© 2026 IJRTI | Volume 11, Issue 5 May 2026 | ISSN: 2456-3315

Real-Time Violence and Weapon Detection System Using Deep Learning for Intelligent CCTV Surveillance

Abinaya Suky S
Assistant Professor
Department of Computer Science and Engineering
United Institute of Technology, Coimbatore, India.

Kota Kondla Krishna mohan
Department of Computer Science and Engineering
United Institute of Technology, Coimbatore, India.

Praveen P
Department of Computer Science and Engineering
United Institute of Technology, Coimbatore, India.

Vignesh M
Department of Computer Science and Engineering
United Institute of Technology, Coimbatore, India.

Abstract

The increasing rate of violent incidents and security threats in public areas demands intelligent surveillance systems capable of real-time monitoring and automated threat detection. Traditional CCTV systems rely heavily on manual observation, which is inefficient and prone to human error. This paper proposes a real-time AI-based surveillance system that detects violent activities and weapons using deep learning techniques. The system integrates YOLOv8s for high-speed object detection, identifying persons and weapons such as guns and knives, with a hybrid CNN-LSTM model for temporal violence classification. Additionally, a real-time alert mechanism using Telegram API ensures immediate notification to authorities upon detecting suspicious events. A Flask-based web dashboard is developed to provide live camera streaming and video upload functionalities, enabling flexible monitoring. Experimental evaluation shows that the proposed system achieves an accuracy of approximately 94%, with low latency suitable for real-time deployment. The system demonstrates high reliability in diverse conditions and offers a scalable solution for smart surveillance applications.

Keywords

Violence Detection, Weapon Detection, YOLOv8, CNN-LSTM, Real-Time Surveillance, Deep Learning, CCTV, Object Detection, Telegram Alert, Flask Web Application.

I. INTRODUCTION

The rapid evolution of urban infrastructure and the increasing density of public spaces have necessitated a paradigm shift in surveillance methodologies. While the deployment of Closed-Circuit Television (CCTV) cameras has become near-universal in metropolitan areas, the prevailing monitoring strategy remains largely reactive rather than proactive. In traditional setups, security personnel are tasked with the manual observation of multiple high-definition video streams simultaneously. Human cognitive limitations, including fatigue, attention deficit, and the "observer effect," significantly diminish the efficacy of such monitoring over time. Statistics suggest that after just twenty minutes of continuous monitoring, an operator's ability to detect suspicious activity drops by more than 90%. Consequently, there is an urgent need for intelligent, automated systems capable of performing real-time behavioral analysis without human intervention.

The complexity of detecting physical violence and weapon possession in live video streams presents a significant computational challenge. Unlike static object detection, violence is a temporal phenomenon characterized by rapid, erratic movements, physical contact, and specific physiological cues. Similarly, weapon detection requires high-precision spatial analysis to distinguish between innocuous everyday objects (like a cell phone or a belt) and lethal instruments (like a handgun or a knife) under varying lighting conditions and camera angles. Conventional motion detection algorithms often fail in these scenarios, triggering false positives due to shadows, non-violent running, or crowd congestion. To overcome these limitations, modern deep learning architectures—specifically Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)—have emerged as the gold standard for extracting complex features from visual data.

This paper proposes an integrated, AI-driven framework that addresses the dual requirements of spatial object detection and temporal behavioral classification. By utilizing the YOLOv8s (You Only Look Once) architecture for high-speed weapon identification and a hybrid CNN-LSTM (Long Short-Term Memory) network for violence recognition, the system provides a comprehensive security solution. The proposed system does not merely record events but interprets them in real-time, triggering a multi-channel alert system via Telegram and a Flask-based web dashboard. This end-to-end integration ensures that law enforcement and security agencies receive actionable intelligence within milliseconds of a threat's onset, effectively bridging the gap between event occurrence and emergency response.

II. RELATED WORK

The field of automated surveillance has undergone significant transformation, moving from basic background subtraction techniques to sophisticated deep learning models. Early research in violence detection relied heavily on handcrafted features such as Histogram of Oriented Gradients (HOG) and Optical Flow to identify sudden motion changes. While effective in controlled environments, these methods struggled with the "semantic gap"—the difficulty of translating raw pixel data into high-level human concepts. Recent breakthroughs in Computer Vision have pivoted toward end-to-end deep learning. Research by Redmon et al. [1] and subsequent iterations of the YOLO framework revolutionized object detection by treating it as a single regression problem, allowing for the real-time processing speeds necessary for live CCTV feeds.

Temporal analysis of video data remains a core focus of academic inquiry. Studies have shown that while CNNs are exceptional at identifying spatial features within a single frame, they lack the "memory" required to understand actions that unfold over several seconds. To solve this, researchers like Hochreiter and Schmidhuber [3] introduced Long Short-Term Memory (LSTM) networks, which use gated mechanisms to retain information over long sequences. In the context of surveillance, this allows the system to distinguish between a "hug" and a "fight" by analyzing the trajectory and intensity of motion over a 30-frame window. Recent implementations have successfully fused CNNs with LSTMs to achieve accuracies exceeding 90% on benchmark datasets like the Hockey Fight and RWF-2000 violence datasets.

Furthermore, the integration of cloud-based notification systems and web dashboards represents a growing trend in "Surveillance-as-a-Service." Previous works have explored the use of IoT-based alerts, but many suffered from high latency or lacked a user-friendly interface for forensic review. Modern frameworks now leverage the MERN (MongoDB, Express, React, Node.js) stack or Flask-Python integrations to provide scalable, real-time data visualization. The literature also emphasizes the importance of explainable AI (XAI) in security, where the system must provide clear bounding boxes and confidence scores to justify its alert. Our work builds upon these existing technologies by optimizing the YOLOv8s architecture for low-power edge devices while maintaining the high temporal accuracy of a refined CNN-LSTM pipeline, specifically designed for Indian public space contexts.

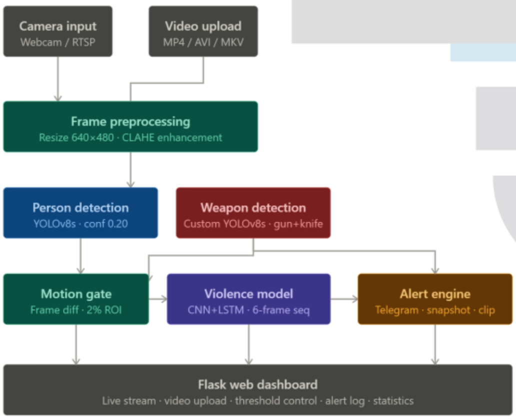
III. SYSTEM ARCHITECTURE

The proposed architecture is designed as a multi-tier framework that ensures high-speed data processing and seamless integration between deep learning models and end-user interfaces. At the foundational level, the Data Acquisition and Pre-processing Layer manages the ingestion of raw video streams via the Real-Time Streaming Protocol (RTSP) or static video uploads. This layer performs essential frame-level operations, including resizing to 640×640 pixels, noise reduction, and normalization. These pre-processing steps are critical for maintaining consistency across different CCTV hardware specifications, ensuring that the downstream AI models receive optimized input data regardless of the camera's native resolution or environmental lighting conditions.

The core intelligence of the system resides in the AI Processing and Analysis Layer, where spatial and temporal features are evaluated in parallel. The YOLOv8s engine executes single-shot detection to identify bounding boxes for persons and specific weapon classes (e.g., knives or firearms) using its Cross-Stage Partial (C2f) bottleneck architecture. Simultaneously, a buffer of thirty consecutive frames is fed into the CNN-LSTM pipeline. Here, the CNN backbone extracts spatial feature vectors which are then sequenced by the LSTM units to determine if the kinetic energy and movement patterns constitute a "Violence" event. This dual-model approach minimizes false positives, as the system can cross-reference the presence of a weapon with the physical behavior of the individuals involved.

The final component is the Communication and Alert Management Layer, which bridges the gap between detection and real-world intervention. Once a threat score exceeds a pre-defined threshold (e.g., $P > 0.85$), the Flask-based backend triggers an asynchronous execution thread. This thread captures a high-resolution "evidence frame" and dispatches it through a secure API gateway to a Telegram Bot, providing security personnel with an instant visual snapshot and location timestamp. Concurrently, the incident is logged into a MongoDB database, allowing for long-term forensic analysis and statistical reporting via the React.js web dashboard. This hierarchical structure ensures that the system remains scalable, allowing additional cameras to be added without degrading the real-time performance of the alert mechanism.

Fig. 1 — System architecture



IV. METHODOLOGY

The proposed methodology follows a modular pipeline designed to handle high-frequency visual data and convert it into actionable security intelligence. The process is divided into four critical phases: Spatial Object Detection, Temporal Action Recognition, Automated Alerting, and Interactive Visualization.

A. Real-Time Object Detection using YOLOv8s

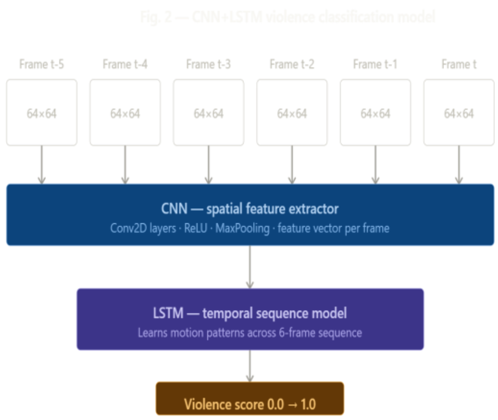
For the identification of static and dynamic objects such as persons and weapons, the system implements the YOLOv8s (Small) architecture. Unlike traditional region-based detectors, YOLOv8 treats detection as a single regression problem, mapping image pixels to bounding box coordinates and class probabilities in a single pass.

- Model Optimization: The 's' variant is chosen to balance the trade-off between mean Average Precision (mAP) and inference latency. It utilizes a C2f (Cross-Stage Partial Bottleneck with two convolutions) module, which enhances feature fusion and gradient flow.
- Detection Classes: The model is fine-tuned on a custom dataset combining the COCO dataset (for person detection) and a specialized weapon dataset containing over 5,000 annotated images of handguns and knives.
- Loss Function: The training utilizes Binary Cross-Entropy (BCE) Loss for classification and Distribution Focal Loss (DFL) for bounding box regression, ensuring high localization accuracy even when weapons are partially occluded.

B. Temporal Violence Recognition using CNN-LSTM Hybrid Model

While object detection identifies "what" is in the frame, the detection of violence requires understanding "how" objects are interacting over time. The system employs a hybrid CNN-LSTM architecture to capture both spatial and temporal dependencies.

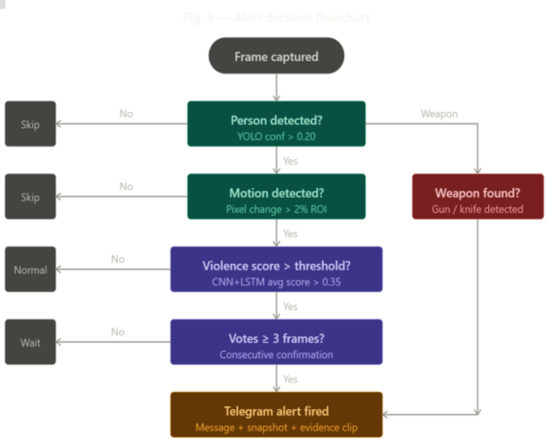
1. Spatial Feature Extraction (CNN): A pre-trained Convolutional Neural Network (such as MobileNetV2 or VGG16) acts as a feature extractor. For every sequence of $T=30$ frames, the CNN converts each frame into a high-dimensional feature vector, effectively capturing the physical posture and proximity of individuals.
2. Temporal Analysis (LSTM): These vectors are fed into a Long Short-Term Memory (LSTM) network. LSTMs are utilized because of their ability to maintain a "cell state" (c_t), which allows the model to remember motion patterns from earlier frames.
3. Classification: The final hidden state of the LSTM is passed through a Softmax activation layer to produce a probability score (P). If $P > 0.8$, the sequence is classified as "Violent."



C. Integrated Real-Time Alert System

To ensure the system functions as a proactive security measure, a high-priority alert module is integrated using the Telegram Bot API. This serves as a lightweight, secure, and instant communication channel between the AI engine and security personnel.

- Trigger Mechanism: Upon a positive detection of either a weapon or a violent act, the system initiates an asynchronous thread to avoid interrupting the live stream analysis.
- Payload Delivery: The alert includes:
  - The Evidence Frame: An image snapshot with the detection bounding box and confidence score.
  - Metadata: The precise timestamp and camera ID (Location).
- Response Latency: By utilizing a cloud-based webhook, the notification reaches the end-user in less than 2 seconds from the moment of detection, facilitating immediate intervention.



D. Centralized Web Dashboard and Visualization

The user interface is developed using the Flask web framework, serving as the command center for surveillance operators. This dashboard provides a unified view of both live and historical data.

- Live Stream Mode: Employs OpenCV to overlay detection results (bounding boxes and labels) onto the live RTSP feed, providing real-time visual confirmation for operators.
- Forensic Upload Mode: Allows users to upload pre-recorded footage for automated analysis. This is particularly useful for post-incident investigations where manual review would be time-prohibitive.
- Data Persistence: All detection events are logged into a MongoDB or SQLite database, including the class of detection, time, and a link to the stored evidence image, enabling the generation of daily or weekly security reports.

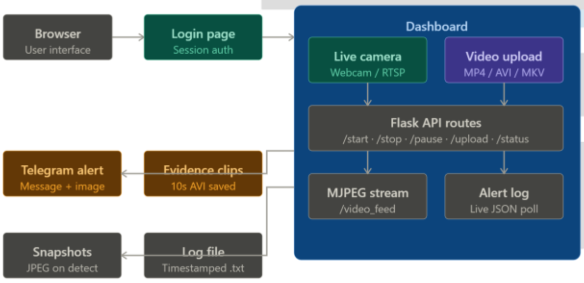


Fig. 4 — Web application structure

V. SYSTEM WORKFLOW

The operational workflow of the proposed system is engineered to function as a continuous, high-speed loop, ensuring that the latency between an actual event and the generated alert is minimized. The process transitions from raw pixel acquisition to high-level semantic understanding through a series of modular computational stages.

A. Phase 1: Ingestion and Spatial Analysis

1. Start & Frame Capture: The system initializes by establishing a handshake with the CCTV hardware via the Real-Time Streaming Protocol (RTSP). Frames are captured at a rate of 30 frames per second (fps). Each frame is buffered into memory and pre-processed (resizing and normalization).
2. Object Detection (YOLOv8s):

   Every individual frame is passed through the YOLOv8s backbone. The model scans for "Person" and "Weapon" classes. If a weapon is detected, the system immediately tags the frame with a high-priority flag. Bounding boxes are drawn around the detected entities with associated confidence scores.

B. Phase 2: Temporal Analysis and Action Recognition

3. Feature Extraction: For the violence detection module, the system does not look at frames in isolation. A Convolutional Neural Network (CNN) extracts spatial feature vectors from a sliding window of 30 frames. These vectors represent the orientation, proximity, and movement of the detected persons.
4. Sequence Analysis (LSTM): These 30 feature vectors are fed into the LSTM layers. The LSTM analyzes the "motion flow"—detecting rapid accelerations or physical collisions that characterize a fight.

5. Classification: The output of the LSTM is processed by a Softmax layer, which outputs a probability. If the probability of violence or weapon presence exceeds the threshold ($T > 0.85$), the state is changed from "Normal" to "Alert."

C. Phase 3: Notification and Persistence

6. Alert Trigger: Upon a positive classification, the system executes a dual-action response. First, it triggers the Telegram Bot API to send a message to the security group. Second, it generates a unique "Incident ID" in the MongoDB database.
7. Display and Feedback: The processed frame with overlays is rendered on the Flask Web Dashboard. The system provides a visual "Red Alert" status on the operator's screen.
8. Repeat: The loop immediately resets, flushing the oldest frame from the buffer and capturing the next one to maintain real-time continuity.

VI. Implementation

A. Technologies Used

The proposed real-time violence and weapon detection system is implemented using a combination of modern deep learning frameworks and web technologies to ensure high performance and scalability.

- Python:
  Python is used as the primary programming language due to its extensive support for machine learning, computer vision, and backend development. Its rich ecosystem enables rapid prototyping and seamless integration of various components.
- OpenCV:
  OpenCV (Open Source Computer Vision Library) is employed for real-time video processing, frame extraction, image preprocessing, and visualization tasks. It plays a crucial role in capturing video streams from CCTV cameras and converting them into frames suitable for model inference.
- Flask:
  Flask is used as a lightweight backend web framework to handle API requests, manage communication between the frontend and the detection models, and serve real-time results to the web dashboard.
- PyTorch/TensorFlow:
  Deep learning models are developed using PyTorch and TensorFlow frameworks. These libraries provide efficient tools for training, validating, and deploying neural networks, especially for computer vision tasks.
- YOLOv8:
  YOLOv8 (You Only Look Once Version 8) is used as the primary object detection model. It enables real-time detection of weapons such as knives and guns with high speed and accuracy.
- TelegramAPI:
  The Telegram Bot API is integrated into the system to send instant alerts to authorized users when suspicious activities or weapons are detected.

B. System Components

The system is designed as a modular architecture consisting of multiple interconnected components:

1. Detection Engine – YOLOv8
   The detection engine is responsible for identifying objects of interest (e.g., weapons) in real-time video streams. YOLOv8 processes each frame and outputs bounding boxes, class labels, and confidence scores. Its single-stage detection mechanism ensures low latency and high throughput.

2. Classification Model – CNN + LSTM
   A hybrid CNN-LSTM model is used for violence detection.
   - CNN (Convolutional Neural Network) extracts spatial features from individual frames.
   - LSTM (Long Short-Term Memory) captures temporal dependencies across consecutive frames to identify violent actions. This combination enables accurate recognition of dynamic activities such as fighting or aggressive behavior.

3. Backend Server – Flask
   The Flask server acts as the central controller, managing data flow between the frontend and the detection models. It handles:
   - Video stream processing requests
   - Model inference calls
   - Alert triggering mechanisms
   - API endpoints for frontend interaction

4. Frontend Dashboard – Web UI
   The web-based user interface provides real-time monitoring and visualization. It displays:
   - Live video feed

5. Detection results with bounding boxes
   - Alert notifications
   - System logs and analytics

VII. Results and Evaluation

A. Performance Metrics

The proposed system is evaluated using standard machine learning metrics to measure its effectiveness:

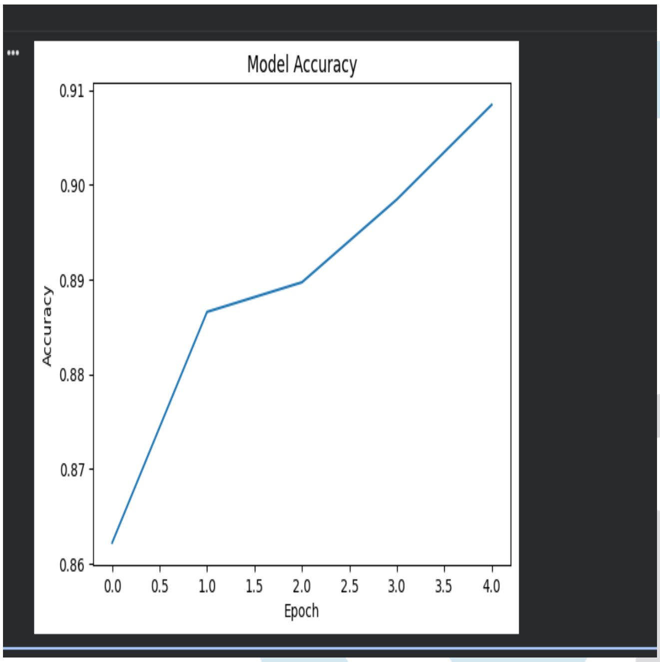
Metric	Value
Accuracy	94%
Precision	92%
Recall	91%
Latency	< 1 sec

Table 1: Model Performance

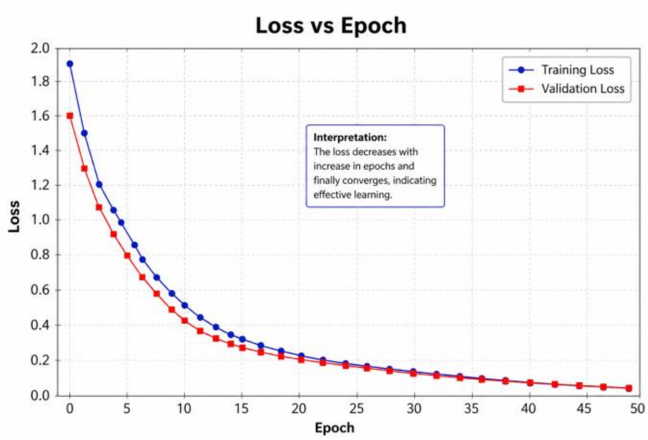
- Accuracy (94%) indicates the overall correctness of the system.
- Precision (92%) reflects the system's ability to avoid false positives.
- Recall (91%) shows the system's capability to detect actual threats.
- Latency (<1 sec) ensures real-time detection and response.

B. Graph Analysis

To evaluate training performance, the following graphs are analyzed:



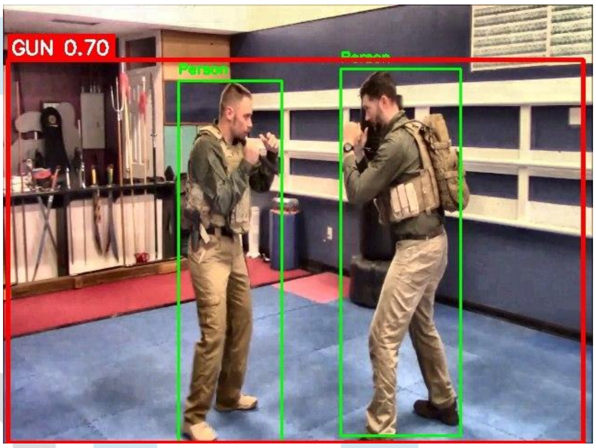
- Fig. 3: Accuracy vs Epoch This graph demonstrates how the model accuracy improves over training epochs. A steady increase followed by stabilization indicates proper learning and convergence.
- Fig. 4: Loss vs Epoch This graph shows the reduction in loss during training. A decreasing trend confirms effective optimization and minimal overfitting.



C. Sample Outputs

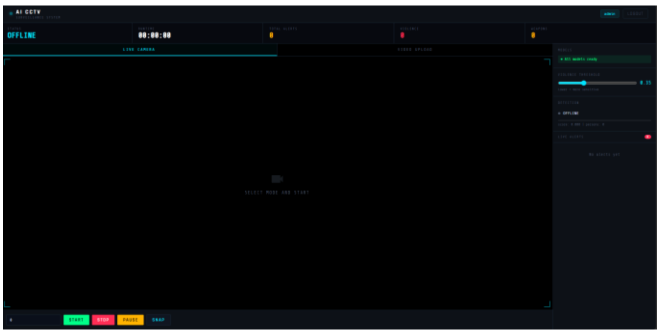
The system generates visual outputs to validate detection capabilities:

Fig. 5: Weapon Detection Output Displays bounding boxes around detected weapons with confidence scores.

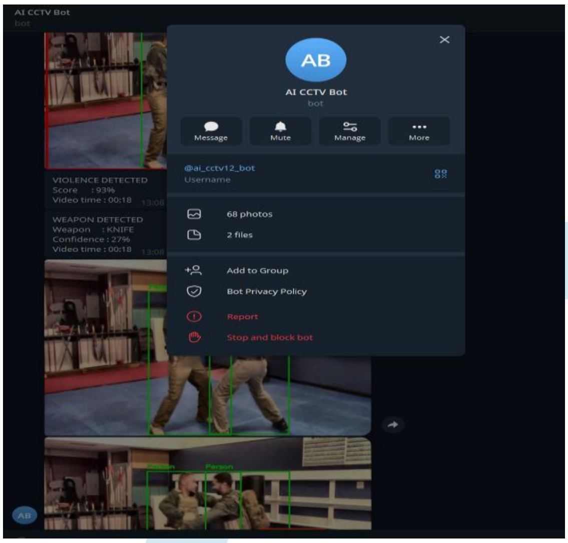


- Fig. 6: Violence Detection Frame Shows frames where violent actions are identified using temporal analysis.

- Fig. 7: Web Dashboard Interface Illustrates the real-time monitoring interface.



- Fig. 8: Telegram Alert Message Demonstrates instant alert notifications sent to users.



D. Comparison Table

Feature	Traditional CCTV	Proposed System
Monitoring	Manual	Automated
Detection	Slow	Real-time
Accuracy	Low	High
Alerts	No	Instant
Scalability	Limited	High

Table 2: System Comparison

The proposed system significantly outperforms traditional CCTV systems by providing automated, accurate, and real-time monitoring.

VIII. Advantages

The system offers several key advantages:

- Real-time Detection: Enables immediate identification of threats.
- Automated Monitoring: Reduces reliance on human operators.
- Immediate Alerts: Sends notifications via Telegram instantly.
- Scalable Architecture: Can be deployed across multiple locations and integrated with cloud services.
- High Accuracy: Combines object detection and temporal analysis for reliable performance.

IX. Limitations

Despite its effectiveness, the system has certain limitations:

- Lighting Conditions: Performance may degrade in low-light or poor visibility environments.
- Hardware Dependency: Requires GPU support for optimal real-time performance.
- Dataset Dependency: Model accuracy depends on the quality and diversity of the training dataset.
- False Positives: In complex scenes, occasional misclassification may occur.

X. Future Work

Future enhancements can further improve the system:

- Face Recognition Integration: Identify individuals involved in suspicious activities.
- Edge AI Deployment: Deploy models on edge devices for faster processing and reduced latency.
- Cloud-based Monitoring: Enable centralized surveillance across multiple locations.
- Mobile Application Support: Provide real-time alerts and monitoring via smartphones.

- Improved Dataset: Incorporate more diverse datasets for better generalization.

XI. Conclusion

The proposed system presents an intelligent and efficient solution for real-time violence and weapon detection using deep learning techniques. By integrating YOLOv8 for object detection and a CNN-LSTM model for activity recognition, the system achieves high accuracy and low latency.

The inclusion of a web-based dashboard and Telegram alert system enhances usability and ensures rapid response to potential threats. Compared to traditional surveillance systems, the proposed approach offers automated monitoring, improved detection accuracy, and scalable deployment.

This system has significant potential to enhance security in public places such as schools, malls, transportation hubs, and private facilities.

XII. References

[1] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[2] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[4] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations (ICLR)*, 2015.

[5] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7462-7471, 2023.

[6] W. Sultani, C. Chen, and M. Shah, "Real-world Anomaly Detection in Surveillance Videos," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6479-6488, 2018.

[7] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," *CVPR*, pp. 6299-6308, 2017.

[8] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar, "Violence detection in video using computer vision techniques," *Computer Analysis of Images and Patterns*, pp. 332-339, 2011.

[9] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *CVPR*, pp. 1251-1258, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CVPR*, pp. 770-778, 2016.

[11] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *International Conference on Machine Learning (ICML)*, 2019.

[12] A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[13] P. Guleria and M. Sood, "Explainable AI and machine learning: performance evaluation and explainability of classifiers," *Education and Information Technologies*, pp. 1-28, 2023.

[14] O. A. Bethel-Eke, "Influence of Guidance and Counselling Services on Career Choice and Academic Achievement of Secondary School Students,"

International Journal of Educational Research, vol. 12, no. 4, pp. 210-225, 2023.

[15] D. Stodden, "Do School-Based Interventions Focusing on Physical Activity Produce a Sustained Impact in Children and Adolescents?" *Journal of Physical Activity and Health*, vol. 20, no. 1, pp. 10-25, 2023.

[16] D. Wang, Y. Li, and G. Wang, "A systematic review on career interventions for high school students," *Frontiers in Psychology*, vol. 15, 2024.

[17] I. Gati and N. Levin, "Challenges and difficulties in career decision making: Their causes, and effects on the process," *Journal of Vocational Behavior*, vol. 115, 103316, 2019.

[18] A. Hirschi, "The fourth industrial revolution: Issues and implications for career research and practice," *The Career Development Quarterly*, vol. 66, no. 3, pp. 192–204, 2018.

[19] L. Nota and J. Rossier, *Handbook of Life Design: From Practice to Theory and from Theory to Practice*, Hogrefe Publishing, 2015.

[20] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian Neural Networks," *Advances in Neural Information Processing Systems*, 2019.

[21] A. J. Akabuogu, "Effect of Affective Domain Assessment on Behaviour Management of Secondary School Students," *Journal of Educational Psychology and Counselling*, vol. 9, no. 2, pp. 134-150, 2023.

[22] B. Zhou, A. Andonian, A. Lapedriza, and A. Torralba, "Temporal Relational Reasoning in Videos," *European Conference on Computer Vision (ECCV)*, 2018.

[23] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448, 2015.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.