International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

DeepGuard: Enhancing Violence Detection in Smart Cities Through Deep Learning

K. Mahesh Lakshmana Kumar
Reg. No. 24Q71F0021
maheshlakshmankaduluri@gmail.com
Department of Master of Computer Applications
Avanthi Institute of Engineering and Technology (Autonomous)
Vizianagaram, Andhra Pradesh, India
Under the guidance of Mr. P. Satyanarayana, M.Tech., (Ph.D.), Associate Professor
pukkallasatya84@gmail.com

***Abstract*—*DeepGuard is a deep-learning-based violence detection system designed to enhance public safety in smart-city surveillance environments. The system uses computer vision and a hybrid neural architecture—a Convolutional Neural Network (ResNet18) for spatial feature extraction combined with a Bi-directional Long Short-Term Memory (Bi-LSTM) network for temporal modelling—to analyse video footage from CCTV and uploaded sources. Frames are sampled at fixed intervals, resized, normalised, and converted to tensors before being passed through the hybrid model, which classifies the sequence as Violence or Non-Violence. When a violent activity is detected, the system triggers an immediate alert to designated personnel through an interactive Streamlit-based interface. The implementation is realised in Python using PyTorch, OpenCV, NumPy, and Pillow. By combining spatial and temporal cues, the system reduces dependence on continuous manual monitoring and provides a scalable, automated solution suitable for integration with existing smart-city surveillance infrastructure. Functional, integration, system, performance, and accuracy tests confirm the correctness of all modules and end-to-end behaviour of the prototype.*

***Keywords*—*Violence Detection; Smart-City Surveillance; Deep Learning; Convolutional Neural Network; Long Short-Term Memory; ResNet18; Streamlit; PyTorch.*

I. INTRODUCTION

The rapid growth of urban populations and the expansion of smart-city infrastructure have created a pressing demand for advanced surveillance systems capable of safeguarding public spaces in real time. Traditional video surveillance depends almost entirely on human operators continuously monitoring large numbers of camera feeds. This approach is fatiguing, error-prone, and inherently unable to scale to the volume of data generated in modern urban environments. Critical violent incidents, such as fights, assaults, and abnormal crowd behaviour, are therefore often missed or recognised only after the fact.

Deep learning, a subfield of artificial intelligence, has demonstrated strong performance in computer vision tasks including image recognition, object detection, and activity classification. By leveraging deep neural networks, surveillance systems can be enhanced to interpret complex human behaviour and to

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

identify unusual or violent activities directly from video streams, replacing passive recording with proactive threat detection.

The proposed system, DeepGuard, addresses these limitations by combining a Convolutional Neural Network (CNN) for spatial feature extraction with a sequence model (LSTM) for temporal pattern analysis. The system accepts video input from CCTV feeds or uploaded files, classifies the activity as Violence or Non-Violence, and generates instant alerts whenever a violent incident is detected. The contributions of this work are: (i) a hybrid CNN–Bi-LSTM model tailored to smart-city violence detection; (ii) a modular processing pipeline covering input, preprocessing, classification, and alerting; and (iii) a usable Streamlit-based interface that allows operators to upload models and video and view predictions with confidence scores.

II. LITERATURE SURVEY

Early work in violence and action recognition relied on handcrafted features such as the Histogram of Oriented Gradients (HOG) and optical flow. While these methods captured basic motion patterns, they struggled with complex human interactions and real-world variability. The emergence of Convolutional Neural Networks (CNNs) significantly improved spatial feature extraction, with architectures such as AlexNet and ResNet establishing strong baselines on image classification tasks that were later extended to video.

To capture temporal dynamics, which CNNs alone cannot model, researchers introduced hybrid approaches combining CNNs with Long Short-Term Memory (LSTM) networks, as well as 3D CNNs and two-stream networks that handle motion and appearance jointly. More recent studies have explored attention-based and transformer-based architectures for richer video understanding. Despite these advances, real-time processing, computational cost, and robustness under varying lighting and crowd conditions remain open challenges. DeepGuard builds on this body of work by integrating CNN-based spatial feature extraction with Bi-LSTM-based temporal modelling, targeting a balance between accuracy and deployability in smart-city environments.

TABLE I. SUMMARY OF REPRESENTATIVE PRIOR WORK

S.No	Author(s) / Year	Method	Description	Limitation
1	Doshi et al., 2019	CNN-based detection	Classifies violent vs. non-violent frames	Lacks temporal analysis
2	Shariq et al., 2020	CNN + LSTM	Combined spatial and temporal features for activity recognition	High computational cost
3	Hassner et al., 2012	Violent Flows	Motion-based features for violence detection on a new dataset	Limited dataset diversity

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

S.No	Author(s) / Year	Method	Description	Limitation
4	Simonyan & Zisserman, 2014	Two-Stream CNN	Separate spatial and temporal streams for action recognition	Complex architecture
5	Tran et al., 2015	3D CNN	Captures spatiotemporal features directly from videos	Requires large training data
6	Carreira & Zisserman, 2017	I3D	Improved 3D CNN using ImageNet pretraining	High resource consumption

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

Existing surveillance deployments rely primarily on continuous human monitoring of multiple video feeds. This model has well-documented limitations: it is tedious and error-prone over long shifts, vulnerable to fatigue, lacks real-time automated alerting, and struggles to extract useful information from the large volumes of video data produced by city-scale camera networks. Such systems also have limited ability to detect complex or subtle violent activity.

Disadvantages of the existing approach:

- Continuous multi-camera monitoring is tedious and error-prone.
- Critical events are easily missed due to operator fatigue.
- No real-time automated alert mechanism.
- Difficulty in analysing large volumes of video data.
- Limited ability to detect complex or subtle violent activities.

B. Proposed System

DeepGuard introduces an AI-based detector that automatically classifies video activity using a hybrid deep-learning model. Key features include automated video analysis, spatial feature extraction with a CNN, temporal modelling with an LSTM, real-time inference, instant alert generation, and a reduced reliance on manual monitoring. The system processes incoming video, samples frames, classifies the activity, and surfaces the result—together with a confidence score—through an interactive interface.

Advantages of the proposed system:

- Fully automated video analysis using deep learning.
- Hybrid spatial-temporal model (CNN + LSTM) captures both appearance and motion.
- Near real-time detection of violent activity.
- Instant alerts to authorities for faster intervention.
- Integrates with existing CCTV / surveillance infrastructure.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

IV. SYSTEM DESIGN AND ARCHITECTURE

A. Architecture Overview

The DeepGuard architecture follows a modular pipeline composed of eight cooperating layers: a Video Input Layer that accepts CCTV feeds or uploaded video; a Frame Extraction Module that converts video to discrete frames at fixed intervals; a Preprocessing Layer that resizes, normalises, and converts frames to tensors; a Spatial Feature Extraction stage based on a CNN; a Temporal Analysis stage based on an LSTM; a Classification Layer producing Violence or Non-Violence with confidence; an Alert and Notification System that informs personnel when violence is detected; and a User Interface that displays uploaded video, sampled frames, predictions, and confidence levels.

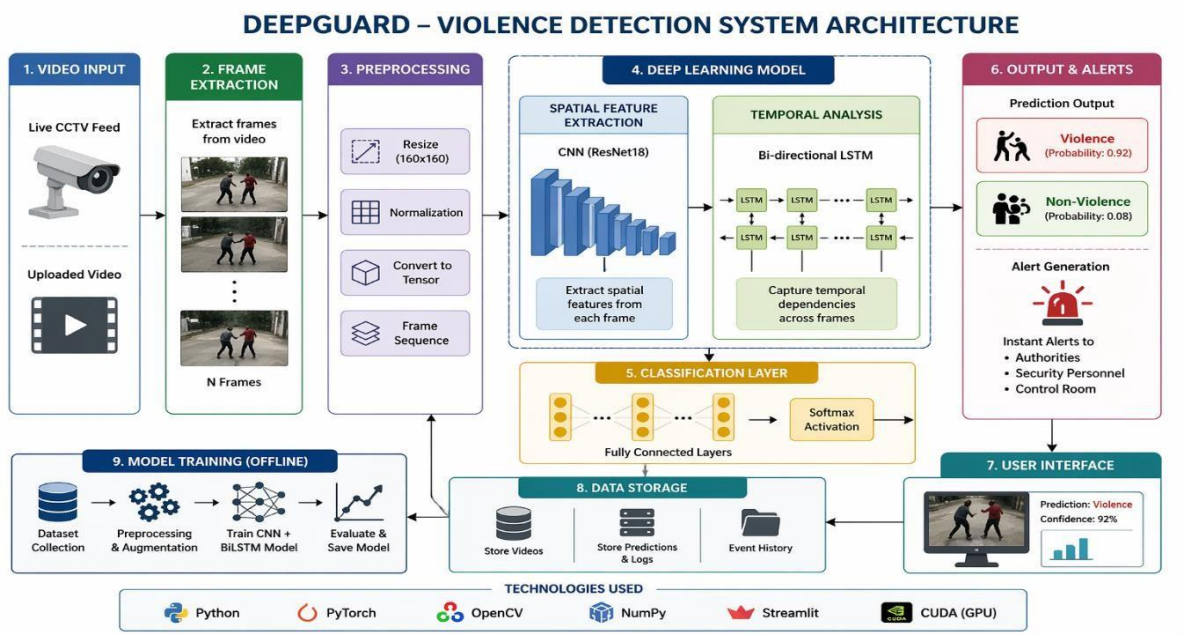

Fig. 1. System Architecture

B. Working Flow

Video input is supplied to the system from either a CCTV camera or an uploaded file. The Frame Extraction Module samples frames at regular intervals. Each frame is preprocessed and passed to the CNN, which extracts spatial features. The resulting feature sequence is fed to the LSTM, which captures temporal dependencies across frames. The classifier produces a prediction with a confidence score; if the prediction is Violence, the alert module is triggered, and the results are displayed on the UI.

C. Functional and Non-Functional Requirements

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Functionally, the system must accept video input, extract frames, preprocess them, run inference, classify activities, display predictions with confidence, and generate alerts on detection. Non-functional requirements include fast processing for near real-time detection, high precision in identifying violent activities, scalability to handle multiple video streams, reliability under varying conditions, an easy-to-use interface, and safe handling of surveillance data.

V. SYSTEM IMPLEMENTATION

A. Technology Stack

TABLE II. TECHNOLOGY STACK

Component	Technology / Tool
Programming Language	Python
Deep-Learning Framework	PyTorch
Computer Vision Library	OpenCV
Web Framework / UI	Streamlit
Image Processing	Pillow
Numerical Computation	NumPy
Compute Device	CUDA GPU when available, otherwise CPU

B. Model Architecture

The model is implemented as a PyTorch nn.Module named DeepGuard. The spatial branch uses ResNet18 with the final classification head removed; the remaining convolutional backbone serves as a frozen feature extractor producing a 512-dimensional descriptor per frame. These per-frame features are arranged as a sequence and fed to a Bi-directional LSTM with hidden size 256, configured with batch_first=True and bidirectional=True. The Bi-LSTM output is passed through a fully connected layer that maps to two classes: NonViolence and Violence.

C. Preprocessing and Inference Pipeline

Videos are read with OpenCV (cv2.VideoCapture). A fixed number of frames—eight per clip in the prototype—are sampled at regular intervals to ensure a manageable input size. Each sampled frame is resized to 160×160 pixels, normalised using standard mean and standard-deviation values, and converted to a tensor with torchvision transforms. The stacked frame tensors are passed through the model, and a softmax is applied to the output logits to obtain class probabilities, from which the predicted label and confidence score are extracted.

D. User Interface

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

The user interface is implemented in Streamlit. Operators can upload a trained model (.pth file), upload a video file or capture from a webcam, preview the input, and view sampled frames alongside the prediction and confidence value. The interface is intentionally minimal so that personnel without machine-learning expertise can operate it. Internally, the prototype was reachable during development at http://10.23.195.132:8501.

VI. SYSTEM TESTING AND RESULTS

Testing was conducted at four levels. Unit tests verified the correctness of the Frame Extraction, Preprocessing, Model Prediction, and UI modules in isolation. Integration tests confirmed that data flows smoothly from video input through frame extraction, preprocessing, model inference, and finally to the user interface. System tests exercised the complete pipeline end-to-end with real video inputs, including the triggering of alerts when violence was detected. Performance tests measured processing time across CPU and GPU environments and behaviour under longer video inputs. Accuracy testing used the qualitative metrics of accuracy, precision, recall, and F1-score, evaluated on test videos.

TABLE III. REPRESENTATIVE TEST CASES

ID	Description	Input	Expected Output	Result
TC01	Upload valid video	MP4 file	Video displayed	**Pass**
TC02	Extract frames	Video input	Frames extracted correctly	**Pass**
TC03	Preprocess frames	Raw frames	Normalised tensors	**Pass**
TC05	Predict violence	Fight video	Violence detected	**Pass**
TC07	Alert generation	Violence detected	Alert triggered	**Pass**
TC08	Webcam input	Image capture	Prediction displayed	**Pass**

A. Observed Results

The prototype detects violent activity with good qualitative accuracy on the test videos used during evaluation. Real-time prediction works efficiently for short video clips, alerts are triggered correctly when violence is detected, and the Streamlit interface presents clear, easily-interpretable outputs. Performance benefits noticeably from GPU execution; on CPU, longer videos exhibit a slight delay. Quality degrades on low-resolution or heavily-blurred input, and final accuracy depends on the diversity of the training dataset.

Representative screenshots from the prototype implementation:

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

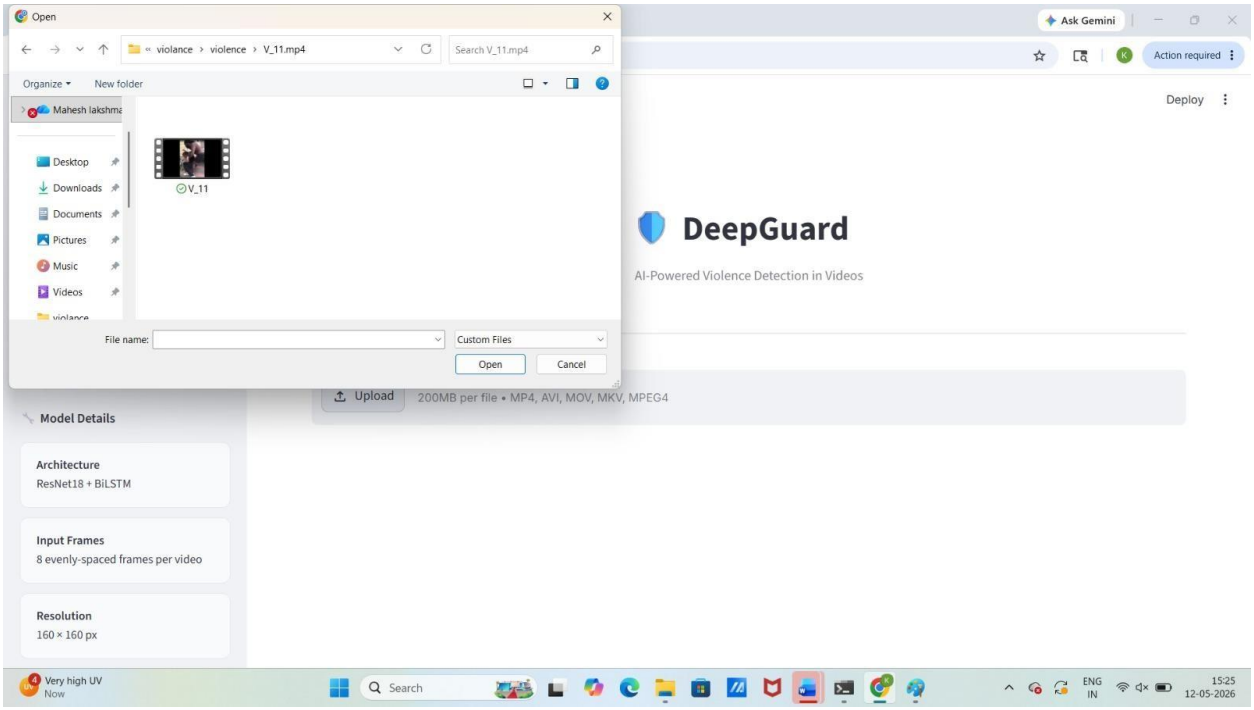

Fig. 2. Streamlit input screen for model and video upload.

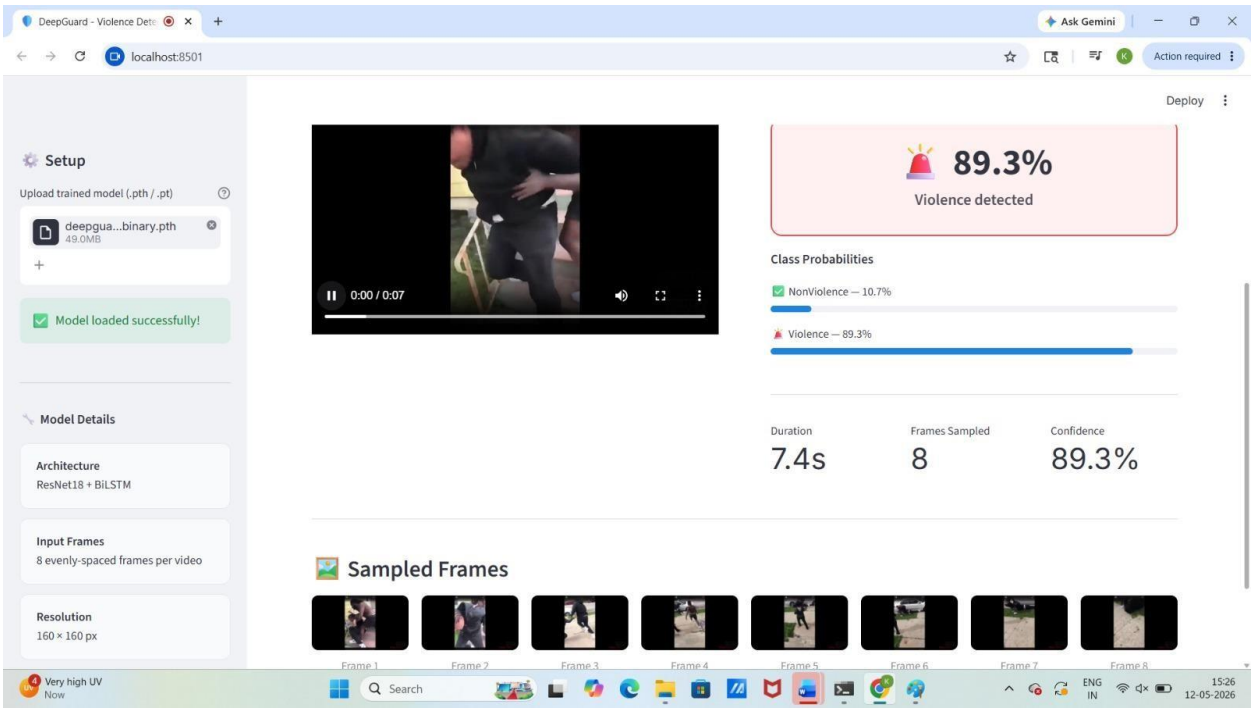

Fig. 3. Sampled frames preview from an uploaded video and Prediction output with class label and confidence score.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

VII. CONCLUSION AND FUTURE SCOPE

DeepGuard demonstrates how modern AI techniques can be applied to public-safety problems in smart-city environments. By combining a CNN-based spatial feature extractor (ResNet18) with a Bi-LSTM temporal analyser, the system delivers automated violence detection on video streams together with confidence-aware alerting. The prototype, implemented in PyTorch and Streamlit, reduces reliance on continuous human monitoring, lowers the risk of missed incidents, and supports faster response through automated notification.

Several extensions are planned. Direct integration with live CCTV feeds will enable continuous monitoring without manual uploads. The model can be upgraded to transformer-based architectures or 3D CNNs for richer spatiotemporal modelling. Multi-class classification will allow the system to flag additional event types such as theft, accidents, suspicious behaviour, and crowd anomalies. Cloud deployment and a companion mobile application will improve scalability and remote access, while richer alerting (SMS, email, location tracking) will tighten the loop with emergency services. Training on larger, more diverse datasets and porting inference to edge devices are further priorities for reducing latency and improving real-world robustness.

REFERENCES

[1] K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos,” in Advances in Neural Information Processing Systems (NIPS), 2014.

[2] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2015.

[3] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.

[5] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in Advances in Neural Information Processing Systems (NIPS), 2012.

[7] T. Hassner, Y. Itcher, and O. Kliper-Gross, “Violent Flows: Real-Time Detection of Violent Crowd Behavior,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops, 2012.

[8] F. Chollet, Deep Learning with Python. Manning Publications, 2017.

[9] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[10] PyTorch Documentation. [Online]. Available: https://pytorch.org/

[11] OpenCV Documentation. [Online]. Available: https://opencv.org/

[12] Streamlit Documentation. [Online]. Available: https://streamlit.io/

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

[13] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018.

[14] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes," Computer Vision and Image Understanding, 2018.