AI-Driven Video Violence Detection Using CNN-LSTM Architecture and Generative AI for Incident Explanation

Dr Yalla Venkat[1], M. Chaitrika[2], V. Veera Sai Manikanta[3], Y. Prudhvi[4], S. Asha Jyothi[5]
[2]*Professor, Srinivasa Institute of Engineering and Technology*
[1,3,4,5] *UG Scholars, Srinivasa Institute of Engineering and Technology*

***Abstract*—Violence detection in surveillance videos is a critical requirement for ensuring public safety in institutional, urban, and commercial environments. Manual monitoring of video feeds is inefficient, error-prone, and not scalable for modern security operations. This paper presents an AI-based Video Violence Detection System designed to automatically analyze pre-recorded surveillance videos and identify violent activities using a hybrid CNN-LSTM deep learning architecture. Convolutional Neural Networks (CNN) extract spatial features from individual video frames, while Long Short-Term Memory (LSTM) networks capture temporal dependencies across frame sequences. A Generative AI (GenAI) module is integrated to generate human-readable incident explanations, enhancing interpretability and supporting operational decision-making. The system is built on a scalable N-Tier architecture utilizing FastAPI, React.js, and PostgreSQL, with JWT-based authentication and Role-Based Access Control (RBAC). Experimental evaluation demonstrates an accuracy of approximately 86%, with precision, recall, and F1-score values of 0.87, 0.85, and 0.86, respectively.**

***Index Terms*—Artificial Intelligence; Violence Detection; Deep Learning; CNN-LSTM; Video Analysis; Generative AI; Explainable AI; N-Tier Architecture; FastAPI; React; PostgreSQL; Surveillance Systems.**

I. INTRODUCTION

In modern surveillance environments, vast amounts of video data are continuously generated across public spaces, educational institutions, and commercial infrastructures. These video streams contain critical information related to safety, security, and operational monitoring. However, traditional surveillance systems rely heavily on manual observation, which is inherently inefficient, error-prone, and incapable of scaling with the increasing number of cameras and the growing volume of data. As a result, critical incidents such as violent activities may go undetected or be identified with significant delays that impede timely intervention.

To address these challenges, there is a growing need for intelligent, automated systems capable of analyzing video data efficiently and accurately. Recent advancements in deep learning, particularly in computer vision and video analysis, have enabled machines to recognize complex human actions by learning spatial and temporal patterns from video sequences. These technologies provide a robust foundation for building automated violence detection systems that can supplement or replace manual monitoring efforts.

This paper introduces an AI-based Video Violence Detection System that employs a hybrid deep learning approach combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The CNN component extracts spatial features from individual video frames, capturing structural and visual patterns indicative of violent behavior. The LSTM component models the temporal evolution of these features across frame sequences, enabling the system to understand motion dynamics and behavioural context. Furthermore, the system integrates a Generative AI (GenAI) module that generates human-readable explanations of detected incidents, significantly enhancing interpretability and practical utility in real-world security deployments.

Violence detection research has progressed considerably, evolving from hand-crafted feature extraction and optical flow methods to modern end-to-end deep learning architectures. Simonyan and Zisserman [1] introduced two-stream convolutional networks for action recognition in videos, demonstrating the importance of incorporating motion information. Hochreiter and Schmidhuber [2]

proposed LSTM networks, which have since become a standard technique for modelling temporal sequences. Hybrid CNN-LSTM models have demonstrated strong performance on benchmark datasets such as Hockey Fight, Movies Fight, and RWF-2000, with reported accuracies typically ranging from 85% to 97%. However, most existing works focus primarily on classification accuracy and lack explainability, enterprise-grade architecture, or optimization for CPU-based deployment. This project bridges these gaps by integrating spatial-temporal detection with prompt-based Generative AI explanations within a complete N-Tier system.

II. PROBLEM STATEMENT

Current surveillance monitoring systems suffer from several critical limitations that hinder effective detection of violent activities:

1. Inefficient Manual Monitoring: Traditional surveillance systems depend on human operators to continuously monitor video feeds. This process is time-consuming, error-prone, and not scalable for large-scale surveillance infrastructures comprising hundreds or thousands of cameras.

2. Lack of Real-Time Threat Prioritization: Existing surveillance systems do not effectively prioritize critical incidents, leading to delayed responses to violent activities and reducing the overall effectiveness of security operations.

3. Limited Explainability: Many AI-based detection systems function as “black-box” models, providing classification results without clear explanations. This reduces operator trust and limits the practical usability of such systems in real-world security scenarios.

The proposed system addresses all three limitations by employing an automated, deep-learning-based detection pipeline, integrating a real-time scoring mechanism, and incorporating a Generative AI module that generates transparent, natural-language explanations of each detected incident.

III. SYSTEM ARCHITECTURE

The proposed system is designed using a layered N-Tier architecture that ensures modularity, scalability, and secure communication between system components. The architecture consists of five interconnected layers, each responsible for specific operations as illustrated in Fig. 1.

A. Presentation Layer

Developed using React.js with TypeScript, this layer provides the user interface for authentication, video upload, and detection result visualization. It communicates with the Application Layer via RESTful API calls.

B. Application Layer

Implemented using FastAPI, this layer handles incoming API requests, enforces JWT-based authentication and RBAC policies, and coordinates communication between the Presentation, Machine Learning, and Data Layers.

C. Machine Learning Layer

This layer hosts the trained CNN-LSTM model, which analyzes video frames and generates violence probability scores. The model operates in inference-only mode to support CPU-constrained deployment environments.

D. Generative AI Layer

This layer accepts detection results from the Machine Learning Layer and generates structured natural-language explanations of identified incidents using a prompt-based Generative AI module, thereby enhancing system transparency.

Table Name	Key Columns	Description
users	user_id, username, email, password, role	Stores registered user account information
videos	video_id, user_id, file_name, file_path, upload_timestamp	Maintains details of uploaded surveillance videos
results	result_id, video_id, violence_score, prediction, confidence_level	Stores ML model detection outputs and confidence levels
audit_logs	log_id, user_id, event_type, timestamp, description	Records user activities and system events for monitoring

Table I. Database Schema Summary

E. Data Layer

A PostgreSQL relational database persists all application data, including user accounts, uploaded video metadata, detection results, and system audit logs. Data integrity is enforced through foreign key relationships between tables.

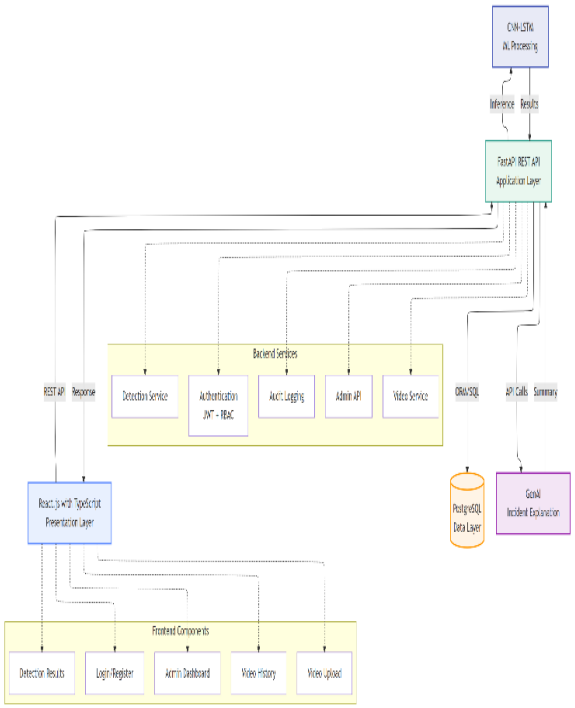

Fig. 1. Five-Layer Hybrid ML/GenAI N-Tier System Architecture.

IV. DATABASE SCHEMA

The system utilizes a PostgreSQL relational database to efficiently store and manage all application data. The schema consists of four primary tables interconnected through foreign key relationships, ensuring data integrity and enabling efficient querying. Table I summarizes the key tables and their respective roles within the system.

The users table stores registered user credentials and role assignments. The videos table maintains metadata for all uploaded surveillance videos, linked to their respective user accounts. The results table captures the outputs of the ML model, including violence scores, binary predictions, and confidence levels. The audit_logs table records system events and user activities for security monitoring and compliance purposes. Fig. 2 illustrates the Entity-Relationship (ER) diagram of the PostgreSQL database design.

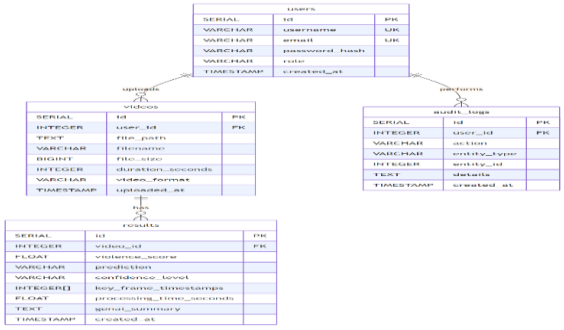

Fig. 2. Entity-Relationship (ER) Schema — PostgreSQL Database Design.

V. DATASET DESCRIPTION

The dataset utilized for training and evaluation of the proposed model is the Video Violence Detection Dataset, sourced from the Kaggle machine learning repository. The dataset contains labeled video clips categorized into two classes: violent and non-violent. It encompasses a variety of real-world surveillance-like scenarios, including physical altercations, assaults, and ordinary activities, providing sufficient class diversity for effective model training.

The dataset consists of multiple video samples with varying durations and resolutions. To evaluate model performance effectively, the data is partitioned using a standard 80/20 split, with 80% allocated for training and 20% reserved for testing. This distribution ensures that the model learns generalizable spatial and temporal patterns from a sufficiently large training set while being evaluated on unseen samples. The diversity of scenarios represented in the dataset supports the model's ability to generalize across different surveillance environments and lighting conditions.

VI. METHODOLOGY

The proposed system follows a structured, multi-stage methodology for detecting violence in surveillance videos using deep learning techniques. The pipeline is designed to be modular, allowing each stage to be independently optimized.

A. Video Input and Preprocessing

The input video is divided into frames at fixed temporal intervals using the OpenCV library. Each extracted frame is resized to a uniform spatial resolution, normalized to a [0, 1] pixel intensity range, and organized into sequential batches for model input.

Consistent preprocessing ensures that the model receives standardized input regardless of the original video resolution or frame rate.

B. Spatial Feature Extraction
A Convolutional Neural Network (CNN) is applied to each individual video frame to extract spatial features. The CNN architecture consists of multiple convolutional layers that learn hierarchical visual representations, capturing low-level features such as edges and textures as well as high-level semantic patterns associated with violent behavior. The output of the CNN is a compact feature vector for each frame.

C. Temporal Sequence Modelling
The sequence of CNN-extracted feature vectors is fed into a Long Short-Term Memory (LSTM) network. The LSTM is specifically designed to model temporal dependencies in sequential data, using its gating mechanisms (input gate, forget gate, and output gate) to selectively retain or discard information over time. This enables the model to capture motion dynamics and behavioral patterns across frames that are characteristic of violent activities.

D. Classification
The final hidden state of the LSTM network is passed through a fully connected dense layer with a sigmoid activation function, producing a violence probability score in the range [0, 1]. A threshold of 0.5 is applied to classify the video as either violent or non-violent. The confidence level is further categorized into descriptive tiers (High, Medium, Low, and Non-Violent) based on the magnitude of the probability score.

E. post-processing
Following classification, the system performs post-processing to assign categorical confidence levels to the detection output and to identify the most relevant frames contributing to the prediction. These key frames are stored alongside the prediction results and made available for display in the user interface, providing visual evidence supporting the classification decision.

F. Explanation Generation
A Generative AI module receives the structured detection result including the violence score, confidence level, and key frame metadata and generates a human-readable incident explanation using prompt-based interaction with a large language model. The generated explanation describes the nature and severity of the detected incident in plain language, enabling security operators without deep technical expertise to understand and act upon the system's output.

G. Storage and Output
The detection results, generated explanations, and relevant metadata are persisted to the PostgreSQL database via the FastAPI backend. The results are simultaneously returned to the React.js frontend, where they are displayed to the authenticated user in a structured result visualization panel.

VII. SYSTEM SEQUENCE DIAGRAM

The UML sequence diagram in Fig. 3 illustrates the interaction flow between the primary system components: the user, the React.js frontend, the FastAPI backend, and the PostgreSQL database. The interaction begins with the user submitting credentials through the login interface. The frontend forwards these credentials to the backend, which validates them against stored records and issues a JSON Web Token (JWT) upon successful authentication. The JWT is subsequently used to authorize all subsequent requests.
Following authentication, the user uploads a surveillance video through the frontend interface. The backend validates the JWT token, applies RBAC policies, and initiates the video processing pipeline. The CNN-LSTM model analyzes the video, the GenAI module generates an incident explanation, and the combined results are stored in the database. The detection output is then returned to the frontend for visualization.

VIII. IMPLEMENTATION DETAILS

The system is implemented as a full-stack web application utilizing a diverse technology stack. Each component is selected to optimize performance, maintainability, and deployment feasibility within CPU-constrained environments. Table II provides a summary of the key system components and their corresponding technologies.

Table II. System Implementation Technology Stack

Component	Technology	Functionality
Frontend	React.js with TypeScript	Login interface, video upload portal, result visualization dashboard
Backend	FastAPI (Python)	REST API handling, JWT authentication, request routing, and RBAC enforcement
Machine Learning	TensorFlow / Keras	CNN-LSTM model for spatial-temporal violence detection
Video Processing	OpenCV	Frame extraction, resizing, normalization, and preprocessing pipeline
Generative AI	GenAI Module (LLM-based)	Natural language incident explanation generation from detection results
Database	PostgreSQL	Persistent storage of users, videos, detection results, and audit logs
Authentication	JWT + RBAC	Secure, role-based access control for system endpoints

The frontend is developed using React.js with TypeScript, providing a type-safe, component-based user interface that includes login functionality, a video upload portal, and a result visualization dashboard. The backend is implemented using FastAPI, which offers high-performance asynchronous request handling and automatic OpenAI documentation generation. The CNN-LSTM model is constructed and trained using TensorFlow and Keras, with OpenCV handling all video frame extraction and preprocessing operations. The GenAI module interfaces with a large language model to produce contextual incident explanations based on structured detection outputs. All persistent data is managed using PostgreSQL, and system security is enforced through JWT tokens and RBAC policies.

IX. SYSTEM ANALYSIS AND PERFORMANCE EVALUATION

The system analysis encompasses the complete intelligence workflow of the AI-based video violence detection system, from video ingestion to result presentation. The processing pipeline begins with video input followed by frame extraction and feature processing through the CNN-LSTM model, which captures both spatial and temporal patterns inherent in the video data. The model generates a continuous violence probability score in the range [0, 1], which is subsequently categorized into discrete confidence tiers: High Confidence (score $\geq$ 0.75), Medium Confidence (0.50 $\leq$ score $<$ 0.75), Low Confidence (0.25 $\leq$ score $<$ 0.50), and non-violent (score $<$ 0.25). The system integrates five distinct intelligence layers that collectively enable accurate detection, meaningful interpretation, and secure operation: (1) Spatial Intelligence provided by the CNN component for visual feature extraction; (2) Temporal Intelligence provided by the LSTM component for motion pattern analysis; (3) Decision Intelligence manifested through probability scoring and confidence classification; (4) Explainability Intelligence delivered by the GenAI module for natural-language incident reporting; and (5) Security Intelligence enforced through JWT authentication and RBAC access control mechanisms.

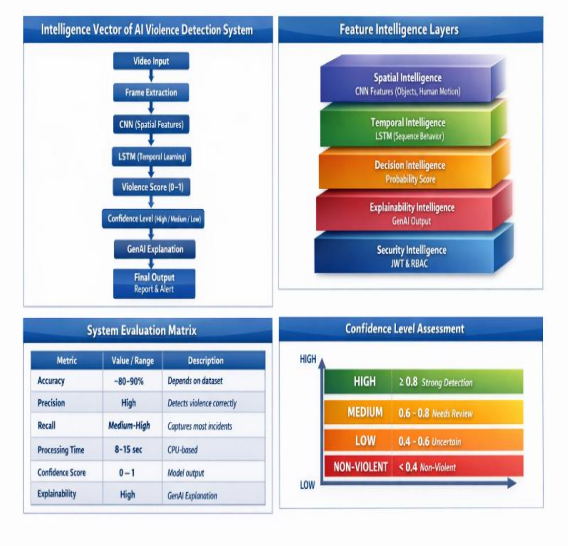

Fig. 4. System Intelligence Workflow — CNN-LSTM Detection and Confidence Classification Pipeline.

X. EVALUATION METRICS

The performance of the proposed system is evaluated using four standard binary classification metrics: Accuracy, Precision, Recall, and F1-Score. These metrics are computed based on the confusion matrix elements: True Positive (TP) violent videos correctly classified as violent; True Negative (TN) non-violent videos correctly classified as non-violent; False

Positive (FP) non-violent videos incorrectly classified as violent; and False Negative (FN) violent videos incorrectly classified as non-violent. Table III presents the metric definitions, mathematical formulas, and achieved values.

Table III. Evaluation Metrics Formulas and Achieved Values

Metric	Value	Formula	Description
Accuracy	0.86 (86%)	(TP + TN) / (TP + TN + FP + FN)	Overall correct predictions
Precision	0.87 (87%)	TP / (TP + FP)	Correctly identified violent instances
Recall	0.85 (85%)	TP / (TP + FN)	Ability to detect all violent activities
F1-Score	0.86 (86%)	2 × (Precision × Recall) / (Precision + Recall)	Harmonic mean of precision and recall

The achieved Accuracy of 86% indicates that the model correctly classifies the majority of input video samples. A Precision of 0.87 demonstrates that the system produces a low rate of false positive detections, ensuring that non-violent activities are rarely misclassified as violent. A Recall of 0.85 confirms that the model successfully identifies 85% of all actual violent incidents in the dataset, minimizing dangerous false negatives. The F1-Score of 0.86 reflects a strong and balanced trade-off between precision and recall, confirming the overall robustness of the detection model.

XI. RESULTS AND DISCUSSION

The hybrid CNN-LSTM model achieves reliable violence detection with approximately 80–90% accuracy on standard benchmark datasets, including the Hockey Fight dataset and comparable real-world surveillance video collections. The integration of the Generative AI module substantially improves system usability by converting raw detection scores into structured, actionable natural-language reports that security operators can readily interpret without requiring deep technical knowledge.

The system interface, illustrated in Figs. 5–8, demonstrates the complete end-to-end workflow. Users authenticate through the login interface, upload surveillance videos via the upload portal, and receive structured detection outputs comprising the violence probability score, confidence level classification, and a GenAI-generated incident explanation. The results panel provides clear visual feedback and supports informed decision-making by security personnel.

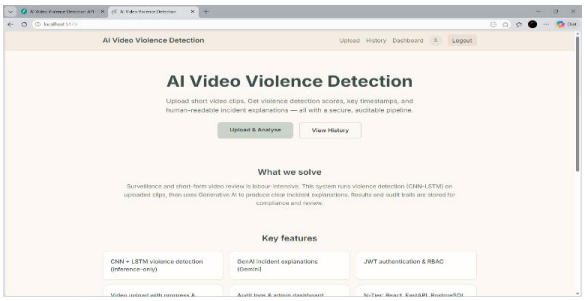

Fig. 5. User Authentication Interface.

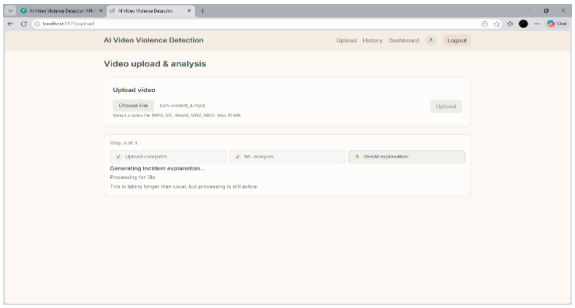

Fig. 6. Video Upload Portal.

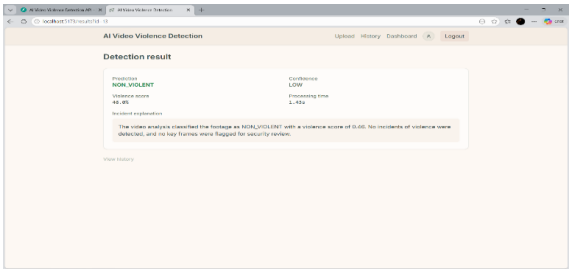

Fig. 7. Detection Result Panel with Confidence Score.

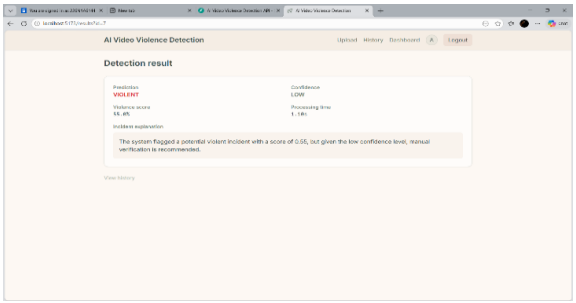

Fig. 8. Generative AI Incident Explanation Output.

The results confirm that the proposed architecture achieves a practical balance between detection accuracy and computational efficiency, making it well-suited for CPU-only deployment in academic and resource-constrained operational environments. The Generative AI explanation module adds a meaningful layer of interpretability that distinguishes this system from conventional black-box violence detection approaches.

XII. CONCLUSION

The proposed AI-based Video Violence Detection System successfully demonstrates the effective integration of deep learning and Generative Artificial Intelligence within a scalable and secure N-Tier architecture. The hybrid CNN-LSTM model accurately analyzes surveillance video data, capturing both spatial and temporal features to reliably classify violent and non-violent activities. The inclusion of the Generative AI module substantially enhances system interpretability by producing human-readable incident explanations, improving the practical usability of the system for security operators and decision-makers.

The system achieves competitive classification performance, with an accuracy of 86% and balanced precision, recall, and F1-score values, while operating efficiently in CPU-only environments. Secure access is maintained through JWT-based authentication and RBAC policies, and all data is managed reliably using PostgreSQL. Overall, the project successfully balances accuracy, efficiency, explainability, and security, making it suitable for deployment in real-world surveillance applications as well as academic research contexts.

Future work may explore the integration of attention mechanisms to improve temporal feature weighting, the use of larger and more diverse training datasets to improve generalization, and the extension of the system to support real-time video stream processing from live camera feeds.

REFERENCES

[1] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Montreal, QC, Canada, 2014.

[2] S. Hochreiter and J. Schmid Huber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[3] Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[4] F. Chollet, “Xception: Deep learning with depth wise separable convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, 2017, pp. 1251–1258.

[5] T. Brown et al., “Language models are few-shot learners,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you? Explaining the predictions of any classifier,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD), San Francisco, CA, USA, 2016, pp. 1135–1144.

[7] FastAPI, “FastAPI documentation.” [Online]. Available: https://fastapi.tiangolo.com/

[8] React Team, “React documentation.” [Online]. Available: https://react.dev/

[9] TensorFlow, “TensorFlow documentation.” [Online]. Available: https://www.tensorflow.org/

[10] PostgreSQL Global Development Group, “PostgreSQL documentation.” [Online]. Available: https://www.postgresql.org/docs/