Artificial Intelligence-Based Floating Rescue Robot for Victim Detection in Water Emergencies

Aishwarya[1,*], Ankitha V[1], Bhavana M B[1], Keerthana D P[1], Jayashree S[1]

[1] Department of Electronics and Communication Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Karnataka, India

***Corresponding author email: [add corresponding author email]**

Abstract — Rescue operations after floods, drownings in lakes and canals, and boating mishaps need a fast first assessment, but sending responders into the water exposes them to concealed hazards such as debris and strong flow. We build a low-cost floating robot that helps responders find people in water by joining remote driving, live video, automatic person detection, and position reporting. Its hull is a twin-pipe catamaran frame made of polyvinyl chloride, propelled by two geared direct-current motors that turn paddle wheels. A smartphone sends direction commands over Bluetooth to an STM32 microcontroller, which switches the motors through an L298N driver board. A camera module (ESP32-CAM) sends live video by Wi-Fi to a laptop, where a Python program built on the OpenCV library applies a MobileNet single-shot detector and outlines any person found with a box and a status message. A NEO-6M Global Positioning System receiver sends the robot's latitude and longitude back to the phone. Tests in a water tank showed that driving, video, detection, and position reporting can run together in a modular design that students can reproduce. The robot is steered by hand and serves only as an observation aid: its detector finds the generic person class and cannot tell whether someone is drowning, and detection accuracy, delay, wireless range, and battery life remain unmeasured. Self-navigation, obstacle avoidance, and a rescue aid are left for future work.

Keywords — floating rescue robot; MobileNet single-shot detector; person detection; ESP32-CAM; GPS; wireless control

I. INTRODUCTION

A person caught in a flood, a canal, a lake, or a boating accident can be in serious danger within minutes. Rescuers first need to know where the victim is and what the surroundings look like, yet reaching the spot may mean entering water with debris, submerged obstacles, or strong currents. Recent work on water safety applies deep learning to this problem. Fixed-camera systems detect drowning in swimming pools in real time on low-cost hardware [2], [5], and vehicle-based systems combine detection with an unmanned surface vehicle [1] or with aerial images [4]. Remotely operated rescue robots with a live camera have also been reported [3]. However, the systems that combine detection with a moving platform use several deep-learning models and a monitoring station, and the low-cost robots leave identification entirely to the operator.

This paper addresses that gap with a simple, affordable, student-buildable floating platform. The contributions are:

- A low-cost catamaran-style floating platform made of polyvinyl chloride pipes, with differential paddle-wheel propulsion.
- A smartphone-controlled movement sub-system built around an STM32 microcontroller, a Bluetooth module, and an L298N motor driver.
- A Wi-Fi video link from an ESP32-CAM to a laptop that runs MobileNet single-shot detector person detection and shows a bounding box and a status message.
- Global Positioning System (GPS) location reporting of the robot, and a documented test procedure with its limitations stated clearly.

The system is a proof-of-concept observation aid. The robot is steered manually, and the word "autonomous" in earlier descriptions of the project refers only to the automatic detection of persons in the video. The remainder of the paper is organised as follows. Section II reviews related work. Section III describes the proposed methodology. Section IV presents the results and discussion. Section V concludes.

II. RELATED WORK

Fixed-camera drowning detection. Amer et al. run a YOLOv11 model on a Raspberry Pi 5 and send alerts to lifeguards when a possible drowning is seen [2]. Zhang et al. improve the YOLO11n model for pools and report a drowning-class mAP50 of 94.1% with 2.02 million parameters [5]. These systems depend on fixed boundaries, clear water, and controllable lighting, and have no moving platform or location output.

Detection combined with a vehicle. Srijiranon et al. propose Robobuoy, which uses YOLO12m to recognise drowning individuals and YOLOv5m to track an unmanned surface vehicle that intervenes autonomously [1]. Bai et al. propose a lightweight YOLOv5s model for finding persons in unmanned aerial vehicle images, working together

with unmanned surface vessels [4]. Both approaches are more complex than a student-level build, and the second uses aerial rather than surface-level images.

Remotely operated rescue robots. Shahria et al. describe an underwater robot controlled from a laptop that streams webcam video to the operator [3]. The operator must inspect the video unaided, and automatic person detection is not the focus. TABLE I summarises the five works.

TABLE I. Summary of the reviewed works

Work	Technique	Drawback for this project
Srijiranon et al. [1]	YOLO12m drowning detection and YOLOv5m tracking of an autonomous surface vehicle	Two deep-learning models and a monitoring station; complex to build
Amer et al. [2]	YOLOv11 on Raspberry Pi 5 with alarm and mobile alerts	Fixed pool surveillance; no moving platform or location output
Shahria et al. [3]	Raspberry Pi robot with webcam and brushless motors	Operator-only observation; person detection not the focus
Bai et al. [4]	Lightweight YOLOv5s for small targets in aerial images	Aerial images; no surface-level camera
Zhang et al. [5]	Improved YOLO11n for pools (94.1% drowning mAP50)	Fixed-camera pool scenes; no mobile platform

The present work differs from each of these in three ways. It uses a moving surface platform, unlike the fixed-camera systems [2], [5]. It uses one pre-trained detector on a laptop rather than several deep-learning models and a monitoring station [1], [4]. And it adds automatic person detection and GPS reporting to a remotely driven camera robot [3]. It does not claim autonomous rescue, guaranteed victim identification, or field-validated rescue performance.

III. METHODOLOGY

The design has five sub-systems, namely movement control, video, human detection, GPS, and power (Fig. 1). Keeping them separate lets each be tested alone and keeps the main controller simple.

A. System Architecture

Commands tapped in the Serial Bluetooth Terminal app on a smartphone reach an HC-05 Bluetooth module, which passes them as serial data to the STM32 microcontroller. The microcontroller sets the logic inputs of the L298N driver, and the driver supplies the two motors that rotate the paddle wheels. The ESP32-CAM [9] records video and serves it as a Wi-Fi stream to a laptop, where Python and OpenCV [8] apply the MobileNet single-shot detector. Position data from the NEO-6M GPS receiver [10] travels back over the Bluetooth link and is displayed as text in the phone terminal. Two separate wireless paths are therefore used: Bluetooth carries the short movement commands and the GPS text, and Wi-Fi carries the video, which needs a much higher data rate.

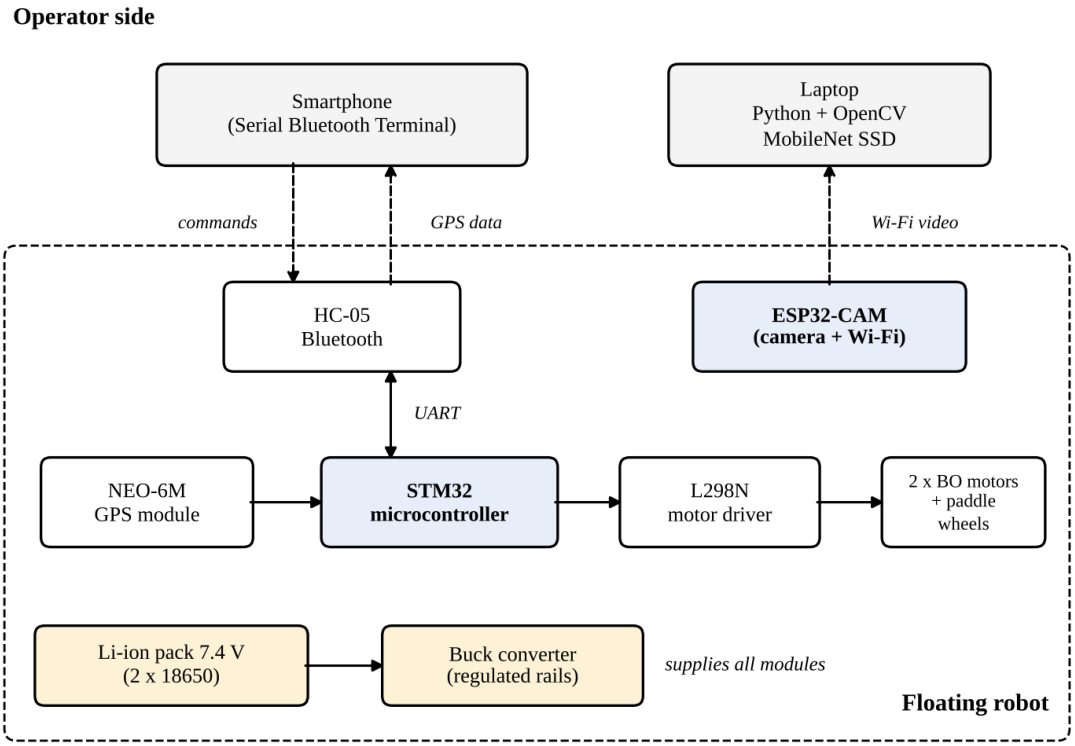

Fig. 1. Block diagram of the floating rescue robot.

B. Mechanical Design and Hardware

The hull consists of two parallel polyvinyl chloride pipes joined by cross pieces, forming a catamaran (Fig. 2). The gap between the pipes provides sideways stability, and a central deck holds the electronics, battery, camera, and motor mounts. The battery and other heavy items sit low near the middle, with the electronics kept clear of the water line. Each of the two 100 RPM geared direct-current motors turns a paddle wheel. Running both in the same direction drives the robot ahead or astern, while slowing or reversing a single motor swings it to one side.



Fig. 2. Top view of the fabricated floating rescue robot prototype.

The STM32 microcontroller takes each character arriving on its UART and turns it into motor outputs, and it ignores invalid or incomplete characters so the robot stays stopped. Because the microcontroller cannot supply motor current, an L298N dual H-bridge [11] sits between them; its internal voltage drop leaves the motors with slightly less than the battery voltage. Power comes from two 18650 lithium-ion cells in series (7.4 V), and a buck converter lowers this to the level needed by the STM32 and the ESP32-CAM. The NEO-6M sends serial NMEA sentences, and its coordinates count only when the receiver reports a valid fix.

C. Software and Control Algorithm

On start-up the embedded program keeps the motors off and waits for a character from the HC-05 (Fig. 3). A valid character selects a movement state, and the matching L298N inputs and PWM signals are set; anything else is

discarded. The robot echoes the name of the command it applied. TABLE II lists the five buttons configured in the app. The same program also sends the GPS latitude and longitude, or a “no fix” message, to the terminal.

TABLE II. Application buttons, commands, and expected platform response

Button	Reply shown	Motor action	Expected platform response
M1	Command: FORWARD	Both motors forward	Both paddle wheels give forward thrust; robot moves ahead
M2	Command: BACKWARD	Both motors reversed	Both paddle wheels give reverse thrust; robot moves back
M3	Command: RIGHT	Differential action	Thrust difference turns the robot to the right
M4	Command: LEFT	Differential action	Thrust difference turns the robot to the left
M5	Command: STOP	Both motors off	Motor movement ceases

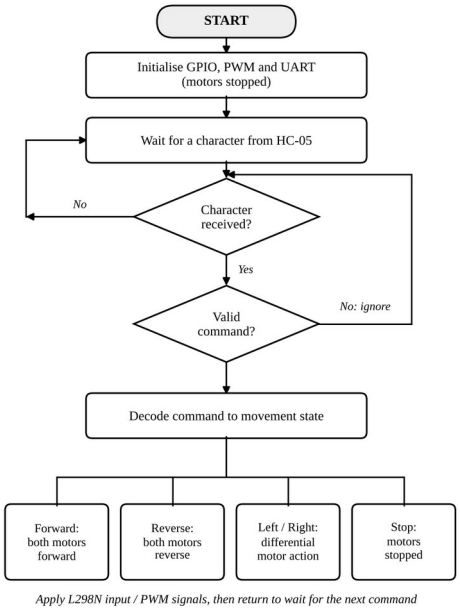

Fig. 3. Flow chart of the motor-control program.

The detection program runs on the laptop and follows Fig. 4. It connects to the ESP32-CAM stream and loads the MobileNet single-shot detector [6], [7] with its class labels. For each frame it resizes the image, builds the input blob, and runs inference. Any person detection above the confidence threshold gets a green bounding box with a confidence label, and the status “presence detection = human detected” is displayed; if there is none, the status “presence detection = no human detected” appears in red.

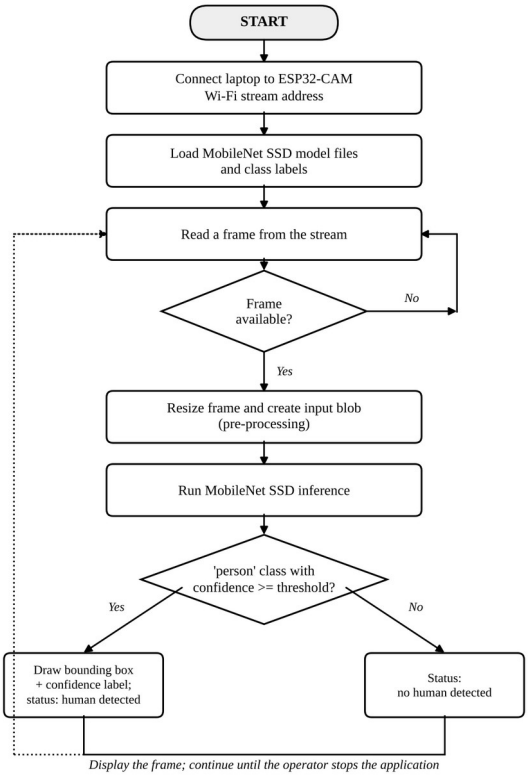

Fig. 4. Flow chart of the human-detection application.

D. Dataset and Pre-processing

This work neither collected a dataset nor trained a model. It uses a pre-trained MobileNet single-shot detector (ssd_mobilenet_v3_large_coco), trained on the Common Objects in Context dataset, and keeps only the person class. Before inference each frame is resized to the 320 × 320 pixel network input, scaled, and reordered to the expected colour channels. The confidence threshold is set to 0.5; lowering it detects more people at the cost of more false alarms, and raising it has the reverse effect.

E. Experimental Setup

The prototype was built and checked in stages. First each sub-system was tested separately: the motors with the robot lifted off the water, the Bluetooth link with simple commands, the video stream on the laptop, and the GPS module outdoors. The complete system was then run in a controlled water tank. The checks were functional observations of movement, video, detection, and GPS output, kept as photographs and terminal screens. Because detection accuracy, latency, Bluetooth range, and battery runtime were not measured, the paper gives no numerical performance values.

IV. RESULTS AND DISCUSSION

In the test tank the prototype floated and responded to the smartphone (Fig. 5). Each of the buttons M1 to M5 returned the matching reply (FORWARD, BACKWARD, RIGHT, LEFT, or STOP). Fig. 6 shows the terminal: the HC-05 connection is reported first, followed by green lines with GPS coordinates refreshed roughly every second, interleaved with the command replies.

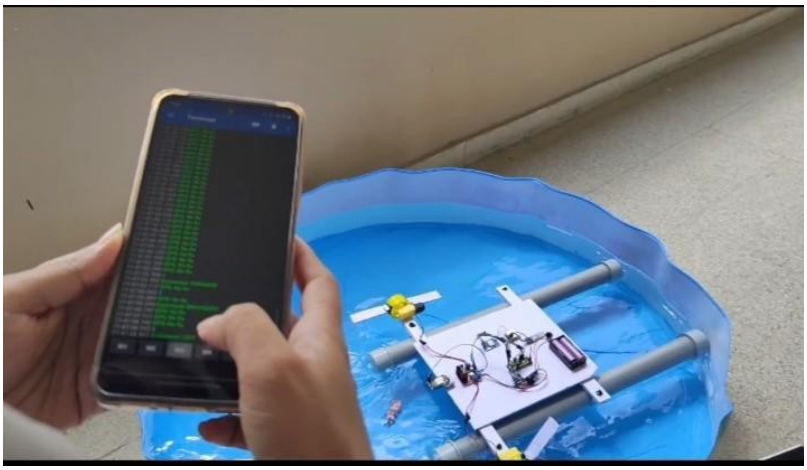

Fig. 5. The prototype floating in the test tank and controlled from a smartphone using the Serial Bluetooth Terminal application.



Fig. 6. Terminal screen showing the HC-05 connection, GPS readings, and replies to the movement commands.

Live video from the ESP32-CAM reached the laptop and was processed frame by frame. A person above the threshold produced a green box and the status “presence detection = human detected” (Fig. 7), and a scene without a person switched the status to the red “presence detection = no human detected” (Fig. 8). This confirms that the video-processing pipeline works at a basic level. TABLE III summarises the functional results.

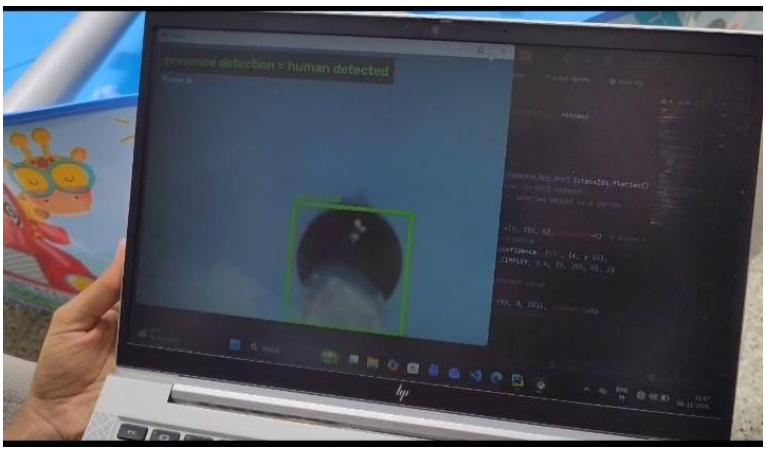

Fig. 7. Detection output when a person is present: green bounding box and status “presence detection = human detected”.

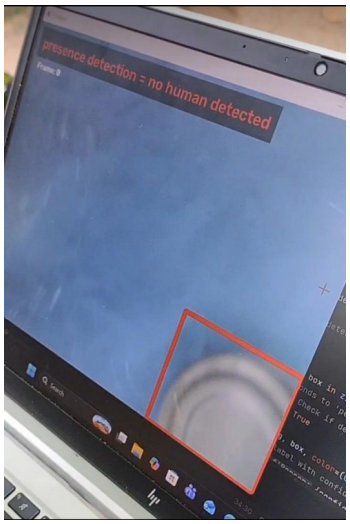

Fig. 8. Detection output when no person is detected: status “presence detection = no human detected” in red.

TABLE III. Summary of the functional results

Function	Observation	Quantitative metric
Floating platform	Fabricated and tested in water	Not measured
Live video	Video displayed on the laptop	Frame rate and latency not measured
Person detection	Bounding box and “human detected” status shown	Accuracy not measured
GPS location	Coordinates displayed on the smartphone terminal	Accuracy and fix time not measured
Remote movement	Direction control from buttons M1 to M5	Range and latency not measured
Power	7.4 V lithium-ion pack with buck converter	Runtime not measured

These results indicate that inexpensive surface mobility, a live camera view, person detection, and position reporting fit together in one modular prototype. Using different links for control and video keeps the architecture simple, and running the detector on a laptop gives far more computing power than a microcontroller could. TABLE IV compares the prototype qualitatively with conventional manual observation. It does not prove better rescue outcomes, response time, or detection performance, and no baseline detectors or ablation studies were run because no models were trained or evaluated.

TABLE IV. Qualitative comparison with conventional manual observation

Aspect	Conventional manual observation	Proposed prototype
Observation position	Rescuer observes from shore or boat, or enters the area	Camera platform operated remotely
Visual feedback	Direct human view	Live video on a laptop
Detection support	Human visual identification	Bounding box and status from a person detector
Location awareness	Landmarks or a manual report	GPS coordinates of the robot
Autonomy	Human-directed	Human-directed; no autonomous path planning

Several limitations apply. Steering is manual, with no path planning or obstacle avoidance. The detector only knows the generic person class, cannot judge whether someone is drowning, and may miss partly submerged or hidden people or flag other objects by mistake. Performance depends on the quality and range of the Wi-Fi link. The GPS value is the robot’s position rather than the person’s, and it needs a valid fix; indoors the terminal shows “no fix”. Late or partial Bluetooth commands can make the robot move differently from what the operator intended, so a fail-safe stop on lost communication is advisable. The prototype is a proof-of-concept and has not been validated in the field.

V. CONCLUSION AND FUTURE WORK

This paper presented a low-cost floating rescue robot that was built and demonstrated in a water tank. It joins a polyvinyl chloride catamaran hull, smartphone control through an STM32 microcontroller, live ESP32-CAM video, MobileNet single-shot detector person detection on a laptop, and GPS position reporting. Its main outcome is that these functions operate together in a modular system made from easily available parts. The robot assists the operator’s observation; it is not an autonomous life-saving device.

Future work includes: (i) autonomous navigation using GPS waypoints and obstacle avoidance; (ii) a detector trained on water-specific images, with accuracy measured on a labelled test set; (iii) on-board processing and a fail-safe stop with return to shore; (iv) a flotation aid or rescue buoy; and (v) measured evaluation of detection accuracy, latency, communication range, and battery runtime, with larger batteries, longer-range links, and waterproof enclosures.

VI. DECLARATIONS

A. Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

B. Conflict of Interest

The authors declare no conflict of interest.

C. Data Availability

The data that support the findings of this study are available from the corresponding author on reasonable request.

D. Ethics Statement

This study did not involve human or animal subjects.

E. Author Contributions

[To be completed by the authors using the CRediT taxonomy, with author initials.]

VII. ACKNOWLEDGMENTS

The authors are grateful to the Management, Principal, and Head of the Department of Electronics and Communication Engineering at GSSS Institute of Engineering and Technology for Women, Mysuru, for their support and a favourable learning environment, and to the project co-ordinators and staff of the department for their guidance and co-operation.

VIII. REFERENCES

[1] K. Srijiranon, N. Varisthanist, T. Tardtong, C. Pumthurean, and T. Tanantong, “Towards intelligent water safety: Robobuoy, a deep learning-based drowning detection and autonomous surface vehicle rescue system,” Applied System Innovation, vol. 9, no. 1, Art. no. 12, 2026.

[2] D. A. Amer et al., “Intelligent eyes on water: YOLOv11-based real-time drowning detection system,” The Journal of Supercomputing, 2025, doi: 10.1007/s11227-025-07732-7.

[3] T. Shahria, K. T. Zaman, S. Rabbi, and M. M. Khan, “Underwater research and rescue robot,” in Proc. Int. Conf. on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1–5.

[4] J. Bai, J. Dai, Z. Wang, and S. Yang, “A detection method of the rescue targets in the marine casualty based on improved YOLOv5s,” Frontiers in Neurorobotics, vol. 16, Art. no. 1053124, 2022, doi: 10.3389/fnbot.2022.1053124.

[5] W. Zhang, L. Chen, and J. Shi, “A pool drowning detection model based on improved YOLO,” Sensors, vol. 25, no. 17, Art. no. 5552, 2025, doi: 10.3390/s25175552.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in Proc. European Conf. on Computer Vision (ECCV), 2016.

[7] A. G. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017.

[8] OpenCV, “OpenCV documentation.” [Online]. Available: https://docs.opencv.org/

[9] Espressif Systems, “ESP32 / ESP32-CAM technical documentation.”

[10] u-blox, “NEO-6 series GPS receiver data sheet.”

[11] STMicroelectronics, “L298 dual full-bridge driver datasheet.”

© 2026 The Author(s). Published by the International Journal of Engineering and Techniques (IJET). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. License: https://creativecommons.org/licenses/by/4.0/