Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

AI-Assisted Surveillance Robot for Lab Monitoring

Om Khairnar
Dept. of Electronics and Computer Science
Pillai College of Engineering
New Panvel, India

Ayush Dugade
Dept. of Electronics and Computer Science
Pillai College of Engineering
New Panvel, India

Pranav Vetale
Dept. of Electronics and Computer Science
Pillai College of Engineering
New Panvel, India

Dr. Karpagavalli S.
Dept. of Electronics and Computer Science
Pillai College of Engineering
New Panvel, India

***Abstract*—A wheeled surveillance robot augmented with artificial intelligence was built and tested for use in laboratory settings where unattended monitoring and hazard detection are required. Three distinct operating modes were incorporated: manual joystick-driven control via a web browser, fully autonomous obstacle-reactive navigation, and a pre-programmed mapped-patrol sequence that traces a fixed route through the monitored space. Motor actuation and sensor interfacing are handled by an ESP8266 (NodeMCU) module, which also hosts a lightweight private web dashboard over Wi-Fi. Temperature, humidity, and air-quality readings are gathered by DHT11 and MQ135 sensors respectively, while an HC-SR04 ultrasonic transducer supplies proximity data for collision avoidance. A smartphone secured to the chassis runs IP-Webcam and forwards a live video stream over the local network to a Flask/OpenCV server running on a separate machine; this server performs motion detection, face-based attendance recording using Local Binary Patterns Histograms (LBPH), and conversational inventory assistance through a rule-based chatbot. Onboard power is supplied by a custom pack built from fifteen recycled 18650 lithium-ion cells wired in a 5P×3S arrangement, protected by a 3S BMS and regulated to 5 V via a 7805 linear regulator. Testing inside a $6\,\mathrm{m} \times 4\,\mathrm{m}$ indoor laboratory yielded 88–93% face recognition accuracy, 93% motion-detection precision, and a 90% patrol completion rate across ten runs. The outcome is a low-cost, modular platform that validates the hybrid embedded-plus-server approach for intelligent laboratory oversight.**

***Index Terms*—ESP8266, IoT, Surveillance Robot, OpenCV, Flask, Face Recognition, Autonomous Navigation, Environmental Sensing, Attendance Management**

I. Introduction

Laboratories house expensive instruments, sensitive reagents, and personnel whose wellbeing depends on diligent environmental oversight. Beyond asset protection, regulatory frameworks governing chemical and biological laboratories impose requirements on continuous air-quality monitoring, access logging, and rapid response to hazardous conditions. Static camera networks and periodic manual rounds have historically been the primary mechanisms for meeting these obligations; however, each brings structural limitations. A fixed camera can surveil only the fraction of the room within its field of view, offers no environmental telemetry, and cannot physically respond to a detected event. Scheduled human patrols are labour-intensive, intermittent, and subject to human error [1].

A mobile robotic platform overcomes these limitations by combining on-demand spatial reach with continuous sensor sampling. Equipped with appropriate actuators and communication modules, such a robot can traverse the laboratory autonomously, collect temperature, humidity, and air-quality data at multiple locations, and relay both sensor readings and live video to a central operator — all without interrupting ongoing work. The addition of computer-vision analytics on the received video stream further enables automated personnel identification and attendance logging, replacing paper-based or swipe-card systems that are easily circumvented.

Affordable wireless microcontrollers, particularly the ESP8266 family, have matured to the point where functional IoT-driven mobile systems can be assembled at a fraction of the cost of industrial alternatives [3], [5]. The trade-off is that computationally intensive workloads — video analytics and natural-language processing — exceed what such resource-constrained devices can handle directly. Splitting responsibilities between an embedded node that governs movement and telemetry, and an external server that executes AI pipelines, resolves this tension without sacrificing analytical capability or substantially increasing system cost.

The robot described in this paper realises exactly that split. An ESP8266 manages vehicle dynamics, reads the onboard sensors, and serves a browser-accessible control panel, while a Flask/OpenCV server running on a separate workstation processes the smartphone video stream, records attendance through face recognition, and handles inventory queries via a chatbot. The specific contributions of this work are:

- A **two-layer hybrid architecture** in which low-level vehicle control and sensor telemetry reside on the ESP8266, and all vision and NLP workloads are offloaded to a Flask/OpenCV server, reducing both onboard complexity and overall system cost.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

- A **custom power subsystem** built from fifteen recycled 18650 cells (5P×3S, 3S BMS, 7805 regulator) paired with DHT11 and MQ135 sensors for sustained, multi-parameter environmental monitoring without a commercial battery pack.
- Verified **autonomous and mapped patrol behaviour** using HC-SR04-based reactive obstacle avoidance and a timed square-route sequence, evaluated over ten representative indoor test runs.
- A fully integrated **face-recognition attendance system** (LBPH, CSV logging) and a **rule-based inventory chatbot**, both accessed through the AI dashboard alongside the live video feed.

Section II surveys the relevant prior art and identifies the gap addressed by this work. Section III describes the system architecture and implementation in detail. Section IV presents the experimental evaluation and discussion, and Section V summarises the findings and outlines future directions.

II. Related Work

A. Mobile Surveillance Platforms and IoT Monitoring

Sensor-rich mobile robots and IoT-based monitoring architectures have attracted sustained research interest. A wide-ranging survey by Sharma and Kanwal catalogues how heterogeneous sensor arrays and video analytics are integrated within IoT surveillance frameworks to deliver real-time anomaly detection in smart facilities [1]. The survey emphasises that combining mobility with sensing is essential for comprehensive indoor coverage, a conclusion that directly motivates the mobile form factor adopted here. Valentino and Leonen constructed a wheeled security robot featuring night-vision imaging, onboard threat classification, and an Android companion app; the demonstrated ability to patrol and detect intrusions without a fixed vantage point underscores the practical value of mobile platforms for security applications [2].

B. ESP8266 and NodeMCU-Based Embedded Systems

The ESP8266's combination of Wi-Fi connectivity, adequate GPIO throughput, and low unit cost has made it a popular choice for IoT-driven control and telemetry tasks. Its use as the primary controller in an IoT-based fire-fighting robot confirms its suitability for reactive embedded robotics [3]. Home-automation and remote-sensing deployments further establish the reliability of ESP8266 and NodeMCU boards in long-running, network-connected scenarios [5], [6]. An in-depth study of GPIO and networking characteristics by Ali and Bao provides additional evidence that the module is well suited to low-latency actuator control [7]. Babu *et al.* demonstrated a remotely operated multipurpose surveillance rover whose layered hardware architecture — separating sensing from actuation — influenced several design decisions in the present work [4].

C. Face Recognition and Attendance Systems

Server-side face recognition using Flask and OpenCV has been validated for attendance management in institutional settings by Shukla and Deoli, whose pipeline closely mirrors the approach taken in this paper [9]. Munirathinam *et al.* extended the same architectural pattern with liveness-detection capability to counter photograph-based spoofing, demonstrating that the Flask/OpenCV stack can accommodate security enhancements without architectural redesign [8]. Palankar *et al.* reported a deployed attendance system in a comparable academic environment, achieving recognition rates consistent with those observed in the present evaluation [10].

D. Chatbot-Driven Laboratory and Inventory Management

Conversational interfaces have been shown to reduce the clerical burden of laboratory equipment management: Lin *et al.* demonstrated that a chatbot integrated into a laboratory management system could handle routine queries and update records without manual database interaction [11]. Amalraj *et al.* corroborated these findings in a general inventory-management context, noting measurable reductions in operator workload when natural-language queries replaced form-based data entry [12].

E. Positioning the Present Work

Table I contrasts the key capabilities of representative prior systems with the robot described in this paper. While individual prior works address one or two of the functional requirements — mobile patrol, environmental sensing, face recognition, or chatbot assistance — none combines all four within a single low-cost platform. The hybrid ESP8266-plus-server architecture proposed here is the specific mechanism that makes this integration feasible without dedicated onboard compute hardware.

TABLE I
Capability Comparison with Representative Prior Work

Work	Mobile Platform	Env. Sensing	Face Recog.	Auto Nav.	Chat-bot
Valentino & Leonen [2]	✓	–	–	–	–
Nagarajan *et al.* [3]	✓	–	–	✓	–
Babu *et al.* [4]	✓	–	–	–	–
Shukla & Deoli [9]	–	–	✓	–	–
Lin *et al.* [11]	–	–	–	–	✓
Proposed	✓	✓	✓	✓	✓

III. System Design and Methodology

The overall architecture comprises two cooperating layers: an **Embedded Control Layer** that pairs the ESP8266 microcontroller with the motor driver and onboard sensors, and an **AI Services Layer** that runs Flask, OpenCV, the LBPH face-recognition pipeline, and the inventory chatbot on a separate compute device. Fig. 1 illustrates the data and control pathways between these layers.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

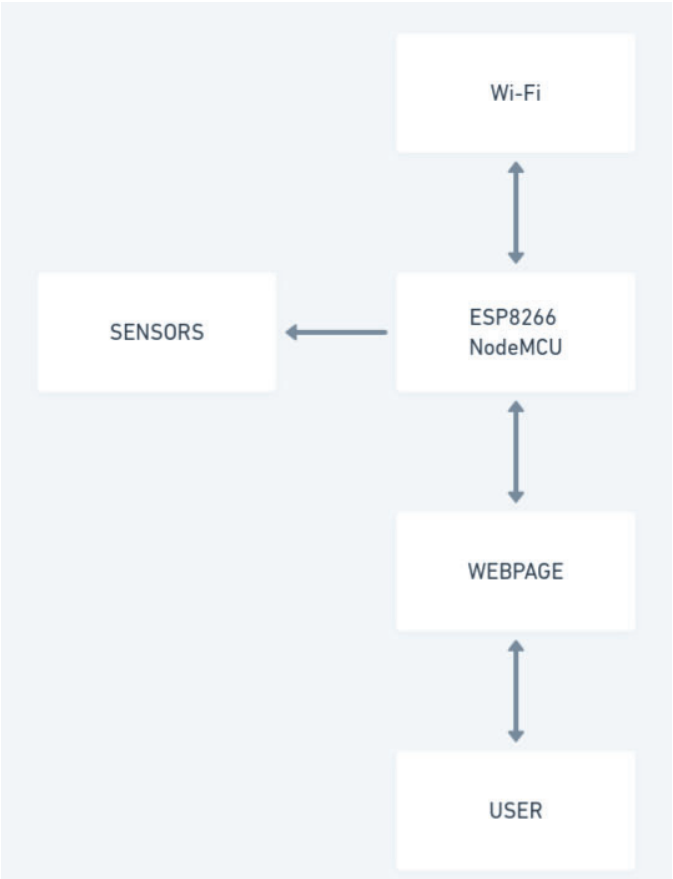

Fig. 1. Top-level hardware block diagram showing data and control pathways between the ESP8266, onboard sensors, the Wi-Fi network, and the user-facing web dashboard.

A. Hardware Architecture

Chassis and Drive Train. Four 12 V, 100 RPM DC geared motors — two per side — are mounted to a flat-plate chassis in a differential-drive configuration. Each motor is driven by one output channel of an L298N dual H-bridge module. The L298N accepts TTL-compatible direction signals (IN1/IN2, IN3/IN4) and a PWM signal on each enable pin (ENA, ENB), allowing independent speed and direction control of the left and right motor pairs directly from the ESP8266 GPIO outputs.

Embedded Controller. The ESP8266 NodeMCU module serves as the sole onboard processing unit. Its integrated 802.11 b/g/n radio enables both the browser-accessible dashboard and outbound AJAX telemetry delivery, while its GPIO complement is sufficient for motor direction signals, PWM generation, one-wire DHT11 communication, analogue MQ135 sampling, and the HC-SR04 trigger/echo interface [5], [7]. The module operates at 3.3 V logic; level-compatible connection to the 5 V L298N control lines is achieved via the NodeMCU's on-board 3.3 V regulator, which is supplemented by the dedicated 7805 rail for sensors and logic.

Sensors. Three sensors cover distinct monitoring roles. An HC-SR04 ultrasonic ranging module measures the forward clearance to obstacles; a 40 kHz burst is emitted on the TRIG pin and the echo pulse width on the ECHO pin is timed to derive distance. A DHT11 capacitive sensor reports ambient temperature and relative humidity on a one-wire digital bus at two-second intervals. An MQ135 resistive gas sensor outputs an analogue voltage that varies with the concentration of CO_2, ammonia, and volatile organic compounds; the signal is read by the ESP8266's single ADC input and scaled to a dimensionless AQI proxy value for dashboard display. Table II summarises the key specifications of all principal hardware components.

TABLE II
PRINCIPAL COMPONENT SPECIFICATIONS

Component	Key Parameter	Value
ESP8266 NodeMCU	CPU clock	80 MHz
	Wi-Fi standard	802.11 b/g/n
	GPIO count	17
	Operating voltage	3.3 V
L298N	Max motor voltage	35 V
	Max output current	2 A/channel
	Logic voltage	5 V
HC-SR04	Measuring range	2–400 cm
	Accuracy	±3 mm
	Operating voltage	5 V
DHT11	Temp. range	0–50 °C
	Temp. accuracy	±2 °C
	Humidity accuracy	±5% RH
MQ135	Target gases	CO_2, NH_3, VOC
	Output	Analogue voltage
	Operating voltage	5 V
18650 cell	Nominal voltage	3.7 V
	Typical capacity	2000–2500 mAh
Pack (5P×3S)	Nominal voltage	11.1 V
	Est. capacity	12–15 Ah
7805 regulator	Output voltage	5 V (fixed)
	Max output current	1 A

Power Architecture. Fifteen 18650 lithium-ion cells salvaged from retired laptop battery modules are connected as five parallel strings of three series cells (5P×3S), yielding a nominal bus voltage of 11.1 V and an estimated usable capacity of 12–15 Ah. A three-cell BMS board provides over-charge protection, over-discharge cutoff, short-circuit protection, and passive cell balancing. The 11.1 V bus powers the L298N motor driver directly; a 7805 linear regulator derives the 5 V logic rail for the ESP8266 board, the sensor modules, and the HC-SR04.

Imaging. A commodity smartphone is held in a 3D-printed clamp mounted centrally on the chassis. Rather than adding a dedicated camera module — which would demand additional GPIO lines and driver firmware — the handset runs the IP-Webcam application, which exposes a Motion-JPEG (MJPEG) HTTP endpoint over the local Wi-Fi network. The AI server

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

consumes this stream via OpenCV's `VideoCapture` class. This arrangement exploits the smartphone's auto-exposure, optical image stabilisation, and on-chip ISP without adding to the embedded firmware complexity.

B. Embedded Firmware and Operating Modes

The ESP8266 firmware is written in Arduino-compatible C++ and serves a single-page web dashboard to any browser connected to the same local network [6]. The dashboard is delivered as a self-contained HTML/CSS/JavaScript page stored in PROGMEM flash; subsequent sensor updates and command acknowledgements are exchanged as JSON over lightweight AJAX calls, avoiding full-page reloads.

HTTP Endpoint Structure. The firmware exposes the following named GET endpoints:

- `/data` — Returns a JSON object containing the latest DHT11 temperature and humidity readings and the current MQ135 ADC value. Polled by the dashboard JavaScript every 2 seconds.
- `/forward`, `/backward`, `/left`, `/right`, `/stop` — Immediately apply the corresponding L298N IN/ENA pin states for manual drive control.
- `/auto` — Toggles autonomous obstacle-avoidance mode on or off.
- `/mapped` — Activates the pre-programmed square-patrol sequence.

Operating Mode State Machine. At any instant the firmware occupies one of three mutually exclusive modes, as illustrated conceptually in Algorithm 1. Transitions between modes are triggered exclusively by incoming HTTP commands from the dashboard.

Autonomous Mode. In autonomous mode the HC-SR04 TRIG pin is pulsed every 60 ms and the echo pulse width converted to a distance estimate. When the measured clearance falls below a configurable threshold (default: 20 cm), the firmware sets the motor driver to halt, executes a reverse-and-turn manoeuvre, and resumes forward travel, effectively implementing a reactive wall-following behaviour without any map knowledge.

Mapped-Patrol Mode. Mapped mode encodes a square patrol route as an ordered array of {direction, duration} tuples stored in firmware flash. Each entry drives the motors in the specified direction for the specified time before advancing to the next entry; the sequence repeats cyclically. Ultrasonic checks are woven into the inter-entry gaps: a detected obstacle suspends sequence progression and triggers a back-up routine until the path is clear.

Fig. 2 shows the dashboard as rendered on a mobile browser, with live sensor readings visible alongside the mode toggles and directional drive buttons.

C. AI Services Layer

A standard workstation connected to the same local network hosts the Flask application. Three functional modules execute in parallel within the server process.

Algorithm 1 Firmware operating-mode state machine

```
Initialise: mode ← MANUAL
loop
  if HTTP command received then
    Update mode accordingly
  end if
  if mode = MANUAL then
    Apply last received motor command
  else if mode = AUTO then
    Read HC-SR04 distance d
    if d < threshold then
      Execute reverse-and-turn avoidance manoeuvre
    else
      Drive forward
    end if
  else if mode = MAPPED then
    Step through patrol sequence entry i
    Read HC-SR04 distance d
    if d < threshold then
      Pause; reverse if necessary; retry segment
    else
      Execute timed motor command for entry i
      i ← i + 1; wrap at sequence end
    end if
  end if
end loop
```

Motion Detection. Each frame retrieved from the smartphone MJPEG stream is passed to OpenCV's MOG2 adaptive Gaussian background-subtraction algorithm, which maintains a statistical model of the static background scene and segments foreground blobs. A frame is classified as a motion event when the ratio of foreground pixels exceeds a configurable threshold. Motion events activate the face-recognition pipeline and initiate clip recording to disk; non-motion frames bypass recognition entirely, reducing unnecessary compute load during unoccupied periods.

Face Recognition and Attendance Logging. The recognition pipeline operates in two phases. During the offline *training phase*, image samples for each registered individual are collected, grey-scaled, and resized to a uniform resolution. A Haar-cascade face detector localises face regions within each sample image; the extracted face patches are labelled with integer identity indices and used to fit an LBPH model, which is then serialised to disk with OpenCV's `face.LBPHFaceRecognizer` API. During the online *inference phase*, the server loads the serialised model at startup. For every motion-triggered frame, the Haar cascade detects face regions and the LBPH model returns a predicted identity label and a confidence score for each region [8], [9]. Predictions below a confidence threshold are written to `attendance.csv` alongside a UTC timestamp and the associated frame thumbnail. Detections whose confidence exceeds the threshold — indicating low certainty — are held in a "pending faces" queue rendered on the dashboard for manual

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

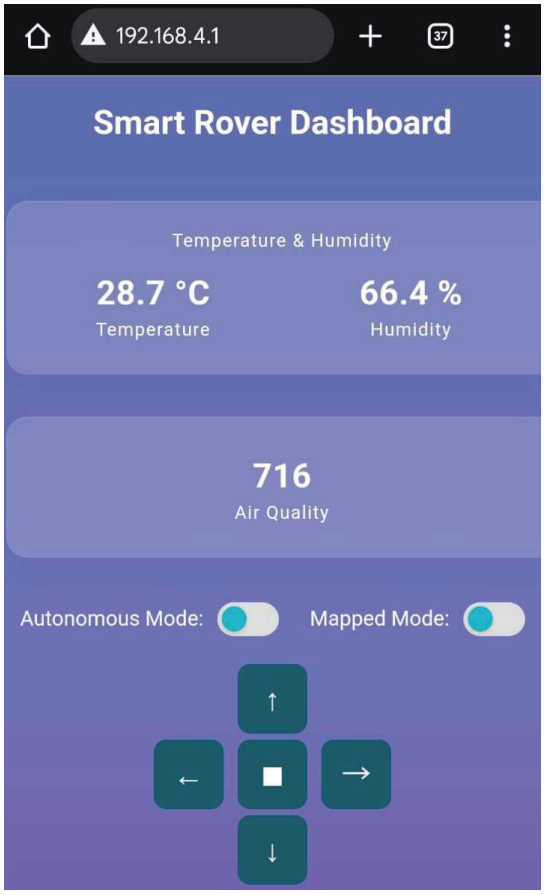

Fig. 2. Mobile view of the ESP8266 web dashboard during a live session, showing temperature (28.7 °C), relative humidity (66.4%), and air-quality index (716), together with autonomous/mapped mode toggles and directional drive controls.

review and optional manual confirmation.

Inventory Chatbot. A rule-based natural-language module serves the dashboard's chatbot text box. The underlying data store is a JSON file mapping item names to integer quantities. On receiving a query, the module applies keyword-matching rules to classify the intent as one of four operation types: *query* (report current quantity), *add* (create a new item record), *update* (modify an existing quantity), or *delete* (remove a record). Matched records are displayed inline in the chat panel; unrecognised queries return a clarification prompt [11], [12].

Fig. 3 shows the AI dashboard during a live session, with the face-detection annotation, confirmed attendance log, inventory panel, and chatbot all visible simultaneously.

IV. Experimental Evaluation

A. Experimental Setup

All tests were conducted inside a 6 m×4 m indoor laboratory representative of a typical academic instrumentation space. The workstation hosting the Flask server was connected to the same 2.4 GHz Wi-Fi access point as the robot. Three distinct evaluation scenarios were executed: (i) face-recognition trials with ten registered participants who each presented

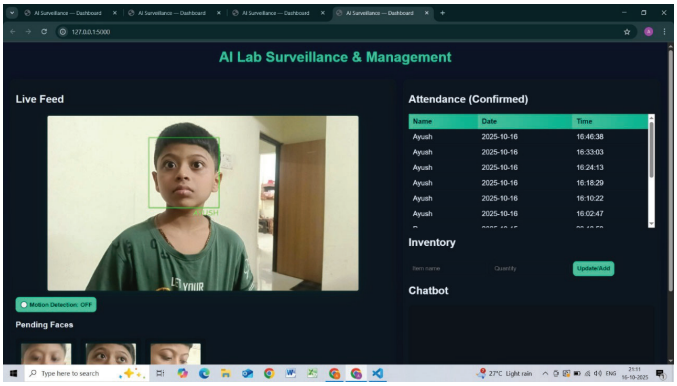

Fig. 3. AI surveillance dashboard in operation. Left panel: live MJPEG feed with Haar-cascade bounding box and LBPH identity label overlaid. Right panel (top to bottom): confirmed attendance log with timestamps, inventory management input fields, and chatbot conversation interface.

both frontal and partial-profile views to the camera; (ii) ten consecutive mapped-patrol runs with one to two physical obstacles placed at different positions along the patrol path for each run; and (iii) short-duration environmental logging under controlled occupancy conditions to characterise the repeatability and trend-detection responsiveness of the DHT11 and MQ135 sensors.

B. Quantitative Results

Table III summarises the five principal performance metrics obtained across all evaluation scenarios.

TABLE III
Summary of Measured Performance Metrics

Metric	Value	Notes
Face recognition accuracy	88–93%	Varies with occlusion/pose
Motion detection precision	93%	Low false-alarm rate
Patrol completion rate	90%	10 runs, 1–2 obstacles
DHT11 temp. stability	±0.8 °C	Within datasheet tolerance
MQ135 AQI response	Qualitative	Calibration required

Face Recognition. Accuracy across the ten registered participants spanned the 88–93% band. Frontal presentations under adequate ambient lighting tended toward the upper end of that range; partially occluded or laterally angled views returned lower LBPH confidence scores, which is a well-documented characteristic of histogram-based texture descriptors when the viewing geometry deviates from the training pose [8]–[10]. All correctly recognised detections were successfully written to the attendance CSV with accurate timestamps.

Motion Detection. The MOG2-based gating stage attained a precision of 93%, meaning that nearly all frames forwarded to the recognition pipeline represented genuine subject presence rather than illumination artefacts or minor camera vibration. The high precision rate is significant because it directly governs the computational load on the Flask server: low precision would invoke the LBPH recogniser unnecessarily on background frames, increasing average processing latency.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Autonomous Navigation. Nine of the ten mapped-patrol runs concluded without operator intervention, yielding a 90% completion rate. In the single unsuccessful run the robot was unable to clear an obstacle within its coded avoidance routine and required manual resumption. In the nine successful runs, whenever the avoidance routine was invoked it correctly reversed, turned, and rejoined the patrol path without assistance. The timed nature of the patrol sequence means that accumulated timing errors can cause slight positional drift over multiple laps; this is identified as an area for improvement through odometry or encoder feedback.

Environmental Sensors. DHT11 temperature measurements remained within $\pm 0.8\,^{\circ}\mathrm{C}$ across the logging session, consistent with the manufacturer's published tolerance of $\pm 2\,^{\circ}\mathrm{C}$ (the tighter observed value reflects the stable indoor environment rather than an improved device). MQ135 analogue output responded to changes in ventilation and occupancy in a qualitative manner; the resistance shift was perceptible in the trend but could not be converted to a calibrated parts-per-million concentration without a reference instrument, a limitation common to low-cost resistive gas sensors of this type.

C. Implementation Challenges

Several non-trivial engineering issues were encountered during development and are worth documenting for practitioners attempting similar builds.

Battery Pack Assembly. Constructing a reliable pack from recycled cells required individual cell capacity testing with an electronic load before assembly; cells whose capacity had degraded below a minimum threshold were discarded to ensure uniform discharge behaviour across the parallel strings. Spot-welding nickel strips rather than soldering minimised heat exposure to the cell terminals.

Wi-Fi Stability. The ESP8266's single-core architecture must service the TCP/IP stack, the AJAX web server, PWM generation, sensor sampling, and the ultrasonic ranging interrupt within the same execution loop. Under heavy dashboard polling rates, watchdog resets were occasionally observed. This was resolved by introducing a `yield()` call in the main loop and reducing the AJAX polling interval to 2 seconds, giving the network stack sufficient scheduling time.

LBPH Model Drift. The LBPH recogniser's accuracy is sensitive to changes in ambient illumination between the training and inference sessions. Retraining the model under the specific lighting conditions of the target laboratory, rather than using a generic training set, was found to be important for achieving the reported accuracy range.

MQ135 Warm-Up. The MQ135 requires a burn-in period of approximately 24–48 hours on first use and a shorter warm-up of several minutes at each power-on before its analogue output stabilises. The firmware was modified to delay the first AQI reading by 60 seconds after boot to avoid reporting transient out-of-range values.

D. Discussion

Keeping low-level vehicle control on the ESP8266 while routing computationally demanding AI tasks to the server proved to be an effective division of labour: the embedded firmware remained lean and responsive, while the server delivered sophisticated analytics without resource contention on the robot [1], [2]. The principal vulnerability of the current design is its dependence on an uninterrupted Wi-Fi link; a dropped connection simultaneously disables manual override capability and the video feed. A firmware watchdog that halts the robot safely upon link timeout is planned for the next hardware revision. On the sensing side, the MQ135 suits indicative trend monitoring but falls short of quantitative accuracy; replacement with a calibrated electrochemical sensor would address this gap. The DHT11 provides adequate coarse thermal data, though the DHT22 would offer improved accuracy ($\pm 0.5\,^{\circ}\mathrm{C}$) for environments where tighter thermal tolerances must be maintained.

Fig. 4 shows the fully assembled prototype with all hardware components in their final positions.

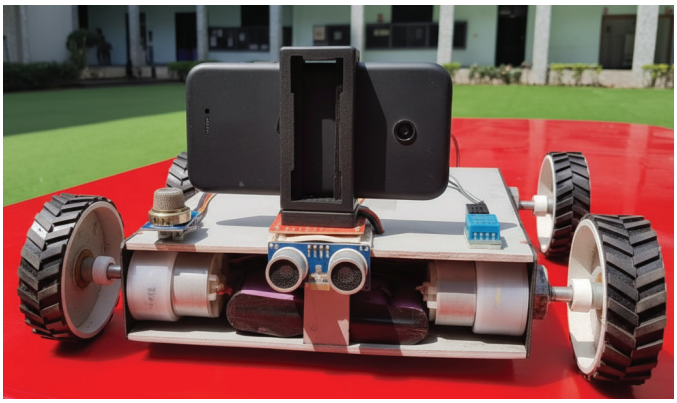

Fig. 4. Fully assembled robot prototype. Components visible include the four 12 V DC geared motors, HC-SR04 ultrasonic sensor (front centre), MQ135 gas sensor (front left), DHT11 temperature-humidity sensor (rear right), ESP8266 NodeMCU board (underside), and the smartphone camera mount (centre top).

V. Conclusion and Future Work

A functional AI-assisted surveillance robot tailored to laboratory monitoring requirements was designed, assembled, and experimentally evaluated. By confining vehicle control and environmental telemetry to an ESP8266 while delegating face recognition, motion detection, and chatbot inference to a Flask/OpenCV server, the system achieves a productive balance between hardware cost and analytical capability. Face recognition attained 88–93% accuracy across ten registered users, the motion-detection stage achieved 93% precision, and the mapped patrol completed successfully in 90% of obstacle-laden test runs. The inventory chatbot rounds out the platform's utility as a general-purpose laboratory management tool, offering a conversational interface for stock queries and updates without requiring database expertise from the operator.

Several targeted improvements are planned for future iterations of the system:

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

- **On-device inference.** Migrating the face-recognition pipeline to a Raspberry Pi paired with a Coral USB Accelerator or equivalent neural processing unit would reduce network round-trip latency and confine biometric data to the robot, improving privacy.
- **Calibrated gas sensing.** Replacing the MQ135 with a verified electrochemical sensor and implementing a standardised AQI calculation would upgrade air-quality reporting from qualitative trend observation to quantitative measurement.
- **SLAM-based navigation.** Replacing the fixed-duration patrol sequence with a simultaneous localisation and mapping (SLAM) framework driven by a 2-D LiDAR or depth camera would allow the robot to build an accurate floor map and plan collision-free paths dynamically, removing the need to reprogram routes when the laboratory layout changes [2].
- **Odometry and closed-loop position control.** Adding wheel encoders to the drive train would allow the firmware to track cumulative displacement and correct heading drift in the mapped-patrol sequence without relying solely on timed motor commands.
- **Secured remote connectivity.** Introducing TLS-encrypted telemetry, authenticated internet-facing access via a reverse proxy, and cloud-backed storage of attendance and sensor logs would prepare the platform for deployment beyond a single local network segment.
- **Deep-embedding recognition.** Upgrading from LBPH to a metric-learning approach such as FaceNet or ArcFace would improve robustness to challenging pose, partial occlusion, and variable illumination conditions encountered in real laboratory environments.

ACKNOWLEDGMENT

The authors are grateful to the Department of Electronics and Computer Science, Pillai College of Engineering, New Panvel, for providing access to laboratory facilities and for institutional support throughout this project.

REFERENCES

[1] H. Sharma and N. Kanwal, "Smart surveillance using IoT: a review," *Radioelectron. Comput. Syst.*, no. 1(109), pp. 116–126, Feb. 2024. DOI: 10.32620/reks.2024.1.10.

[2] A. M. Valentino and J. T. Leonen, "IoT-based smart security robot with Android app, night vision and enhanced threat detection," in *Proc. IEEE 15th Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Bhimtal, India, Dec. 2023, pp. 415–420. DOI: 10.1109/CICN59264.2023.10402359.

[3] D. Nagarajan *et al.*, "IoT based surveillance and automatic fire extinguisher robot," in *Proc. Int. Conf. Pervasive Comput. Social Netw. (ICPCSN)*, Salem, India, 2023, pp. 1–6.

[4] B. R. Babu, P. M. A. Khan, S. Vishnu, and K. L. Raju, "Design and implementation of an IoT-enabled remote surveillance rover for versatile applications," in *Proc. IEEE Int. Conf. Interdiscip. Approaches Technol. Manage. Social Innov. (IATMSI)*, Gwalior, India, 2022, pp. 1–6.

[5] P. O. Ayeni and O. C. Adesoba, "IoT-based home control system using NodeMCU and Firebase," *J. Edge Comput.*, vol. 4, no. 1, pp. 17–34, 2025.

[6] R. Ambekar *et al.*, "Home automation system using ESP8266 and Blynk mobile app," *Int. J. Sci. Res. Eng. Mgmt.*, vol. 8, no. 4, pp. 1–5, Apr. 2024.

[7] A.-S. A. Ali and X. N. Bao, "Design and research of infrared remote control based on ESP8266," *Open Access Libr. J.*, vol. 8, pp. 1–14, 2021.

[8] T. Munirathinam *et al.*, "Face recognition with liveness detection login on Flask web application," *Int. Res. J. Adv. Eng. Hub*, vol. 3, no. 4, pp. 1321–1327, Apr. 2025.

[9] M. Shukla and S. Deoli, "Facial recognition attendance system using Flask and OpenCV," in *Proc. Int. Conf. Innov. Data Commun. Technol. (ICCIDT)*, Coimbatore, India, 2023.

[10] P. Palankar *et al.*, "Face recognition attendance system," *Int. J. Future Mod. Res.*, vol. 7, no. 2, pp. 1–8, Mar.–Apr. 2025.

[11] S.-H. Lin, R.-S. Run, and J.-Y. Yan, "Chatbot application in laboratory equipment management and e-assistant," in *Proc. Int. Symp. Comput. Consumer Control (IS3C)*, Taichung, Taiwan, 2020, pp. 1–4.

[12] T. Amalraj Victoire *et al.*, "Smart inventory management system using chatbot," *Int. J. Innov. Res. Technol.*, vol. 10, no. 12, pp. 919–924, May 2024.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)