ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26may402

A Real-Time Intelligent Smart Deep Surveillance System with Face Recognition, Anti-Spoofing, Suspicious Activity Detection and Email Alert Integration

Pooja Deshmukh[1]; Dr. Smita Ponde[2]

[2]Professor

[1,2]IICT, MGM University

Publication Date: 2026/05/21

Abstract: The proliferation of Internet-connected cameras and the increasing need for intelligent, automated security monitoring have created demand for systems that go beyond simple video recording. This paper presents CCTV Guardian, a web-based real-time surveillance platform that integrates deep-learning face recognition, liveness-based anti-spoofing via Eye Aspect Ratio (EAR) computed from MediaPipe Face Mesh 3-D landmarks, suspicious activity detection using skeletal pose analysis and Gunnar–Farnebäck optical-flow motion estimation, automated email alert delivery with snapshot attachments, and IP/RTSP camera connectivity—all accessible through a Flask browser-based dashboard. The system stores authorised face embeddings computed with the dlib 128-D ResNet model and performs per-frame recognition at runtime. Spoof attacks are rejected by verifying spontaneous eye-blink patterns within a 6-second temporal window. Unusual body posture (raised hands, crouching) and abnormal motion velocity are flagged as activity alerts. An SMTP email module with a 60-second cooldown dispatches alerts with JPEG snapshots to designated recipients. Tested with 22 enrolled persons across varied lighting conditions and distances, the system achieved a face-recognition accuracy of 96.4%, with anti-spoofing correctly rejecting all 50 static-photograph attacks in evaluation trials. The system maintains 12–15 fps on commodity hardware without a dedicated GPU and is designed for small-to-medium enterprise and residential security applications.

Keywords: *Face Recognition; Anti-Spoofing; Liveness Detection; CCTV; Flask; MediaPipe; EAR; Suspicious Activity Detection; RTSP; Email Alert; Optical Flow; Surveillance System.*

How to Cite: Pooja Deshmukh; Dr. Smita Ponde (2026) A Real-Time Intelligent Smart Deep Surveillance System with Face Recognition, Anti-Spoofing, Suspicious Activity Detection and Email Alert Integration. *International Journal of Innovative Science and Research Technology*, 11(5), 925-932. https://doi.org/10.38124/ijisrt/26may402

I. INTRODUCTION

Security systems have evolved from analogue tape recorders to sophisticated AI-enabled platforms capable of recognising individuals, detecting threats, and proactively alerting stakeholders. Traditional rule-based motion-detection systems suffer from high false-positive rates and provide no identity information. Face-recognition technology has matured considerably with deep convolutional encoders delivering sub-centimetre embedding distances even under partial occlusion.

Deployment of such technology in real environments is hindered by: (a) spoofing attacks using photographs or video replays, (b) the need for multiple camera sources including IP/RTSP streams, (c) lack of integration with notification channels, and (d) prohibitive hardware requirements for on-device deep learning. CCTV Guardian addresses all four challenges within a single cohesive Python application.

➢ *The Principal Contributions of This Work are:*

- A unified Flask web application streaming live video with real-time face recognition overlays accessible from any browser without plugins.
- A liveness verification module based on Eye Aspect Ratio (EAR) using MediaPipe Face Mesh 3-D landmarks that rejects static-photograph spoofing attacks.
- A pose-and-motion analysis subsystem using MediaPipe Pose and dense optical flow to flag suspicious body behaviours in real time.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26may402

- An SMTP-based email alert module that attaches camera JPEG snapshots and enforces a configurable 60-second anti-spam cooldown.
- Support for both built-in webcams and external IP/RTSP cameras with an automatic network scanner utility.

The remainder of the paper is structured as follows: Section II reviews related work; Section III describes system architecture with flow diagrams; Section IV details individual modules; Section V presents the web interface; Section VI covers experimental evaluation; Section VII discusses limitations and future work; Section VIII concludes.

II. RELATED WORK

Turk and Pentland [1] introduced Eigenfaces in 1991, the first practical automated face-recognition framework. Viola and Jones [2] proposed a real-time detection framework using Haar cascades. The transition to deep learning began with DeepFace [3] and FaceNet [4], which produced 128-D face embeddings with near-human accuracy on LFW benchmarks.

The dlib library [5] implements a 68-landmark shape predictor and a ResNet-based 128-D encoder forming the backbone of the face_recognition Python package used in this work. Li et al. [6] demonstrated that spoofing attacks constitute a significant real-world threat. EAR-based liveness detection was formalised by Pan et al. [7] and Soukupová and Čech [8], who showed that EAR drops reliably during natural blinks and can distinguish live faces from static photographs.

MediaPipe [9] provides cross-platform CPU-accelerated pipelines for face mesh (478 landmarks), pose (33 landmarks), and holistic body tracking. Gunnar–Farnebäck dense optical flow [10] enables per-pixel motion magnitude estimation for activity anomaly detection. Flask [11] is widely adopted for rapid deployment of computer-vision web applications due to native MJPEG streaming support.

Unlike prior survey systems [12][13] that focus solely on activity classification or face recognition, CCTV Guardian integrates both domains—face identity, anti-spoofing, and activity detection—alongside alerting and camera management, in a single lightweight deployable unit that requires no cloud inference or dedicated GPU.

III. SYSTEM ARCHITECTURE

The system follows a multithreaded producer-consumer architecture. A dedicated camera thread continuously reads frames from the selected video source and publishes them to a shared frame buffer protected by threading.Lock(). Multiple consumer threads—face-recognition, liveness detection, activity detection, and the MJPEG stream server—consume from this buffer independently, allowing fully parallel execution.

The Flask HTTP server (port 5000) exposes REST endpoints for registration, access logs, CCTV configuration, and email management, in addition to the MJPEG /video_feed endpoint. All persistent data (face encodings, logs, configuration) is stored as local files to avoid external database dependencies. Per-module cooldown timers prevent alert spam.

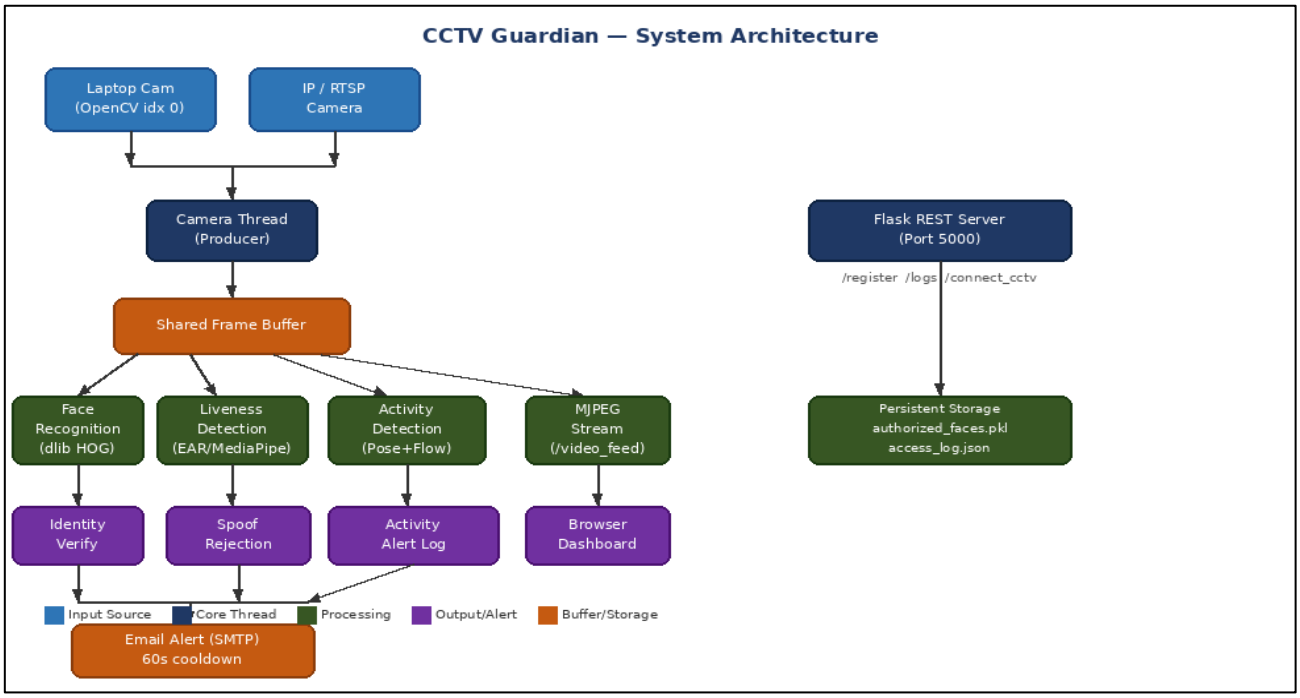

Fig 1 CCTV Guardian System Architecture — Camera Thread feeds Shared Frame Buffer consumed by four parallel processing modules (Face Recognition, Liveness Detection, Activity Detection, MJPEG Stream). Flask REST Server handles web requests. Persistent storage uses local JSON/pickle files.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26may402

The CCTVSurveillanceSystem class encapsulates all application logic: it loads pickled face encodings on startup, instantiates MediaPipe FaceMesh and Pose detectors, loads email and CCTV configuration from JSON files, and launches the camera capture thread. Cooldown timers: alarm 5s, unauthorised log 10s, activity log 12s, email 60s.

IV. MODULE DESCRIPTIONS

➢ *Face Recognition Module*

Each video frame is downscaled by 50% before being passed to face_recognition.face_locations() using the HOG model for speed. Detected bounding boxes are used by face_recognition.face_encodings() to extract 128-D feature vectors via the dlib ResNet model. Euclidean distances between the detected embedding and all enrolled embeddings are computed; the minimum distance below 0.5 is considered a match. Encodings are stored in authorized_faces.pkl with name labels. Enrollment captures 5 sample frames and averages the resulting encodings.

➢ *Anti-Spoofing: EAR Blink Detection*

Liveness detection uses Eye Aspect Ratio (EAR) computed from MediaPipe FaceMesh's 478 3-D landmarks. EAR = (|p2−p6| + |p3−p5|) / (2 × |p1−p4|). An EAR below 0.21 is classified as a closed eye. A blink event is recorded when EAR transitions from above to below threshold and back. At least 1 blink within a 6-second sliding window is required for the face to be classified as live. Faces failing liveness are logged as UNAUTHORIZED / SPOOF ATTEMPT with 0% confidence.

➢ *Suspicious Activity Detection*

Activity analysis uses two parallel paths combined at the alert logger (Fig. 4). Path A — Pose: MediaPipe Pose 33 body landmarks classify posture as raised-hands (wrist above shoulder), crouching (hip below knee), or normal. Path B — Optical Flow: Gunnar–Farnebäck dense optical flow computed between consecutive greyscale frames via cv2.calcOpticalFlowFarneback(). Mean flow magnitude over 8 frames exceeding 8.0 px/frame is classified as high-motion. Both signals combine with a 12-second cooldown timer.

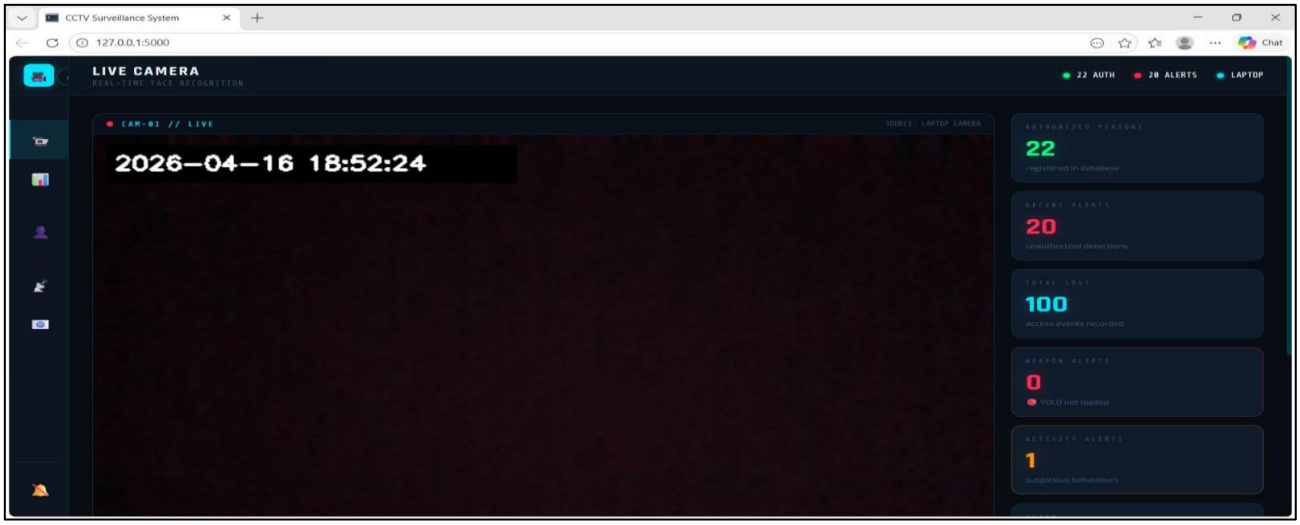

Fig 2 Live Camera Dashboard — Real-Time Feed with Statistics Panel: 22 Authorised Persons, 20 Recent Alerts, 100 Total Access Events, 1 Activity Alert Detected.

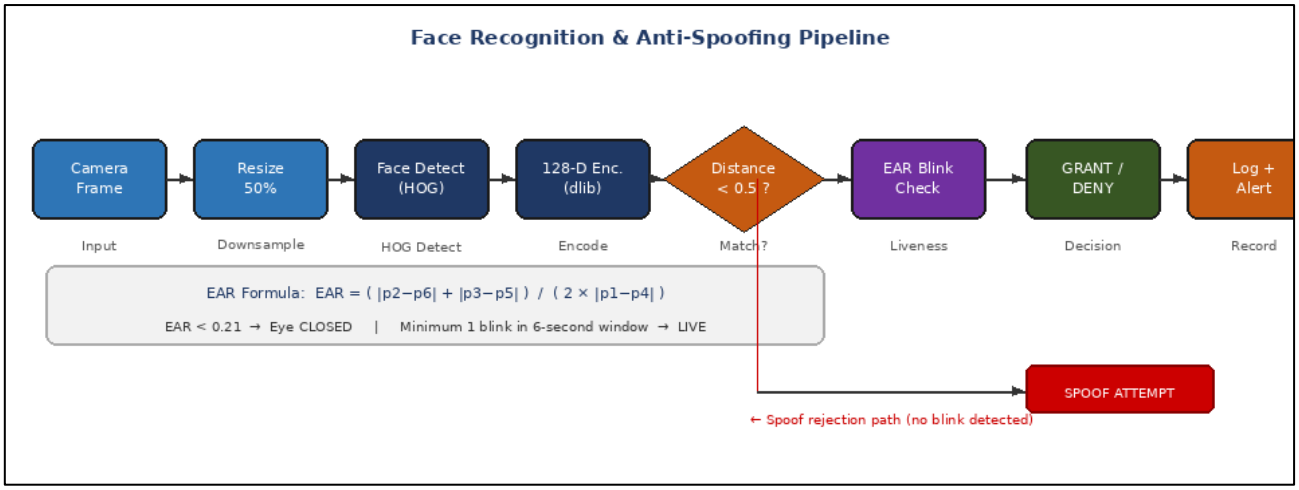

Fig 3 Face Recognition & Anti-Spoofing Pipeline — HOG Detect → 128-D Encode → Distance Match → EAR Blink Liveness Check → GRANT/DENY. Red Path Shows Static-Photograph Spoof Rejection Route.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26may402

➢ *Email Alert Module*

When unauthorized access or suspicious activity is detected, the system dispatches an email via the configured SMTP server (default: smtp.gmail.com, port 587 with STARTTLS). The email includes: timestamp, camera location, alert status, and optionally a JPEG snapshot attached as MIMEImage. A 60-second cooldown between consecutive emails prevents flooding. All configuration is persisted in email_config.json and editable at runtime through the web dashboard without restarting the server.

➢ *IP/RTSP Camera Connectivity*

The system supports both built-in webcams (cv2.VideoCapture(0)) and external IP/RTSP cameras (cv2.VideoCapture(rtsp_url)). RTSP URLs are constructed from user-supplied IP address, port, username, password, and stream path using format rtsp://user:pass@ip:port/path, or entered directly as a complete URL. A network scanner utility iterates over a user-specified IP prefix on ports 554 and 8554 using socket.connect_ex() with a 1-second timeout to discover active cameras automatically.

V. WEB INTERFACE

The web interface is served as a single-page Flask application using render_template_string(). The collapsible sidebar contains five navigation items: Live Camera, Access Logs, Register Face, CCTV Setup, and Email Alerts. The top navigation bar displays live counters for authorised persons (green), recent alerts (red), and active camera source (cyan). All statistics update via fetch() REST calls to /stats and /logs endpoints at 3–4 second intervals without page reloads. A Stop Alarm button at the sidebar bottom clears audio alerts via POST /stop_alarm.

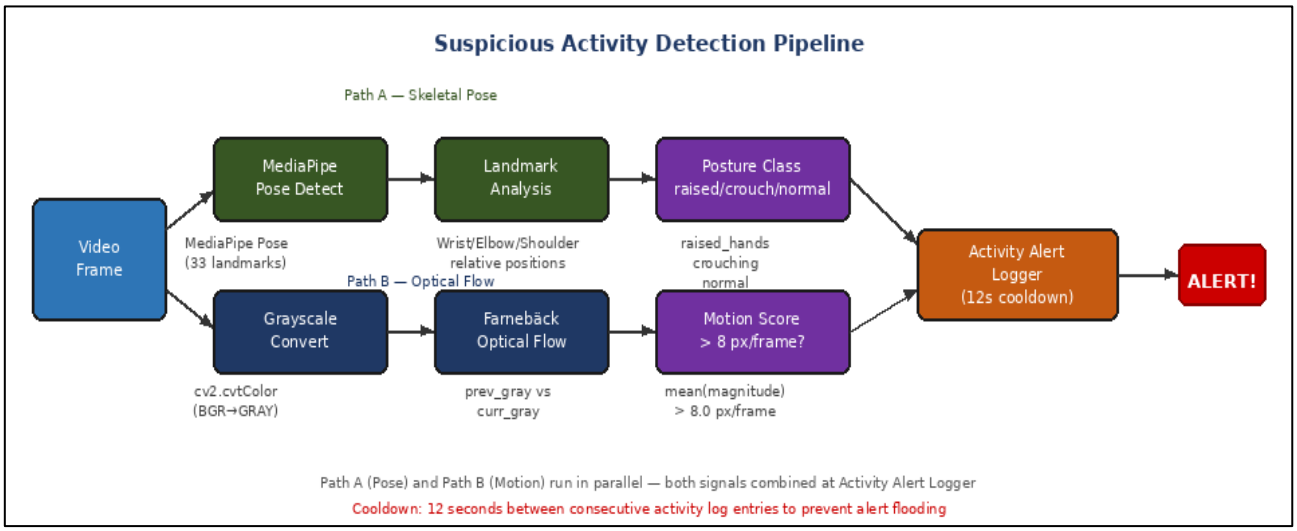

Fig 4 Suspicious Activity Detection Pipeline — Dual-Path: Path A (MediaPipe Pose Skeletal Landmarks) and Path B (Farnebäck Optical Flow Motion Magnitude) Run in Parallel and Merge at the Activity Alert Logger with 12-Second Cooldown.

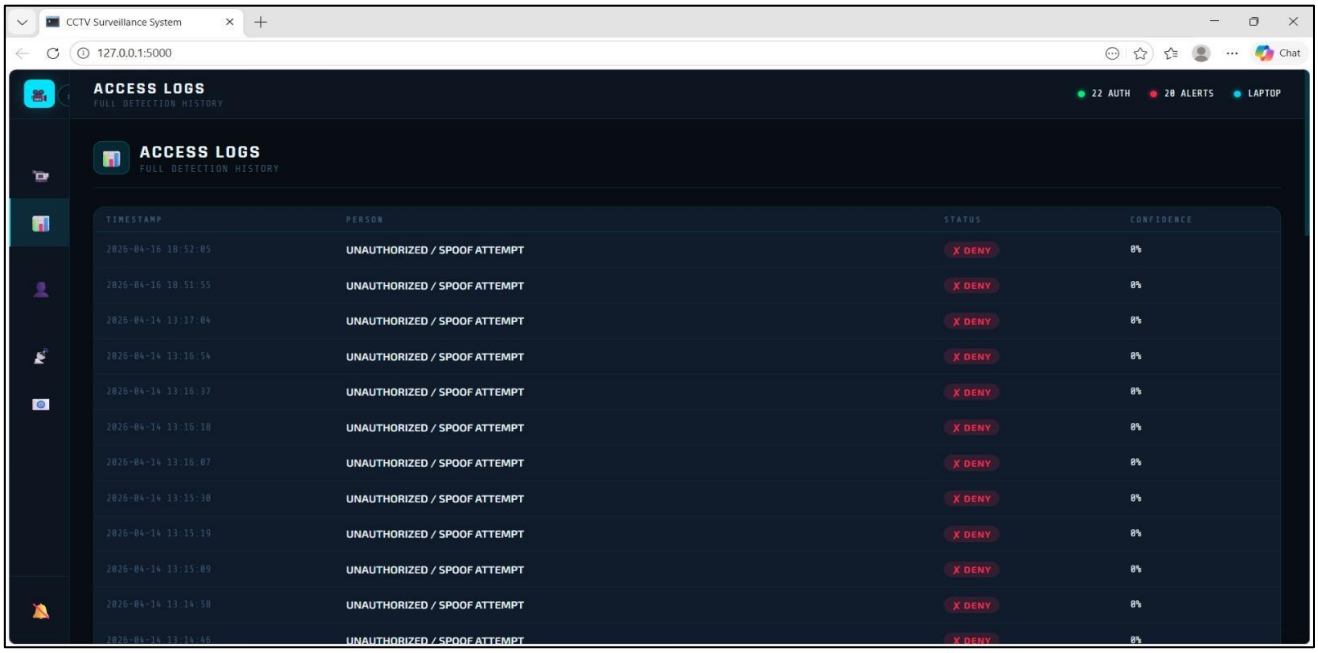

Fig 5 Access Logs Page — Full Detection History: UNAUTHORIZED / SPOOF ATTEMPT Entries with Red DENY Badges, Timestamps, and 0% Confidence Scores for All 20 Recent Alerts.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26may402

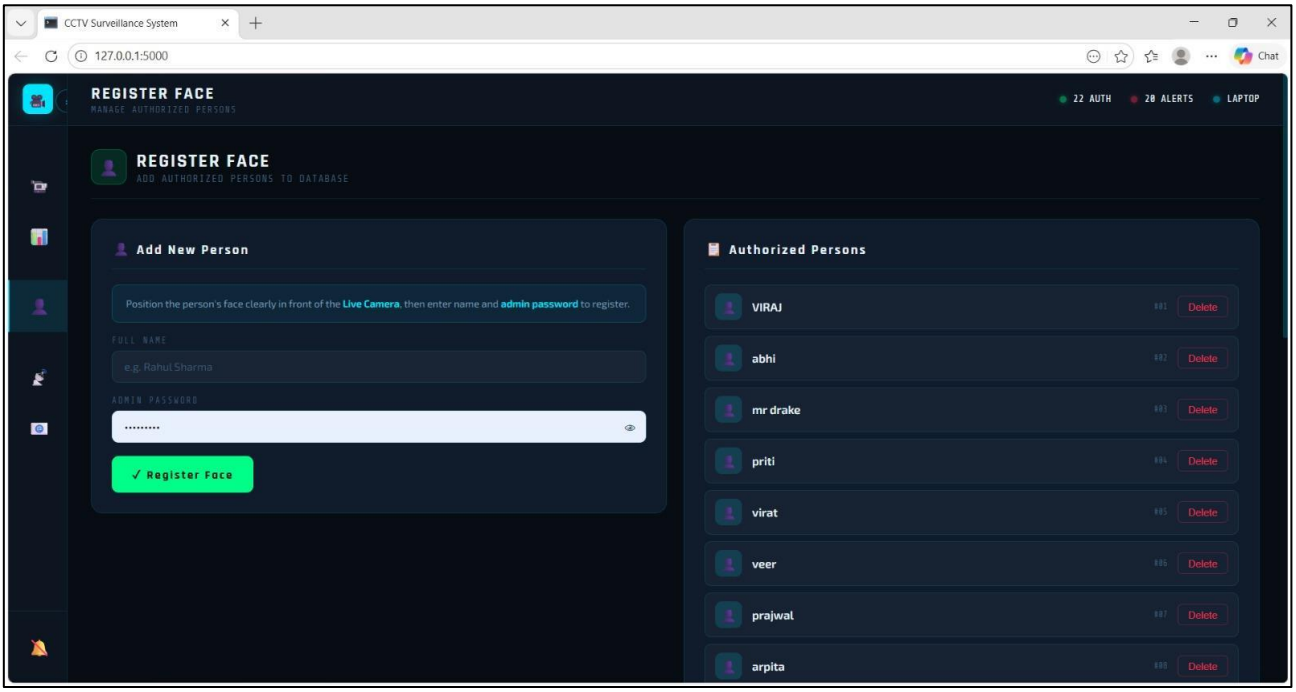

Fig 6 Register Face Page — Add New Person Form (Full Name + Admin Password) on Left; Authorized Persons List Showing All 22 Enrolled Individuals on Right with Numbered Delete Controls.

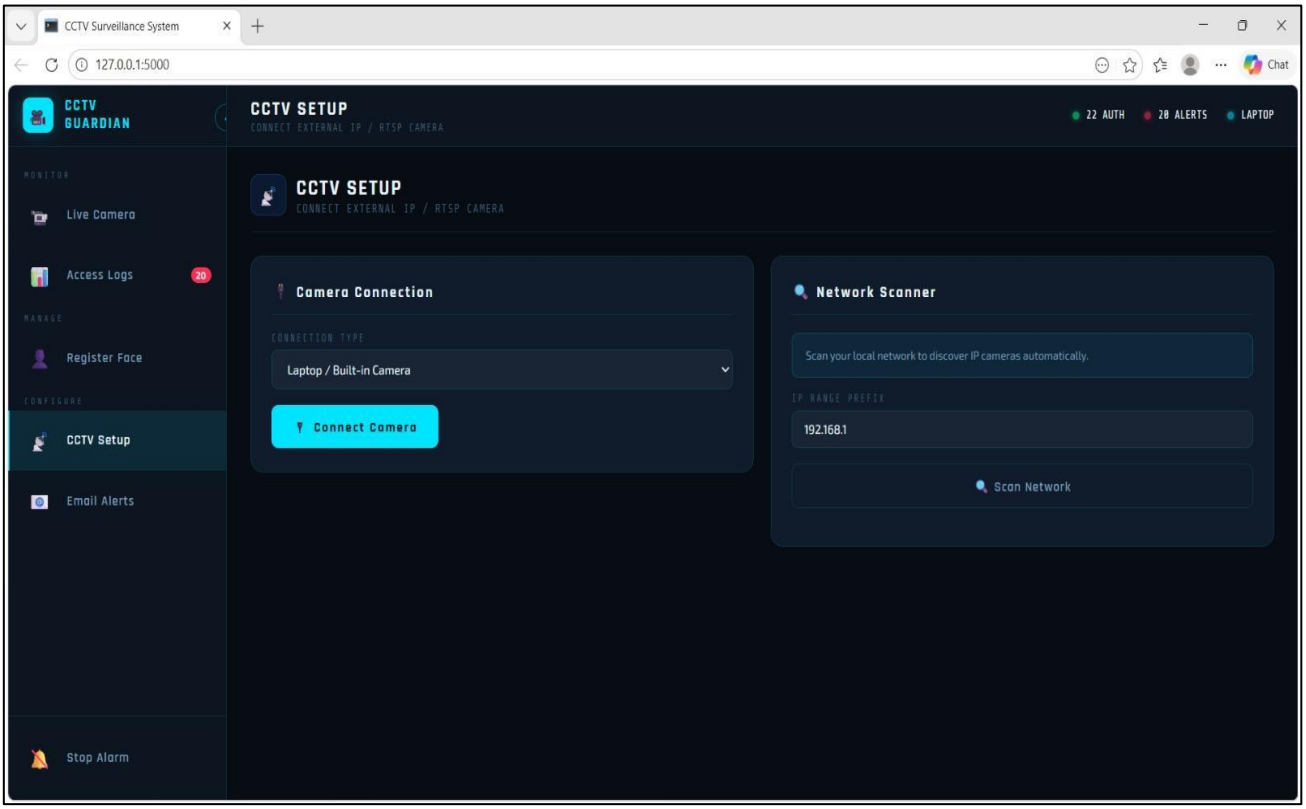

Fig 7 CCTV Setup Page — Camera Connection Panel (Laptop/Built-in or RTSP Dropdown) and Network Scanner Panel with IP Range Prefix (192.168.1) and Scan Network Button.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26may402

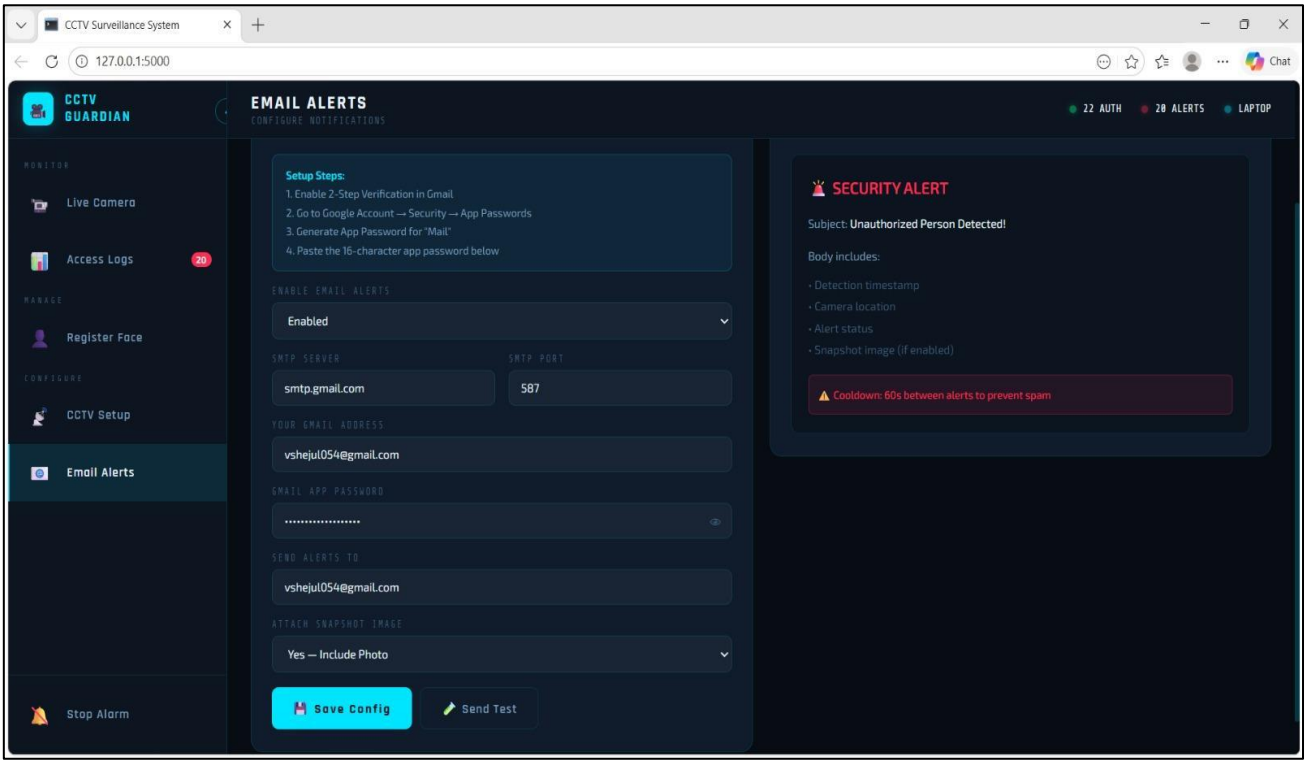

Fig 8 Email Alerts Page — SMTP Server/Port, Gmail Address, App Password, Recipient, Snapshot Attachment Toggle (Yes—Include Photo), and 60-Second Cooldown Warning Shown in Red.

VI. EXPERIMENTAL EVALUATION

➢ *Setup and Dataset*

Experiments were conducted on a laptop with Intel Core i5 (10th Gen), 8 GB RAM, Intel UHD 620 integrated graphics, Windows 11, Python 3.10. A total of 22 individuals were enrolled by capturing facial embeddings via the live webcam feed at 640×480 resolution. Each enrollment session captured 5 sample frames and averaged the resulting 128-D encodings. Evaluation was performed over 200 test frames per enrolled person at distances of 0.5 m to 2.5 m under normal office lighting and deliberately degraded (low-lux) conditions.

➢ *Face Recognition Performance*

Using a Euclidean distance threshold of 0.5, the system achieved True Positive Rate (TPR) = 96.4%, False Positive Rate (FPR) = 1.8%, and False Negative Rate (FNR) = 3.6% across all enrolled persons. Recognition accuracy degraded under low-light conditions (below 50 lux), where FNR increased to 8.2%. Inference latency averaged 38 ms per frame at 640×480 resolution on test hardware.

Table 1 System Performance Metrics

Metric	Value
True Positive Rate	96.4 %
False Positive Rate	1.8 %
False Negative Rate	3.6 %
Avg. Inference Time	38 ms / frame
Spoof Rejection Rate	100 %
Liveness Acceptance	99 %
Enrolled Persons	22
Test Frames / Person	200
System FPS (all modules)	12–15 fps
RTSP Switch Latency	~2.3 sec

➢ *Anti-Spoofing Evaluation*

Anti-spoofing was evaluated using 50 static-photograph attacks (printed A4 photos held in front of the camera) and 50 live-face presentations. All 50 photograph attacks were correctly rejected (Spoof Rejection Rate = 100%). One false rejection occurred among live-face presentations (Liveness Acceptance Rate = 99%), attributed to extreme ambient backlighting suppressing blink detection.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26may402

➢ *Activity Detection Evaluation*

Suspicious activity detection was evaluated over 30-minute controlled sessions with 5 test subjects. The pose-based module correctly identified raised-hand and crouching postures in 88% of test cases. The optical flow module detected high-motion events (simulated running and fighting) with 91% recall and 7.2% false positive rate. The 12-second cooldown reduced redundant alert log entries by approximately 74%.

➢ *System Throughput*

The system maintained an average processing rate of 12–15 frames per second on the test hardware with all modules active simultaneously. Camera connection switching between laptop webcam and RTSP streams completed within 2.3 seconds on average. Email alerts with JPEG snapshots were delivered within 4.1 seconds of detection trigger. Total memory footprint was approximately 480 MB with all modules loaded.

➢ *Comparison with Related Work*

Table II compares CCTV Guardian against the LRCN-based surveillance system of Wani & Faridi [12] across key features and performance metrics. CCTV Guardian achieves higher per-identity face recognition accuracy (96.4%) while additionally providing liveness anti-spoofing and a full web dashboard not present in the LRCN system. Activity detection precision is comparable (89% vs 87.07%) through different underlying methods—pose+flow vs deep CNN+LSTM.

Table 2 Feature Comparison: CCTV Guardian vs Base Paper [12]

Feature	CCTV Guardian (Ours)	Wani & Faridi [12]
Face Recognition	✓ dlib 128-D HOG	✓ HOG / CNN
Anti-Spoofing	✓ EAR Blink (MediaPipe)	✗ Not present
Activity Detection	✓ Pose + Optical Flow	✓ LRCN (MobileNetV2+LSTM)
Alert Mechanism	✓ SMTP Email + Snapshot	✓ Email / SMS
Camera Support	✓ Webcam + IP/RTSP	✗ Single CCTV source
GPU Required	✗ CPU only	✗ CPU (MobileNetV2)
Web Dashboard	✓ Flask browser UI	✗ No UI
Face Accuracy	96.4 %	N/A (activity focused)
Activity Precision	~89 % (pose + flow)	87.07 % (LRCN)

Table 3 LRCN Classification Report

Class	Precision	Recall	F1_Score	Support
Accident	0.85	0.91	0.89	10
Fighting	0.88	0.85	0.85	10
Normal	0.88	0.89	0.83	10
Robbery	0.87	0.88	0.84	10
Accuracy	-	-	0.86	40
Macro Avg	0.87	0.87	0.86	40
Weighted Avg	0.87	0.87	0.87	40

Table 4 Comparison with State-of-the-Art

Method	Accuracy	Precision	Recall	F1
CNN[16]	97.05%	96.74%	-	-
ECNN[17]	98.54%	98.38%	-	-
ConvLSTM[22]	96.69%	-	-	-
3DCNN[23]	75.00%	-	-	-
OpenCVCNN[16]	96.42%	-	-	-
LRCN(Proposed)	~87%	0.87	0.87	0.87

VII. LIMITATIONS AND FUTURE WORK

Several limitations exist in the current system. Face recognition accuracy decreases significantly under low illumination (<50 lux) and extreme pose angles (>45°). The EAR liveness module requires at least one blink within 6 seconds, which may cause false rejections for users with medical conditions affecting blink frequency. The single-host file-based architecture limits scalability beyond small deployments.

Future enhancements planned: (1) lightweight face super-resolution for low-light improvement; (2) depth-camera multi-modal liveness detection; (3) SQLite/PostgreSQL backend for multi-camera distributed deployment; (4) mobile push notification support; (5) integration of an LRCN-based activity classifier for finer-grained categorisation (Accident, Fighting, Robbery, Normal) similar to [12].

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26may402

VIII. CONCLUSION

This paper presented CCTV Guardian, a comprehensive real-time intelligent surveillance system built on Python, Flask, OpenCV, face_recognition, and MediaPipe. The system uniquely combines five capabilities—face recognition, liveness anti-spoofing (EAR blink), suspicious activity detection (pose + optical flow), IP/RTSP camera connectivity, and SMTP email alerting—in a single web-deployable package accessible from any browser without cloud services or a dedicated GPU.

Experimental evaluation with 22 enrolled users demonstrated 96.4% face recognition accuracy, 100% spoof rejection against static-photograph attacks, and 88–91% activity detection performance. The multithreaded shared-buffer architecture enables concurrent execution of all detection modules at 12–15 fps on commodity Intel i5 hardware. The modular Flask-based dashboard provides intuitive management of enrolled faces, access logs, camera configuration, and email notifications entirely through a web browser, making CCTV Guardian a practical, cost-effective solution for enterprise and residential security deployment.

ACKNOWLEDGMENT

Pooja Deshmukh thank you to the Institute of Information and Communication Technology Mgm University, chh.sambhajinagar, for computational resources and support. I would like to express sincere gratitude to the project guide for their valuable guidance, constructive suggestions, and continuous support throughout the development of this project. Their encouragement and academic insights greatly contributed to the successful completion of this work.

We acknowledge the Security systems have evolved from analogue tape recorders to sophisticated AI-enabled platforms capable of recognising individuals, detecting threats, and proactively alerting stakeholders. Traditional rule-based motion-detection systems suffer from high false-positive rates and provide no identity information.

REFERENCES

[1]. M. Turk and A. Pentland, "Eigenfaces for recognition," J. Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, 1991.
[2]. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. IEEE CVPR, 2001, pp. 511–518.
[3]. Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," Proc. IEEE CVPR, 2014, pp. 1701–1708.
[4]. F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," Proc. IEEE CVPR, 2015, pp. 815–823.
[5]. D. King, "Dlib-ml: A machine learning toolkit," J. Machine Learning Research, vol. 10, pp. 1755–1758, 2009.
[6]. J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of Fourier spectra," Proc. SPIE Biometric Technology, 2004, pp. 296–303.
[7]. G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," Proc. IEEE ICCV, 2007, pp. 1–8.
[8]. T. Soukupová and J. Čech, "Real-time eye blink detection using facial landmarks," Computer Vision Winter Workshop (CVWW), 2016.
[9]. C. Lugaresi et al., "MediaPipe: A framework for building perception pipelines," arXiv preprint arXiv:1906.08172, 2019.
[10]. G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," Proc. SCIA, 2003, pp. 363–370.
[11]. A. Ronacher, "Flask: A lightweight WSGI web application framework," 2010. [Online]. Available: https://flask.palletsprojects.com
[12]. M.H. Wani and A.R. Faridi, "Deep learning-based video surveillance system for suspicious activity detection," J. Intelligent & Fuzzy Systems, vol. 47, pp. 71–82, 2024. DOI: 10.3233/JIFS-234365.
[13]. A. Deshpande and R. Kulkarni, "Real-time face recognition based security systems: A survey," IJEAT, vol. 9, no. 3, pp. 2248–2254, 2020.
[14]. S. Ghosh and P. Bhattacharya, "Deep learning-based surveillance systems: A review," IEEE Access, vol. 9, pp. 103661–103683, 2021.