IOT – Based Under Water Object Detection

Swathi[1], Nidhi V[2], Pooja BM[3], Pooja Mahesh Gangnaik[4], Prakruthi G[5]

[1]Asst Prof. Dept. of Electronics and Communication Engineering, Bangalore Institute of Technology, Bangalore, VTU, Karnataka.

[2,3,4,5,] Dept. of Electronics and Communication Engineering, Bangalore Institute of Technology, Bangalore, VTU, Karnataka, India.

[1]swathi@bit-bangalore.edu.in, [2]nidhiv1103@gmail.com , [3]poojabm1908@gmail.com, [4]poojagangnaik@gmail.com, [5]prakruthi274@gmail.com

Abstract

An IoT-based real-time underwater object detection system is developed to identify marine debris, aquatic species, and submerged structures by integrating edge computing with machine learning. A waterproof camera coupled with a Raspberry Pi captures live underwater footage, processed locally using OpenCV and YOLO algorithms for efficient and low-latency object recognition. The detected information is transmitted via wireless IoT protocols such as Wi-Fi or LoRa for remote monitoring and analysis. The system is cost-effective, scalable, and adaptable to various underwater conditions, enabling continuous data acquisition for marine research, pollution detection, and environmental monitoring. Its modular design supports additional sensors like temperature and pH modules for enhanced ecological insight. Future developments aim to incorporate advanced deep learning and computer vision models to further improve detection accuracy and robustness under challenging underwater conditions, contributing to smart and sustainable marine observation.

INTRODUCTION

The advancement of the Internet of Things (IoT) and artificial intelligence (AI) has enabled the creation of intelligent systems for marine monitoring and environmental analysis. This paper presents an IoT-based real-time underwater object detection system designed to identify marine debris, aquatic species, and submerged structures. A waterproof camera integrated with a Raspberry Pi captures live underwater footage, which is processed locally using OpenCV and the YOLO algorithm for efficient, low-latency detection. The system transmits results wirelessly via IoT protocols for remote monitoring and analysis.

Designed for cost-effectiveness and scalability, the system supports additional sensors such as temperature and pH modules to enhance environmental insight. Experimental testing demonstrated reliable detection performance under varying water conditions. Future enhancements include integrating deep learning models and cloud analytics to improve detection accuracy and adaptability. The proposed solution offers a sustainable, intelligent approach to real-time underwater monitoring and marine ecosystem management.

LITERATURE SURVEY

Underwater monitoring systems have come a long way—from simple sensor-based setups to advanced IoT and AI-driven detection networks. Researchers have explored everything from embedded alert systems to deep learning–based object detection, all aiming to make underwater environments safer, smarter, and more observable. Still, challenges like visibility, real-time responsiveness, and cost-effective deployment continue to surface. What follows is a look at some of the most important related works and how this project builds on them to create a more complete, efficient underwater detection system.

IoT-Based Underwater Monitoring with Arduino: One early effort, “Monitoring and Alert Systems for Underwater Data Centres using Arduino” by Yashas Kadambi et al. (2021), showed how embedded microcontrollers could handle underwater sensing. Their MASUD system tracked parameters like temperature and corrosion using Arduino and basic sensors. It was not focused on vision-based detection but laid the groundwork for reliable IoT monitoring underwater. It proved that embedded systems could survive harsh marine conditions and trigger real-time alerts—an idea that carries directly into today’s smart IoT detection platforms.

Machine Learning for Fish Detection: Another big leap came from Radha N et al. (2022), who built a YOLOv5-based system to automatically detect fish species in underwater videos. They trained several YOLOv5 variants and found YOLOv5M to be the best performer, hitting an F1-score of 94.9%. The project tackled problems like low visibility and lighting variation—exactly the kind of issues this current system also faces. Their results validated YOLO’s strength in real-time underwater detection and shaped later model choices.

IoT-Enabled Marine Litter Classification: In 2023, Ramakrishnan Raman and Shreyasi Bhattacharya took things further with an IoT-based ocean cleanup system. Their setup used CNN models to detect and classify marine litter from underwater images, sending data wirelessly to a central dashboard. It proved how IoT and AI can team up for large-scale environmental monitoring. The concept of connecting underwater vision systems to cloud dashboards directly inspired the IoT-based architecture in the present project.

Improved YOLO Frameworks for Underwater Detection: J. Zhang et al. (2023) pushed YOLO even further with an improved YOLOv5 model customized for underwater conditions. They used special augmentation and loss functions to overcome turbidity and poor lighting which are common problems in aquatic imagery. Their study showed that tailoring models for underwater data can dramatically improve detection accuracy. This insight guided the dataset preparation and fine-tuning strategies used in our system.

An AIoT-Based Trash Detection with YOLOv8: Finally, Biplov Paneru et al. (2024) combined YOLOv8 with IoT in a system for real-time underwater trash management. With edge computing on a Raspberry Pi and data streaming via ThingSpeak, they achieved scalable, low-latency performance. Their work demonstrated that even small, affordable setups could support powerful AI-driven underwater monitoring. This project builds on that same principle—deploying YOLO models on the Raspberry Pi for real-time, cost-effective detection and IoT-based data sharing.

Table I. Comparative Survey of Parking Spot Availability and Reservation System.

Author (s)	Techniques	Outcomes
Yashas Kadambi et al. (2021)	Designed an Arduino-based monitoring and alert system for underwater data centres using multiple sensors.	Demonstrated effective real-time condition monitoring and alert generation, but lacked vision-based detection and AI integration.
Radha N et al. (2022)	Implemented real-time fish detection using YOLOv5 variants trained on custom underwater datasets.	Achieved high accuracy (F1-score 94.9%) in fish detection; validated YOLO's capability in low-visibility environments.
Ramakrishnan Raman & Shreyasi Bhattacharya (2023)	Developed an IoT-enabled CNN-based marine litter detection and classification system.	Enabled real-time debris monitoring with IoT dashboard visualization; focused mainly on pollution tracking.
J. Zhang et al. (2023)	Proposed an improved YOLOv5 framework with underwater-specific data augmentation and loss functions.	Enhanced detection accuracy and robustness under turbid and low-light conditions through model customization.
Biplov Paneru et al. (2024)	Built an AIoT-based underwater trash detection system using YOLOv8 and edge computing on Raspberry Pi.	Achieved efficient, real-time detection and IoT-based visualization; demonstrated low-latency, scalable underwater monitoring.

Table I gives a quick overview of what researchers have explored so far with IoT-based underwater object detection and monitoring systems. Most of the work focuses on combining IoT, embedded hardware, and deep learning models to make underwater observation smarter, faster, and more reliable. The main goal? Automate underwater monitoring, detect marine debris or aquatic species in real time, and send those results wirelessly for analysis and environmental action.

Early systems leaned heavily on basic embedded setups using microcontrollers like Arduino. They worked well for sensing parameters such as temperature or pressure, but they couldn't "see" underwater objects or recognize patterns visually. Then came the shift toward machine learning and computer vision—projects started using YOLO and CNN-based models for detecting fish, litter, and other objects in underwater videos. These new approaches made detection far more accurate and autonomous but also increased the need for powerful processors and well-annotated datasets.

Recent studies have taken things a step further by merging AI with IoT, creating real-time, edge-powered underwater detection networks. Systems running YOLOv5 and YOLOv8 on Raspberry Pi boards have shown that high accuracy and low latency are possible even with compact, low-cost hardware. However, challenges like water turbidity, lighting variations, and the cost of maintaining underwater equipment still make deployment tricky. The ongoing research, including the system proposed in this work, aims to overcome these limitations through optimized models, modular IoT designs, and scalable data communication solutions.

I.SYSTEM ANALYSIS

A. Functional Requirements

The system uses a waterproof camera and Raspberry Pi running YOLO and OpenCV to detect underwater objects like debris, fish, and structures in real time, displaying results on both the monitor and LCD screen.

An ultrasonic sensor measures object distance, and the ESP32 transmits detection data wirelessly via IoT protocols such as Wi-Fi or LoRa for remote monitoring and analysis.

The setup ensures real-time performance with minimal latency, reliable operation under varied underwater conditions, and secure data transmission through encrypted communication.

Its modular design supports adding sensors like temperature and pH modules, making the system scalable, adaptable, and suitable for future deep learning and multi-node expansions.

B. Non-Functional Requirements

The system delivers real-time performance with minimal latency (1–2 seconds), ensuring smooth detection and display of underwater objects without delays or interruptions.

It operates reliably under varied underwater conditions such as low light and moderate turbidity, maintaining stability during continuous, long-term monitoring.

The architecture is scalable and secure, supporting encrypted communication (TLS/WPA2) between devices while allowing easy integration of additional sensors or upgraded deep learning models.

Designed for efficiency and adaptability, the system remains cost-effective, power-efficient, and

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

modular—making maintenance simple and enabling expansion for larger monitoring networks.

C. Major System Modules

Detection Module: Captures live underwater footage using a waterproof camera and processes it on the Raspberry Pi with OpenCV and YOLO for real-time object recognition.
IoT Communication Module: Uses the ESP32 to transmit detection data and distance readings wirelessly via Wi-Fi or LoRa to a remote monitoring dashboard.
Sensor Module: Employs an ultrasonic sensor to measure object distance and supports additional sensors like temperature and pH for environmental monitoring.
Monitoring and Data Module: Displays detection results and logs data for analysis, allowing remote users to track performance and system status in real time.

II.PROPOSED METHODOLOGY

The proposed system combines IoT technology, machine learning, and edge computing for real-time underwater object detection. A waterproof camera connected to a Raspberry Pi captures underwater footage, processed locally using OpenCV and the YOLO model for object recognition. Detected objects and distance readings are transmitted via the ESP32 module to a remote monitoring dashboard, ensuring fast, low-latency data communication.

A. System Workflow

The camera captures live underwater images, the YOLO model detects and classifies objects, and the ultrasonic sensor measures their distance. The ESP32 transmits detection results and sensor data wirelessly to the IoT dashboard for real-time visualization.

B. Image Processing and detection Module:

The Raspberry Pi runs the YOLO model and OpenCV library to identify and classify underwater objects in live video. This module performs frame-by-frame analysis, drawing bounding boxes and labels on detected items. It ensures accurate recognition even under low-light or turbid conditions, enhancing detection reliability for marine research and monitoring applications.

C. Sensor and IoT Communication Module:

An ultrasonic sensor measures the distance of detected objects and sends readings to the Raspberry Pi. The ESP32 microcontroller then transmits both detection and distance data wirelessly using Wi-Fi or LoRa protocols. This real-time communication enables remote monitoring and storage of results on a cloud-based platform for further analysis or visualization.

D. Data Management and Monitoring Module

The system stores detection logs, timestamps, and sensor readings in a local or cloud-based database. The remote dashboard displays object detections, distance values, and status indicators in real time. Administrators can review performance, monitor system health, and analyze environmental data to track underwater activity and pollution trends.

E. System Control and Expansion Module

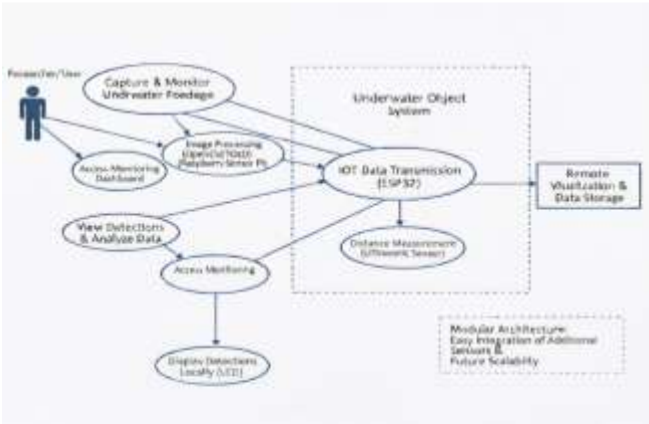
The modular design supports easy integration of additional sensors like temperature, pH, or dissolved oxygen sensors. These inputs provide a more comprehensive understanding of underwater environments. The system can also be expanded into a network of multiple IoT nodes, each capturing and analysing data independently for large-scale marine observation.

III.SYSTEM DESIGN

The proposed system follows an integrated IoT architecture where the Raspberry Pi acts as the main processing unit, handling live underwater image capture and object detection using OpenCV and the YOLO algorithm. The ESP32 microcontroller manages IoT communication, transmitting detection data and sensor readings wirelessly to the monitoring dashboard. The ultrasonic sensor measures object distances. All detected information is displayed locally on an LCD and transmitted for remote visualization, enabling real-time monitoring and analysis.

A. Use Case Diagram

The Use Case diagram shows how the user interacts with the underwater object detection system. The researcher or user captures and monitors underwater footage, while the system



handles image processing, object recognition, distance measurement, and IoT data transmission. The dashboard allows users to view detections and analyze stored data for environmental studies. The modular architecture ensures easy integration of additional sensors and future scalability.

Fig. 1. Use Case Diagram

B. Sequence Diagram

The sequence diagram below represents the flow of operations in the IoT-based real-time underwater object detection system, showing how different components interact step by step during the detection and monitoring process.

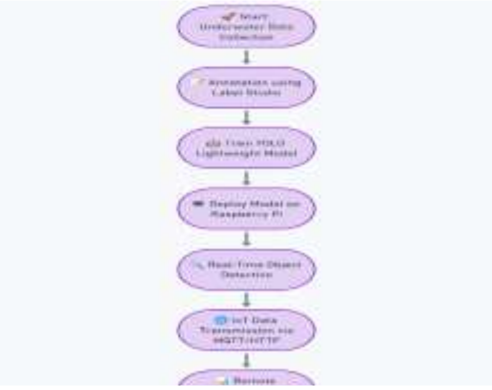


Fig. 2.

Sequence Diagram

C. Activity Diagram

The activity diagram represents the workflow of the IoT-based real-time underwater object detection system. The process starts with live video capture from a waterproof camera connected to the Raspberry Pi. The captured frames are processed using OpenCV and the YOLO model to detect underwater objects such as debris, aquatic species, or submerged structures. Once detection occurs, the Raspberry Pi triggers the ultrasonic sensor to measure the distance of the detected object. The processed results and distance data are sent to the ESP32, which handles IoT communication. The ESP32 displays the results on the local LCD screen and transmits the data wirelessly to a remote monitoring dashboard for real-time analysis. If an object is detected within a set distance, the buzzer alerts nearby users. The system continuously logs and updates all detection events, ensuring accurate and consistent underwater monitoring.

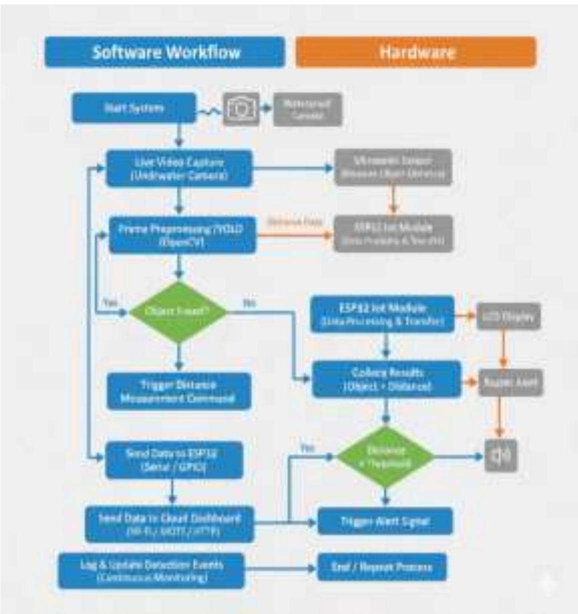


Fig. 3. Activity Diagram

IV.IMPLEMENTATION AND RESULTS

This section outlines the tech components forming the setup,how, they rebuilt, also the process flow seen in the diagrams.

A. Hardware Implementation

1. Raspberry Pi (4GB)

The Raspberry Pi (4GB) serves as the main processing unit of the system. It runs the operating system and executes the YOLO-based object detection model for real-time analysis of underwater video streams. The Raspberry Pi processes frames received from the waterproof camera, performs object detection, and generates bounding boxes and labels. Its compact size, sufficient memory, and support for Python and OpenCV make it suitable for edge computing applications.

Fig. 4. Raspberry Pi (4GB)

2. ESP32 Microcontroller:

The ESP32 microcontroller is used for IoT communication and control operations. It receives detection status information from the Raspberry Pi and manages wireless data transmission using built-in Wi-Fi. The ESP32 also controls peripheral devices such as the LCD display and ultrasonic sensor. Its low power consumption and reliable connectivity make it ideal for real-time IoT-based monitoring systems.

Fig. 5. ESP32 Microcontroller

3. Waterproof Camera

The waterproof camera is used to capture live underwater images and video. It is enclosed in a waterproof casing to ensure reliable operation in submerged conditions. The camera continuously captures underwater visuals and sends the video feed to the Raspberry Pi for real-time processing and object detection.

Fig. 6. Waterproof Camera

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

4. **Ultrasonic Sensor**

The ultrasonic sensor is used for obstacle detection and proximity measurement. It works by transmitting ultrasonic waves and measuring the time taken for the echo to return after hitting an object. This sensor helps in detecting nearby physical obstacles, enhancing system awareness and safety.

Fig. 7. Ultrasonic Sensor

5. **LCD Display**

The LCD (Liquid Crystal Display) is used to provide local visual feedback in the proposed IoT-based underwater object detection system. It displays important system information such as object detection status, alert messages, and system operation updates. The LCD allows users to monitor the system output without relying on a monitor or remote dashboard, which is especially useful during field deployment and testing.

Fig. 8. LCD Display

B. Software Implementation

1. **Raspberry Pi OS**

Raspberry Pi OS, a Linux-based operating system optimized for Raspberry Pi, serves as the primary platform for the underwater object detection system. It provides a stable and lightweight environment for running Python-based machine learning models and hardware control. With built-in support for USB cameras, GPIO interfaces, and networking, it efficiently manages real-time video processing and IoT communication. The OS also ensures security, reliability, and scalability for continuous underwater monitoring operations.

Fig. 8. Raspberry Pi OS

2. **Programming Languages:**

Python is used on the Raspberry Pi for image capture, frame processing, YOLO-based object detection, and visualization. Its rich libraries and simplicity make it ideal for computer vision tasks. Arduino C is used for programming the ESP32 microcontroller to manage IoT communication, LCD display, ultrasonic sensor, and buzzer. The combination enables efficient coordination between high-level processing and low-level hardware control.

3. **Development Tools: Arduino IDE and Thonny IDE**

The Arduino IDE is used to write, compile, and upload ESP32 programs, offering built-in support for Wi-Fi communication and peripheral control. Thonny IDE is used on the Raspberry Pi for Python development, allowing easy debugging, package management, and interaction with hardware for real-time testing and execution.

4. **Libraries and Frameworks:**

YOLO serves as the core object detection framework due to its high speed and accuracy. OpenCV handles video capture, image preprocessing, and bounding box rendering. NumPy supports numerical computations, while Pandas manages detection data and logs. MQTT and HTTP protocols enable IoT-based communication for transmitting detection results to remote dashboards.

5. **Dataset Tools: Label Studio for Annotation**

Label Studio, an open-source annotation tool, is used to label underwater images by drawing bounding boxes around objects such as debris and aquatic organisms. The platform supports YOLO format exports, ensuring seamless integration with the training pipeline. Its project management features enhance labeling consistency and data quality, improving model accuracy for underwater detection.

Fig. 9. Label Studio Annotation

6. **Model Training: YOLO-Based Training Environment**

YOLO (You Only Look Once) is employed for real-time object detection, processing entire images in a single pass to predict bounding boxes and class probabilities efficiently. Its lightweight design allows deployment on the Raspberry Pi for real-time underwater detection.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

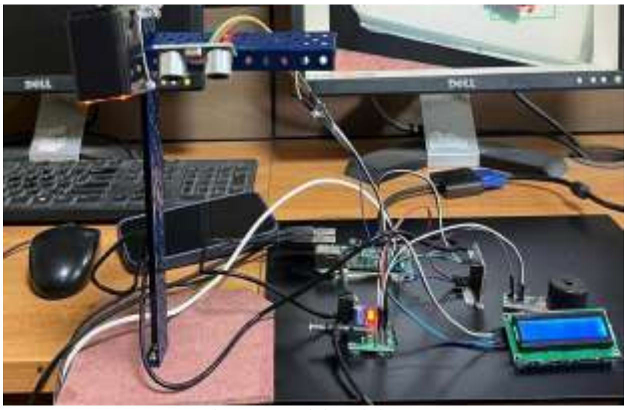


Fig. 10. Model Training

V.OUTPUT WORKFLOW

1. HARDWARE IMPLEMENTATION RESULTS

The IoT-based underwater object detection system was successfully designed and implemented, integrating all hardware components such as the Raspberry Pi, waterproof camera, ESP32, LCD, and sensors into a unified framework. The waterproof camera captured live underwater footage, which was processed by a YOLO-based model running on the Raspberry Pi for real-time object detection. The system accurately identified and labelled various underwater objects, including marine debris, fish, and submerged structures, with bounding boxes and class names. An ultrasonic sensor measured the distance of detected objects, and the values were displayed on the LCD screen in centimetres for immediate reference. Through IoT-based communication, detection results were transmitted wirelessly to a remote monitoring dashboard, enabling real-time observation and analysis. Overall, the system demonstrated high efficiency, cost-effectiveness, and scalability, making it suitable for marine research, pollution monitoring, and underwater exploration applications.

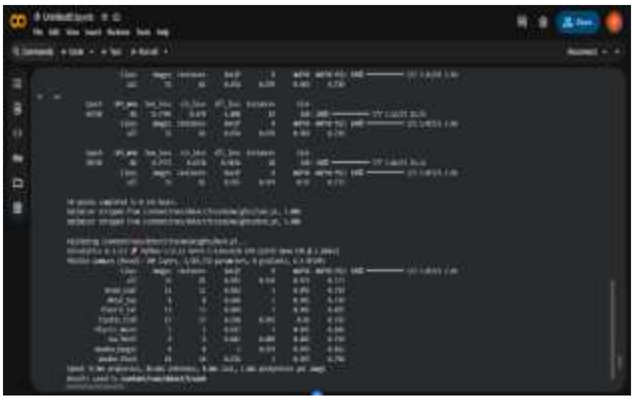


Fig. 11. Final Prototype Hardware Output

2. Vehicle Reservation

The system was continuously tested in real-time underwater conditions, successfully capturing and monitoring live footage without interruption. High detection accuracy was achieved for trained underwater objects, although performance varied slightly with changes in water clarity and object distance. The Raspberry Pi maintained excellent real-time processing speed with minimal delay, ensuring quick and reliable object detection. The system also demonstrated stable performance under different lighting and environmental conditions, though a slight reduction in accuracy was noted in turbid or low-visibility water. Throughout extended testing, both hardware and software functioned reliably with no crashes or data losses. Repeated trials produced consistent detection results, with stable distance measurements and accurate labeling, confirming the system's overall reliability, robustness, and efficiency in continuous underwater monitoring scenarios.

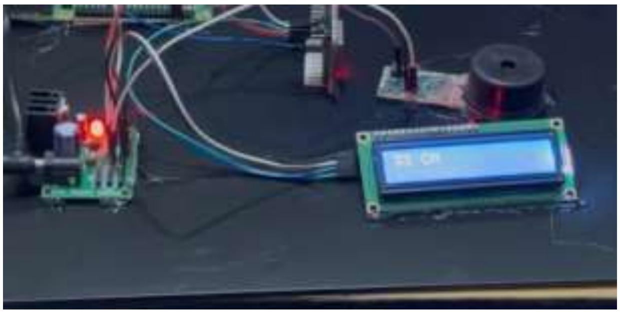


Fig. 12. Ultrasonic sensor distance output shown on LCD module

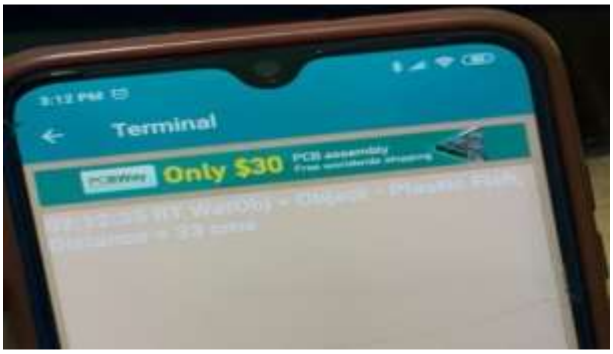


Fig. 13. Bluetooth terminal output showing detected object type and measured distance

VI.CONCLUSION

The IoT-based underwater object detection system presented in this project was successfully designed and implemented using a combination of Raspberry Pi, ESP32, a waterproof camera, and machine learning techniques. The system effectively captures real-time underwater video footage and processes it using a YOLO-based object detection model deployed on edge hardware. The detected underwater objects are accurately identified and displayed with bounding boxes and labels, demonstrating the feasibility of real-time underwater monitoring using low-cost embedded devices. The integration of ultrasonic distance measurement further enhances the system by providing real-time object proximity information, which is displayed on the LCD screen for improved safety and awareness. The seamless coordination between hardware and software components validates the reliability and practical usability of the proposed system. The project highlights a cost-effective and scalable solution for underwater surveillance, environmental monitoring, and marine research applications. IoT-based wireless communication enables remote monitoring and continuous data transmission, reducing the need for manual inspection or expensive underwater equipment. Although factors such as water clarity, lighting conditions, and dataset limitations influence detection accuracy, the system maintains stable and consistent performance under normal conditions. Overall, the project provides a strong foundation for future enhancements, including improved deep learning models, cloud-based data analytics, automated alert systems, and integration with autonomous underwater vehicles. This work contributes meaningfully toward sustainable underwater exploration and environmental protection using intelligent IoT solutions. Overall, this project demonstrates how low-cost IoT hardware combined with deep learning techniques can be effectively used for real-time underwater object detection. The system reduces dependence on manual monitoring and expensive equipment while improving efficiency and safety in underwater observation. By enabling continuous monitoring and intelligent detection.

VII.FUTURE SCOPE

In the future, the underwater object detection system will be enhanced to perform efficiently under challenging conditions such as turbid, muddy, or polluted waters where visibility is poor. Real-world datasets from rivers, lakes, and coastal areas will be used along with advanced data augmentation and image enhancement techniques to improve clarity and detection accuracy. The system's capabilities will be expanded to identify a wider range of underwater objects, including various plastic types, debris, cables, pipelines, and marine species, by training on larger and more diverse datasets. Future versions will integrate improved YOLO architectures, cloud-based analytics, and GPS modules for real-time tracking, data storage, and long-term monitoring. The system could also be paired with autonomous underwater vehicles (AUVs) or remotely operated vehicles (ROVs) for large-scale, continuous exploration. These advancements will make the solution more intelligent, scalable, and practical—supporting marine research, pollution detection, and environmental protection efforts to promote sustainable ocean management. In addition, incorporating AI-driven decision support and automated alert systems can enhance real-time responsiveness and operational safety. Overall, these future improvements will increase the system's reliability and impact, making it a valuable tool for large-scale marine conservation and underwater exploration initiatives.

. REFERENCES

1. Yashas Kadambi, Vishal R., and Roopa Ravish, "Monitoring and Alert Systems for Underwater Data centres using Arduino," IEEE Conference, 2021.
2. Radha N., Swathika R., and Shreya P. S., "Automatic Fish Detection in Underwater Videos using Machine Learning," Sixth International Conference on I-SMAC, 2022.
3. Ramakrishnan Raman and Shreyasi Bhattacharya, "IoT-Based Ocean Cleanup: Detecting and Classifying Marine Litter using CNN," Global Conference on Information Technologies and Communications (GCITC), 2023.
4. Hongyi Xia and Lixin Tan, "Research on Underwater Object Detection Algorithm Based on Improved YOLOv5," International Conference on SMC-IOT, 2023.
5. Biplov Paneru et al., "Towards Underwater Sustainability with AIoT: Underwater Trash Management using YOLOv8," IEEE, 2024.
6.] Joseph Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
7. S Raspberry Pi Foundation, "Raspberry Pi Documentation," Official Documentation.
8. OpenCV Documentation, "Real-Time Computer Vision using OpenCV," Open Source Computer Vision Library.