International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Smart Home Safety System Based on IOT with Image Verification

Arun T[1], Akshay Gowda G[2], Dhiviek[3]
Department of Computer Science and Engineering
East West Institute of Technology
Bengaluru, Karnataka, India

Prof. Harsh Verma
Project Guide
Department of Computer Science and Engineering
East West Institute of Technology
Bengaluru, Karnataka, India

Abstract - This paper presents an Internet of Things (IoT)-based Smart Home Safety System that brings door access control, face verification, household-condition monitoring, and voice interaction into one web-based platform. The prototype uses an ESP32 to communicate with sensors and control a relay-driven 12 V solenoid lock. Face verification runs on a MacBook using its built-in camera, OpenCV Haar Cascade face detection, and Local Binary Patterns Histograms (LBPH) recognition; an ESP32-CAM is not used. The system also includes PIN-based authentication, a Flask backend, a MySQL database, a web dashboard, and NOVA, a voice assistant powered by OpenAI. A DHT11 measures temperature and humidity, a PIR sensor detects motion, a reed switch monitors the door, and gas/safety readings are checked against a configured threshold. The backend handles application requests and stores user records, sensor data, access logs, door events, and alerts. In functional testing, a registered face was recognized and the software reported the door-unlock action. The prototype demonstrates how IoT hardware, computer vision, database-backed services, and voice interaction can work together. However, its dependence on the MacBook and network services, along with the lack of advanced liveness detection, limits its readiness for real-world deployment.

Key Words: Smart Home, Internet of Things, Face Recognition, ESP32, PIN Authentication, Voice Assistant.

1. INTRODUCTION

Smart-home systems increasingly combine connected sensors, access-control devices, and software interfaces to help users monitor household conditions. Traditional keys and standalone alarms provide basic protection, but they do not always offer a single place to check door status, motion, temperature, humidity, and gas-related readings. Bringing these functions together in a web dashboard can make the system easier to monitor and test.

This project brings access control, face verification, environmental monitoring, and voice interaction into one working prototype. The ESP32 connects to the sensors and door-control hardware, while a MacBook runs the face-recognition program and the main software services, including the Flask backend and voice-assistant integration. Face verification uses the MacBook's built-in camera rather than an ESP32-CAM. OpenCV performs face detection, LBPH handles recognition, and MySQL stores structured application records [1], [2], [4], [6].

The voice assistant, called NOVA in the project, provides a spoken way to request supported information, such as the current door state. Its voice-assistant functionality uses OpenAI services [5]. The aim is to demonstrate an integrated prototype, not to present the system as a certified or fully hardened commercial security product.

2. METHODOLOGY

2.1 System Architecture

The system has three main layers: hardware, application software, and data storage. The hardware layer includes the ESP32, DHT11, PIR sensor, reed switch, gas/safety sensing, LCD, LEDs, relay, and solenoid lock. The application layer includes camera-based face recognition, PIN authentication, Flask APIs, NOVA voice interaction, and the browser dashboard. MySQL stores user information and system events. The MacBook and ESP32 communicate over the same local Wi-Fi network or mobile hotspot.

The main data path is: sensors and lock circuit ↔ ESP32 ↔ local Wi-Fi/hotspot ↔ Flask backend ↔ MySQL and web dashboard. For face verification, the flow is: MacBook camera → Haar Cascade detection → LBPH recognition → access decision → ESP32 door-control path.

2.2 Authentication and Face Verification

The project design includes PIN-based authentication alongside face verification. For face verification, the MacBook camera captures image frames. OpenCV's Haar Cascade classifier locates the face, and an LBPH recognizer compares its patterns with the trained data [1], [2], [6]. The trained model is stored in trainer.yml. When the configured decision logic identifies a registered user, the application starts the authorized door-unlock workflow. The project test log records a successful verification followed by a door-unlock action.

Face detection and face recognition serve different purposes: detection locates a face in an image, while recognition compares that face with trained identity data. Recognition performance can vary with lighting, distance, pose, and the number and quality of training samples. The current prototype does not include advanced liveness detection, so resistance to photographs or replayed images has not been demonstrated.

2.3 Hardware and Door Control

The ESP32 provides Wi-Fi connectivity and interfaces with the attached hardware [7]. The DHT11 measures temperature and humidity, the PIR sensor detects motion, and the reed switch reports the door state. Gas/safety readings are compared with a configured threshold. An I2C LCD and LEDs provide local indications. A relay switches the solenoid-lock circuit. The 12 V solenoid requires a suitable external power supply and must not be powered directly from an ESP32 GPIO.

2.4 Backend, Database, and NOVA

Flask provides the backend API, with endpoints including /login, /register, /esp32-status, /door-status, /door-control, /create-alert, and /test-db [3]. The MySQL database is named smart_home_safety and contains the tables users, registered_faces, access_logs, door_events, safety_alerts, and sensor_data [4]. Together, these records support

authentication, monitoring, access-history review, and safety-event tracking.

NOVA runs as part of the software system rather than directly on the ESP32. OpenAI provides the voice-assistant functionality [5]. The supported commands should be described according to the implemented code. If the assistant calls OpenAI's hosted API, voice processing needs internet access, even though basic communication between the ESP32 and MacBook can remain on the local network.

2.5 Network Configuration and Testing

The prototype was first developed on the home Wi-Fi network. For a college demonstration, the MacBook and ESP32 can instead connect to the same mobile hotspot. The ESP32 firmware must contain the hotspot credentials, and its configured backend URL must point to the MacBook's current local IP address and Flask port. Since the local IP may change when the network changes, the complete workflow should be tested on the intended hotspot before the demonstration.

Testing covered the main modules and their interaction, including face verification, the door-unlock action, hardware communication, backend and database connectivity, and local-network communication. The registered-face test confirms that the main recognition workflow operated under the conditions tested. Numerical accuracy, latency, false-acceptance rate, and large-scale reliability were not measured and are therefore not claimed.

3. MODELING AND ANALYSIS

3.1 Research Gap and Problem Analysis

Home monitoring, access control, biometric verification, and voice interaction are often handled as separate functions. Combining them in one prototype introduces practical challenges because each part has different hardware, data, communication, and security requirements. A dashboard can display sensor readings, but users may still need to navigate it manually. A voice assistant provides another way to request supported system information.

This project addresses the integration challenge by connecting authentication, face verification, sensor monitoring, door control, event storage, a web dashboard, and voice interaction. The contribution is not that these individual technologies are new, but that they are brought together in a single working prototype.

3.2 Main Modules

The system contains six main modules: (i) user authentication, including PIN-based access as specified in the project design; (ii) face verification through the MacBook camera; (iii) monitoring of door, temperature, humidity, gas/safety, and motion information; (iv) a web dashboard; (v) the NOVA voice assistant; and (vi) backend and database services that connect the modules.

TABLE I. KEY COMPONENTS AND IMPLEMENTATION

Function	Implementation
Hardware interface and Wi-Fi	ESP32 sensor communication and door-control interface
Face image capture	MacBook built-in camera
Face verification	OpenCV Haar Cascade detection; LBPH recognition; trained model trainer.yml
Backend/API	Flask application endpoints
Persistent records	MySQL database: users, registered_faces, sensor_data, access_logs, door_events, safety_alerts
Voice-based interaction	NOVA voice assistant using OpenAI services; internet required for hosted API calls
Door lock	Relay-controlled 12 V solenoid with external power supply

3.3 Safety Threshold as Configured

The current backend uses 750 as its configured gas/safety threshold. This value is an implementation setting, not a universal danger limit. A real deployment would require calibration for the particular sensor, its measurement units, and the intended environment.

4. RESULTS AND DISCUSSION

The implemented system brings the main access-control and home-monitoring functions into one software arrangement. The clearest confirmed functional result is the face-verification test: the MacBook camera and LBPH recognizer identified a registered user, after which the software reported that the door was unlocked. This indicates that camera capture, recognition, and the access-control path worked together under the tested conditions.

The ESP32 serves as the hardware-facing controller, Flask handles application requests, and MySQL stores persistent records. The backend supports registration and login, ESP32 status, door status and control, and alert creation. The dashboard is intended to present information such as lock and door status, temperature, humidity, gas/safety condition, and motion. NOVA adds voice-based access to supported information.

The system remains a prototype with several limitations. Face recognition and multiple backend services depend on the MacBook being available. The ESP32 also needs a working local connection to the Flask server, while voice processing through the OpenAI API requires internet access. The current face-recognition implementation lacks advanced liveness detection, and the sensor thresholds and hardware safety need further validation before deployment.

The available project records do not include a formal numerical test dataset or measured accuracy and response-time results. Further testing should record genuine and rejected access attempts, vary lighting and face angles, measure end-to-end delay, and examine how the system behaves when Wi-Fi, the backend, or the database becomes unavailable. Recording these results would make the evaluation easier to reproduce.

5. CONCLUSION

This paper presented an IoT-based Smart Home Safety System with Image Verification. It combines ESP32-based sensor communication and door control with face verification, PIN-based authentication as outlined in the project design, a Flask backend, MySQL storage, a web dashboard, and the NOVA voice assistant powered by OpenAI. The final face-recognition setup uses the MacBook's built-in camera rather than an ESP32-CAM. In the reported functional test, a registered face was verified and the unlock workflow completed.

The prototype demonstrates the practical value of combining access control, home-condition monitoring, and voice interaction in one application. Future work should focus on more independent operation, stronger anti-spoofing, secure communication and credential handling, calibrated safety sensing, backup power, and systematic testing under different network and environmental conditions.

ACKNOWLEDGMENT

The authors would like to thank Prof. Harsh Verma, the project guide, for his guidance and support throughout the development of this project. The authors also acknowledge the Department of Computer Science and Engineering, East West Institute of Technology, Bengaluru, for providing the academic setting in which the work was carried out.

REFERENCES

1. P. Viola and M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. I-511–I-518, 2001, doi: 10.1109/CVPR.2001.990517.
2. T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Description with Local Binary Patterns: Application to Face Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037–2041, Dec. 2006, doi: 10.1109/TPAMI.2006.244.
3. Pallets Projects, "Flask Documentation." [Online]. Available: https://flask.palletsprojects.com/en/stable/. [Accessed: Oct. 21, 2026].
4. Oracle, "MySQL Documentation." [Online]. Available: https://dev.mysql.com/doc/. [Accessed: Oct. 21, 2026].
5. OpenAI, "API Documentation." [Online]. Available: https://platform.openai.com/docs/. [Accessed: Oct. 21, 2026].
6. OpenCV, "OpenCV Documentation." [Online]. Available: https://docs.opencv.org/. [Accessed: Oct. 21, 2026].
7. Espressif Systems, "ESP32 Documentation." [Online]. Available: https://docs.espressif.com/. [Accessed: Oct. 21, 2026].