ISSN:2455-9679
SJIF Impact Factor: 6.190

IJRTSM

INTERNATIONAL JOURNAL OF RECENT TECHNOLOGY SCIENCE & MANAGEMENT

"IoT-ENABLED SMART SECURITY CAMERA WITH MOTION DETECTION AND FACE RECOGNITION USING RASPBERRY PI"

Sonali Verma[1], Vaishnavi Dharse[2], Gangotri Singroul[3], Prashant Kumar[4]

[1-4]Department of Electronics & Communication Engineering, Sagar Institute of Science, Technology & Research (SISTec-R), Bhopal, Madhya Pradesh, India

Sonaliverma698@gmail.com, dharsevaichnavi904@gmail.com, singraulgangotri@gmail.com

ABSTRACT

The rapid growth of Internet of Things (IoT) technology has created new opportunities for developing intelligent, low-cost, and automated security systems. Conventional security systems generally depend on continuous human monitoring and may provide limited remote accessibility. This research presents an IoT-enabled smart security camera system based on Raspberry Pi 3B+, a Passive Infrared (PIR) motion sensor, camera module, buzzer, wireless communication, and face-recognition technology. The proposed system continuously monitors a protected area and uses the PIR sensor to detect human movement. When motion is detected, the Raspberry Pi activates the camera and processes the captured image for face detection and recognition.

The system uses a three-stage processing pipeline: face-image acquisition, face-encoding/model preparation, and real-time security monitoring. Known individuals are identified and logged, whereas an unknown individual causes the system to generate a visual warning, activate the buzzer, save the captured image, and send an email alert. A Flask-based interface is also incorporated to support live camera monitoring through a browser. The system therefore combines motion sensing, computer vision, embedded processing, local alarm generation, event logging, and wireless notification in a single platform. The project provides a cost-effective and flexible approach to residential and small-area security monitoring and can be extended using cloud storage, artificial intelligence, mobile applications, smart locks, multiple cameras, and advanced biometric authentication.

Keywords: *Internet of Things, Smart Security Camera, Raspberry Pi 3B+, PIR Sensor, Motion Detection, Face Recognition, Computer Vision, Wireless Monitoring, Flask, Home Security.*

I. INTRODUCTION

Security and surveillance have become increasingly important because of unauthorized access, theft, and security risks in residential and other restricted environments. Conventional approaches such as locks, alarms, and manually monitored cameras may not provide automatic identification, immediate notification, or convenient remote access. IoT technology provides an opportunity to overcome these limitations by connecting sensors, processing units, cameras, and communication systems into an integrated security platform.

The proposed project, titled "IoT-Enabled Home Security System Using Raspberry Pi 3B+ with Camera, PIR Sensor, and Wireless Display," uses Raspberry Pi 3B+ as the central processing and control unit. The system combines a PIR motion sensor with a camera module to detect and visually verify suspicious activity. According to the attached project, the

ISSN:2455-9679
SJIF Impact Factor: 6.190

Raspberry Pi coordinates the sensors, camera, alarm, display, and internet communication.

The PIR sensor provides the first level of detection by identifying changes in infrared radiation associated with human movement. Once motion is detected, a signal is sent to the Raspberry Pi through GPIO. The camera then captures an image or video of the detected event. The captured image can be processed using a face-recognition algorithm to determine whether the detected individual belongs to the registered group.

The proposed system further improves security by incorporating a buzzer and wireless notification mechanism. For an unknown person, the system displays an "Unknown" label, activates the buzzer, saves the image, sends an email alert, and records the event.

The main objectives of this research are:

1. To develop a low-cost IoT-based smart surveillance system.
2. To detect human movement automatically using a PIR sensor.
3. To capture visual evidence when motion occurs.
4. To recognize registered individuals using face recognition.
5. To identify unknown individuals and generate security alerts.
6. To provide local and wireless remote monitoring.
7. To maintain event logs for later analysis.
8. To provide an expandable architecture for future AI-based security applications.

II. Review of Literature

IoT-based security systems have become an important application area of embedded computing and smart-home technology. The basic concept is to connect sensing devices, processing units, communication networks, and user interfaces so that security events can be detected and reported automatically.

2.1 IoT-Based Security Systems

IoT technology enables physical devices to collect information and exchange it through communication networks. In security applications, sensors can continuously monitor an environment while an embedded processor analyzes sensor signals and initiates appropriate actions. The attached project follows a sensor-to-processor-to-user/cloud architecture, in which environmental information is collected by sensors, processed by Raspberry Pi, and communicated to users through wireless connectivity.

2.2 Raspberry Pi-Based Surveillance

Raspberry Pi is widely used in embedded, IoT, automation, and computer-vision applications because it provides processing capability, GPIO interfaces, networking, and camera connectivity in a compact platform. The project uses Raspberry Pi 3B+ as the central controller. Its interfaces support sensor inputs, buzzer outputs, camera connection, display output, and network communication.

2.3 PIR-Based Motion Detection

PIR sensors are commonly used for motion detection because they detect changes in infrared radiation without actively emitting energy. In the proposed system, the PIR sensor acts as the first stage of security detection. When motion is detected, it sends a digital signal to the Raspberry Pi, which initiates the subsequent camera and alert operations. The project indicates a typical PIR detection range of approximately 3–7 metres, depending on the sensor and environment.

2.4 Camera-Based Surveillance

Camera modules provide visual evidence that cannot be obtained from motion sensors alone. The Raspberry Pi camera is connected through the CSI interface and can capture images or video after motion detection. The captured information can be stored locally or transmitted for remote monitoring.

ISSN:2455-9679
SJIF Impact Factor: 6.190

2.5 Face Recognition

Face recognition provides an additional security layer by distinguishing registered individuals from unknown individuals. The supplied implementation creates a dataset of face images, generates face encodings, stores them in an encoding file, and compares newly detected faces with the stored encodings. The live system classifies the detected face as either "KNOWN" or "UNKNOWN."

2.6 Alert and Remote Monitoring

An effective security system should not only detect an event but also communicate it to the user. The proposed system uses a buzzer for local warning and Wi-Fi for remote communication. The project also incorporates a Flask web interface for browser-based monitoring and email notification for unknown-person detection.

Overall, the literature and project architecture indicate a movement from conventional passive security systems toward integrated systems combining sensing, computer vision, embedded processing, and IoT communication.

III. PROBLEM IDENTIFICATION

Traditional surveillance systems face several limitations:

3.1 Continuous Human Monitoring

Conventional camera systems often require a person to continuously observe the video feed. This increases human workload and creates the possibility of missing important events.

3.2 Lack of Automatic Identification

A conventional motion detector can indicate movement but cannot determine whether the person is authorized or unauthorized. The proposed system addresses this limitation by adding face recognition.

3.3 Delayed Security Response

If an intrusion is detected only after reviewing recorded footage, the response may be delayed. An automated buzzer and remote notification mechanism can provide faster awareness.

3.4 Limited Remote Access

Traditional systems may require physical access to the surveillance location. IoT connectivity enables users to monitor security information remotely.

3.5 False Motion Alerts

Motion sensors may respond to unwanted environmental movement. The project therefore uses camera-based visual verification and face recognition after motion detection to improve the decision process.

3.6 Resource Constraints

The system operates on Raspberry Pi hardware with limited computational resources. The attached implementation therefore uses reduced streaming dimensions, a moderate frame rate, image scaling, and the HOG face-detection model to reduce processing requirements.

3.7 Need for Event Records

Security events should be recorded for later examination. The proposed software maintains text and CSV-based detection logs and saves captured images associated with security events.

ISSN:2455-9679
SJIF Impact Factor: 6.190

To address the identified problems, an integrated IoT-enabled smart security camera system is proposed.

The Raspberry Pi 3B+ serves as the central controller. The PIR sensor continuously monitors the environment and acts as an event trigger. When motion is detected, the camera captures an image. The image is processed using face detection and face recognition.

IV. SOLUTION

The proposed solution can be represented as:

PIR Sensor → Raspberry Pi → Camera → Face Detection → Face Recognition → Decision → Alert/Logging/Monitoring

For a **known person**, the system displays the person's name, provides a visual identification frame, and records the event.

For an **unknown person**, the system:

- Displays an "Unknown" identification label.
- Activates the buzzer.
- Saves the captured image.
- Generates an email notification.
- Records the event in the log.
- Displays a warning through the monitoring interface.

This decision structure is directly represented in the supplied project algorithm.

The solution also supports wireless monitoring. Wi-Fi allows captured images, alerts, and video streams to be communicated to remote devices such as smartphones and computers.

The major advantages of the proposed solution are:

- Low-cost embedded implementation.
- Automated motion detection.
- Face-based identification.
- Immediate local alarm.
- Remote wireless notification.
- Visual evidence collection.
- Event logging.
- Browser-based monitoring.
- Expandability for AI and cloud applications.

V. PROPOSED METHODOLOGY

5.1 System Architecture

The proposed architecture consists of five major layers:

Layer 1 – Sensing

The PIR sensor detects movement in the protected area. An ultrasonic sensor may additionally be used for distance and obstacle detection.

Layer 2 – Image Acquisition

ISSN:2455-9679
SJIF Impact Factor: 6.190

When motion occurs, the Raspberry Pi activates the camera to capture an image or video frame.

Layer 3 – Processing

The Raspberry Pi performs face detection and generates face encodings. The detected encoding is compared against the stored database of known individuals.

Layer 4 – Decision and Alert

The recognition result is classified as either known or unknown.

Layer 5 – Monitoring and Storage

The result is displayed locally or through the Flask web interface. Security events, images, and detection information are stored for future reference.

The supplied implementation specifically uses Raspberry Pi, OpenCV, NumPy, pandas, the face-recognition library, Flask, Picamera2, and GPIO-related functionality.

5.2 Face Dataset Preparation

First, photographs of authorized individuals are collected using the Raspberry Pi camera. The photographs are stored in person-specific dataset folders. The implementation allows multiple images to be collected for each registered person.

5.3 Face Encoding and Training

The collected images are processed using face detection and face encoding. The generated encodings and corresponding names are stored in encodings.pickle. During operation, these stored encodings are loaded into memory for comparison with newly detected faces.

5.4 Motion-Triggered Recognition

The system continuously waits for a PIR motion signal. The use of motion as a trigger avoids performing unnecessary face-recognition operations continuously and helps reduce computational requirements.

5.5 Decision Process

When a face is detected, its encoding is compared against known encodings. The implementation calculates face distances and determines the best matching candidate. The result is classified as either **KNOWN** or **UNKNOWN**.

5.6 Alert Generation

For an unknown person, the system generates a buzzer alert and stores the captured event. Email notification can also be enabled. The implementation provides configurable parameters for the PIR sensor, buzzer, face-recognition tolerance, image capture, streaming, and email alert functions.

5.7 Web-Based Monitoring

A Flask server provides a browser-based monitoring interface. The system can provide a live camera feed and security information through the web interface, allowing remote monitoring over the connected network.

Algorithm

Step 1: Start the system.

ISSN:2455-9679
SJIF Impact Factor: 6.190

Step 2: Initialize Raspberry Pi components including:

- Camera
- PIR sensor
- GPIO
- Buzzer
- Flask server

Step 3: Load stored face encodings.

Step 4: Enter continuous monitoring mode.

Step 5: Check the PIR sensor.

Step 6: If motion is not detected, continue monitoring.

Step 7: If motion is detected, capture an image using the camera.

Step 8: Detect faces in the captured image.

Step 9: If no face is detected, return to monitoring.

Step 10: If a face is detected, generate its face encoding.

Step 11: Compare the new encoding with stored known-face encodings.

Step 12: If the person is known:

- Display the person's name.
- Mark the face as known.
- Store the detection event.
- Continue monitoring.

Step 13: If the person is unknown:

- Display an “Unknown” label.
- Activate the buzzer.
- Save the captured image.
- Send an email alert.
- Store the event in the log.
- Display a warning.

Step 14: Return to monitoring mode.

Step 15: Continue the process until the system is stopped.

This algorithm follows the process described in the supplied project document.

VI. RESULT

The developed system provides an integrated approach to motion-triggered surveillance and face-based security identification.

ISSN:2455-9679
SJIF Impact Factor: 6.190

The implementation establishes three major operational stages: image capture, model training, and live face-recognition security monitoring. In the live stage, a known person is displayed with a name label and recorded in the event log, while an unknown person produces a warning response involving the buzzer, popup, captured image, and email notification.

The major observed functional results of the proposed system are summarized below.

Test Condition	System Response	Expected Result
No motion	System remains in monitoring mode	Continuous surveillance
Motion detected	Camera activation	Image/video acquisition
Motion with no detectable face	Return to monitoring	No false identity assignment
Known face detected	Name and known status displayed	Authorized-person identification
Unknown face detected	Red identification frame and warning	Unauthorized-person detection
Unknown person	Buzzer activated	Local security alert
Unknown person	Image saved	Visual evidence
Unknown person	Email notification	Remote security awareness
Security event	Log updated	Event history
Active network	Flask interface available	Remote/live monitoring

The supplied project states that the system continuously returns to monitoring after completing an alert cycle, thereby supporting continuous surveillance.

The software configuration also demonstrates an effort to balance recognition performance and Raspberry Pi processing limitations. The implementation uses a reduced streaming resolution, 15 FPS streaming, image scaling, and a configurable face-recognition tolerance.

VII. CONCLUSION

This research presents an IoT-enabled smart security camera system using Raspberry Pi 3B+, PIR motion detection, camera-based surveillance, face recognition, buzzer alerts, wireless communication, and web-based monitoring.

The proposed system provides a multi-stage security mechanism in which motion detection acts as the initial trigger, camera capture provides visual information, and face recognition provides identity-based classification. Known users can be recognized and logged, while unknown individuals can trigger multiple security responses including visual warnings, buzzer activation, image storage, and email notification.

The system demonstrates the usefulness of combining IoT, embedded computing, computer vision, and wireless communication for smart surveillance. Its modular architecture makes it suitable for residential environments as well as offices, educational institutions, restricted areas, warehouses, and other locations requiring automated monitoring. The attached project also identifies low cost, real-time monitoring, wireless access, automation, expandability, and easy installation as important advantages.

The research also identifies the importance of optimizing computational resources when computer-vision algorithms are executed on resource-constrained embedded platforms. The proposed architecture provides a practical foundation for more advanced intelligent security systems.

VIII. FUTURE SCOPE

The proposed system can be further enhanced in several directions:

1. **Advanced Face Recognition:** Improve recognition performance using more robust deep-learning-based models.
2. **Cloud Storage:** Store security images and event records securely in cloud platforms.
3. **Mobile Application:** Develop an Android/iOS application for real-time notifications and monitoring.

ISSN:2455-9679
SJIF Impact Factor: 6.190

4. **AI-Based Motion Detection:** Use artificial intelligence to distinguish humans, animals, vehicles, and irrelevant motion.
5. **Smart Lock Integration:** Connect the security system with electronic locks for automated access control.
6. **Voice Assistant Integration:** Enable voice-based security status and control.
7. **Multiple Camera Support:** Add multiple cameras to increase surveillance coverage.
8. **Night Vision:** Add infrared/night-vision capability for low-light environments.
9. **SMS and Call Alerts:** Provide alternative communication channels when internet-based notification is unavailable.
10. **Event Analytics:** Develop dashboards for analyzing historical security events.
11. **Biometric Authentication:** Combine face recognition with other biometric methods.
12. **Fire and Smoke Detection:** Integrate additional safety sensors.
13. **False-Alarm Reduction:** Develop intelligent classification methods to reduce unnecessary alerts.
14. **Improved Video Streaming:** Optimize network and video-processing performance.
15. **Power Optimization:** Reduce energy consumption and support backup or solar-powered operation.
16. **Emergency Response:** Integrate automated emergency notification mechanisms.
17. **Smart-City Integration:** Extend the architecture to larger surveillance networks.

IX. REFERENCES

1. Raspberry Pi Foundation, Raspberry Pi Documentation.
2. Raspberry Pi Foundation, Raspberry Pi Camera Module Documentation.
3. Raspberry Pi Foundation, Raspberry Pi OS Documentation.
4. Raspberry Pi Foundation, Getting Started with Raspberry Pi.
5. Python Software Foundation, Python Documentation.
6. OpenCV, Open Source Computer Vision Library Documentation.
7. GPIO Zero, GPIO Zero Documentation.
8. HC-SR501, PIR Motion Sensor Datasheet.
9. Flask, Flask Web Framework Documentation.
10. IEEE, IEEE Xplore Digital Library.
11. Raspberry Pi, GitHub Documentation and Repositories.
12. OpenCV, Official OpenCV Resources.