Shinde Karan Laxman ,2026, 14:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Smart Security Cameras Using Iot: A Review Of Raspberry PI Video Streaming, Opencv-Based Detection, Face Recognition And Secure Alerting

Shinde Karan Laxman[1]; Prof. S. R. Borhade[2]; Prof. M. C. Kolse[3]; Prof. S. K. Said[4]
Department Of Artificial Intelligence And Data Science Jaihind College Of Engineering Pune, Maharashtra, India

Abstract—IoT-based smart security cameras offer intelligent, automated monitoring at low cost, making them attractive for homes and small businesses. This paper reviews smart security camera systems built on single-board computers such as the Raspberry Pi with a Pi Camera module, which stream video over the local network and to remote users through a web interface. It examines the computer vision pipeline, from classical motion detection by background subtraction and pedestrian detection with histogram-of-oriented-gradients features to deep-learning object detectors and lightweight networks for embedded devices, as well as face recognition based on learned embeddings for distinguishing known people from strangers. Edge processing for real-time video analytics, alert mechanisms such as automated e-mail and push notifications with captured snapshots, and storage of event clips are discussed. The security of the camera itself is reviewed, since poorly configured IoT devices are routinely compromised. A five-layer architecture comprising capture, edge vision, event management, secure communication and user interface layers is proposed, together with research gaps, objectives, a comparison of detection methods and challenges such as false alarms, low light, privacy and limited computing power. The review concludes that edge-based intelligent cameras can provide affordable, responsive and privacy-conscious residential security.

Index Terms—Smart security camera, IoT, Raspberry Pi, OpenCV, object detection, face recognition, edge computing, e-mail alerts

I. INTRODUCTION

A conventional CCTV system records hours of footage that someone must watch to find an incident, often after the damage is done. A smart security camera analyses video as it is captured, recognises when a person appears and alerts the owner immediately with a snapshot. Affordable hardware such as the Raspberry Pi and open-source libraries such as OpenCV [1] make such systems feasible for ordinary households.

Edge computing lets video be analysed close to the camera, reducing bandwidth and latency. Ananthanarayanan *et al.* [2] described real-time video analytics as a key application of edge computing, and Satyanarayanan [3] explained why processing near the data source matters.

This paper reviews smart security camera technologies and proposes a secure, edge-based architecture.

II. THEORETICAL BACKGROUND

1. Functions Of A Smart Security Camera

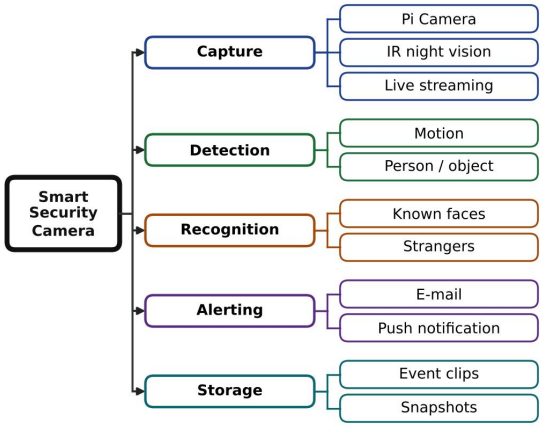


Fig. 1 organizes the functions into capture and streaming, detection, recognition, alerting and storage.

© 2026 Shinde Karan Laxman , This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

Fig. 1. Functions of an IoT-based smart security camera.

2. Hardware Platform

A Raspberry Pi with a Pi Camera captures video and runs Python with OpenCV. Video is streamed over HTTP (for example MJPEG) to browsers on the local network, and optionally through a secure tunnel to remote users.

III. SURVEY OF EXISTING APPROACHES

Motion and person detection. Stauffer and Grimson [4] modelled each pixel with a mixture of Gaussians to separate moving foreground from background, the basis of many motion detectors. Viola and Jones [5] introduced fast cascade detectors, and Dalal and Triggs [6] proposed HOG features for pedestrian detection, available in OpenCV.

Deep detection on embedded devices. YOLO [7] performs real-time object detection in a single pass, and MobileNets [8] make convolutional networks efficient enough for embedded processors such as the Raspberry Pi.

Face recognition. Schroff *et al.* [9] proposed FaceNet, which maps faces to embeddings where distance reflects identity, and Deng *et al.* [10] improved discrimination with an additive angular margin loss.

Security of IoT cameras. Kolias *et al.* [11] showed how the Mirai botnet compromised IP cameras with default passwords, and Alrawi *et al.* [12] systematically evaluated home IoT deployments and found widespread weaknesses in devices, apps and cloud endpoints. Table I summarizes these works.

Ref.	Author(s) / Year	Technique / Approach	Key Contribution	Limitation / Gap
[4]	Stauffer and Grimson, 1999	Gaussian mixture background	Robust motion detection	Lighting changes
[5]	Viola and Jones, 2001	Cascade detector	Real-time face detection	Frontal faces
[6]	Dalal and Triggs, 2005	HOG pedestrian detection	Accurate person detection	Slow on Pi
[9]	Schroff et al., 2015	FaceNet embeddings	Compact face recognition	Needs good images
[7]	Redmon et al., 2016	YOLO detector	Real-time detection	Heavy for Pi
[8]	Howard et al., 2017	MobileNets	Efficient embedded CNNs	Lower accuracy
[11]	Kolias et al., 2017	Mirai botnet analysis	Cameras exploited via defaults	Attack study
[12]	Alrawi et al., 2019	Home IoT security SoK	Widespread weaknesses	Devices change fast

TABLE I Summary of Literature Survey

IV. COMPARATIVE STUDY

Table II compares detection methods. Background subtraction is fast but noisy, HOG is moderate, and deep detectors are accurate but need more computation or accelerators.

Method	Speed on Pi	Accuracy	False Alarms
Frame differencing	Very fast	Low	High
Gaussian mixture BG	Fast	Medium	Medium
HOG + SVM	Moderate	Medium	Medium
MobileNet-SSD	Moderate	High	Low
YOLO (with accelerator)	Fast	High	Low

TABLE II Comparison of Detection Methods for Smart Cameras

V. RESEARCH GAPS

- **G1:** Simple motion detection triggers false alarms from pets, shadows and trees.
- **G2:** Deep models are slow on low-cost boards without acceleration.
- **G3:** Face recognition raises privacy and consent issues.
- **G4:** Many cameras ship with weak credentials and unencrypted streams.
- **G5:** Performance in low light and bad weather is limited.

VI. AIMS AND OBJECTIVES

- To review hardware and streaming methods for IoT security cameras.
- To compare motion, person and object detection techniques.
- To examine face recognition and alerting mechanisms.
- To study security weaknesses of IoT cameras.
- To propose a secure, edge-based smart camera architecture.

VII. PROPOSED ARCHITECTURE

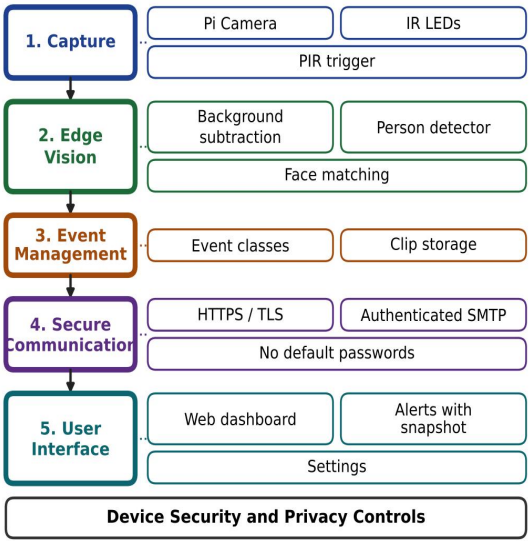


Fig. 2 shows the proposed architecture.

Fig. 2. Proposed five-layer edge-based smart security camera architecture.

1. Capture Layer

A Pi Camera (or USB camera) with optional infrared illumination captures video; a PIR sensor can wake processing.

2. Edge Vision Layer

Background subtraction finds motion regions; a lightweight detector confirms persons; face embeddings are matched against a list of known residents.

3. Event Management Layer

Events are classified as known person, unknown person or object; short clips and snapshots are stored locally.

4. Secure Communication Layer

Streams and alerts use HTTPS or TLS; e-mail alerts are sent through an authenticated SMTP server; default credentials are disabled.

5. User Interface Layer

A web dashboard provides live view, event history and settings; push or e-mail notifications include snapshots.

VIII. PROCESS FLOW AND EVALUATION

Fig. 3 describes the processing loop. Motion is detected by comparing the current frame with a background model,

$$M_t(p) = 1 \text{ if } |I_t(p) - B_t(p)| > \tau \quad (1)$$

and a face is recognised as a known person when the embedding distance satisfies

$$\|f(x) - f(x_ref)\| < \delta \quad (2)$$

Otherwise, an alert is raised.

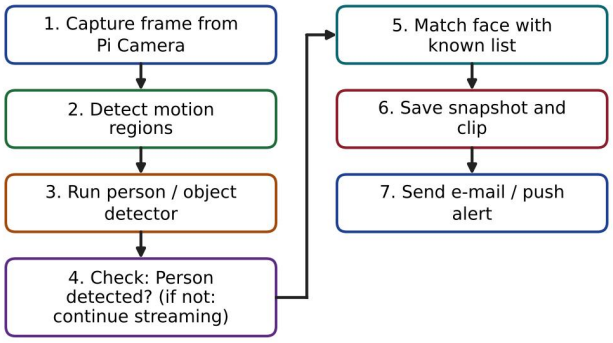


Fig. 3. Smart camera processing and alert loop.

IX. LIMITATIONS AND FUTURE SCOPE

Table III summarizes challenges. Future directions include on-device AI accelerators, activity recognition (for example, detecting loitering or falls), privacy masking of neighbours' property and integration with smart locks.

Challenge	Mitigation
False alarms	Person detection after motion
Low computing power	Lightweight models; accelerators
Low light	IR illumination; tuned exposure
Privacy	Local processing; masking; consent
Device compromise	Strong passwords; TLS; updates

TABLE III Challenges and Mitigation

X. CONCLUSION

IoT-based smart security cameras use low-cost hardware, computer vision and connectivity to deliver real-time monitoring and alerts. The reviewed literature provides efficient detection and recognition methods and highlights serious security risks of poorly configured devices. An edge-based design with person detection, secure communication and user-friendly alerts can provide effective and privacy-conscious home security.

1. Acknowledgment

The authors express their sincere gratitude to the Department of Artificial Intelligence and Data Science, Jaihind College of Engineering, Pune, for providing the facilities, guidance and encouragement needed to carry out this work. The authors also thank the faculty members and colleagues whose valuable suggestions helped improve the quality of this review.

REFERENCES

1. G. Bradski, "The OpenCV library," Dr. Dobb's J. Softw. Tools, vol. 25, no. 11, pp. 120–125, 2000.
2. G. Ananthanarayanan et al., "Real-time video analytics: The killer app for edge computing," Computer, vol. 50, no. 10, pp. 58–67, 2017.
3. M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.
4. C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in Proc. IEEE CVPR, 1999, vol. 2, pp. 246–252.
5. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proc. IEEE CVPR, 2001, pp. I-511–I-518.
6. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE CVPR, 2005, pp. 886–893.
7. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE CVPR, 2016, pp. 779–788.
8. A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.
9. F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proc. IEEE CVPR, 2015, pp. 815–823.
10. J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in Proc. IEEE/CVF CVPR, 2019, pp. 4690–4699.
11. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," Computer, vol. 50, no. 7, pp. 80–84, 2017.
12. O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "SoK: Security evaluation of home-based IoT deployments," in Proc. IEEE Symp. Security Privacy, 2019, pp. 1362–1380.