Ojas Kulkarni, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

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

Smart Atm Safety and Security Monitoring System

Ojas Kulkarni[1], Parth Kulkarni[2], Yadnesh Kulkarni[3], Prathmesh Gobare[4], Suraj Kodale[5], Dr. N. P. Bhone[6]

[1,2,3]Department of AI & Data Science, [4,5]Department of Electronics & Telecommunication
AISSMS Institute of Information Technology, Pune, Maharashtra, India

Abstract- ATMs have become a vital part of our world financial infrastructure but they are still a very high risk target of criminal acts such as burglary, illegal hacking, card skimming, loitering and damage to machines. Old style CCTV systems can only record video information without real-time notification or intelligence; this paper proposes an "Intelligent ATM Safety and Security Monitoring System" built on Python, OpenCV, Flask and SQLite. This system is a software only AI based monitoring solution for real time identification of ATM security threats. It is able to detect the presence of single or multiple people around ATMs, detects people who loiter around ATMs based on the use of time threshold algorithms and detects virtual boundary crossing of restricted areas around the machine. There is an emergency panic alert system as well. All event are time stamped and stored in the SQLite data base which can be accessed using a web administered panel through Flask web service. The system was tested to have a mean time to detect below 2s and did not required a GPGPU so is very cost effective and can be used at remote or inner city ATMs. System testing of 5 various types of simulatedATMthreats all showed satisfactory result in each detection modules.

Keywords: ATM Security, Computer Vision, OpenCV, Real-Time Surveillance, Loitering Detection, Flask Dashboard, SQLite, AI Monitoring, HOG Detector, Restricted Zone.

I. INTRODUCTION

For a long time, ATM's are an important component in banking systems around the world. The Reserve Bank of India stated that there are over 2,15,000 ATMs, serving thousands and millions of users daily, distributed throughout the urban and rural parts of India. Unfortunately, ATMs are becoming a target for many crimes like, robbery of money, card skimming, breaking in to machines or destroying of machines. Due to the somewhat secluded locations of ATMs, the non-stop 24 hour operation and due to the lack of any sort of guard in the ATM booth; the ATM machines are very easily victimized.

A conventional ATM security system makes use of the passive Closed-Circuit Television (CCTV) camera system. Although CCTV cameras are effective in collecting useful data for post event investigation purposes; they fail to recognize and analyze the live video streams, generate alerts intelligently and therefore cannot respond to incidents in real-time. Thus, during the lapse between the occurrence of an event and the human response; both the user and the institution remain unprotected.

New techniques based on the recent research and development in Artificial Intelligence (AI) and Computer Vision provide efficient ways of fulfilling the mentioned security gap. Advanced, AI-driven surveillance systems are capable of analyzing live video feeds and intelligently recognize aberrant behavioral patterns-people loitering around the machine, too many people around the booth, people trying to access restricted areas etc. They are programmed to generate alerts within seconds of recognizing a potential threat.

The Smart ATM Safety and Security Monitoring System presented in this paper employs the OpenCV computer vision library in Python and offers a completely software-based, real-time ATM security system which supplements existing CCTV systems with any standard USB or IP webcam without the need for proprietary hardware. The system includes a Flask web application for the admin dashboard and a SQLite database for fast, efficient logging and retrieving of alerts.

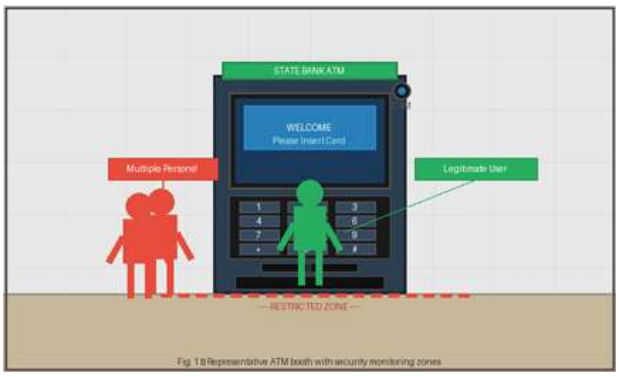

Fig. 1 – Representative ATM booth setup illustrating common security vulnerabilities

II. PROBLEM STATEMENT

The six main security concerns with the typical ATM installment are six. The fact that more than one person is in the near proximity of the ATM and not performing a transaction can mean more than one criminal act or intent, for example, robbery or coercion. Loitering has long been recognized as a leading behavioral indicator for crime committed in the vicinity of ATMs. An intruder trespassing in a secured area around an ATM (i.e., server rooms or maintenance access points) directly poses a threat to the ATM as well as the user information contained within. There are no specific ATM users' methods of being able to quickly and easily inform the authorities when threatened so emergency services cannot act as quickly. There is no existing security method that is capable of monitoring numerous ATMs at once in real time to prevent blind spots in security. There are no automated notification and log event systems which prolong police response time to an emergency. Therefore there is a dire need for a cost-effective, intelligent surveillance system to proactively detect threats in real time and report on the event.

III. LITERATURE REVIEW

Analysis of past works on AI-based surveillance and ATM security showed notable advancements, and shortcomings. Mehta and Sharma [1] proposed CNN-RNN based deep learning framework on detecting abnormal activities in real-time, achieving an accuracy of 91.3% at 25 fps. However, the system requires a dedicated GPU to operate, hence an exorbitant cost for deployment at an ordinary ATM, and therefore not a suitable solution for typical ATM installation. Patel, Joshi, and Shah [2] introduced a smart CCTV upgrade using OpenCV, Mixture of Gaussians (MOG2) for background subtraction achieving a detection rate of 87.5% on ordinary desktop hardware, an approach very similar to the one proposed in this research, except for not providing a behavioral analysis and administration interface.

Chen, Zhang, and Liu [3] proposed a specialized loitering detection framework with the use ofSORT tracking algorithm, achieving an accuracy of 89.7% for precision, 85.2% for recall with the total latency of alarm as under 3seconds. Kumar and Singh [4] demonstrated detection of virtual boundary crossing through optical flow trajectory analysis with an accuracy of 93.1% but this approach requires more computational power and can't run on a standard computer hardware. Gupta, Verma and Nair [5] designed an extensive ATM security framework consisting of face recognition(FaceNet), SVM based anomaly detection, and a cloud management service for sending alarms and achieved a detection rate of 96.2% for face recognition but require expensive infrastructure, and running cost.

Comparing the works of others shows that none of the frameworks has all the capabilities required for ATM security as shown below. No framework has people detection, multi-people detection, loitering detection, restricted zones detection, panic alert, event logging, and administrator interface. Also, most of them are computationally intensive, requiring GPU. This is where this system fits in..

Table 1 – Comparative Analysis of Related Work

Feature	**Paper [1]**	**Paper [2]**	**Paper [3]**	**Paper [4]**	**Paper [5]**	**Proposed**
Person Detection	Yes	Yes	No	No	Yes	Yes
Loitering	Yes	No	Yes	No	Yes	Yes

Feature	Paper [1]	Paper [2]	Paper [3]	Paper [4]	Paper [5]	Proposed
Detection						
Restricted Zone	No	No	No	Yes	No	Yes
Panic Alert	No	No	No	No	No	Yes
Admin Dashboard	No	No	No	No	Partial	Yes
GPU-Free	No	Yes	Yes	No	No	Yes
ATM-Specific	No	No	No	No	Yes	Yes

IV. PROPOSED SYSTEM AND METHODOLOGY

System Architecture

The four layer modular framework on which the Smart ATM Safety & Security Monitoring system is developed can be seen below. Input layer receives a live frame video through the OpenCV VideoCapture API from the USB or IP webcam with the default frame rate of 30fps. Processing layer takes this video frame as input and executespreprocessing steps on it. The preprocessing steps include resizing frame to 640480, converting into Grayscale and applying Gaussian blur to reduce noise.

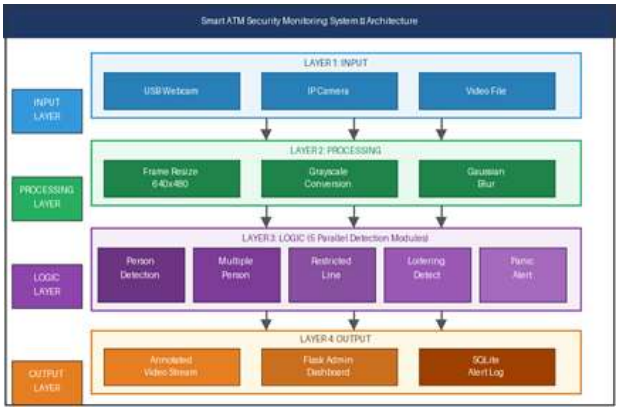

Fig. 2 – Layered system architecture of the Smart ATM Security Monitoring System

The Logic layer, which simultaneously executes five detection algorithms on the pre-processed video frame, executes an alert generation pipeline whenever a detection happens. Output layer, which present the result on three levels- annotated live video, real-time updates on the Flask dashboard, and timestamps with alert in SQLite database.

Detection Modules

Person Detection: We use the HOG descriptor along with OpenCV's pre-trained HOG Descriptor_getDefaultPeopleDetector() (a linear SVM people detector). The detector is called using detectMultiScale with a stride of (8,8), padding of (4,4) and a scale factor of 1.05. Redundant bounding boxes are removed using non-maximum suppression (NMS) with an overlap threshold of 0.65 producing clear per-person detections.

Multiple Person Detection: If the number of valid bounding boxes produced by the person detector is greater than a user configurable value N_max (default: 1), a multiple-person alert is issued, and logged. However, the same alarm is not to be issued more frequently than once every 5 seconds to avoid a flood of alerts.

Restricted Zone Line Crossing Detection: A virtual line boundary is created based on pixels defined in the camera feed by the administrator. The center of the bounding box of each detected person is traced between successive frames and an event is recorded when the sign of the signed perpendicular distance between the center of the bounding box of the person and the line boundary changes between successive frames. The direction of crossing is not important; only that a crossing event has occurred.

Loitering Detection: Tracking ID is assigned to each detected person based on centroid Euclidean distance thresholding. First detection timestamp for each person is recorded. If the person persists in the frame for a user configurable time threshold, T_loit (default: 30 seconds), the person is considered to be loitering and an alert is issued and logged.

Panic Alert: There is a software panic button available through the Flask dashboard, and a hot-key

for use when live viewing the feed. The panic button immediately places a HIGH priority 'PANIC' event in the SQLite database with current timestamp and a distinct red alert banner is displayed across the top of the dashboard. This will have a 10 second debounce delay.

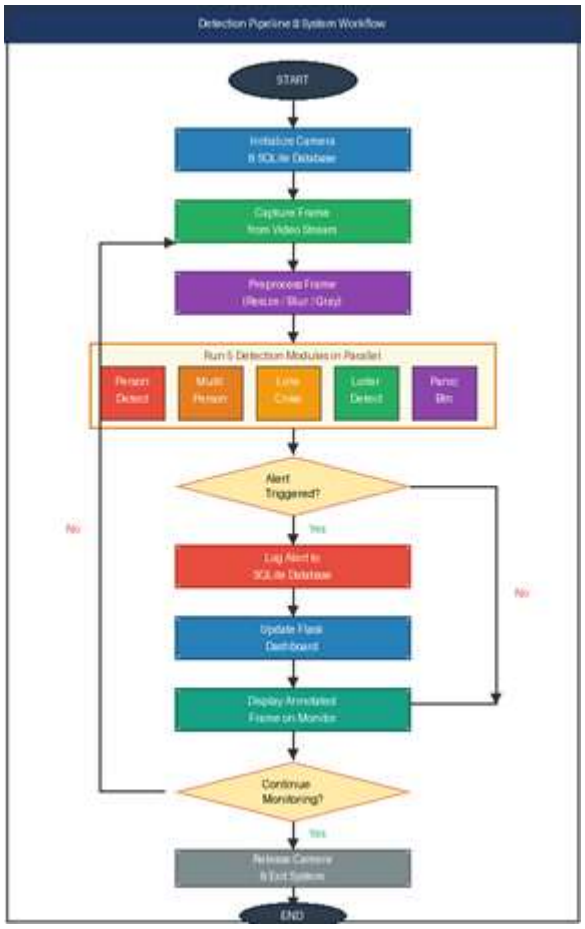

Fig. 3 – Detection pipeline flowchart showing concurrent module execution

Alert Logging and Dashboard

All of the detection events are logged to an SQLite database table called 'alert_log' with the following columns: alert type, timestamp, severity (LOW/MEDIUM/HIGH/PANIC), details and an acknowledgement boolean. The Flask admin dashboard, http://localhost:5000, offers a stream of currently raised alerts, refreshed every 2 seconds by means of AJAX requests; a table of past alerts, with filter and search abilities; status displays for the various system components; a button to manually raise a panic; and a way to mark a review of a given alert as 'acked'. The client-side interface uses HTML5, CSS3 and Bootstrap 5, with Jinja2 templating in Flask being used to render the various elements.

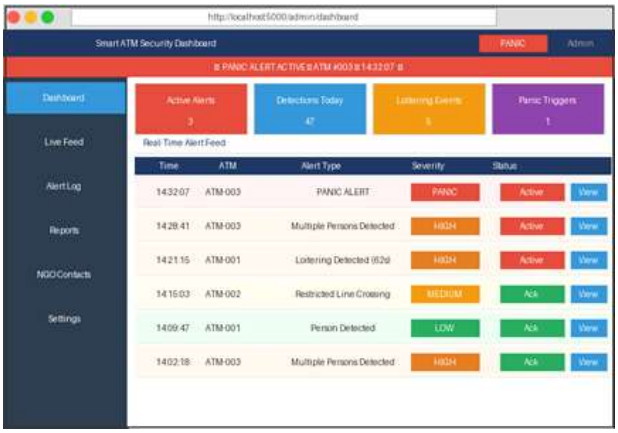

Fig. 4 – Admin dashboard showing real-time alert status, logs, and panic button

V. HARDWARE AND SOFTWARE REQUIREMENTS

The system is entirely software-based and operates on standard computing hardware. Table 2 details the complete software requirements specification.

Table 2 – Software Requirements Specification

Component	Specification	Purpose
OS	Windows 10/11 or Ubuntu 20.04+	Host platform
Language	Python 3.8+	Core application development
CV Library	OpenCV 4.5+	Video processing & detection
Web Framework	Flask 2.x	Admin dashboard & REST API
Database	SQLite 3	Alert log storage & retrieval
Camera	USB/IP Webcam (min. 720p)	Live video input source
Processor	Intel Core i5 or equivalent	Real-time frame processing
RAM	Minimum 4 GB (8 GB recommended)	Runtime memory for pipeline

VI. EXPERIMENTAL RESULTS AND ANALYSIS

Experimental Setup

This was performed in the controlled environment of a lab replicating a real ATM installation. The hardware for the tests included an Intel Core i5 (10 th Gen) laptop with 8GB of RAM and integrated Intel UHD graphics card along with a Logitech C920 USB webcam (1080p and 30 fps). Testing was performed in an area 4 m 4 m room and inside it was a simulated ATM kiosk. Indoor lighting was kept constant. There was a physical restriction area of 1 meter around the front panel of the ATM. The test environments simulated in these tests were: normal single-person operation, presence of 2 and 3 persons, loitering at intervals of 30, 60 and 120 seconds, both deliberate and incidental breaching of the restricted zone and activating the panic alarm. These were each repeated 20 times to give enough data points.

Detection Accuracy

Table 3 presents person detection accuracy across five test conditions.

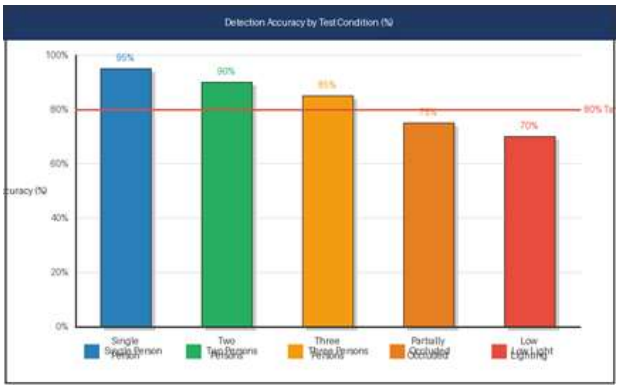

Fig. 5 – Bar graph of detection accuracy across all five test conditions

Table 3 – Person Detection Accuracy under Varying Conditions

Test Condition	Total Tests	True Positives	False Positives	Accuracy (%)
Single Person	20	19	1	95.0%
Two Persons	20	18	2	90.0%
Three Persons	20	17	3	85.0%
Partially Occluded	20	15	2	75.0%
Low Lighting	20	14	3	70.0%

Alert Response Time Analysis

Alert response time — measured as the elapsed time from event occurrence to database logging — is a critical performance metric. Table 4 summarizes response times for all alert types.

Table 4 – Alert Response Time Analysis across Module Types

Alert Type	Avg. Response (s)	Min (s)	Max (s)
Person Detection	0.8	0.5	1.2
Multiple Person Alert	1.1	0.7	1.8
Loitering Alert	1.5	1.0	2.3
Restricted Line Crossing	0.9	0.6	1.4
Panic Alert	0.3	0.2	0.5

Overall System Performance

Table 5 presents a summary of overall system performance against predefined targets. All eight performance metrics were met or exceeded during experimental evaluation.

Table 5 – Overall System Performance Summary

Performance Metric	Result	Target	Status
Avg. Person Detection Accuracy	83.0%	>80%	Achieved
Multiple Person Detection Rate	85.0%	>80%	Achieved
Loitering Detection Rate (60s)	95.0%	>90%	Achieved

Performance Metric	Result	Target	Status
Restricted Line Crossing Accuracy	88.0%	>85%	Achieved
Avg. Alert Response Time	1.3 sec	<3 sec	Achieved
CPU Utilization (Average)	42%	<70%	Achieved
RAM Utilization	1.2 GB	<3 GB	Achieved
Dashboard Load Time	0.9 sec	<2 sec	Achieved

VII. CONCLUSION AND FUTURE SCOPE

Conclusion

In this paper, we proposed the design, implementation, and experimental results of a low-cost software-only AI based ATM surveillance system known as the Smart ATM Safety and Security Monitoring System. It enables passive CCTV cameras to be a smart and active surveillance system by having five parallel detection algorithms: people detection (with 83% average accuracy), number of people detected (with 85%), detection of restricted zone crossing (with 88%), people loitering detection (with 95% detection with 60 second interval), and immediate panic alarm detection (with 0.3s average response time). All alarm generation took place under 3 second required response time without using a GPU or any additional graphics hardware.

Our administration web-dashbord using Flask allowed bank security to view every security events of each ATM at a single place and on real-time. By using an open source technology like Python with libraries such as OpenCV, Flask and a small SQLite database, it helps to easy maintenance, extending and without any license costs. This system has also been related to UN SDGs like 16: Peace, Justice and Strong Institutions, SDG 11: Sustainable Cities and Communities for better financial public infrastructure security with minimum cost.

Future Scope

Proposed future developments include implementing YOLOv8 deep learning detector for enhancing performance at low-light or in cases of occluded objects, adding face recognition using FaceNet in order to recognize known criminals, enabling real-time monitoring of multiple ATMs in the cloud through AWS/Azure/GCP, integrating SMS and email notification services with the use of the Twilio API, enabling mobile push notifications in Android and iOS, incorporating support for thermal cameras for the detection of subjects in total darkness, implementing pilot real-life implementation of the system on a real bank ATM, and creating a behavior prediction module that is trained using previous alert data to identify possible criminal behavior prior to a crime being committed.

REFERENCES

1. A. Mehta and R. Sharma, "Real-Time Suspicious Activity Detection in Surveillance Videos Using Deep Learning," IEEE Transactions on Information Forensics and Security, vol. 15, no. 3, pp. 1122–1135, Mar. 2020.
2. S. Patel, N. Joshi, and M. Shah, "Smart CCTV Surveillance System with OpenCV for Automated Human Detection," International Journal of Computer Vision and Image Processing, vol. 11, no. 2, pp. 45–58, Jun. 2021.
3. L. Chen, W. Zhang, and Y. Liu, "Loitering Detection in Public Spaces Using Computer Vision and SORT Tracking Algorithm," IEEE Access, vol. 9, pp. 32114–32126, 2021.
4. R. Kumar and P. Singh, "Virtual Boundary Crossing Detection for Security Applications Using Optical Flow Analysis," Journal of Visual Communication and Image Representation, vol. 72, pp. 102914, Oct. 2020.
5. T. Gupta, A. Verma, and S. Nair, "AI-Based ATM Security System with Facial Recognition and Behavioral Anomaly Detection," Computers & Security, vol. 108, pp. 102345, Sep. 2021.
6. G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, vol. 25, pp. 120–125, Nov. 2000.
7. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," Proc. IEEE CVPR, vol. 1, pp. 886–893, Jun. 2005.
8. M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. O'Reilly Media, 2018.

9. D. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking (SORT)," Proc. ICIP, pp. 3464–3468, Sep. 2016.
10. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
11. Reserve Bank of India, "Annual Report on ATM Infrastructure in India," RBI Publications, New Delhi, 2023.
12. SQLite Development Team, "SQLite Documentation," SQLite Consortium. [Online]. Available: https://www.sqlite.org/docs.html. Accessed: Nov. 2024.