(opt))

if __name__ == "__main__":
    opt = parse_opt()
    main(opt)
```

APPENDIX D

QUESTIONNAIRES

Name: Ranjoy Javier

Date: June 10 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: Try to adjust the speed of the conveyor belt to collect more water waste

Name: JOSHUA SUNGA

Date: 6/10/23

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: TRY TO DO A STUDY IN SEGREGATING THE COLLECTED WASTE MORE EFFICIENTLY AND CONVENIENT.

Name: Julius Bibay Date: June 07, 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: You should do a regular testing and calibration of the computer vision to ensure that it is accurately detecting and collecting waste

Name: Mejilla, Rovalyn P. Date: 05 June 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - $\boxed{4}$ - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - $\boxed{3}$ - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - $\boxed{3}$ - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - $\boxed{4}$ - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - $\boxed{4}$ - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - $\boxed{5}$
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - $\boxed{4}$ - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - $\boxed{4}$ - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - $\boxed{4}$ - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - $\boxed{3}$ - 4 - 5
Comments and Suggestions: Do more study on the Kinds of waste that are typicaly found in water bodies in the target area.

Name: RODFERT FERNANDEZ Date: JUNE 10, 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: CONDUCT A PILOT TEST OF THE VEHICLE IN A SMALL BODY OF WATER BEFORE SCALING UP THE PROJECT TO LARGER BODIES OF WATER.

Name: Kristchan C. Lopez Date: June 3, 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: I suggest to add extra small holes in the trash bin for the drainage of the water.

Name: KHRESHIA JARILLA Date: June 3, 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: the vehicle can collect trash but takes time to detect in their webpage.

Name: Helen Mejilla Date: June 7, 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: If possible try to a feature to the vehicle that allows it to detect and collect oil spills or other hazardous material in the water.

Name: Connie Hipolito

Date: June 7. 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: To broaden the project's impact and reach, consider cooperating with regional org. or governmental org.

Name: Paul Leandro Lanot

Date: June 7, 2023

Questionnaire:
1. The prototype is user-friendly and easy to use.
1 - 2 - 3 - 4 - 5
2. Utilizing the prototype is preferable than the traditional way for cleaning and collecting solid waste in water.
1 - 2 - 3 - 4 - 5
3. The boat is made to operate safely and effectively alongside users so they may concentrate on tasks of high value.
1 - 2 - 3 - 4 - 5
4. The prototype's operation and functioning don't require many steps to locate and gather waste in water.
1 - 2 - 3 - 4 - 5
5. The prototype is cost-effective and sustainable.
1 - 2 - 3 - 4 - 5
6. The webpage shows the location, and the user can easily monitor the prototype.
1 - 2 - 3 - 4 - 5
7. The interface of the webpage can be accessed easily.
1 - 2 - 3 - 4 - 5
8. The prototype is capable of recognizing and detecting different kinds of waste that are floating in the water and can also avoid obstacles in its path.
1 - 2 - 3 - 4 - 5
9. Even with a full trash container, the prototype's weight consideration is sufficient to keep the boat afloat.
1 - 2 - 3 - 4 - 5
10. The power management system is enough to maintain the prototype working for a certain period of time.
1 - 2 - 3 - 4 - 5
Comments and Suggestions: Making the boat bigger for large wastes and a functionality in the website where they can control the boat and view the live feed on the camera.

APPENDIX E

HOW TO DUPLICATE THE ROBOAT

RoBoat: An Automated Aquatic Vehicle for Collecting Water Waste Using Raspberry Pi 4 Microcontroller with Computer Vision for Real-time Waste Detection and Monitoring Via Web Page Application

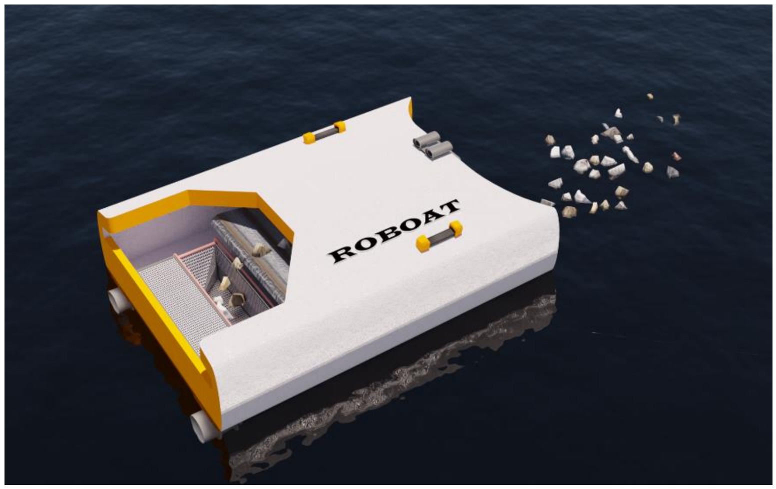
DUPLICATION MANUAL

Version 1.0
July 2023

PART 1 - HARDWARE DUPLICATION

1.1 3D Model of the RoBoat

The proponents used a cloud-based 3D Modelling and CAD software to design the body structure of the RoBoat. The aquatic vehicle will be composed of stainless steel for the structure, DC motors and four air blowers for the movements, a garbage bin to store the waste collected, and three Lead-acid batteries responsible for the robot's power source.

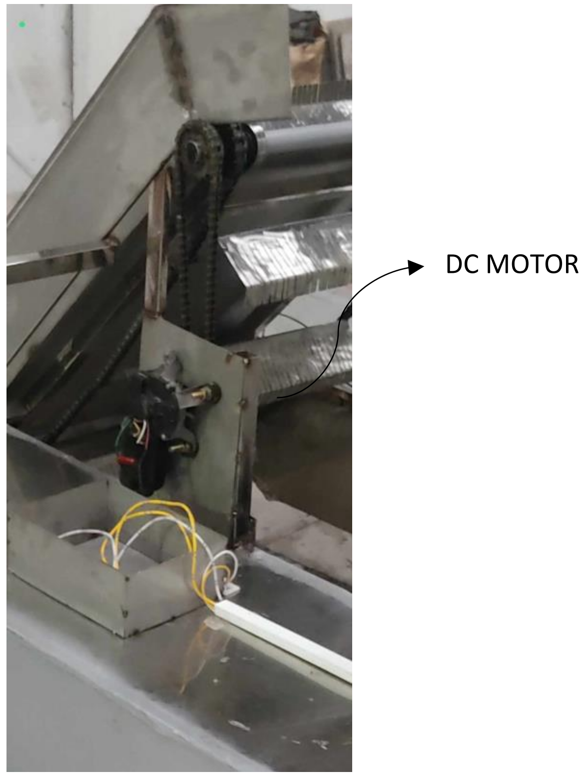


3D Model Design of the RoBoat

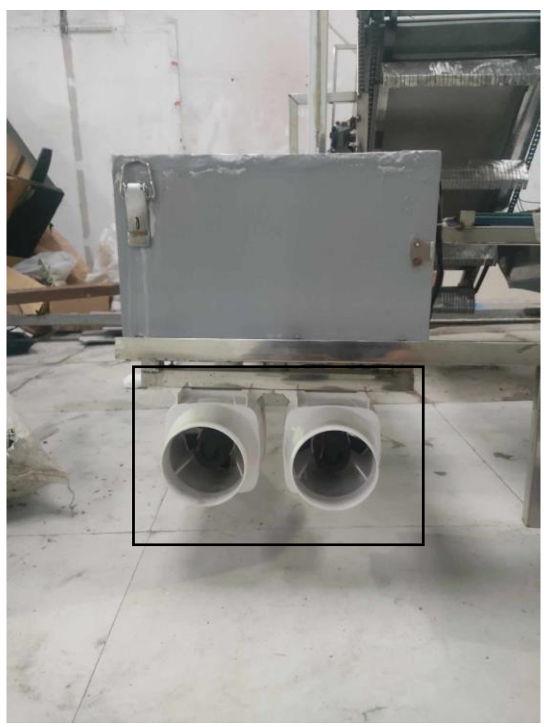
1.2 Parts for the RoBoats Structure

The aquatic vehicle will be comprised of 3mm stainless steel sheets for the frame and the body of the roboat, DC motors for the conveyor belt and four air blowers for the movement of the RoBoat, a trash can for storing the waste collected, weighing scale for the load detection and three Lead-acid batteries for power.

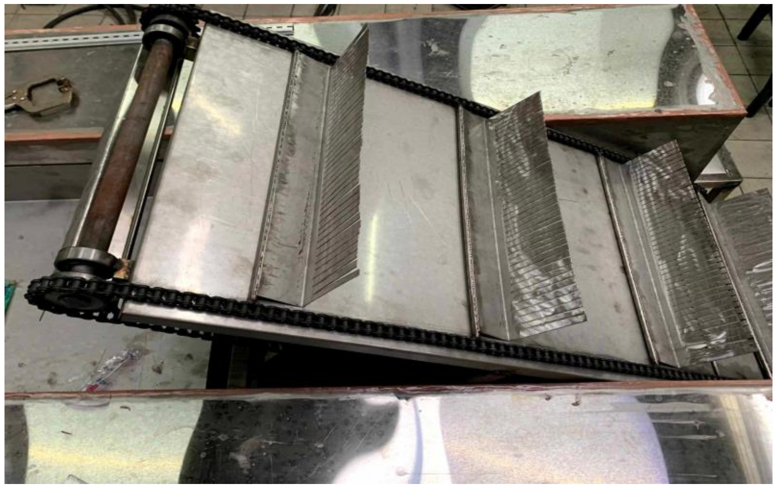
3mm Stainless Steel Sheets



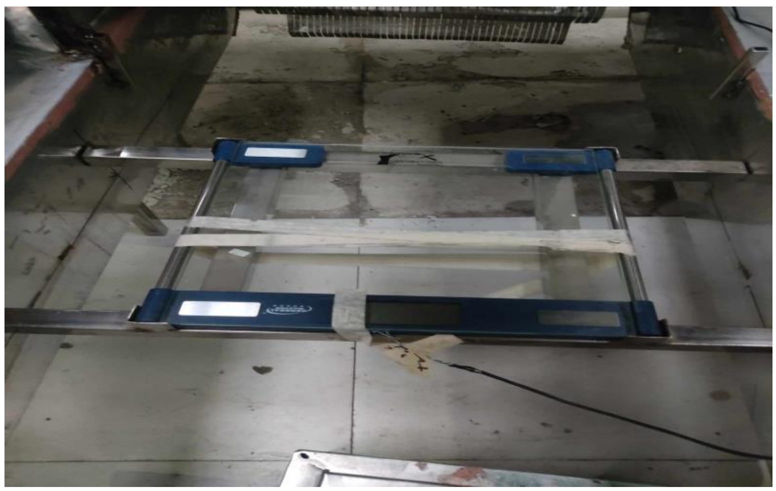
Wiper Motor for the Conveyor Belt



Air Blowers for the Movement



Conveyor Belt with stainless steel teeths



Weighing Scale

Lead Acid Batteries

1.3 Electronic Components

The vehicle automation utilized a Raspberry Pi 4B microcontroller. A PK-910H 1080p Full-HD Webcam for image processing. It also uses an Arduino Mega, sensors (including a ultrasonic sensor for the obstacle detection, GPS module for locating and retrieving, a GSM module for notifying when the garbage storage was full. A weight sensor for trash detection, and relays), an Arduino Nano connected to the Arduino Mega for boat control

A4tech Web Camera, 5V 6 Channel Relay, Electromechanical Relay, Ultrasonic Sensor, Raspberry Pi 4B, Arduino Mega, GPS Module GSM Module, and Arduino Nano

1.4 Assembly

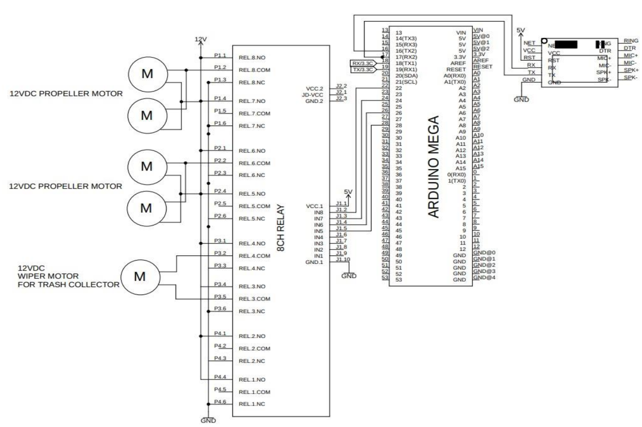
To assemble the structure of the RoBoat we are going to weld the the base of the RoBoat which is made out of Stainless Steel Plates into a 2 rectangles with a length of 5ft , a width of 1.5ft and a height of .5 ft. Assemble and add the conveyor belt in between the 2 rectangle which are going to be the base of the RoBoat with an angle of 45 degrees. As for the garbage bin, you're also going to weld a length of 3ft, width of 1ft and a height of 1ft. The air blowers were located at the bottom at the back of the prototype for easier maneuvering and movement.

Front View of the RoBoat

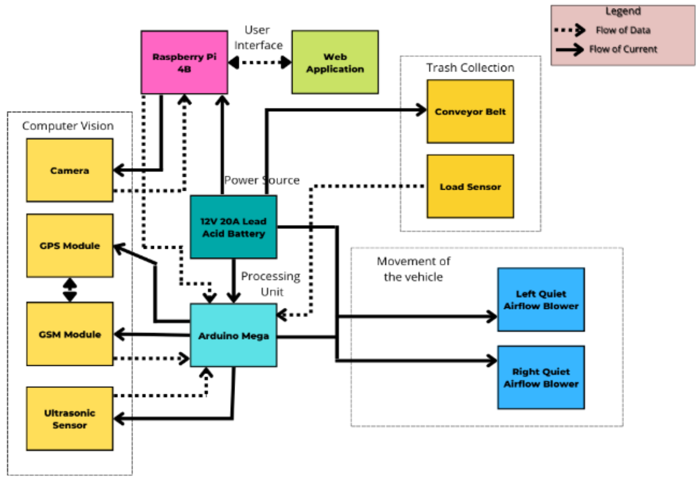
Back View of the RoBoat with the Garbage Bin

For the Electronic components parts and wirings, follow the schematic diagram shown below and the actual sample wirings of the RoBoat's Component box.

Component Box of the Roboat



Schematic Diagram



Block Diagram

PART 2 – SOFTWARE

2.1 Arduino

This study uses Arduino MEGA to receive the data from the raspberry pi 4B to control the aquatic vehicle.

Codes:

```
#include <NewPing.h>2
NewPing Sonar1(8, 9, 400);
float d1;
float distanceThreshold = 30; //30cm
float WeightFull = 5; //30kg

int debug = 0;
#include <SoftwareSerial.h>

// SoftwareSerial Serial3(11, 10); //(RX, TX)
#include <TinyGPS.h>
float flat = 0.0;
float flon = 0.0;
TinyGPS gps;

unsigned long age;
static void smartdelay(unsigned long ms);

#include "HX711.h"
HX711 cell(A0, A1);
```

```
float weight = 0;

// #include <SoftwareSerial.h>
// SoftwareSerial Serial1(11, 10); // TX Pin, RX Pin

#define Button (2)

#define LeftForward_Pin (30)
#define LeftBackward_Pin (32)
#define RightForward_Pin (26)
#define RightBackward_Pin (28)

#define Conveyor_Pin (24)

#include <millisDelay.h>
millisDelay Timer1;
millisDelay CheckWeightTimer;

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);

long RotationInterval = 60000;
long CheckWeightInterval = 10000;

void setup() {
Serial.begin(9600);
Serial1.begin(9600);
Serial3.begin(9600);
// Serial3.listen();
Serial2.begin(115200);
delay(1000);
Serial1.print('Z');
//Buttons
pinMode(Button, INPUT_PULLUP);

//Relays
pinMode(LeftForward_Pin, OUTPUT); digitalWrite(LeftForward_Pin, HIGH);
pinMode(LeftBackward_Pin, OUTPUT); digitalWrite(LeftBackward_Pin, HIGH);
pinMode(RightForward_Pin, OUTPUT); digitalWrite(RightForward_Pin, HIGH);
pinMode(RightBackward_Pin, OUTPUT); digitalWrite(RightBackward_Pin, HIGH);
pinMode(Conveyor_Pin, OUTPUT); digitalWrite(Conveyor_Pin, HIGH);

// lcd.init();
// lcd.backlight();
```

```
Serial2.print("AT\r"); delay(1000);

Serial2.print("ATE0"); delay(1000);
Serial2.print("AT+CMEE=2\r"); delay(1000);
Serial2.print("AT+CMGF=1\r"); delay(1000); //Because we want to send the SMS in
text mode
Serial2.print("AT+CMGDA=\"DEL ALL\"\r"); //Because we want to send the SMS in
text mode

cell.set_scale(22000);
cell.tare(); delay(1000);
// while (1) {
// weight = cell.get_units();
// Serial.println(weight);
// delay(1000);
// }
CheckWeightTimer.start(CheckWeightInterval);

// GetGPS();
// while (flat == 1000) {GetGPS(); delay(1000);}
// DBG("Successful");
}
int Full = 0;
char receivedChar;
void loop() {

if (Serial1.available()) {
receivedChar = Serial1.read();
Serial.println(receivedChar);
if (receivedChar == 'F') {
Forward();
while (1) {if (Serial1.available()) {break;}}
Stop();
}
else if (receivedChar == 'B') {
Backward();
while (1) {if (Serial1.available()) {break;}}
Stop();
}
else if (receivedChar == 'R') {
RotateClockwise();
while (1) {if (Serial1.available()) {break;}}
Stop();
}
else if (receivedChar == 'L') {
RotateCC();
```

```
while (1) {if (Serial1.available()) {break;}}
Stop();
}

else if (receivedChar == 'C') {
digitalWrite(Conveyor_Pin, LOW);
while (1) {if (Serial1.available()) {break;}}
Stop();
}
}

if (Serial.available()) {
receivedChar = Serial.read();

if (receivedChar == '1') {
d1 = Sonar1.ping_cm(); delay(30);
DBG(String(d1));
if ((d1 < distanceThreshold) && (d1 > 0)) {
DBG("Obstacle");
}
else {
DBG("Forward"); digitalWrite(Conveyor_Pin, LOW);
Forward(); delay(5000); Stop();
}
}
else if (receivedChar == '2') {Serial.println(); Backward(); delay(5000); Stop();}
else if (receivedChar == '3') {RotateClockwise(); delay(6000); Stop();}

else if (receivedChar == '4') {

}
}
receivedChar = ' ';
if (CheckWeightTimer.justFinished()) {
GetWeight();
if ((weight > WeightFull) && (Full == 0)) {
Full = 1;
DBG("Sending");
SendMSG2("09914043637","Bin is full");
}
else if (weight < WeightFull - 0.01) {
Full = 0;
}
DBG(String(weight));
CheckWeightTimer.start(CheckWeightInterval);
GetGPS();
```

```
d1 = Sonar1.ping_cm(); delay(30);
Serial.print("!");
Serial.print(d1); Serial.print(",");
Serial.print(weight); Serial.print(",");
Serial.print(flat,7); Serial.print(",");

Serial.print(flon,7); Serial.println("@");
}
}

void Forward() {
digitalWrite(LeftForward_Pin, LOW);
digitalWrite(RightForward_Pin, LOW);
digitalWrite(LeftBackward_Pin, HIGH);
digitalWrite(RightBackward_Pin, HIGH);
}

void Backward() {
digitalWrite(LeftForward_Pin, HIGH);
digitalWrite(RightForward_Pin, HIGH);
digitalWrite(LeftBackward_Pin, LOW);
digitalWrite(RightBackward_Pin, LOW);
}

void RotateClockwise() {
digitalWrite(LeftForward_Pin, LOW);
digitalWrite(RightForward_Pin, HIGH);
digitalWrite(LeftBackward_Pin, HIGH);
digitalWrite(RightBackward_Pin, LOW);

}

void RotateCC() {
digitalWrite(LeftForward_Pin, HIGH);
digitalWrite(RightForward_Pin, LOW);
digitalWrite(LeftBackward_Pin, LOW);
digitalWrite(RightBackward_Pin, HIGH);
}

void Stop() {
digitalWrite(LeftForward_Pin, HIGH);
digitalWrite(RightForward_Pin, HIGH);
digitalWrite(LeftBackward_Pin, HIGH);
digitalWrite(RightBackward_Pin, HIGH);
digitalWrite(Conveyor_Pin, HIGH);
```

```
}

void ToggleLH(long delayings, int relayName) {
digitalWrite(relayName, LOW); delay(delayings);
digitalWrite(relayName, HIGH);
}

void ToggleHL(long delayings, int relayName) {
digitalWrite(relayName, HIGH); delay(delayings);
digitalWrite(relayName, LOW);
}
void RelayLOWwithLimit(int relayName, int MSName) {
if (digitalRead(MSName) == 1){
digitalWrite(relayName, LOW); delay(50);
while(1) {if (digitalRead(MSName) == 0) {break;}}
digitalWrite(relayName, HIGH);
}
}

void LCDprint(int x, int y, String z) {
lcd.setCursor(x,y); lcd.print(z);
}

void LCDClear() {
LCDprint(0,0," ");
LCDprint(0,1," ");
LCDprint(0,2," ");
LCDprint(0,3," ");
}

void DBG(String str) {
if (debug) {Serial.println(str);}
}

void SendMSG2(String PhoneNumber, String Msg) {
Serial2.print("AT+CMGF=1\r"); delay(500); //Because we want to send the SMS in
text mode
Serial2.print("AT+CMGS=\"" + PhoneNumber + "\"\r"); //to be sent to the number
specified.
delay(1000);
Serial2.print(Msg);
Serial2.println();

delay(1000);
Serial2.write(0x1A); //Equivalent to sending Ctrl+Z
delay(8000);
```

```
}

static void smartdelay(unsigned long ms){
unsigned long start = millis();
// Serial3.listen();
Serial3.flush();

do{
while (Serial3.available())
gps.encode(Serial3.read());
} while (millis() - start < ms);
}

void GetGPS(){
smartdelay(1000);
gps.f_get_position(&flat, &flon, &age);
Serial.println();
}

void GetWeight() {
weight = cell.get_units();
if (weight < 0.02) {weight = 0;}
}
```

Python Codes

```
import argparse
import os
import sys
from pathlib import Path
import time

import requests

import torch
import torch.backends.cudnn as cudnn
import serial
SerialData = serial.Serial("/dev/ttyACM0", 9600, timeout = 0.2)
# SerialData = serial.Serial("com6", 9600, timeout = 0.2)
# SerialData.setDTR(False)
# time.sleep(1)
# SerialData.flushInput()
# SerialData.setDTR(True)
# time.sleep(2)
# SerialData.write(bytes("2", "utf-8"))
# reading = SerialData.readline().decode("utf-8")
# reading = SerialData.readline().decode("utf-8")
```

```
# if reading == 'A':
# pass

class MyTimer:
def __init__(self):
self.StartTime = 0
self.TargetTime = 0
self.Force = 0

def start(self, TargetTime):
self.Force = 0
self.StartTime = time.time()
self.TargetTime = TargetTime

def justFinished(self):
# print(self.Force,time.time()-self.StartTime, self.TargetTime)
if self.Force == 1:
return False
if (time.time()-self.StartTime > self.TargetTime):
self.Force = 1
return True
else:
return False
def elapsed(self):
return time.time()-self.StartTime

def remaining(self):
return self.TargetTime - (time.time()-self.StartTime)

def stop(self):
self.Force = 1

Timer1 = MyTimer()
Timer1.justFinished()
Timer1.start(5)

FILE = Path(__file__).resolve()
ROOT = FILE.parents[0] # YOLOv5 root directory
if str(ROOT) not in sys.path:
sys.path.append(str(ROOT)) # add ROOT to PATH
ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative

from models.common import DetectMultiBackend
from utils.dataloaders import IMG_FORMATS, VID_FORMATS, LoadImages,
LoadStreams
from utils.general import (LOGGER, check_file, check_img_size, check_imshow,
```

```
check_requirements, colorstr, cv2,
increment_path, non_max_suppression, print_args, scale_coords,
strip_optimizer, xyxy2xywh)
from utils.plots import Annotator, colors, save_one_box
from utils.torch_utils import select_device, time_sync

print( torch.version.cuda)

@torch.no_grad()
def run(
weights=ROOT / 'yolov5s.pt', # model.pt path(s)
source=ROOT / 'data/images', # file/dir/URL/glob, 0 for webcam
data=ROOT / 'data/coco128.yaml', # dataset.yaml path
imgsz=(640, 640), # inference size (height, width)
conf_thres=0.25, # confidence threshold
iou_thres=0.45, # NMS IOU threshold
max_det=1000, # maximum detections per image
device='', # cuda device, i.e. 0 or 0,1,2,3 or cpu
view_img=False, # show results
save_txt=False, # save results to *.txt
save_conf=False, # save confidences in --save-txt labels
save_crop=False, # save cropped prediction boxes
nosave=False, # do not save images/videos
classes=None, # filter by class: --class 0, or --class 0 2 3
agnostic_nms=False, # class-agnostic NMS
augment=False, # augmented inference
visualize=False, # visualize features
update=False, # update all models
project=ROOT / 'runs/detect', # save results to project/name
name='exp', # save results to project/name
exist_ok=False, # existing project/name ok, do not increment

line_thickness=3, # bounding box thickness (pixels)
hide_labels=False, # hide labels
hide_conf=False, # hide confidences
half=False, # use FP16 half-precision inference
dnn=False, # use OpenCV DNN for ONNX inference
):
source = str(source)
save_img = not nosave and not source.endswith('.txt') # save inference images
is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)
is_url = source.lower().startswith(('rtsp://', 'rtmp://', 'http://', 'https://'))
webcam = source.isnumeric() or source.endswith('.txt') or (is_url and not is_file)
if is_url and is_file:
source = check_file(source) # download
```

```
# Directories
save_dir = increment_path(Path(project) / name, exist_ok=exist_ok) # increment run
(save_dir / 'labels' if save_txt else save_dir).mkdir(parents=True, exist_ok=True) #
make dir

# Load model
device = select_device(device)
model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data, fp16=half)
stride, names, pt = model.stride, model.names, model.pt

imgsz = check_img_size(imgsz, s=stride) # check image size

# Dataloader
if webcam:
view_img = check_imshow()
cudnn.benchmark = True # set True to speed up constant image size inference
dataset = LoadStreams(source, img_size=imgsz, stride=stride, auto=pt)
bs = len(dataset) # batch_size
else:
dataset = LoadImages(source, img_size=imgsz, stride=stride, auto=pt)
bs = 1 # batch_size
vid_path, vid_writer = [None] * bs, [None] * bs

# Run inference
model.warmup(imgsz=(1 if pt else bs, 3, *imgsz)) # warmup
seen, windows, dt = 0, [], [0.0, 0.0, 0.0]
for path, im, im0s, vid_cap, s in dataset:
t1 = time_sync()
im = torch.from_numpy(im).to(device)
im = im.half() if model.fp16 else im.float() # uint8 to fp16/32
im /= 255 # 0 - 255 to 0.0 - 1.0
if len(im.shape) == 3:
im = im[None] # expand for batch dim

t2 = time_sync()
dt[0] += t2 - t1

# Inference
visualize = increment_path(save_dir / Path(path).stem, mkdir=True) if visualize else
False
pred = model(im, augment=augment, visualize=visualize)
t3 = time_sync()
dt[1] += t3 - t2

# NMS
pred = non_max_suppression(pred, conf_thres, iou_thres, classes, agnostic_nms,
```

```
max_det=max_det)
dt[2] += time_sync() - t3

# Second-stage classifier (optional)
# pred = utils.general.apply_classifier(pred, classifier_model, im, im0s)

# Process predictions
for i, det in enumerate(pred): # per image
seen += 1
if webcam: # batch_size >= 1
p, im0, frame = path[i], im0s[i].copy(), dataset.count

s += f'{i}: '
else:
p, im0, frame = path, im0s.copy(), getattr(dataset, 'frame', 0)

p = Path(p) # to Path
save_path = str(save_dir / p.name) # im.jpg
txt_path = str(save_dir / 'labels' / p.stem) + ('' if dataset.mode == 'image' else
f'_{frame}') # im.txt
s += '%gx%g ' % im.shape[2:] # print string
gn = torch.tensor(im0.shape)[[1, 0, 1, 0]] # normalization gain whwh
imc = im0.copy() if save_crop else im0 # for save_crop
annotator = Annotator(im0, line_width=line_thickness, example=str(names))
label = ""
reading = SerialData.read().decode("utf-8")
AccStr = ""
Acc = 0
if reading == '!':
# print("!")
while True:
reading = SerialData.read().decode("utf-8")
if reading == "":
# print("blank")
pass

elif reading == '@':
# print("end")
llli = AccStr.split(",")
obstacle = llli[0]
weight = llli[1]
lat = llli[2]
long = llli[3]

postHttpLink ="https://2mgpttbt63.execute-api.ap-northeast-
1.amazonaws.com/dev/roboat?obstacle="+str(obstacle)+"&weight="+str(weight)+"&lon
```

```
g="+str(long)+"&lat="+str(lat)+"&status=Ongoing"
httplink =
"http://www.domain.com/store.php?d="+str(obstacle)+",weigth="+str(weight)+",long="+
str(long)+",lat="+str(lat)
try:
response = requests.get(url=httplink, timeout=10)
post_response = requests.post(url=postHttpLink, timeout=10)
print(httplink,response.text)
print(postHttpLink,post_response.text)
except:
print(httplink,"Error")
break
else:
# print(reading)

AccStr = AccStr + reading

if len(det):
# Rescale boxes from img_size to im0 size
det[:, :4] = scale_coords(im.shape[2:], det[:, :4], im0.shape).round()

# Print results
for c in det[:, -1].unique():
n = (det[:, -1] == c).sum() # detections per class
s += f"{n} {names[int(c)]}{'s' * (n > 1)}, " # add to string

# Write results

for *xyxy, conf, cls in reversed(det):
if save_txt: # Write to file
xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-1).tolist()
# normalized xywh
line = (cls, *xywh, conf) if save_conf else (cls, *xywh) # label format
with open(f'{txt_path}.txt', 'a') as f:
f.write(('%g ' * len(line)).rstrip() % line + '\n')

if save_img or save_crop or view_img: # Add bbox to image
c = int(cls) # integer class

label = None if hide_labels else (names[c] if hide_conf else f'{names[c]}
{conf:.2f}')

name = f'{names[c]}'
pred = float(f'{conf:.2f}')
print(name,pred)
```

```
if pred > 0.8:
if Timer1.justFinished():
SerialData.write(bytes("1", "utf-8"))
Timer1.start(10)
annotator.box_label(xyxy, name + " FORWARD", color=colors(c,
True))
LOGGER.info("SYSTEM WILL GO FORWARD. (Send forward
signal to Arduino)")
im0 = annotator.result()
else:
annotator.box_label(xyxy, name + "
COOLDOWN"+str(int(Timer1.remaining())), color=colors(c, True))
else:
annotator.box_label(xyxy, name + " <confidence("+str(pred)+")",
color=colors(c, True))
if save_crop:

save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] /
f'{p.stem}.jpg', BGR=True)

if view_img:
if p not in windows:
windows.append(p)
cv2.namedWindow(str(p), cv2.WINDOW_NORMAL |
cv2.WINDOW_KEEPRATIO) # allow window resize (Linux)
cv2.resizeWindow(str(p), im0.shape[1], im0.shape[0])
cv2.imshow(str(p), im0)
cv2.waitKey(1) # 1 millisecond

# Save results (image with detections)
if save_img:
if dataset.mode == 'image':
cv2.imwrite(save_path, im0)
else: # 'video' or 'stream'
if vid_path[i] != save_path: # new video
vid_path[i] = save_path
if isinstance(vid_writer[i], cv2.VideoWriter):
vid_writer[i].release() # release previous video writer
if vid_cap: # video
fps = vid_cap.get(cv2.CAP_PROP_FPS)

w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
else: # stream
fps, w, h = 30, im0.shape[1], im0.shape[0]
save_path = str(Path(save_path).with_suffix('.mp4')) # force *.mp4 suffix
```

```
on results videos
vid_writer[i] = cv2.VideoWriter(save_path,
cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))
cv2.imwrite("final.jpg", im0)
time.sleep(1)

# Print time (inference-only)
LOGGER.info(f'{s}Done. ({t3 - t2:.3f}s)')

# Print results
t = tuple(x / seen * 1E3 for x in dt) # speeds per image
LOGGER.info(f'Speed: %.1fms pre-process, %.1fms inference, %.1fms NMS per
image at shape {(1, 3, *imgsz)}' % t)
if save_txt or save_img:
s = f"\n{len(list(save_dir.glob('labels/*.txt')))} labels saved to {save_dir / 'labels'}"
if save_txt else ''
LOGGER.info(f"Results saved to {colorstr('bold', save_dir)}{s}")

if update:
strip_optimizer(weights) # update model (to fix SourceChangeWarning)

def parse_opt():
parser = argparse.ArgumentParser()
parser.add_argument('--weights', nargs='+', type=str, default=ROOT /
r'runs/train/exp18/weights/best.pt', help='model path(s)')
parser.add_argument('--source', type=str, default=0, help='file/dir/URL/glob, 0 for
webcam')
parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml',
help='(optional) dataset.yaml path')
parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640],
help='inference size h,w')
parser.add_argument('--conf-thres', type=float, default=0.7, help='confidence
threshold')
parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')
parser.add_argument('--max-det', type=int, default=1000, help='maximum detections
per image')
parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
parser.add_argument('--view-img', action='store_true', help='show results')
parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')

parser.add_argument('--save-conf', action='store_true', help='save confidences in --
save-txt labels')
parser.add_argument('--save-crop', action='store_true', help='save cropped prediction
boxes')
parser.add_argument('--nosave', action='store_true', help='do not save images/videos')
parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or
```

```
--classes 0 2 3')
parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')
parser.add_argument('--augment', action='store_true', help='augmented inference')
parser.add_argument('--visualize', action='store_true', help='visualize features')
parser.add_argument('--update', action='store_true', help='update all models')
parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to
project/name')
parser.add_argument('--name', default='exp', help='save results to project/name')
parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok,
do not increment')
parser.add_argument('--line-thickness', default=3, type=int, help='bounding box
thickness (pixels)')
parser.add_argument('--hide-labels', default=False, action='store_true', help='hide
labels')
parser.add_argument('--hide-conf', default=False, action='store_true', help='hide
confidences')

parser.add_argument('--half', action='store_true', help='use FP16 half-precision
inference')
parser.add_argument('--dnn', action='store_true', help='use OpenCV DNN for ONNX
inference')
opt = parser.parse_args()
opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1 # expand
print_args(vars(opt))
return opt

def main(opt):
check_requirements(exclude=('tensorboard', 'thop'))
run(**vars(opt))
if __name__ == "__main__":
opt = parse_opt()
main(opt)
```

1. Setting up Raspberry Pi for programming the Automation

In this study, the Raspberry Pi 4 Model B was used. In order to start and upload the Raspberry Pi codes, manually set up the following cables and materials:

- **Raspberry Pi 4B**
- **HDMI Cable**
- **LED Monitor**

Process:

1) The Aquatic vehicle was programmed using the Python programming language.
2) A USB cable was connected from the laptop to the raspberry pi microcontroller board to upload the program.
3) After the entire system has been installed and the central processing unit, Raspberry Pi, has been powered up, the program runs, and the ultrasonic sensor is activated.

```python
import argparse
import os
import sys
from pathlib import Path
import time
import requests

import torch
import torch.backends.cudnn as cudnn
import serial
SerialData = serial.Serial("/dev/ttyACM0", 9600, timeout = 0.2)
# SerialData = serial.Serial("com6", 9600, timeout = 0.2)
# SerialData.setDTR(False)
# time.sleep(1)
# SerialData.flushInput()
# SerialData.setDTR(True)
# time.sleep(2)
# SerialData.write(bytes("2", "utf-8"))
# reading = SerialData.readline().decode("utf-8")
# reading = SerialData.readline().decode("utf-8")
#   if reading == 'A':
#       pass

class MyTimer:
    def __init__(self):
        self.StartTime = 0
        self.TargetTime = 0
        self.Force = 0
```

Sample Python Codes of the RoBoat

2. Front End Development

A. Layout Designing:

1. Figma was utilized to create the sample plan for the overall structure and layout of the web page.
2. Using editing softwares such as Adobe Photoshop and Illustrator, design elements suitable for the web page's style were created.

B. HTML Markup:

1. HTML tags were utilized to define the structure and content of the web page.
2. The layout was divided into sections using appropriate tags such as '<header>', '<main>', and '<footer>'.
3. With each section, content was arranged using the aforementioned tags.

C. Integration with Cloud MongoDB:

1. Using appropriate credentials and configurations the MongoDB Database service was set up, and a database was created to store data.
2. MongoDB queries were executed to retrieve, insert, update and delete data from the database.
3. The retrieved data was dynamically displayed on the web page using HTML and ReactJS.

3. Camera Vision

A. Training Images

1. Create Dataset

   YOLOv5 models must be trained on labeled data to learn class objects in that data. There are two options for creating your dataset before you start training:

   Use Roboflow to create your dataset in YOLO format
2. Select a Model

   Select a pretrained model to start training from. Here we select YOLOv5s, the second-smallest and fastest model available.
3. Train

   Train a YOLOv5s model on COCO128 by specifying dataset, batch-size, image size and either pretrained --weights yolov5s.pt (recommended), or randomly initialized --weights '' --cfg yolov5s.yaml (not recommended). Pretrained weights are auto-downloaded from the latest YOLOv5 release.
4. Next Steps

Once your model is trained you can use your best checkpoint best.pt to:

- Execute CLI or Python inference on newly created photos and videos.
- Check accuracy on train, val, and test splits.
- TensorFlow, Keras, ONNX, TFlite, TF.js, CoreML, and TensorRT export formats
- Improve performance via evolving hyperparameters.
- Improve your model by collecting real-world photographs and incorporating them into your dataset.

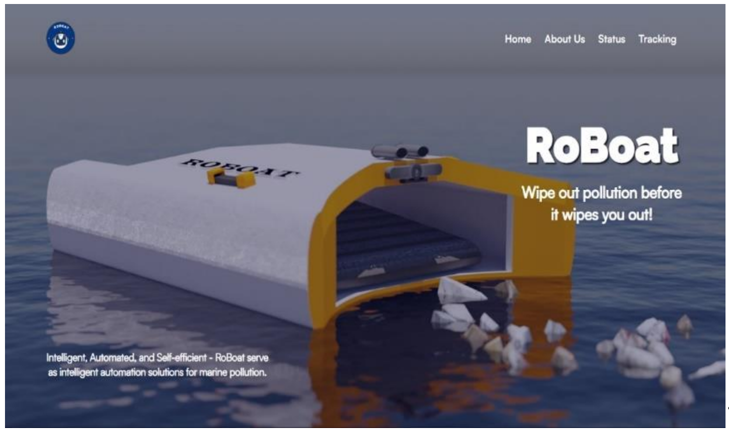
5. Training progress monitoring:

   Progress updates, including loss, learning rate, and elapsed time per iteration, were monitored during training.
6. Model evaluation:

   Following training completion, the model's performance was evaluated using validation or test images.

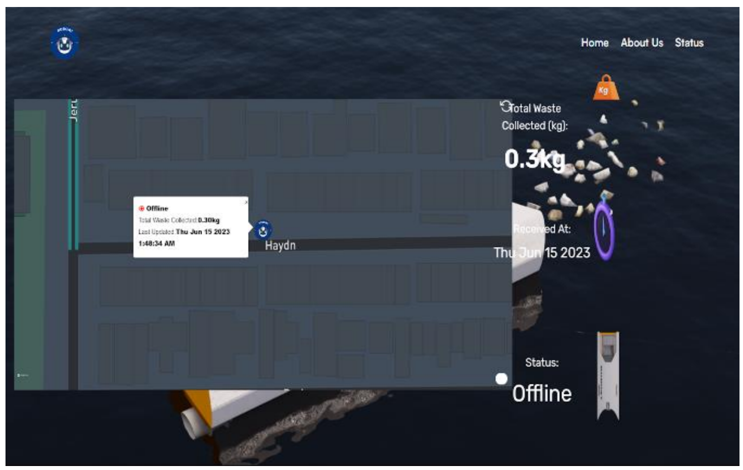
B. Vision to Arm Movement Integration

Using Arduino IDE, the programmed RoBoat movements was paired up with individual coordinate that will represent the location of floating water waste in bodies of water in a captured frame by the Camera of the Raspberry Pi.



,

Landing Page of the RoBoat



Working integrated website to RoBoat

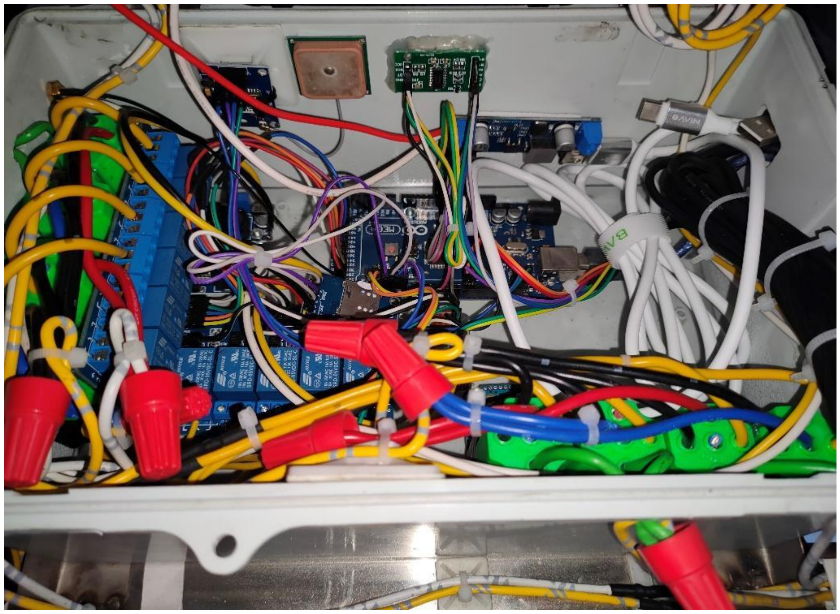
Providing Realtime Data in Waste collected

And Exact Location through GPS Tracker

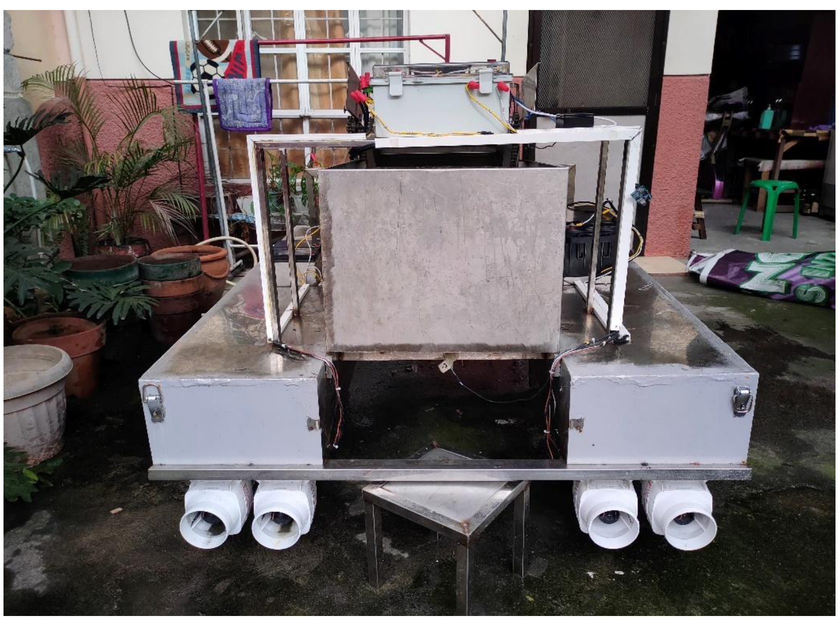
APPENDIX F

THE ROBOAT

I. Component Box



II. Back View



III. Roboat at Appreciate 2023

RoBoat Prototype Displayed at Appreciate 2023

Team RoBoat Picture with Engr. Velasco and Engr. Catipon

‘

APPENDIX F

USER’S MANUAL

RoBoat: An Automated Aquatic Vehicle for Collecting Water Waste Using Raspberry Pi 4 Microcontroller with Computer Vision for Real-time Waste Detection and Monitoring Via Web Page Application

User's Manual

Version 1.0
July 2023

INTRODUCTION

This manual describes, explains, and provides detailed instructions on implementing the project effectively. It has all the information needed to build the RoBoat. Moreover, instructions to create the website for real-time monitoring will also be included in this manual, along with the steps to connect it to the control unit of the RoBoat.

Components attached to the boat cannot typically detect what is happening on the system. As a result, a website is built and created to monitor the device's entire system.

If you have any inquiries or concern that is not explained or covered in this user manual, please kindly contact the Lead Program Developer at 0935-162-7465 or email us at roboat.support@gmail.com

OVERVIEW

The RoBoat's Web Application is an online website that supports the RoBoat's hardware unit. The website consists of a Landing Page, About Us Page, and Status Page.

1. **Landing Page** – This shows the 3D model of the RoBoat to show people insight into how it looks, and it is also shown on the landing page the simple description of the RoBoat and our tagline.

2. **About Us Page** - This shows and tells why we create the RoBoat, our vision, objectives, and goals. On this page, you can also see who the creators are and their roles.

3. **Status Page** – This shows the RoBoat unit's current status, whether the RoBoat is Online or Offline. Here is the total weight collected, the time and date on when the RoBoat received data, and the GPS tracking that locates the exact location of the RoBoat.

SETTING UP THE ROBOAT

STEPS ON SETTING UP THE ROBOAT:

1. Connect the designated wires to the lead acid batteries to power the system. Upon pushing the power button from the electronics component box, you will see a red-light indicator to see if the RoBoat is on.
2. Put the Raspberry Pi 4B Microcontroller in plae and connect it to the component box connector. After noticing the red light indicator, this will turn on the RoBoat's automation.
3. After everything is set, launch the Roboat into the water, test it, start image processing, and start collecting trash nearby.

Figure 1. Setting up the RoBoat upon deployment

GETTING STARTED

USING THE ROBOAT WEB APPLICATION

1. Open your browser and make sure you are connected to the internet. In the search bar, go to www.roboat-tupm.online. Upon getting into the website, you can see the landing page and click on about us for more information about the RoBoat.

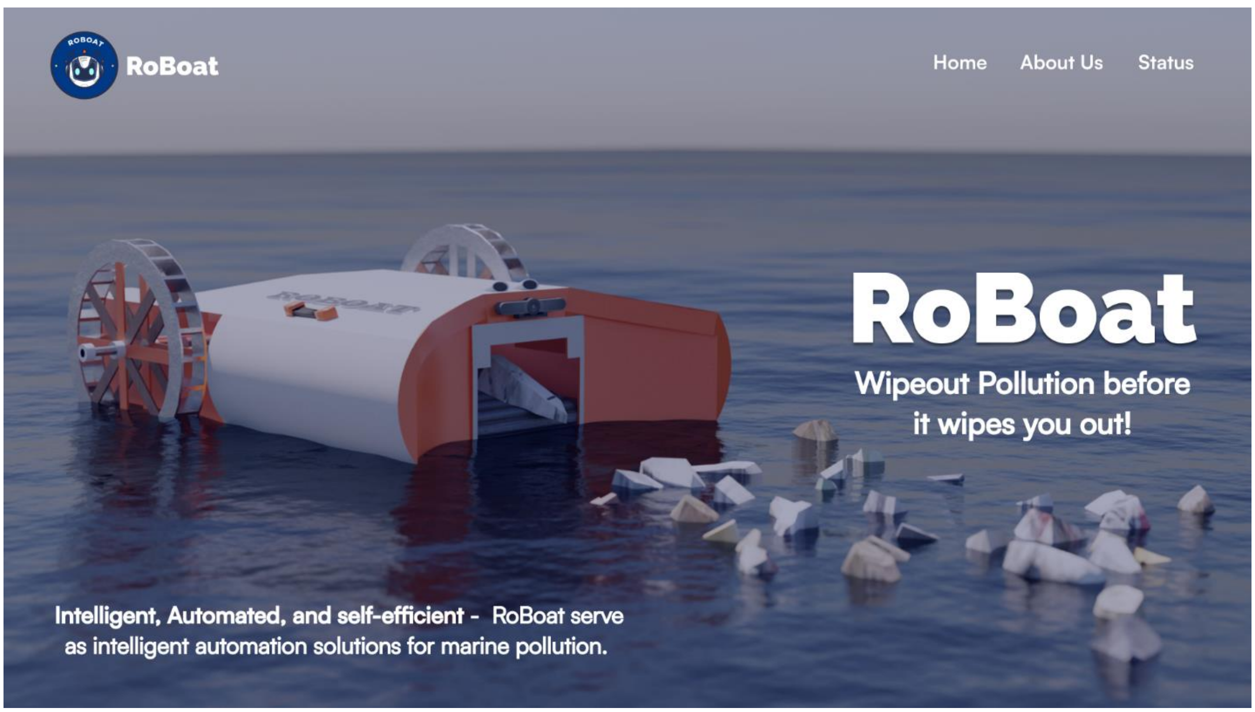


Figure 2. RoBoat Landing Page

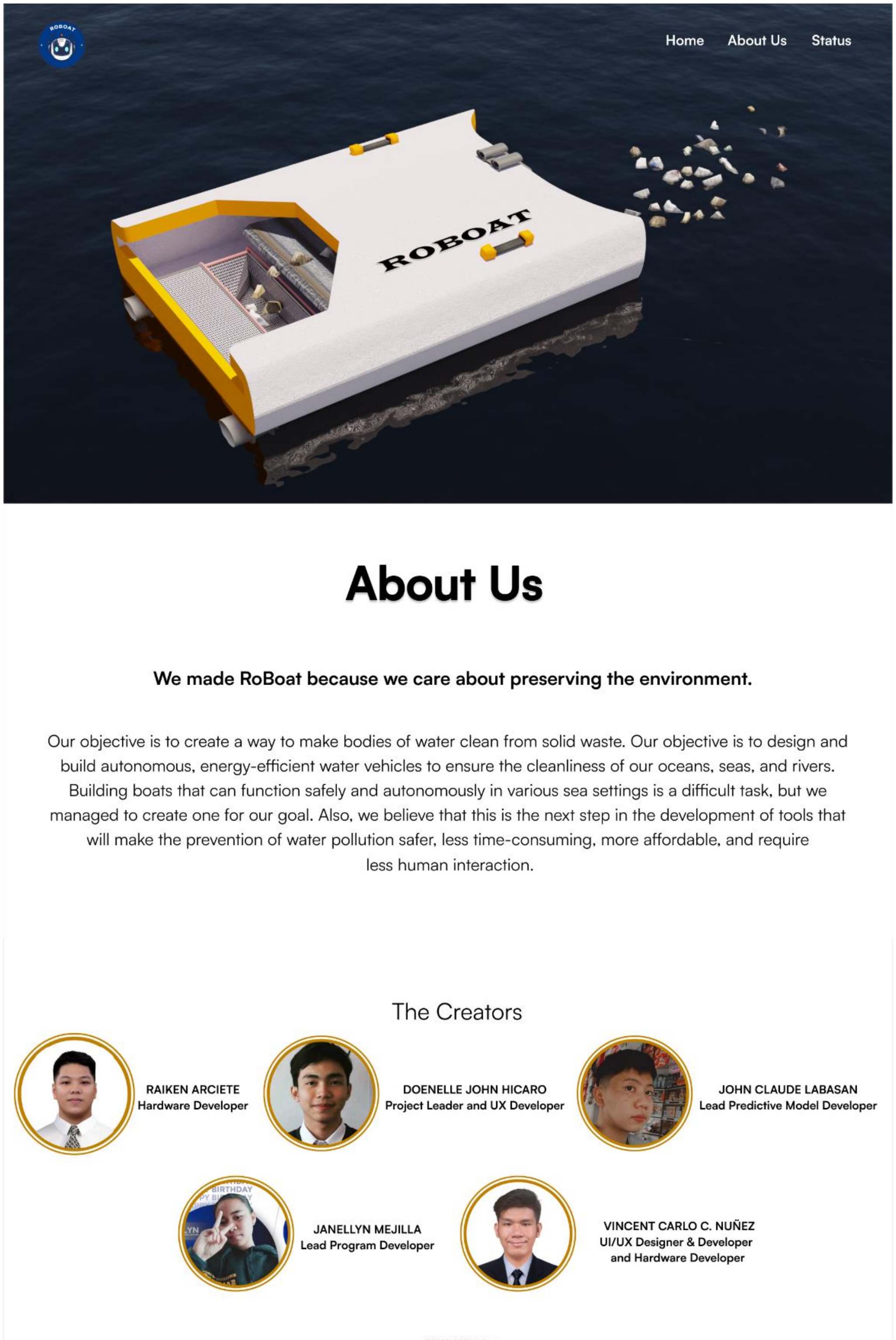


Figure 3. RoBoat About Us Page

STATUS AND MONITORING

2. Click on the Status in the Navigation Bar to monitor the RoBoat. Here you can see the real-time weight collected of the RoBoat, the date and time on when the RoBoat received a data, the current location of the RoBoat through GPS, and if the RoBoat is currently online or offline.

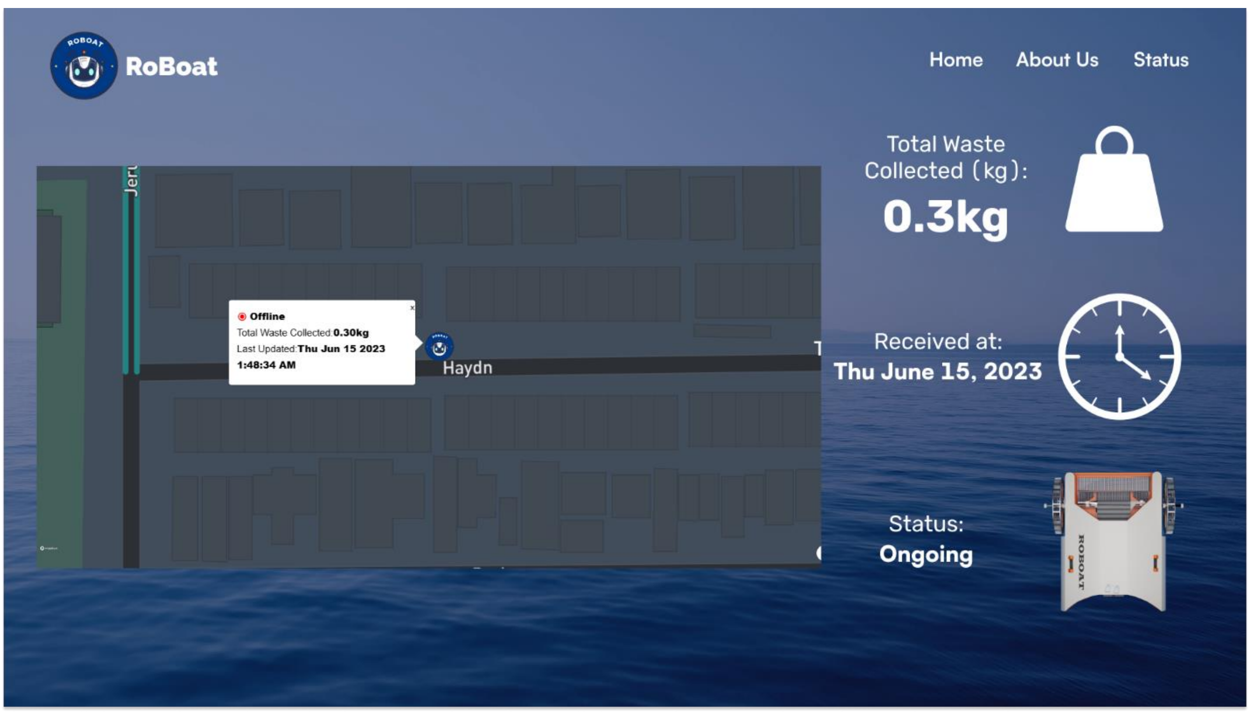


Figure 3. RoBoat Status Page

SUPPORT

Name	Email	Role
Doenelle John Hicaro	doenellejohncarl.hicaro@tup.edu.ph	Project Leader
Raiken Arciete	arcieteraiken@gmail.com	Hardware Developer
John Claude Labasan	labasan.john16@gmail.com	Hardware and Predictive Model Developer
Janellyn Mejilla	janellyn.mejilla@tup.edu.ph	Lead Predictive and Backend Developer
Vincent Carlo C. Nuñez	vincentcarlo.nunez@tup.edu.ph	UI designer and Hardware Developer

APPENDIX I

RESEARCHERS' PROFILE

DOENELLE JOHN CARL HICARO

PROJECT LEAD AND HARDWARE DEVELOPER, ROBOAT

+63 9989386222

doenellejohncarlhicaro@gmail.com

Haydn st. North Olympus Subdivision, Kaligayahan, Quezon City

PROFILE

A highly motivated undergraduate student pursuing a Bachelor of Science in Electronics Engineering, with a solid academic background and a keen interest in technical innovation. He gotten hands-on experience developing and executing electronic circuits, enhancing his practical skills in the sector. His zeal and ambition help them to stay on the cutting edge of technological breakthroughs in the electronics business.

EXPERIENCE

TUP IRTC PEED | INTERN
August 2022 - September 2022

Key responsibilities:

- Assisting in research activities
- Conducting data analysis
- Documenting and reporting

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES-MANILA 2019- Present

BS IN ELECTRONICS ENGINEERING

AMA COMPUTER COLLEGE - FAIRVIEW 2017-2019

SCIENCE, TECHNOLOGY, ENGINEERING AND MATHEMATICS (STEM)

CORE SKILLS

- Able to work with minimum supervision
- Positive Working Attitude
- Up to date knowledge of the IoT products
- Strong communication skills
- Proficient in time management
- Familiarity with Cisco basic routing techniques
- Skilled in Python programming
- Proficient in NI Multisim for circuit design and simulation

CERTIFICATES

Master IP Addressing and Subnetting for CCNA
MNET IT | May 2022

RAIKEN E. ARCIETE

HARDWARE DEVELOPER, ROBOAT

+63 9976898164

arcieteraiken@gmail.com

146 B2 N. Domingo St. Cubao Quezon City

PROFILE

An enthusiastic and driven Electronics Engineering student with an insatiable curiosity for exploring the vast world of technology. Through his years at Technological University of the Philippines, he was able to gain extensive knowledge about electronic systems and circuit, fostering a passion in creating designs that creates cutting-edge solutions in electronics. Dedicated to mastering the art of electronics engineering and strives to build a dynamic force in the ever-evolving realm of technology.

EXPERIENCE

CREOTEC PHILIPPINES, INC. | PCB QUALITY ASSURANCE INSPECTOR
2019

LTO - EAST AVENUE Q.C | INTERN
AUGUST 2022 - SEPTEMBER 20222

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES-MANILA 2019- Present
BS IN ELECTRONICS ENGINEERING

UNIVERSITY OF PERPETUAL HELP - MOLINO, BACOOR 2017-2019
SCIENCE, TECHNOLOGY, ENGINEERING AND MATHEMATICS STRAND

CORE SKILLS

- Skilled in Basic Python programming
- Cadence Virtuoso Layout
- Cisco Packet Tracer
- NI Multisim
- IP Addressing and Subnetting
- MATLAB Programming
- Literate in Microsoft and Google Technologies
- Pleasing personality with capability of arrangement
- Good communication skills and fast learner

CERTIFICATIONS

Master IP Addressing and Subnetting for CCNA
MNET IT | May 2022

JOHN CLAUDE R. LABASAN

HARDWARE AND PREDICTIVE MODEL DEVELOPER

+63 9566327832

labasan.john16@gmail.com

210 Kendi Master Rd., Bagbaguin, Meycauayan, Bulacan

PROFILE

Motivated and dedicated Electronics Engineering student with a passion for innovative technology and a drive to contribute to the field. Seeking opportunities to apply theoretical knowledge, hands-on experience, and problem-solving skills in a dynamic and challenging environment. Committed to continuous learning and professional growth in the realm of electronics and engineering.

EXPERIENCE

DWWW 774 | INTERN
August 2022 - September 2022

Key responsibilities:
- Audio Editing of News Reports
- Arranging Radio Commercial Playback
- Live Broadcasting of News Reports
- Technical and Digital Support

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES-MANILA 2019- Present

BS IN ELECTRONICS ENGINEERING

ADAMSON UNIVERSITY - ERMITA MANILA 2017-2019

SCIENCE, TECHNOLOGY, ENGINEERING AND MATHEMATICS (STEM)

CORE SKILLS

- IP addressing and Subnetting
- Python Programming
- Cisco Basic Routing Technique
- MATLAB Programming
- NI MULTISIM
- Cisco Packet Tracer
- GNS3
- Network Design and Simulation
- Productivity Tools (MS Office, MS Teams, Google Drive)

CERTIFICATES

FORTINET NSE CERTIFICATION PROGRAM
- Certified Associate - Fortinet Level 3 (May 2023)
- Certified Associate - Fortinet Level 2 (May 2023)
- Certified Associate - Fortinet Level 1 (May 2023)

MNET IT
- Master IP Addressing and Subnetting for CCNA (May 2022)
- Neural Networks and Natural Language Processing

VINCENT CARLO C. NUÑEZ

FRONTEND AND HARDWARE DEVELOPER, ROBOAT

+63 9685153421

nunez.vincentcarlo@gmail.com

Barangay Carsadang Bago II Palazzo Bello 1 Imus, Cavite

PROFILE

Highly motivated BS Electronics Engineering student with a strong academic background and a passion for technological innovation. Equipped with comprehensive knowledge of electronic systems, communication networks, and signal processing techniques. Demonstrated ability to apply theoretical concepts to practical situations, coupled with hands-on experience in designing and implementing electronic circuits.

EXPERIENCE

DWWW 774 | INTERN

August 2022 - September 2022

Key responsibilities:

- Audio Editing of News Reports
- Arranging Radio Commercial Playback
- Live Broadcasting of News Reports
- Technical and Digital Support

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES-MANILA 2019- Present

BS IN ELECTRONICS ENGINEERING

PANGASINAN STATE UNIVESITY - URDANETA CITY 2017-2019

SCIENCE, TECHNOLOGY, ENGINEERING AND MATHEMATICS (STEM)

CORE SKILLS

- IP addressing and Subnetting
- Python Programming
- Cisco Basic Routing Technique
- MATLAB Programming
- NI MULTISIM
- Cisco Packet Tracer
- GNS3
- Critical Thinking
- Can work with minimum supervision
- Strong Multitasking Ability
- Good Communicator
- Management Skills
- Positive Working Attitude

CERTIFICATES

FORTINET NSE CERTIFICATION PROGRAM

- Certified Associate - Fortinet Level 3 (May 2023)
- Certified Associate - Fortinet Level 2 (May 2023)
- Certified Associate - Fortinet Level 1 (May 2023)

MNET IT

- Master IP Addressing and Subnetting for CCNA (May 2022)

JANELLYN P. MEJILLA

LEAD PROGRAMMER &
PREDICTIVE MODEL DEVELOPER

+63 9351627465

jahmejilla@gmail.com

437 F. Dela Cruz St. Caridad, Cavite City

PROFILE

As an electronics engineering student, I am passionate about exploring the intricate world of electrical systems, circuits, and devices. I possess a strong foundation in mathematics and physics, which enables me to understand and analyze complex electrical phenomena. My dedication to learning and problem-solving drives me to excel in this field.

ACADEMIC BACKGROUND

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES-MANILA 2019- Present
BS IN ELECTRONICS ENGINEERING

CAVITE NATIONAL HIGH SCHOOL-SENIOR HIGH SCHOOL 2017-2019
SCIENCE, TECHNOLOGY, ENGINEERING AND MATHEMATICS STRAND

EXPERIENCE

QUALITY ANALYST
RADIX TELECOMMUNICATIONS
2022

PHILIPPINE ROBOTICS TEAM
2017-2019

MICROSOFT TECHNICAL SUPPORT
2022

SAMSUNG TECHNICAL SUPPORT
2023

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES-MANILA 2019- Present
BS IN ELECTRONICS ENGINEERING

CAVITE NATIONAL HIGH SCHOOL 2017-2019
SENIOR HIGH SCHOOL
SCIENCE, TECHNOLOGY, ENGINEERING AND MATHEMATICS STRAND

CORE SKILLS

- Skilled in Basic Python programming
- Cadence Virtuoso Layout
- Cisco Packet Tracer
- NI Multisim
- IP Addressing and Subnetting
- MATLAB Programming
- Technical Skills
- Programming skills
- Communication and Teamwork

CERTIFICATES

Master IP Addressing and Subnetting for CCNA
MNET IT | May 2022