International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Violence Detection Network For Video Surveillance Using Deep Learning

P Ganesh Kumar[1], Vaddireddy Lahari[2], Anugu Meghana[3], Nasera Begum[4]

[1]Assistant Professor; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India.

[2,3,4]B.Tech Students; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India.

Mail Id; vaddireddylahari@gmail.com[2], meghanareddy1306@gmail.com[3], 039nasera@gmail.com[4]

Abstract

The rapid expansion of video content generated through surveillance systems and social media platforms has intensified concerns regarding public safety and timely identification of violent activities. Manual monitoring of large-scale video streams is time-consuming, prone to human error, and often leads to delayed responses. To address these limitations, this work proposes an automated violence detection framework based on deep learning techniques for efficient identification of aggressive behavior in video sequences.The proposed approach integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to capture both spatial and temporal information. A pre-trained VGG16 model is utilized for extracting discriminative spatial features from individual frames, while the LSTM network analyzes sequential dependencies to understand motion patterns across time. This hybrid architecture enables robust classification of video segments into violent and non-violent categories.The system is implemented using Python with the Flask framework and provides a user-friendly web interface for uploading video inputs. A preprocessing pipeline performs frame extraction, resizing, normalization, noise filtering, and sequence generation before feeding the data into the model. The output includes classification results, confidence scores, and inference time. To mitigate class imbalance, data augmentation and synthetic sampling techniques are applied.The proposed solution supports scalability, real-time monitoring, and seamless integration with existing surveillance infrastructure. Deployment in public spaces, educational institutions, and transportation hubs can facilitate early detection of violent incidents, thereby enhancing security and enabling faster emergency response.

Keywords: *Violence Detection, Video Surveillance, Deep Learning, CNN-LSTM, VGG16, Temporal Analysis, Video Classification, Real-time Monitoring, Public Safety*

Introduction

The widespread adoption of surveillance systems across urban environments, educational institutions, and transportation facilities has made video monitoring a critical component of modern safety management. Despite the availability of large-scale CCTV infrastructure, surveillance operations still depend heavily on human monitoring, which is inefficient, inconsistent, and difficult to scale. In emergency situations such as public altercations, assaults, or school violence, delayed recognition of harmful activities may result in severe consequences. Furthermore, social media platforms continuously generate vast amounts of video data, making manual moderation impractical.These challenges highlight the need for automated systems capable of analyzing video streams intelligently and detecting violent behavior in real time. This project introduces a deep learning-based Violence Detection System designed to classify video content as violent or non-violent. The framework combines the VGG16 convolutional neural network for spatial feature extraction with Long Short-Term Memory (LSTM) networks for modeling temporal dynamics. This hybrid architecture effectively captures both visual appearance and motion-based patterns.The application includes a web interface built using Python and Flask, allowing users to upload videos for analysis. Each video undergoes preprocessing steps such as frame extraction, normalization, noise reduction, and temporal sequence creation. The system produces classification results along with confidence scores and processing time, ensuring transparency and reliability.The objective of this work is to develop a scalable and accurate violence detection solution that reduces dependence on manual monitoring, improves response time, and enhances security in various environments including public surveillance, educational campuses, and online content moderation platforms.

Purpose of the Project

The main objective of this project is to design an automated system capable of detecting violent activities in video streams with high accuracy. The proposed solution reduces reliance on manual monitoring by leveraging deep learning techniques for intelligent analysis. By combining spatial and temporal feature learning through a CNN-LSTM architecture, the system identifies complex motion patterns associated with violent behavior. The platform provides quick and consistent detection results, supporting faster decision-making.

Ultimately, the project aims to enhance safety in public environments and assist digital platforms in moderating harmful video content.

Existing System

Conventional surveillance systems depend primarily on human operators to monitor CCTV footage. This manual approach is inefficient, error-prone, and unsuitable for large-scale deployment. Basic motion detection algorithms can identify sudden movements but lack contextual understanding, making it difficult to distinguish normal activities from violent behavior. Rule-based systems also fail to capture temporal dependencies and cannot handle complex environments such as crowded scenes or varying illumination conditions. Similarly, content moderation on social media platforms often relies on manual review or simple filters, resulting in delayed detection. These limitations necessitate an intelligent automated solution.

Proposed System

The proposed system introduces an automated violence detection framework based on a hybrid deep learning architecture. A pre-trained VGG16 CNN model extracts spatial features from video frames, while an LSTM network captures temporal motion patterns. The system is deployed as a web-based platform using Python and Flask, enabling users to upload videos for analysis. After processing, the system classifies the input as either "Violence Detected" or "No Violence," along with a confidence score. This approach improves detection accuracy and supports real-time surveillance applications.

Related Work

Recent research in violence detection has focused on deep learning-based spatio-temporal modeling techniques. Early approaches relied on CNN models such as VGG16 to extract spatial features from individual frames. These models demonstrated strong performance in recognizing visual patterns associated with aggressive behavior. Datasets including Hockey Fight and Movies Fight were commonly used for evaluating detection performance.

Subsequent studies integrated handcrafted motion features such as optical flow with CNN architectures, improving behavior recognition accuracy. Advanced models using ResNet architectures further enhanced classification performance by addressing variations in lighting, camera angles, and crowd density. Researchers also explored appearance-based and motion-based feature extraction methods to handle real-world surveillance challenges.

Hybrid architectures combining CNN with LSTM networks gained popularity for capturing both spatial and temporal information. Additionally, 3D CNNs and two-stream networks improved detection accuracy by learning motion dynamics directly from video sequences. More recent approaches incorporate attention mechanisms, transformer-based models, and multi-stream frameworks for enhanced performance in complex environments.

Overall, the evolution from traditional motion detection techniques to deep learning-based spatio-temporal models highlights the effectiveness of hybrid CNN-LSTM architectures for violence detection in real-time surveillance applications.

Requirement Analysis

FunctionalRequirements

The functional requirements define the primary operations that the violence detection system must perform. The system includes a user-friendly interface that allows users to upload video files for analysis. Once the video is uploaded, a preprocessing module extracts frames, resizes them, and applies normalization and noise reduction to prepare the data for model input. The processed frames are then passed to the feature extraction module, where a pre-trained VGG16 convolutional neural network extracts high-level spatial features. These features are subsequently fed into an LSTM-based temporal sequence modeling module, which analyzes motion patterns across frames to understand activity progression. After temporal analysis, the classification module determines whether the video contains violent or non-violent behavior and evaluates the prediction using performance metrics. Finally, the system presents the output through a result visualization module, displaying the classification label along with confidence scores and inference details.

Non-FunctionalRequirements

Non-functional requirements describe the quality attributes and performance expectations of the proposed system. The application must ensure security by protecting uploaded video data and preventing unauthorized access. Reliability is essential so that the system consistently produces accurate detection results with minimal downtime. The platform should provide usability through an intuitive interface that allows even non-technical users to upload videos and view results easily. Scalability is required to support increasing volumes of video data and potential real-time surveillance integration. Portability ensures that the system can operate across different computing environments without major modifications. Additionally, maintainability is important to allow future updates, model improvements, and integration with additional features such as alert systems or cloud deployment.

ComputationalResources

The proposed system requires both software and hardware resources for efficient implementation. The software requirements include Python as the programming language and Flask as the backend framework for web deployment. Essential libraries such as NumPy, Matplotlib, PIL, Torch, and

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Torchvision are used for data processing, visualization, and deep learning model implementation. MySQL is utilized as the database for storing metadata and results, while Anaconda serves as the development environment. GitHub is used for version control and collaboration. The hardware requirements include a system with at least an Intel i3 processor, 4GB of RAM, and 500GB of storage capacity. The recommended operating system for development and deployment is Windows 11, ensuring compatibility with the required software tools and libraries.

Design

System Architecture

The system architecture defines the structural organization of the proposed violence detection framework. It illustrates the interaction between modules including video input, preprocessing, feature extraction, temporal modeling, and classification. The architecture ensures modularity, scalability, and efficient processing. Each component works cohesively to achieve accurate detection results while maintaining performance and reliability.

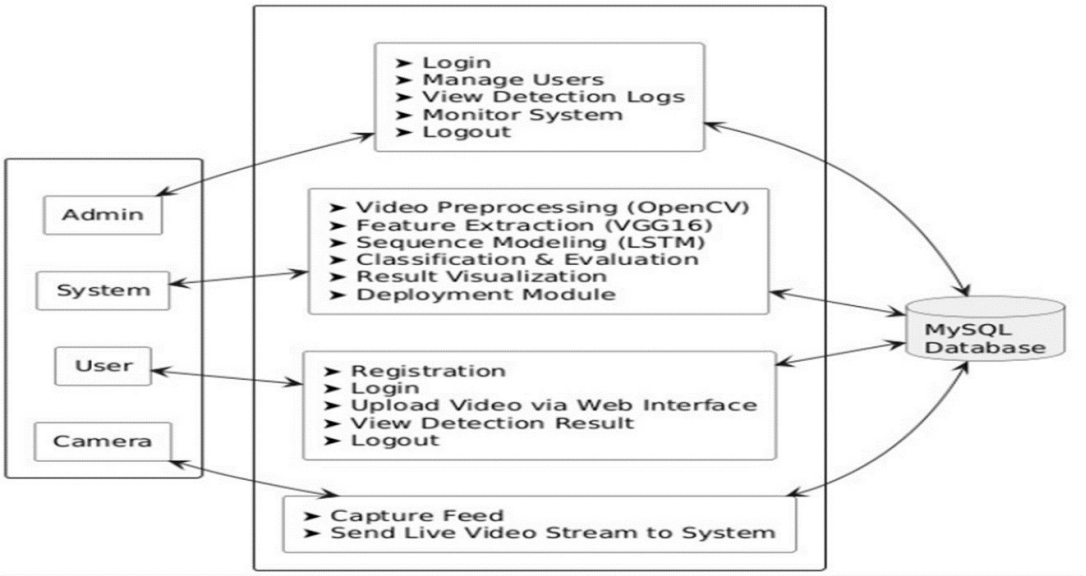


Fig 1 System Architecture

Technical Architecture

The technical architecture consists of a front-end interface developed using HTML, CSS, and JavaScript, allowing users to upload videos and view results. The backend is implemented using Python Flask, which manages routing, preprocessing, and model inference. The deep learning pipeline extracts frames, applies normalization, and feeds data into the VGG16-LSTM model for classification. Results are stored along with metadata for evaluation and future improvements. The modular design supports real-time detection and integration with surveillance systems.

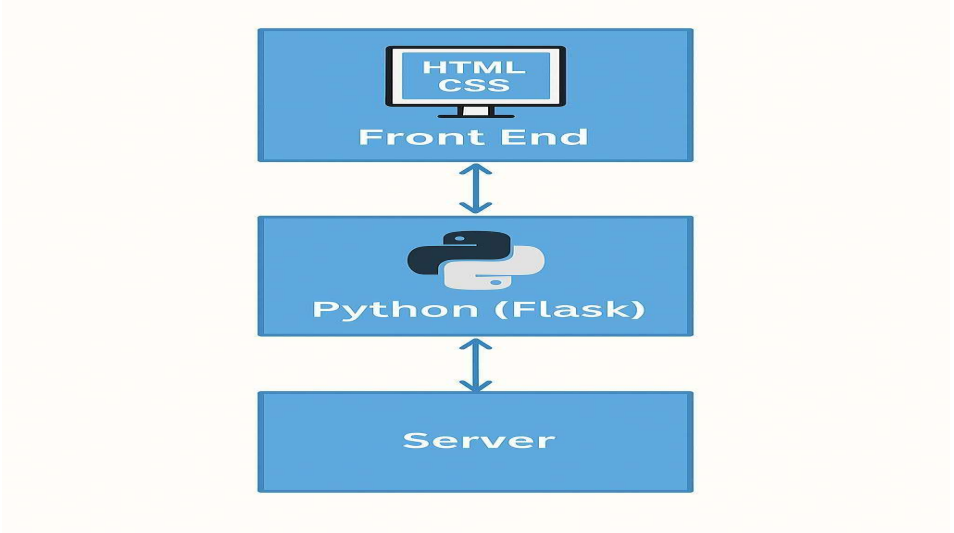


Fig 2 Technical Architecture

Methodology

The proposed violence detection framework employs a hybrid deep learning architecture that integrates Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks. This combination is designed to capture both spatial and temporal characteristics present in video sequences, enabling effective classification of violent and non-violent activities. A pre-trained VGG16 model is utilized for spatial feature extraction due to its deep hierarchical convolutional layers that learn complex visual representations such as human posture, motion cues, and contextual scene information. The VGG16 architecture uses ReLU activation functions and max-pooling layers to refine feature maps and enhance learning stability. By adopting transfer learning, the model leverages pre-trained weights, which reduces computational complexity and accelerates convergence during training.To analyze motion patterns over time, LSTM layers are employed for temporal sequence modeling. The LSTM network processes the ordered feature vectors extracted from individual frames by the VGG16 model. This enables the system to learn temporal dependencies, motion dynamics, and behavioral transitions that are essential for identifying violent interactions. The hybrid CNN–LSTM architecture allows the system to capture subtle cues such as hand movements, body orientation, and interaction intensity, which may not be identifiable using frame-based analysis alone. As a result, both static spatial information and evolving temporal patterns are considered, improving overall classification accuracy.The workflow of the system begins with importing required libraries and extracting frames from uploaded videos. The preprocessing stage includes resizing frames to uniform dimensions, normalization, and noise reduction to improve data quality. Frames are grouped into sequences before being passed to the VGG16 network for spatial feature extraction. The generated feature vectors are then fed into the LSTM model, which analyzes the temporal relationships across frames. Finally, a fully connected layer with sigmoid activation produces probability scores indicating whether the video contains violent or non-violent activity.During implementation, the model is developed using Python with TensorFlow and Keras frameworks. The dataset consists of preprocessed videos containing both violent and non-violent scenes. Data augmentation techniques such as frame skipping, rotation, and brightness adjustment are applied to increase dataset diversity and improve robustness. The CNN and LSTM components are trained jointly in an end-to-end manner. The Adam optimizer is used for efficient weight updates, while binary cross-entropy is selected as the loss function for binary classification. Model performance is evaluated using accuracy, precision, recall, and F1-score. Additionally, confusion matrices are generated to analyze true positives, false positives, and misclassification cases, ensuring a comprehensive performance assessment.

Modules

The proposed system is divided into several functional modules, each responsible for a specific task within the overall violence detection pipeline. The modular design improves maintainability, scalability, and ease of integration. The primary modules include the user interface module, video preprocessing module, feature extraction module, temporal sequence modeling module, evaluation and classification module, and result visualization module.The user interface module provides an interactive platform that allows users to upload videos and view detection results. It supports common video formats and displays analysis status along with final predictions. The video preprocessing module converts raw video inputs into structured data suitable for model processing. This includes extracting frames at fixed intervals, resizing them to standard dimensions, normalizing pixel values, and organizing them into temporal sequences.The feature extraction module is responsible for capturing spatial characteristics from video frames using deep convolutional layers. The VGG16 model extracts high-level features such as body posture, object interactions, and scene context. These features are converted into vectors representing each frame. The temporal sequence modeling module uses LSTM networks to analyze motion patterns across consecutive frames. This module learns action progression and temporal dependencies that help distinguish violent behavior from normal activities.The evaluation and classification module processes the outputs from the LSTM network and classifies the video into violent or non-violent categories. Fully connected layers with sigmoid activation generate probability scores, and evaluation metrics such as accuracy, recall, and F1-score are computed. Finally, the result visualization module presents the outcomes in a user-friendly format. It displays classification results, confidence scores, inference time, and other performance indicators, allowing users to interpret the predictions clearly.

Implementation

The implementation of the proposed violence detection system relies on several software libraries that support deep learning, data processing, and web deployment. TensorFlow and Keras serve as the core frameworks for building and training the CNN–LSTM model. These libraries enable efficient design of neural networks, support GPU acceleration, and provide functionality for saving and loading trained models. OpenCV is used for video processing tasks such as frame extraction, resizing, and noise reduction. It also supports multiple video formats, allowing flexibility in handling input data.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

NumPy is utilized for numerical operations and handling multidimensional arrays, which are essential for image preprocessing and intermediate computations. Pandas is employed for managing structured data, including dataset labels and metadata. Matplotlib is used for visualizing training results, performance metrics, and output frames, which helps in analyzing model behavior. Scikit-learn provides evaluation metrics such as accuracy, precision, recall, and F1-score, along with confusion matrix generation.

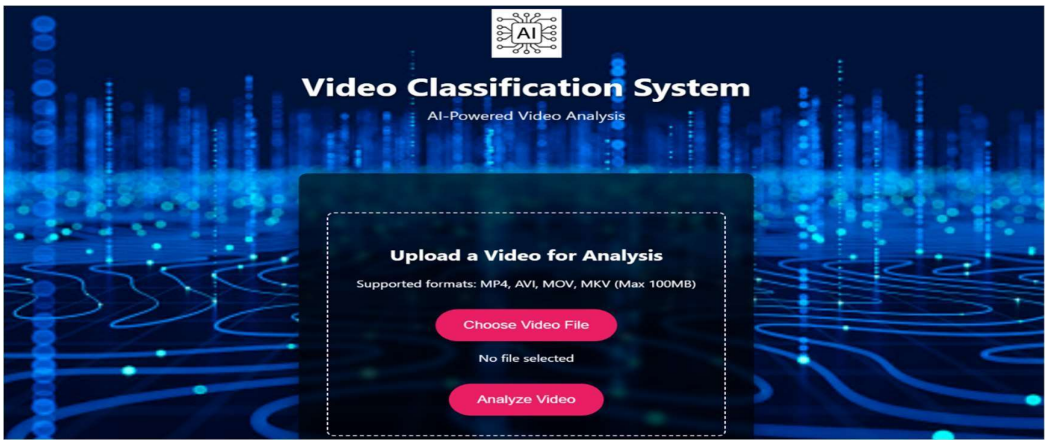
Flask is used as a lightweight backend framework that connects the user interface with the deep learning model. It handles video uploads, processes input data, and returns prediction results. The OS and Glob libraries assist in file management, dataset loading, and directory handling. H5py is used for storing trained models in HDF5 format, enabling efficient model loading during inference. Additionally, ImageDataGenerator is applied for data augmentation, generating additional training samples through transformations such as rotation, flipping, and scaling, which improves model generalization.

Pseudo Code Description

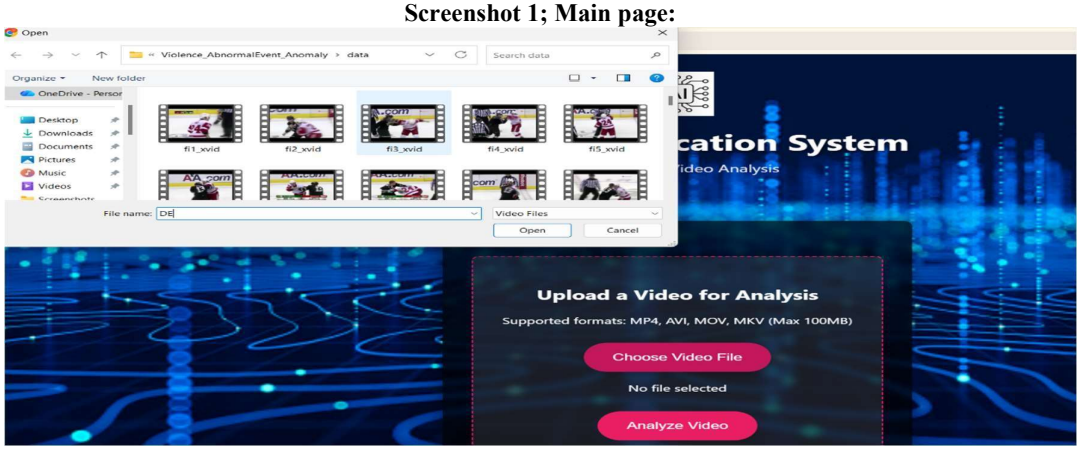
The application begins by importing necessary libraries for deep learning, video processing, and web development. The Flask framework initializes the web application and loads configuration settings. Pre-trained VGG16 and trained LSTM models are loaded for feature extraction and temporal analysis. When a user uploads a video, the system validates the file format and stores it in the upload directory. If required, the video is converted into MP4 format for compatibility.

Frames are extracted from the video using OpenCV and resized to the required input size. These frames are normalized and passed through the VGG16 model to obtain feature vectors. The extracted features are arranged into sequences and fed into the LSTM model for prediction. The model outputs probability scores for violent and non-violent categories. Based on the higher probability, the system determines the final classification. The results, including confidence score and inference time, are displayed to the user through the web interface.

Screenshots

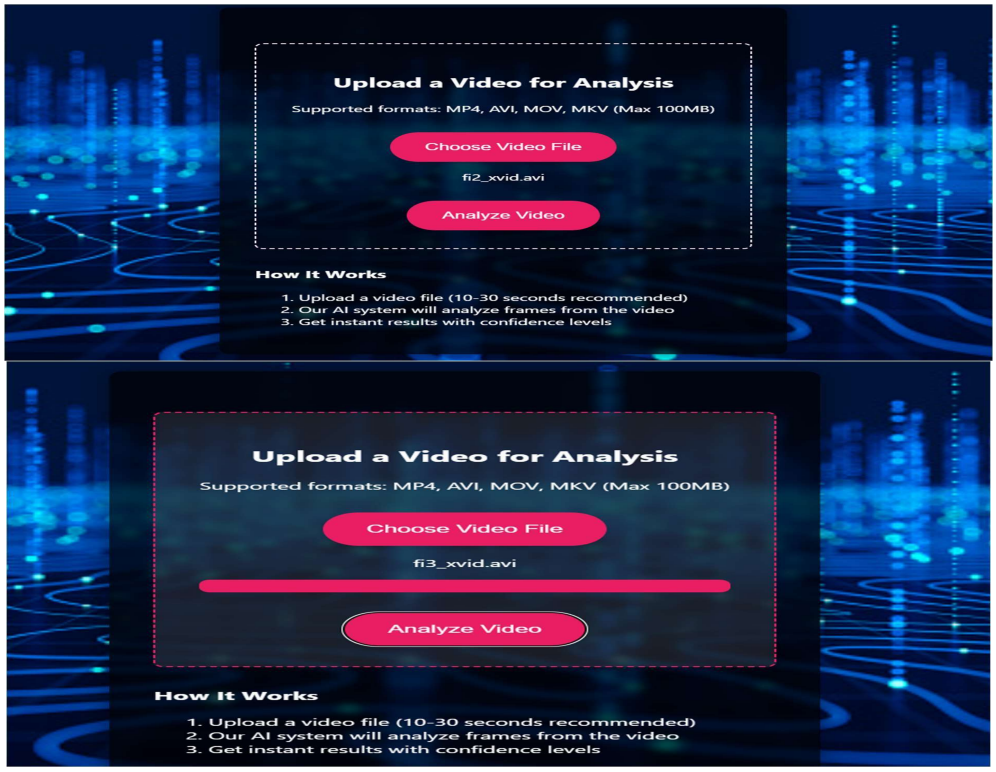


Screenshot 1; Main page:

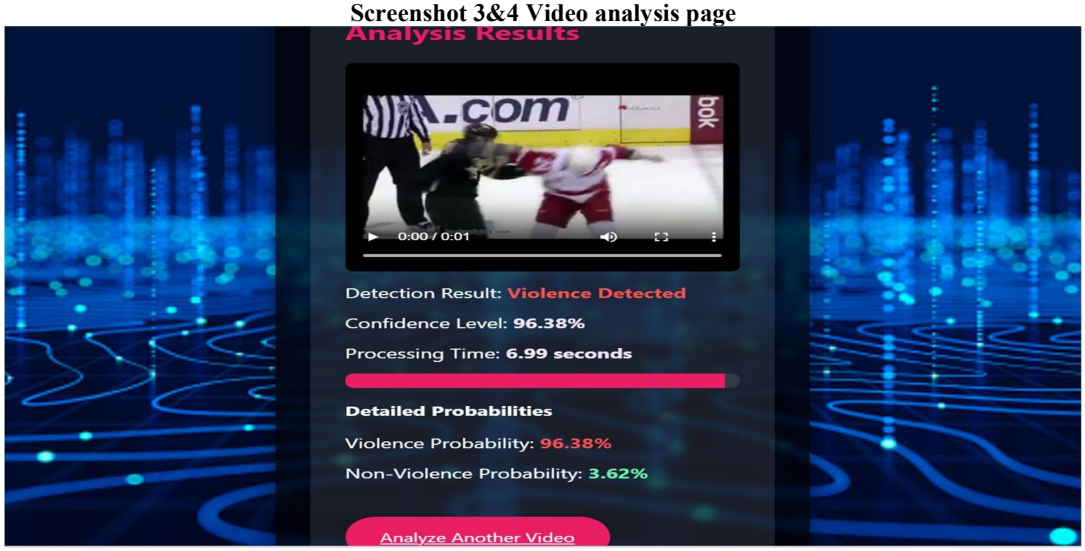


Screenshot 2 Video choosing page

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026



Screenshot 3&4 Video analysis page



Screenshot 5 Result

Conclusion

The proposed Violence Detection Network for Video Surveillance addresses the growing need for automated and reliable identification of violent incidents in video streams. With the widespread deployment of surveillance cameras and the rapid increase in recorded video data, manual monitoring has become inefficient and prone to human error. This work demonstrates how deep learning can enhance traditional surveillance systems by enabling intelligent and proactive threat detection.The system integrates a VGG16-based Convolutional Neural Network for extracting spatial features and a Long Short-Term Memory (LSTM) network for modeling temporal dependencies across video frames. This hybrid architecture allows the model to capture both visual appearance and motion dynamics, which are essential for recognizing violent behavior. The implemented pipeline, including video input, frame preprocessing, feature extraction, sequence learning, and classification, forms a complete end-to-end framework for analyzing video content.Experimental observations indicate that the proposed approach can effectively distinguish

violent and non-violent activities across different scenarios. The model demonstrates strong potential for deployment in surveillance environments such as public spaces, transportation hubs, educational institutions, and smart cities. By automating violence detection, the system reduces reliance on human monitoring and supports faster response to critical situations. Overall, the study confirms that combining CNN and LSTM architectures provides a robust solution for real-time violence detection in video surveillance applications.

Future Scope

The proposed Violence Detection System can be further enhanced to improve performance, scalability, and practical deployment. One important direction is integrating the model with real-time CCTV surveillance systems to enable continuous monitoring and immediate alert generation. Such integration would allow security authorities to respond quickly to potential threats and reduce reaction time during emergencies.Future work can also focus on optimizing the model for edge computing devices, enabling deployment on platforms such as embedded AI hardware. This would allow on-site processing with reduced latency and lower dependency on cloud infrastructure, especially in remote or bandwidth-limited environments. Additionally, extending the system from binary classification to multi-class activity recognition would enable detection of various suspicious behaviors, including fighting, harassment, vandalism, and theft.Improving dataset diversity is another key enhancement. Training with videos captured under different lighting conditions, camera angles, crowd densities, and environmental variations will improve model generalization and robustness. Advanced learning strategies such as semi-supervised learning, transfer learning, and self-supervised learning can also be explored to enhance performance when labeled data is limited.From a usability perspective, a web-based dashboard or mobile application can be developed to display real-time alerts, incident logs, and video monitoring interfaces for security personnel. The system can also be integrated with emergency response mechanisms to automatically trigger alarms or notifications during detected incidents. Privacy-preserving techniques such as anonymization, encrypted data handling, and on-device processing should be incorporated to ensure ethical deployment. Future enhancements may also include gesture analysis and behavioral prediction to identify early signs of aggression, enabling preventive intervention. These improvements will make the system more practical and suitable for large-scale real-world surveillance deployments.

References

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

[2] M. Hassner, Y. Itcher, and O. Kliper-Gross, “Violent flows: Real-time detection of violent crowd behavior,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012.

[3] W. Zhou, Y. Wang, and Z. Wang, “Violence detection in surveillance video using deep learning,” *International Conference on Multimedia Modeling*, 2018.

[4] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” *Advances in Neural Information Processing Systems*, 2014.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *International Conference on Learning Representations*, 2015.

[7] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the Kinetics dataset,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[9] D. Tran et al., “Learning spatiotemporal features with 3D convolutional networks,” *IEEE International Conference on Computer Vision*, 2015.

[10] M. Sultani, C. Chen, and M. Shah, “Real-world anomaly detection in surveillance videos,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.