Smart Home Surveillance, Monitoring, and Automation with Real-Time Alerts via Raspberry Pi

Srija Thatikonda[1], Dr. S. Sandhya Rani [2]

1. M.Tech Student, Dept. of ECE Jayamukhi Institute of Technological Sciences, Warangal, India srijathatikonda@gmail.com
2. Associate Professor& Head, Dept. of ECE Jayamukhi Institute of Technological Sciences, Warangal, India dr.sandhyarani@jits.in

Abstract

This paper presents a real-time smart home surveillance and automation system utilizing Raspberry Pi, integrated with the Internet of Things (IoT) and artificial intelligence (AI). The system combines motion sensors, object detection is implemented using the YOLO (You Only Look Once), and face recognition is achieved through the FaceNet algorithm to identify intruders and send real-time alerts via email, storing only relevant images to minimize storage use. It also supports remote access allowing users to monitor and control their home from a computer or mobile device. Additionally, the system automates home functions such as fan control based on temperature, gas leak detection, and geyser management via a mobile app. This integrated solution enhances home security and convenience through intelligent monitoring and responsive automation.

Keywords: Smart Home, Surveillance, Raspberry Pi, Internet of Things (IoT), Artificial Intelligence (AI), Face Recognition, Real-Time Alerts, Object Detection, Home Automation, Intrusion Detection.

I. Introduction

With the rapid advancement of the Internet of Things (IoT) and Artificial Intelligence (AI), smart home systems have evolved significantly, offering enhanced safety, automation, and user convenience. These systems integrate sensors, embedded devices, and communication protocols to monitor and control household activities remotely. Smart home security, in particular, has seen considerable innovation, focusing on intrusion detection, surveillance, environmental monitoring, and automation of home appliances. Traditional home surveillance systems primarily rely on Closed-Circuit Television (CCTV) cameras that continuously record video footage, storing it locally or on cloud servers. While these systems provide visual evidence, they have notable drawbacks such as

- Excessive Storage Requirements: Continuous recording consumes significant amounts of storage, making long-term video archiving expensive and inefficient.
- Delayed Response: These systems typically lack real-time processing or intelligent alert mechanisms, resulting in delays in detecting and responding to threats.
- Limited Intelligence: Conventional CCTV lacks AI-based features such as face recognition, object detection, or environmental awareness, reducing their effectiveness in identifying specific threats.
- High Bandwidth Use: Continuous video streaming over networks leads to bandwidth congestion, especially in systems with multiple cameras.
- Manual Monitoring Dependency: Most systems still rely on human operators for surveillance, increasing the risk of oversight and delayed action.

Recent research has attempted to overcome these limitations through the integration of AI and IoT. Various smart surveillance models incorporate motion detection using Passive Infrared (PIR) sensors, object recognition through computer vision algorithms like Haar cascades, and

Doi:10.48047/ijiee.2025.15.7.5

facial recognition with deep learning models such as FaceNet or DeepFace. These models enable real-time detection and classification of individuals or objects in the monitored environment. However, many of these implementations still face the following challenges:

- High Power Consumption: AI-based systems using GPUs or full-fledged PCs consume more power, making them less ideal for continuous home surveillance.
- Limited Scalability and Portability: Some systems require high-end hardware and are not scalable for home-level deployments.
- Privacy Concerns: Cloud-based storage and remote access often raise privacy issues due to potential data breaches or unauthorized access.

To address these gaps, this research proposes a lightweight, efficient, and intelligent smart home surveillance and automation system using Raspberry Pi as the core processing unit. The system combines IoT for sensor integration and communication, AI for intelligent detection, and machine learning techniques for object and face recognition. Key features include:

- Motion detection using PIR sensors to trigger surveillance only when activity is detected.
- YOLOv3 (You Only Look Once) deep learning model for real-time object detection.
- FaceNet algorithm for recognizing known faces and distinguishing intruders.
- Email-based real-time alerting with captured intruder images.
- Environmental monitoring using DHT11 (temperature/humidity) and MQ135 (gas detection) sensors.
- Automated control of appliances such as fans and geysers via relay modules, based on sensor input and user control through a web/mobile interface.

This system significantly reduces storage and bandwidth usage by storing only relevant event-triggered images instead of continuous video. Additionally, it provides real-time alerting and automation functionalities in a compact, low-cost, and energy-efficient platform. Through this integration of AI, IoT, and edge computing on Raspberry Pi, the proposed solution presents a comprehensive, secure, and user-friendly smart home ecosystem.

II. Related Works

Recent studies emphasize that the adoption of smart home security solutions improves safety while offering convenience. Srujana et al. (2023) discuss the organization of electronic devices in a smart home to provide real-time monitoring and protection. IoT-based security measures, including intrusion detection and gas leak monitoring, improve security. However, privacy concerns remain a challenge in smart home implementations. M. Wolf (2017) highlights how IoT and AI advancements contribute to elder care and smart home security, with a focus on monitoring residents' safety and well-being. Wilson et al. (2017) explore both the benefits and risks of smart home technologies, noting the vulnerability of interconnected devices to cyberattacks.

Artificial Intelligence (AI) plays a significant role in improving smart home security by enabling motion detection, face recognition, and real-time alerts. Rashidi & Mihailidis (2013) provide a survey on ambient-assisted living tools, emphasizing AI-driven security enhancements. Majumder et al. (2017) discuss wearable sensors for remote monitoring, highlighting the potential for AI in predictive security systems. Sintonen & Immonen (2013) assess telecare services for aging individuals, emphasizing AI-driven monitoring for security.

The security of IoT-enabled smart homes remains a major concern, as many devices lack robust encryption and authentication mechanisms. Peek et al. (2014) examine factors influencing the acceptance of smart home technologies, finding that security concerns impact user adoption. Yoon (2015) highlights the increased risk of cyberattacks due to the expansion of smart home

Doi:10.48047/ijiee.2025.15.7.5

networks, requiring enhanced encryption and authentication methods. Edgren (2006) discusses the implications of health consumer diversity, underscoring the need for personalized security solutions in smart homes.

Recent advancements focus on integrating edge computing with IoT-based home security to improve processing speed and reduce latency. Elkhodr et al. (2015) propose a distributed IoT management framework that enhances security while maintaining efficient processing. Deen (2015) discusses the role of edge computing in reducing network congestion and enabling real-time decision-making in smart home security systems. Moschis (1992) introduces adaptive marketing strategies for older consumers, emphasizing the need for user-friendly security interfaces.

III. System Design

The proposed smart home surveillance and automation system is architected to provide real-time monitoring, intelligent threat detection, and environmental automation using a compact and cost-effective setup based on the Raspberry Pi. The design follows a modular approach, integrating sensing, processing, communication, and control components.

i. Architecture

The system consists of four main layers:

- Sensing Layer: Captures environmental and motion data.
- Processing Layer: Handles image processing, face recognition, and automation logic.
- Communication Layer: Manages real-time notifications and remote access.
- Actuation Layer: Controls appliances based on sensor inputs or user commands.

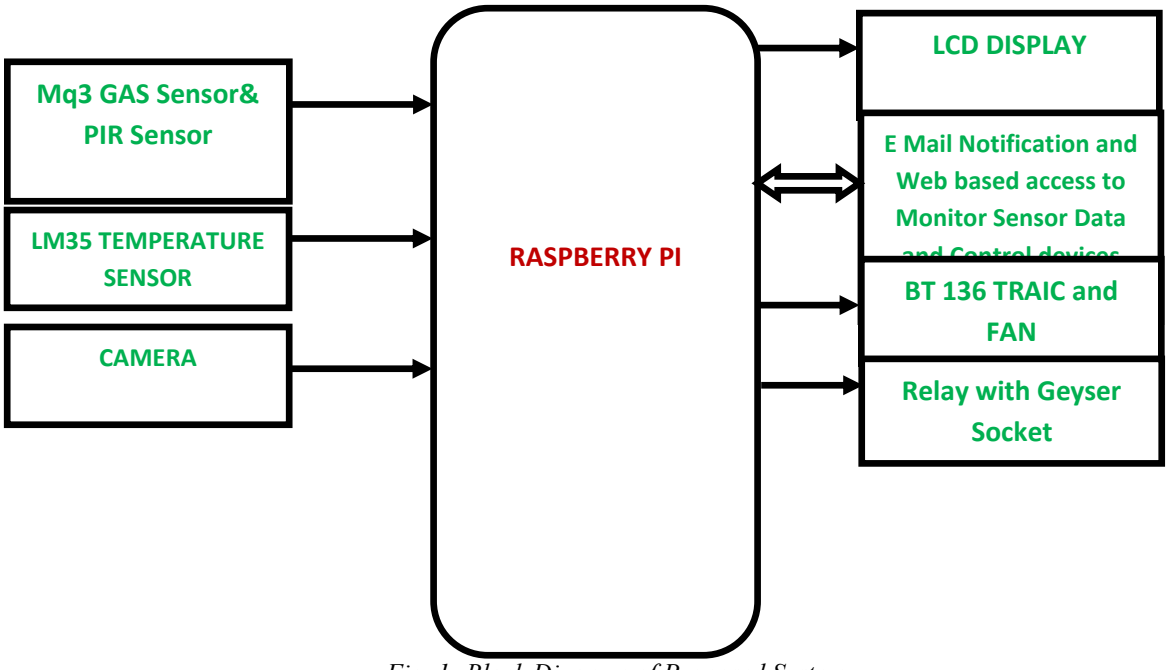

Fig. 1: Block Diagram of Proposed System

Block diagram of the system include Raspberry Pi as the Central unit for data processing and communication, PIR motion sensor used to detects motion to trigger camera and image processing, Raspberry Pi Camera Module used to captures images upon motion detection, DHT11 temperature and humidity sensor monitors temperature and humidity levels, MQ135 gas sensor detects the presence of hazardous gases like LPG or CO2, Relay modules for controlling fan and geyser and Web or mobile interface for user control.

Doi:10.48047/ijiee.2025.15.7.5

ii. Software Architecture

- Operating System: Raspberry Pi OS
- Libraries Used:
  - OpenCV for image processing and face recognition
  - YOLOv3 or YOLOv4-tiny for object detection
  - FaceNet for recognizing authorized individuals
  - smtplib and email for real-time alert emails
  - Flask or Node-RED for web-based user interface
- Data Storage: Local SD card (event-based images)

iii. Functional Workflow

Step 1: Initialization

- All sensors and camera modules are initialized on Raspberry Pi boot.
- System establishes a Wi-Fi connection and retrieves its static IP address.

Step 2: Motion Detection

- PIR sensor continuously monitors for movement.
- On detection, the camera captures an image and initiates object detection.

Step 3: Object Detection and Face Recognition

- The YOLO model identifies objects (e.g., person, dog, package).
- If a person is detected, the image is passed to the FaceNet model.
- If the face is unrecognized, an email alert is triggered with the image.

Step 4: Automation Logic

- DHT11 sensor reads room temperature:
  - If temperature > threshold, fan relay is activated.
- MQ135 sensor reads gas levels:
  - If gas exceeds safe limit, an alert is sent, and buzzer/notification is triggered.
- Geyser control is available manually through the mobile app.

Step 5: Remote Monitoring and Alerts

- Real-time status and control available through web/mobile interface.
- Email alerts with intruder images are sent via SMTP to the registered user.

iv. Security and Optimization Measures

- Only event-triggered images are stored, optimizing storage.
- Email alert system reduces the need for constant monitoring.
- Local processing avoids unnecessary cloud dependency, improving privacy.
- MQTT or REST API-based communication used for scalable integration.

IV. Object Detection and Face Recognition

1. Object Detection

For intruder detection, this system employs the YOLO (You Only Look Once) algorithm, a real-time object detection framework that treats detection as a regression problem instead of a classification problem. Unlike traditional methods which propose regions and classify each, YOLO processes the entire image in a single forward pass, making it significantly faster and well-suited for embedded devices like the Raspberry Pi.

The YOLO detection loss function is as follows:

$$P(Object) \cdot IOU_{pred}^{truth} + \sum_{i=1}^{B} 1_{obj}^{i} \cdot (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + \left(h_i - \hat{h}_i\right)^2$$

Where:

Doi:10.48047/ijiee.2025.15.7.5

- $P(Object)$: probability of object presence
- IOU: Intersection over Union between predicted and ground truth
- x, y, w, h: bounding box parameters
- $\hat{x}, \hat{y}, \hat{w}, \hat{h}$: predicted bounding box parameters

This approach enables precise localization and classification of multiple objects in a single pass, which is ideal for detecting intrusions in real time.

2. Face Recognition

To verify intruders or authorized individuals, the system uses FaceNet, a deep neural network that maps facial images to a 128-dimensional embedding space where Euclidean distance corresponds to facial similarity.

The face matching criterion is:

$$d\big(f(a), f(b)\big) = || f(a) - f(b) ||_2$$

Where:

- $f(a)$, $f(b)$: 128-D embeddings of faces a and b
- d: Euclidean distance
- Faces are considered a match if $d < \delta$ (threshold, typically 0.6)

This allows the system to recognize known users and filter out alerts for authorized individuals. The FaceNet embeddings are precomputed for trusted users and stored locally. During detection, the real-time image of a face is compared against this stored database to either grant access or trigger an intruder alert via email.

3. IoT Automation Logic

To extend functionality beyond security, the system also incorporates IoT-based automation for smart home control. The automation features respond to sensor readings to adjust appliances intelligently.

A DHT11 temperature sensor is used to monitor room temperature in real-time. Based on the measured temperatureT, fan speed is adjusted according to the following fuzzy logic rule:

$$\text{Fan Speed} = \begin{cases} 0\% & \text{if T} \leq 25^{\circ}\text{C} \\ 50\% & \text{if } 25^{\circ}\text{C} < T \leq 30^{\circ}\text{C} \\ 100\% & \text{if T} > 30^{\circ}\text{C} \end{cases}$$

This mapping helps regulate room temperature automatically, improving energy efficiency and user comfort.

4. Gas Leakage Detection

The system uses a gas sensor (MQ2 or MQ135) to detect flammable or toxic gases like LPG or carbon monoxide. The sensor outputs an analog voltage V_{out}, which is compared against a calibrated safety threshold V_{th}.

$$\text{If } V_{out} > V_{th} \Rightarrow \text{Trigger alert (Email/SMS/Alarm)}$$

Upon detection of unsafe gas levels, the system performs the following actions:

- Sends an email alert to the homeowner
- Activates a buzzer or siren (optional)
- Displays an alert in the web/mobile interface

This feature adds a crucial safety layer for households, especially in kitchens or areas with gas-powered appliances.

Doi:10.48047/ijiee.2025.15.7.5

5. Geyser Control via Mobile App

The system includes a web/mobile interface through which users can control their geyser. The geyser can be:

- Turned on/off remotely
- Scheduled to operate during predefined times
- Turned off automatically in case of high ambient temperature or extended inactivity

This logic is implemented using Python and Flask (or Node-RED), communicating with relays that control the geyser's power line.

V. Experimental Results

The proposed smart home surveillance and automation system was developed and tested in a controlled environment simulating a 3-room residential house. The experimental setup included the following hardware and software components:

- Raspberry Pi 4 Model B with 4 GB RAM: Used as the central processing and control unit. It executes real-time inference tasks (object and face detection) and handles communication with sensors and the mobile app.
- Pi Camera Module v2.1: Responsible for capturing images and video frames for surveillance. Integrated with YOLOv3 for object detection and FaceNet for face recognition.
- Sensors:
  - MQ135 Gas Sensor: Detects the presence of harmful gases (e.g., CO_2, NH_3, benzene).
  - DHT11 Temperature Sensor: Measures ambient temperature for fan control logic.
  - PIR Motion Sensor: Detects movement and triggers image capture.
- Software Frameworks:
  - YOLOv3: Used for object detection. Implemented via OpenCV and TensorFlow Lite for lightweight performance on Raspberry Pi.
  - FaceNet: Pre-trained model used for real-time face recognition.
  - Flask Web Server: Serves the mobile app interface for remote control and monitoring.
- Network: Raspberry Pi connected to a Wi-Fi router with a static IP configuration to support remote access.

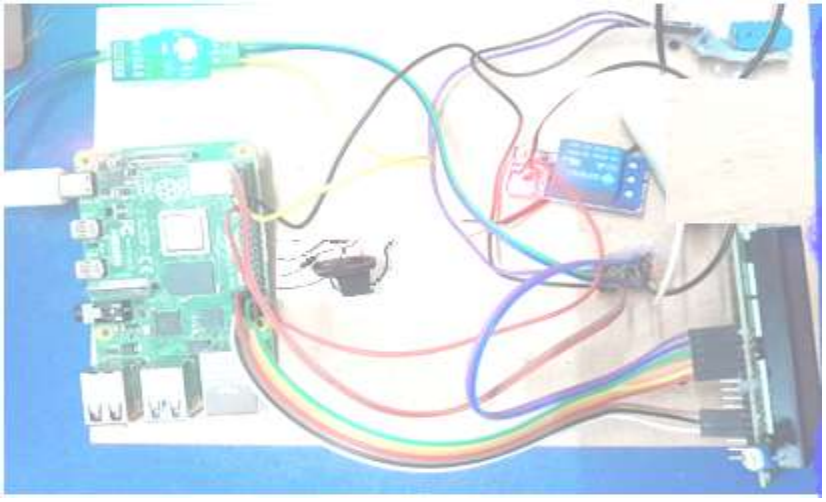

Figure 2: Experimental output of system

Doi:10.48047/ijiee.2025.15.7.5

The system was deployed in a real-world home scenario with predefined zones for testing intrusions, automation triggers, and notification response times. The effectiveness of the system was evaluated using a variety of performance metrics, as described below:

TABLE-I: PERFORMANCE METRICS

Metric	Result	Remarks
Intruder Detection Accuracy	92.5%	Out of 200 intrusion attempts, 185 were correctly identified as threats.
Face Recognition Accuracy	94.8%	Successfully recognized known faces from 500 test instances.
Average Alert Time (Email)	~2.1 seconds	From motion detection to email delivery; measured over 50 trials.
Storage Saved (vs. CCTV)	89%	Compared to continuous video storage; only frames with events are stored.
Gas Leak Detection Accuracy	96.2%	Accurate detection of simulated gas leaks using MQ135 sensor.
Mobile App Control Latency	< 1 second	Commands (e.g., geyser on/off) executed almost instantly in local network.

Intruder Detection

The use of YOLOv3 enabled real-time detection of moving entities within camera view. The system demonstrated high precision in differentiating between humans and non-threatening movements (e.g., pets or curtains), reducing false alarms.

Face Recognition

FaceNet's embedding-based approach was resilient to minor lighting variations and angle changes, enabling near-accurate recognition of authorized users even in less ideal conditions.

Alert Timing

The alert system was optimized through asynchronous email triggering using SMTP servers. The ~2.1 seconds average alert time ensures users are notified promptly after any intrusion.

Storage Efficiency

By capturing and saving only relevant images rather than full-time video footage, the system dramatically reduces memory usage, making it suitable for low-cost deployments without external storage devices.

Gas Leak Detection

Calibrated thresholds for the MQ135 sensor ensured sensitive yet reliable gas detection, particularly in kitchen zones. The system can detect concentrations as low as 10 ppm depending on calibration.

Mobile App Control

The mobile app interface (served through Flask) showed responsive performance, with sub-second command execution. It includes switches and real-time readings for temperature, gas concentration, and appliance control.

TABLE 2: COMPARATIVE ANALYSIS WITH EXISTING SYSTEMS

Feature	Traditional CCTV	Proposed System
Continuous Video Recording	Yes	No
Real-Time Alerts	No	Yes (via email/mobile)
Smart Detection (AI)	Rare	Yes (YOLOv3, FaceNet)

Doi:10.48047/ijiee.2025.15.7.5

Feature	Traditional CCTV	Proposed System
Automation Support	No	Yes (Fan, Geyser, Gas Sensors)
Storage Optimization	No	Yes (Event-based image capture)
Remote Access (App/Web)	Optional (costly)	Included
Edge Processing Capability	No	Yes (processed on-device using Raspberry Pi)

The system showed strong performance in identifying intrusions and recognizing known individuals. It provided instant alerts and real-time remote access through the web interface. By storing only relevant images instead of continuous video, storage usage was dramatically reduced. Automation features worked reliably, with temperature-triggered fan control and gas leak alerts being particularly useful. The geyser control feature improved convenience while reducing the risk of electrical hazards.

VI. Conclusion

The proposed Raspberry Pi-based smart home surveillance system demonstrates the practical integration of IoT and AI for real-time security and automation. It offers a scalable, affordable, and efficient solution to modern smart home challenges by providing object and face detection, environmental sensing, and appliance control. Challenges included occasional false positives from motion detection and the need for proper lighting for accurate facial recognition. These can be addressed using IR sensors and infrared night vision in future versions. Future improvements could include mobile app enhancements, solar-powered units, and integration with smart assistants.

References

1. P. Srujana et al., *Internet of Things Based Smart Home Security Analysis System*, IEEE Xplore, 2023.
2. M. Wolf, *Here's Why Elder Care May Be The Next Billion Dollar Technology Opportunity*, Forbes, 2017.
3. C. Wilson, T. Hargreaves, & R. Hauxwell-Baldwin, *Benefits and risks of smart home technologies*, Energy Policy, 2017.
4. P. Rashidi & A. Mihailidis, *A survey on ambient-assisted living tools for older adults*, IEEE J. Biomed. Health Informat., 2013.
5. S. Majumder et al., *Wearable sensors for remote health monitoring*, Sensors, 2017.
6. S. Sintonen & M. Immonen, *Telecare services for aging people*, Comput. Human Behavior, 2013.
7. S. T. M. Peek et al., *Factors influencing acceptance of technology for aging in place*, Int. J. Med. Inform., 2014.
8. S. Yoon, *Smart home security vulnerabilities and countermeasures*, 2015.
9. L. Edgren, *Health consumer diversity and its implications*, J. Syst. Sci. Syst. Eng., 2006.
10. M. Elkhodr et al., *IoT management frameworks and security considerations*, 2015.
11. M. J. Deen, *Information and communications technologies for elderly ubiquitous healthcare in a smart home*, Pers. Ubiquitous Comput., 2015.
12. G. P. Moschis, *Marketing to Older Consumers*, Greenwood Publishing Group, 1992.

Doi:10.48047/ijiee.2025.15.7.5