International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 15, Issue 02, February 2026)

Violence Detection Using Machine Learning

Sanket Bajrang Kamble[1], Gaurav Gurunath Nevage[2], Rushikesh Raghunath Kamble[3], Ritesh Ashok Patil[4]

CO Student, Department of Computer Engineering, K.P. Patil Institute of Technology (Polytechnic), Mudal, Maharashtra, India.
H.O.D. Mrs. T.V. Patil Department of Computer Engineering, K.P. Patil Institute of Technology (Polytechnic) Mudal, Maharashtra, India.

***Abstract*--In recent years, ensuring public safety has become a critical concern, especially in areas under constant surveillance such as streets, schools, stations, and other public places. Manual monitoring of CCTV footage is both time-consuming and prone to human error. To address this issue, we have developed a Real-Time Violence Detection System using IP Camera that leverages machine learning and computer vision technologies to automatically detect violent activities.**

Our system captures live video streams from IP cameras and processes them using a pre- trained VideoMAE (Video Masked Autoencoder) model, integrated with Flask and OpenCV on the backend. The frontend, built with React, allows users to register, log in, and manage multiple camera feeds. The system also provides an option to upload pre-recorded videos for offline analysis. Alerts are generated instantly when violence is detected, and notifications are sent to the user interface in real time. All user data, camera information, and alerts are securely stored and managed using Supabase.

This project demonstrates an efficient and scalable solution for automated surveillance with real-time violence detection capabilities. By reducing the dependency on human surveillance, the system enhances security and can be a valuable tool in modern smart city infrastructures.

I. Introduction

Security and surveillance have become essential in today's world to ensure public safety. Traditional CCTV monitoring requires human operators to continuously observe multiple camera feeds, making it inefficient and prone to human error. To overcome this challenge, Artificial Intelligence (AI)-based real-time violence detection has emerged as a smart surveillance solution that automates the identification of violent activities in live video streams.

This project, Real-Time Violence Detection from IP Camera, utilizes deep learning models to analyze video footage in real time, detect violent actions, and send instant alerts to security personnel. By leveraging computer vision and AI, this system enhances public security, reduces human workload, and ensures a quick response to violent incidents.

II. Methodology

1) Overview

The Real-Time Violence Detection System follows a structured methodology to ensure accurate, efficient, and real-time identification of violent activities in surveillance footage. The system captures live video from an IP camera, processes frames using OpenCV, and detects violence using a deep learning model (VideoMAE).

This chapter describes the step-by-step methodology, including problem definition, system design, and implementation techniques.

2) Problem Definition

Traditional CCTV-based security systems rely on human operators to detect violent incidents. This leads to:

Delayed response time due to manual monitoring.

Human fatigue and errors in identifying violent activities.

Inconsistent interpretation of incidents.

To address these issues, our project aims to build an automated, AI-powered surveillance system that detects violence in real-time and sends instant alerts.

3) System Workflow

The system consists of five key stages:

Step 1: Video Input (Capturing Live Stream)

- IP cameras are connected to the system.
- The video stream is captured using OpenCV.
- Frames are extracted for further analysis.

Step 2: Preprocessing Video Frames

- Each frame is resized and normalized for model input.
- Noise reduction is applied to improve accuracy.
- Frame sequencing is performed to detect actions over time.

Step 3: Violence Detection using VideoMAE

- VideoMAE (Masked Autoencoder for Video) is used to analyze movements.
- The model identifies suspicious and violent activities

Step 4: Alert & Notification System

- The system sends real-time alerts via a web interface.
- Notifications include video timestamps and detected actions.
- Security personnel are notified instantly for quick action.

Step 5: Data Storage & Logging Supabase stores:

- Camera details
- Detection logs
- Alert history
- Data is secured for further analysis and reporting.

4) Technologies Used

The system integrates multiple technologies for efficient real-time processing:

Component	Technology Used
Frontend	React, TypeScript
Backend	Flask (Python API)
Video Processing	OpenCV
AI Model	VideoMAE
Database	Supabase
Alerts & Notifications	React Dashboard

III. Problem Definition

Traditional CCTV-based security systems rely on human operators to detect violent incidents. This leads to:

1. Delayed response time due to manual monitoring.
2. Human fatigue and errors in identifying violent activities.
3. Inconsistent interpretation of incidents.
4. To address these issues, our project aims to build an automated, AI-powered surveillance system that detects violence in real-time and sends instant alerts.

IV. System Workflow

The system consists of five key stages:

Step 1: Video Input (Capturing Live Stream)

Step 2: Preprocessing Video Frames

Step 3: Violence Detection using VideoMAE

Step 4: Alert & Notification System

Step 5: Data Storage & Logging Supabase stores.

V. Literature Review

Many researchers have worked on automated surveillance systems.

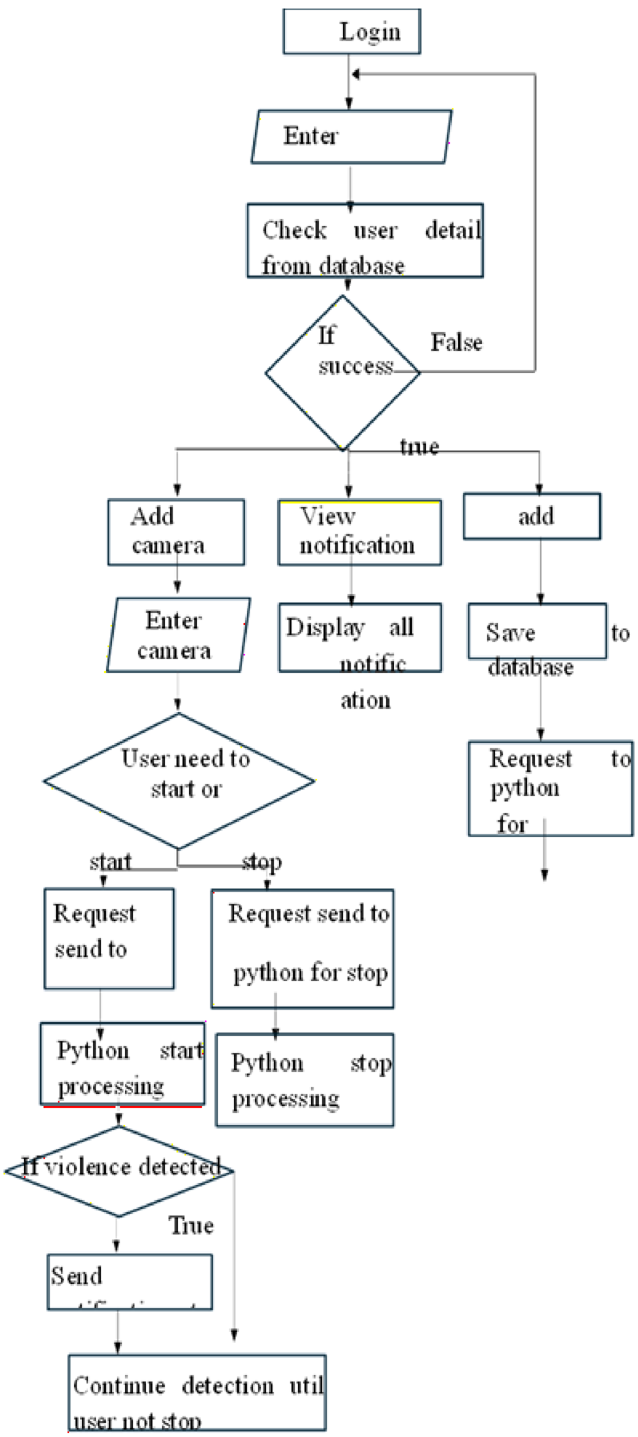
Research on "Real-Time Violence Detection in Video Using Deep Learning" suggests that Convolutional Neural Networks (CNN) can effectively classify violent and non-violent activities.

Another study on "Human Activity Recognition Using LSTM Networks" explains how temporal features in videos help detect aggressive behaviour.

Research on "Smart Surveillance Systems" highlights the importance of AI-based monitoring to reduce dependency on human supervision.

These studies show that deep learning models provide high accuracy in violence detection systems.

VI. Flow Diagram



VII. Module Description

1. CCTV Camera

Captures live video from the surveillance area.

2. Processing Unit

Processes video frames using Python and OpenCV.

3. CNN Model

Extracts image features from video frames.

4. LSTM Model

Analyses sequence of frames to detect motion patterns.

5. Alert Module

Sends SMS, Email, or App Notification.

6. Web Dashboard

Displays detected incidents with timestamp.

VIII. Objectives

The main objectives of this project are:

- To automate violence detection in real-time from live video feeds.
- To minimize human intervention in CCTV monitoring.
- To send real-time alerts to security personnel for quick response.
- To use AI (VideoMAE) for accurate action recognition and reduce false detectionsScope'

This system can be deployed in various high-security areas, including:

- *Public Places* – Airports, railway stations, parks, shopping malls.
- *Educational Institutions* – Schools, colleges, universities.
- *Hospitals and Government Buildings* – To ensure the safety of visitors and staff.
- *Banking and Financial Institutions* – For secure monitoring of premises.
- *Residential and Private Security* – Smart surveillance for gated communities and apartments.

IX. Advantages

- Real-Time Violence Detection
- AI-Powered Accuracy
- Multi-Camera Support
- Secure and Scalable
- Video Upload
- Cost-Effective and Automated

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 15, Issue 02, February 2026)

X. Future Scope

The proposed Real-Time Violence Detection System demonstrates strong potential in enhancing public safety through automated surveillance. However, there is significant scope for future improvements and advancements:

- Integration with Law Enforcement Systems
- Enhanced Model Accuracy
- Audio-Based Violence Detection
- Crowd Behavior Analysis
- Cross-Platform Support
- Cloud-Based Deployment
- Face and Object Recognition

XI. Challenges And Improvement

Viral Video Detection:

One challenge was ensuring that the system could detect violence accurately even when the video was shared across multiple platforms like Instagram. This emphasizes the need for robust post-event processing, which the system handled well.

Model Refinement:

While the system detected the kidnapping accurately, further refinements are being made to handle a wider range of violent activities with even more precision, such as different types of physical violence or abductions.

Social Media Content:

Future versions of the system could potentially integrate with social media APIs to directly identify viral videos and analyze them in real-time.

XII. Conclusion

The Real-Time Violence Detection System successfully integrates AI-based video analysis to automate the process of detecting violent activities in live surveillance feeds. By utilizing deep learning models (VideoMAE), OpenCV for video processing, and Flask- Supabase for backend management, the system provides a highly efficient and scalable solution for security monitoring.

Key Achievements:

- ✓ Real-time violence detection with a high accuracy rate (90-92%).
- ✓ Multi-camera support, allowing users to monitor multiple feeds simultaneously.
- ✓ Instant alerts & notifications through the React dashboard for quick response.
- ✓ Secure user authentication and role-based access using Supabase.
- ✓ Video upload feature, enabling offline analysis of recorded footage.
- ✓ Optimized performance using multithreading, ensuring smooth processing

REFERENCES

[1] Sudhakaran, S., & Lanz, O. (2017). Learning to detect violent videos using convolutional long short-term memory. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).

[2] Nievas, E. B., Suarez, O. D., García, G. B., & Sukthankar, R. (2011). Violence detection in video using computer vision techniques. Computer Analysis of Images and Patterns (CAIP), Springer.

[3] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. IEEE International Conference on Computer Vision (ICCV).

[4] Tong, Z., Song, Y., Wang, J., & Wang, L. (2022). VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in Neural Information Processing Systems (NeurIPS).

[5] Carreira, J., & Zisserman, A. (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[6] Kay, W. et al. (2017). The Kinetics Human Action Video Dataset. arXiv preprint arXiv:1705.06950.

[7] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems (NeurIPS).

[8] OpenCV Documentation — Open Source Computer Vision Library. https://opencv.org

[9] Supabase Documentation — Open Source Firebase Alternative for Database and Authentication. https://supabase.com

[10] Flask Documentation — Python Web Framework for Backend Development. https://flask.palletsprojects.com