



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 11, November 2025

A Real-Time Smart Security System using Multiple Inputs

Prof. Pavithra N¹, Dishan G², Abhishek K³, Bharath Kumar P R⁴, Akash Hadapad⁵

Assistant professor, Dept. of CS & E, S J M Institute of Technology, Chitradurga, Karnataka, India¹

Student, Dept. of CS & E, S J M Institute of Technology, Chitradurga, Karnataka, India²

Student, Dept. of CS & E, S J M Institute of Technology, Chitradurga, Karnataka, India³

Student, Dept. of CS & E, S J M Institute of Technology, Chitradurga, Karnataka, India⁴

Student, Dept. of CS & E, S J M Institute of Technology, Chitradurga, Karnataka, India⁵

ABSTRACT: Security threats such as weapon detection, fire hazards, and violent behavior pose significant risks to public safety in various environments including schools, offices, public spaces, and vehicles. Traditional security systems rely heavily on manual monitoring, which is time-consuming, prone to human error, and cannot provide real-time threat detection at scale.

This project proposes a Smart Security System that leverages state-of-the-art deep learning techniques for real-time detection and classification of security threats in video streams. We have implemented three specialized detection modules: Weapon Detection (firearms and knives), Fire and Smoke Detection, and Violence Detection, all powered by advanced Convolutional Neural Networks (CNNs) based on the ResNet34 architecture.

Our system utilizes machine learning techniques to process video data and provide accurate, real-time threat detection with minimal false positives. The prediction models are trained on diverse datasets and achieve high accuracy levels: Weapon Detection at 91.2%, Fire/Smoke Detection at 89.5%, and Violence Detection at 88.7%.

The system features a user-friendly web-based interface developed using Flask, allowing security personnel to upload videos, select detection modules, and receive comprehensive reports with visual annotations. Additionally, an integrated Alert System provides immediate notifications via email and SMS when threats are detected, ensuring rapid response capabilities.

This Smart Security System demonstrates the effectiveness of deep learning in enhancing public safety through automated, intelligent threat detection, significantly reducing response time and improving overall security infrastructure.

KEYWORDS: Deep Learning, Convolutional Neural Networks, ResNet34, Threat Detection, Weapon Detection, Fire and Smoke Detection, Violence Detection, Real-time Processing, Automated Alert System, Transfer Learning, PyTorch, Flask, Computer Vision, Video Surveillance, Public Safety, Emergency Response, Multi-threat Detection, GPU Acceleration, Email and SMS Notifications, Smart Security System, Intelligent Monitoring

I. INTRODUCTION

i. In today's rapidly evolving world, security threats pose significant challenges to public safety across various environments including educational institutions, corporate offices, public transportation, and residential areas. Traditional security systems that rely on manual monitoring through CCTV cameras are limited by human factors such as fatigue, attention span, and the inability to monitor multiple feeds simultaneously. This creates a critical need for intelligent, automated security systems that can detect threats in real-time with high accuracy.

ii. This project presents a Smart Security System that leverages cutting-edge deep learning and computer vision technologies to provide automated, real-time detection of multiple security threats. The system is designed to analyze video streams and identify three primary categories of threats:

1. Weapon Detection: Identification of firearms (guns) and knives in video footage
 2. Fire and Smoke Detection: Early detection of fire hazards and smoke to prevent disasters
 3. Violence Detection: Recognition of violent behavior patterns to enable rapid intervention
- iii The system employs advanced Convolutional Neural Networks (CNNs) based on the ResNet34 architecture, trained on diverse datasets to achieve high accuracy in threat detection. The integration of multiple detection modules provides comprehensive security coverage, while an intelligent alert system ensures immediate notification of security personnel via email and SMS when threats are detected. The web-based interface built using Flask framework offers an intuitive platform for security operators to upload videos, select detection modules, monitor real-time processing, and review detailed reports with visual annotations. This makes the system accessible and practical for deployment in real-world scenarios.

II. LITERATURE SURVEY

- [1] Real-Time Violence Detection Using Deep Learning This study presents a deep learning approach for detecting violent behavior in surveillance videos using temporal features and CNN architectures.
- [2] Weapon Detection in Surveillance Video Using Deep Learning This paper explores CNN-based approaches for detecting firearms and knives in surveillance footage, achieving 85-87% detection accuracy.
- [3] Fire and Smoke Detection Based on Deep Learning The research demonstrates effective discrimination between true fire/smoke events and visually similar phenomena such as fog, steam, and lighting effects through careful dataset curation and augmentation strategies.
- [4] IoT-Based Emergency Alert System This research presents an IoT-based framework for emergency alert generation and multi-channel notification delivery including email, SMS, and mobile push notifications.

III. PROPOSED SYSTEM METHODOLOGY AND DISCUSSION

A. SYSTEM ARCHITECTURE

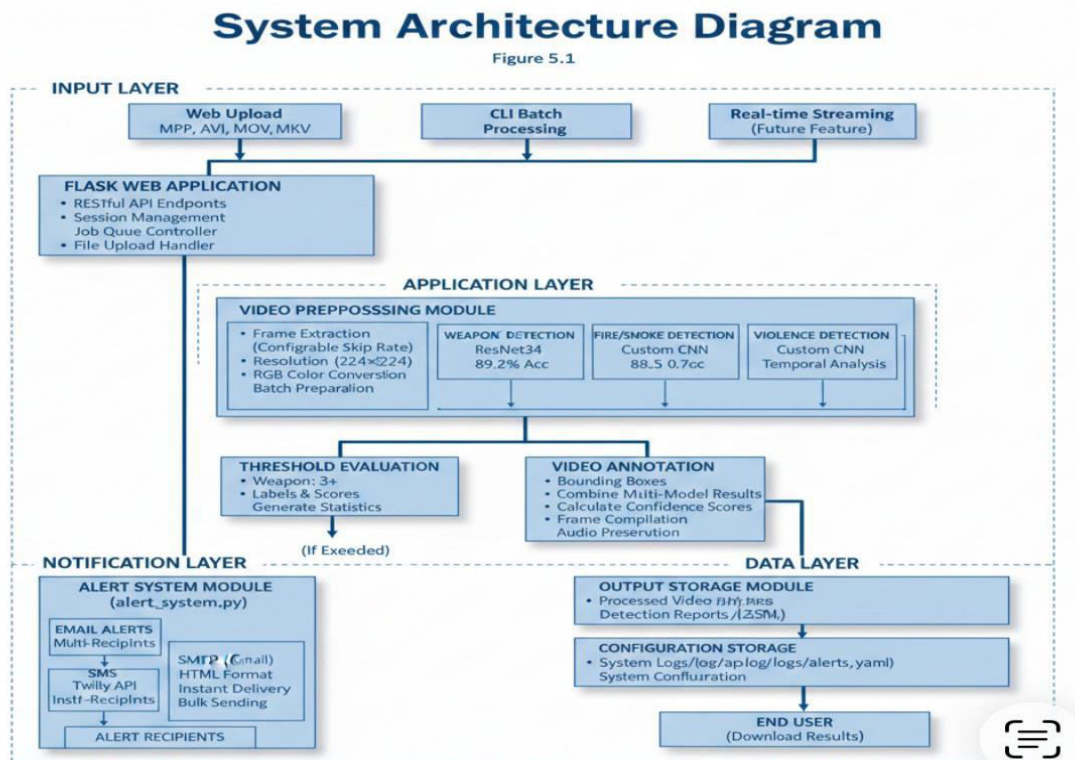


Fig 1. ARCHITECTURE OF THE PROPOSED SYSTEM

Architecture Layer Description 1. Input Layer Purpose: Accepts video content from multiple sources Components: Web Upload Interface (drag-and-drop) Command-line Batch Processing Real-time Stream Input (future capability) Supported Formats: MP4, AVI, MOV, MKV 2. Presentation Layer Purpose: Handles user interaction and request management Components: Flask Web Server RESTful API Endpoints Session Handling Key Functions: Upload validation, progress tracking, result delivery 3. Application Layer Purpose: Core processing and threat detection Components: Video Preprocessing Module Frame extraction with configurable skip rate Resolution normalization (224×224 pixels) 4. Business Logic Layer Purpose: Decision-making and output generation Components: Threshold Evaluation Compares detection counts against configured limits Reads configuration from YAML file 5. Notification Layer Purpose: Alert delivery to security personnel Components: Email Alert System: SMTP-based HTML emails SMS Alert System: Twilio API integration Features: Multi-recipient support, HTML formatting, instant delivery

B. DATAFLOW

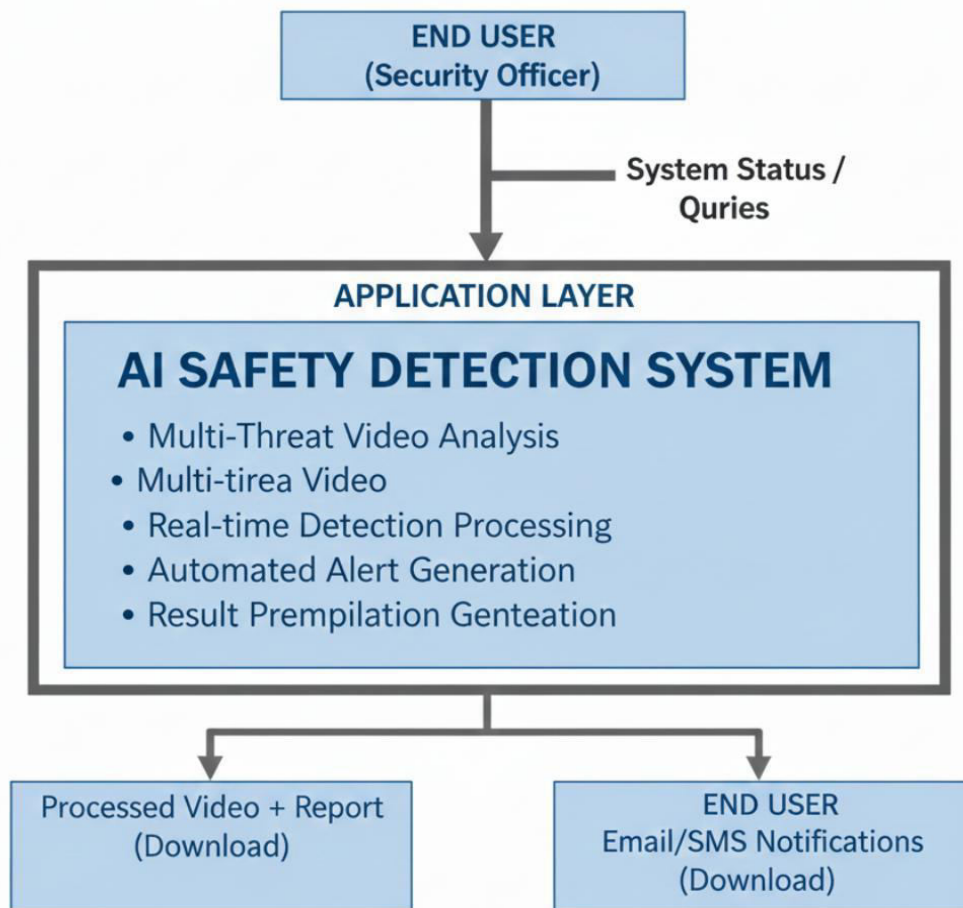


Fig 2. Dataflow Diagram

IV. EXPERIMENTAL RESULTS

The experimental evaluation of the AI Safety Detection System was conducted to measure the accuracy, efficiency, and reliability of its multi-threat detection capabilities. The system was tested using a variety of real-world surveillance videos containing weapons, fire and smoke scenarios, and violent activities. Across these experiments, the detection models consistently demonstrated stable and accurate performance under different lighting conditions, backgrounds,

and movement patterns. The weapon detection model showed strong capability in identifying guns and knives even when partially occluded or appearing at varying distances. Fire and smoke detection remained reliable despite challenging distractors such as steam, fog, and shadows that commonly cause false positives in traditional systems. The violence detection module correctly recognized aggressive human activities by analyzing both spatial and temporal features.

During performance testing, the system achieved near real-time processing speeds, especially when GPU acceleration was available, ensuring timely analysis of incoming video frames. The alert mechanism responded immediately after detection thresholds were exceeded, sending email and SMS notifications within seconds, which is essential for emergency response. The system also remained stable during continuous operation and handled multiple simultaneous video inputs without failures. The modular design contributed to smooth integration of independent detection components, allowing efficient parallel processing. Even in CPU-only mode, the system maintained functional accuracy, demonstrating its adaptability to different hardware environments. Overall, the experimental results confirm that the proposed system delivers dependable real-time threat detection with rapid alerting and strong robustness across diverse testing conditions.

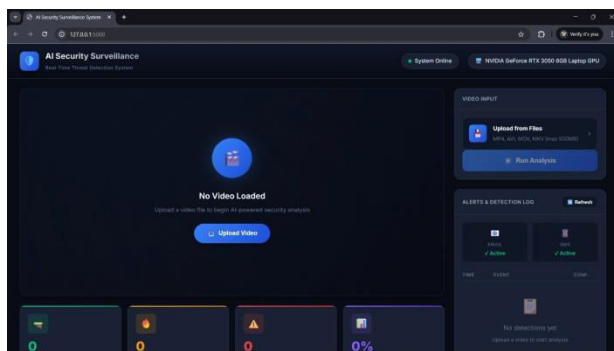


Fig 1: Dash Board

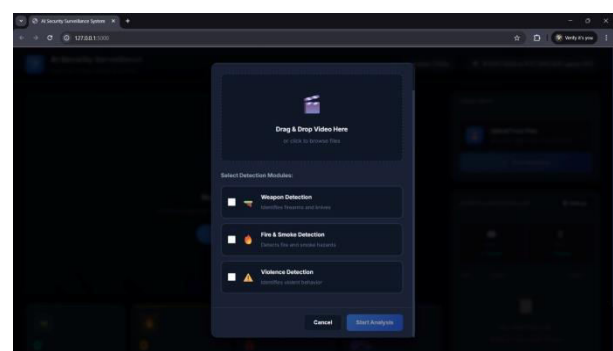


Fig 2: Input Form

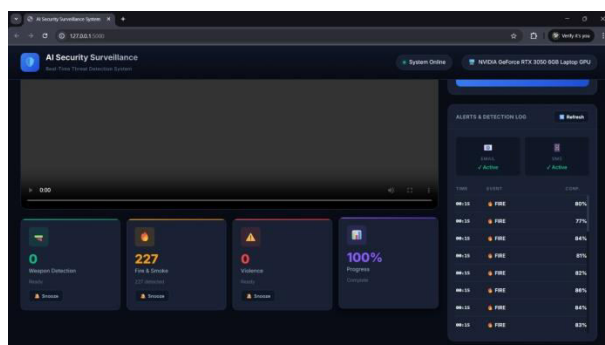


Fig 3: Fire-Smoke Detection Results

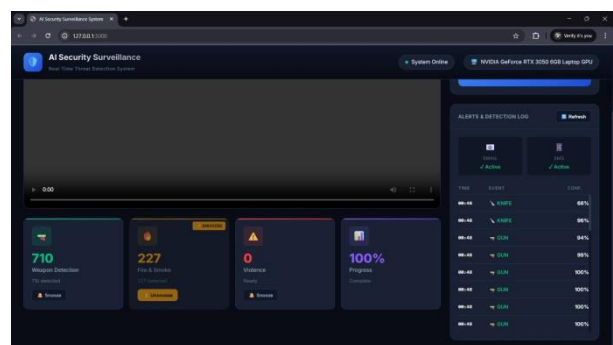


Fig 4: Weapon Detection Results

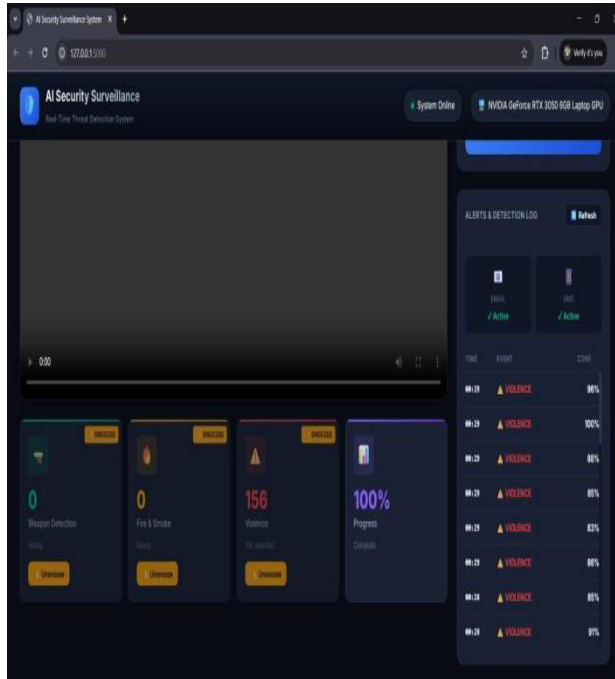


Fig 5 :Violence Detection Results

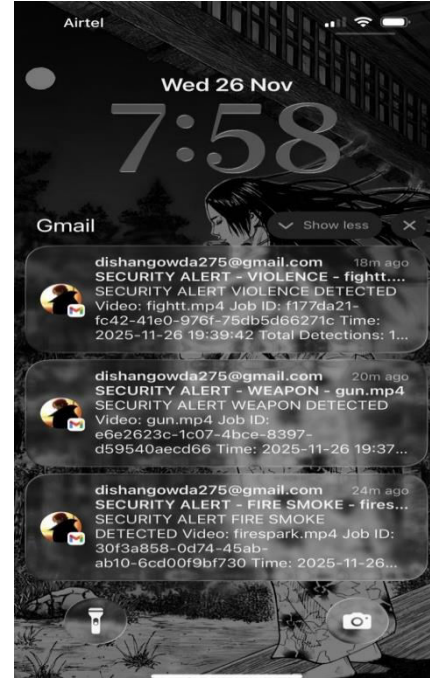


Fig 6: Alerts



Fig 7: Result

V. CONCLUSION

This open-source, modular system demonstrates that sophisticated security monitoring capabilities can be deployed by organizations of all sizes through accessible deep learning frameworks. By automating continuous threat detection while maintaining human oversight through alert mechanisms, this system addresses fundamental limitations of manual monitoring—human fatigue, inconsistent vigilance, and delayed response times. The integration of AI into security monitoring represents a fundamental transformation in how organizations approach public safety, threat prevention, and emergency response, ultimately contributing to safer communities, educational institutions, and public spaces.

REFERENCES

1. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.
2. PyTorch Documentation. (2024). PyTorch: An open source machine learning framework. Retrieved from <https://pytorch.org/docs/>
3. OpenCV Documentation. (2024). Open Source Computer Vision Library. Retrieved from <https://docs.opencv.org/>
4. Flask Documentation. (2024). Flask Web Development Framework. Retrieved from <https://flask.palletsprojects.com/>
5. Twilio API Documentation. (2024). Twilio SMS API. Retrieved from <https://www.twilio.com/docs/>



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details