Project Report:•   Real-Time Violence Detection in CCTV Surveillance
Domain: Deep Learning / Computer Vision / Video Analytics
Application: Public Safety, Smart Surveillance, Crime Prevention
Level: B.Tech Final Year (Moderate to Advanced Complexity)

---

1. Executive Summary

With the increasing deployment of CCTV cameras in public and private environments, continuous manual monitoring of surveillance footage has become inefficient and unreliable. Human operators are prone to fatigue, delayed attention, and errors, especially when monitoring multiple camera feeds simultaneously. As a result, violent activities such as fights, assaults, robberies, and vandalism may go unnoticed until after the incident has occurred. Traditional CCTV systems primarily support post-event analysis and lack intelligent real-time decision-making capabilities.

To address these challenges, this project presents an **Intelligent CCTV-Based Violence & Weapon Detection System**, designed to automatically analyze live surveillance video streams and detect violent activities and weapon presence in near real time. The system captures frames from CCTV feeds and processes them through a structured deep learning pipeline. Initially, humans and potential weapons are detected using a lightweight YOLO-based object detection model to ensure fast inference.

Detected human regions are then passed through spatial feature extractors based on Convolutional Neural Networks (CNNs). Temporal patterns of motion are analyzed using sequence models such as LSTM or BiLSTM to recognize violent behaviors including fighting, assault, robbery, and vandalism. In parallel, weapon detection focuses on identifying firearms and edged weapons. Upon detecting a threat, the system generates alerts, logs event details, and visualizes results on a web-based dashboard. This approach enables proactive surveillance, faster response, and improved public safety.

---

2. Project Objectives

● **Automated Violence Detection:** Identify violent activities such as fighting, assault, robbery, and vandalism from CCTV footage
● **Weapon Detection:** Detect firearms and edged weapons in real-time surveillance scenes
● **Human-Centric Analysis:** Focus analysis on detected persons to reduce background noise
● **Temporal Motion Analysis:** Capture sequential movement patterns using LSTM/BiLSTM models
● **Real-Time Alert Generation:** Provide near real-time alerts for detected threats
● **Efficient Processing:** Ensure low-latency inference suitable for continuous surveillance
● **Incident Logging:** Store detected events for further investigation and analysis

---

3. Datasets & Models (With Access Links)

3.1 Open-Source Datasets

The following fully open-source datasets are used for training, fine-tuning, and evaluation. Large-scale datasets are used through **class-based filtering and subset selection** to ensure computational feasibility.

Dataset Name	Purpose	Access Link
Open Images V7 (Filtered)	Firearm detection (handgun, rifle, shotgun)	https://storage.googleapis.com/openimages/web/index.html
OD-WeaponDetection	Weapon detection fine-tuning	https://datasetninja.com/weapon-detection
Kaggle Weapons Dataset	Knife, dagger, sword detection	https://www.kaggle.com/datasets
SCUT-WMD	Small and occluded weapon detection	https://github.com/SCUT-BIP-Lab/SCUT-WMD
UCF-Crime (Subset)	Violence and anomaly detection benchmarking	https://www.crcv.ucf.edu/projects/real-world/
Real Life Violence Dataset	Primary violence detection training	https://www.kaggle.com/datasets
RWF-2000	Real-time fight detection validation	https://github.com/mchengny/RWF2000-Video-Database

✔ Publicly available for academic use
✔ Widely cited in surveillance and action recognition research
✔ Suitable for student-level projects

3.2 Pre-Trained Models & Frameworks

All tools and models used are fully open-source and research-proven.

Model / Framework	Task	Access / Documentation
YOLOv8n	Person and weapon detection	https://github.com/ultralytics/ultralytics
CNN (ResNet / MobileNet)	Spatial feature extraction	PyTorch / Torchvision
LSTM / BiLSTM	Temporal violence recognition	PyTorch
ArcFace (Pretrained)	Face embedding extraction (optional verification)	https://github.com/deepinsight/insightface
OpenCV	Video processing & frame extraction	https://opencv.org/

✔ Open-source and academically accepted
✔ Compatible with real-time deep learning pipelines
✔ Suitable for deployment on standard hardware

---

4. Methodology & Key Visual Indicators

4.1 Visual Features Analyzed

Feature	What the System Analyzes	Importance
Human Presence	Bounding boxes and motion	Focused analysis
Body Motion	Sudden and aggressive movements	Violence detection
Temporal Change	Movement consistency over time	Action recognition
Weapon Appearance	Shape and context	Threat identification
Interaction Patterns	Person-to-person contact	Fight and assault detection

---

5. System Architecture & Enhanced Workflow

5.1 High-Level Architecture

The system follows a **client–server architecture**. Live CCTV streams are processed by the backend server, which performs detection and classification. Results are returned to the frontend dashboard and stored in a database.

5.2 Detailed Core Logic Workflow (Backend)

Step 1: Video Acquisition & Preprocessing
● Frames captured from CCTV or RTSP streams
● Resizing, normalization, and noise reduction applied

Step 2: Human & Weapon Detection (YOLOv8n)
● Persons and weapons detected in each frame
● Regions of interest extracted

Step 3: Spatial Feature Extraction (CNN)
● Deep features extracted from human ROIs
● Reduces irrelevant background information

Step 4: Temporal Feature Modeling (LSTM / BiLSTM)
● Sequential frame features analyzed
● Violent motion patterns recognized

Step 5: Threat Classification
● Violence category predicted with confidence score
● Weapon presence validated

Step 6: Alert Generation & Logging
● Alerts triggered for detected threats
● Event metadata stored in database

5.3 Detailed Frontend Workflow

Step 1: Live CCTV Visualization
● Video streams displayed in dashboard
● Multiple feeds supported

Step 2: Alert & Event Monitoring
● Detected threats highlighted in real time
● Event history and timestamps shown

---

6. Use Cases

● Public surveillance and safety monitoring
● Smart city security systems
● Campus and office security

● Crime prevention and early intervention
● Post-incident analysis and reporting

---

7. Conclusion

The Intelligent CCTV-Based Violence & Weapon Detection System demonstrates the effective application of deep learning and computer vision techniques for proactive surveillance. By combining YOLO-based object detection, CNN-based spatial feature extraction, and LSTM-based temporal modeling, the system achieves reliable detection of violent activities and weapon threats in real time. The use of selective dataset filtering and open-source frameworks ensures scalability, academic validity, and feasibility for student-level deployment. This project highlights the potential of intelligent surveillance systems in enhancing public safety and security infrastructure.

---

8. Hardware & System Requirements

The proposed system is designed to operate efficiently on standard student-level hardware.

8.1 Laptop / System Requirements

Minimum System Requirements
● Processor: Intel Core i5 (6th Gen+) or AMD Ryzen 5
● RAM: 8 GB
● Storage: 120 GB free space
● Graphics: Integrated graphics
● OS: Windows 10/11 or Ubuntu 20.04+
● Python: Version 3.8 or higher

Recommended System Requirements
● Processor: Intel Core i7 (8th Gen+) or AMD Ryzen 7
● RAM: 16 GB or more
● Storage: SSD with 150 GB free space
● Graphics (Optional): NVIDIA GTX 1650 or higher

8.2 Camera Requirements

Minimum Requirements
● Resolution: 720p
● Frame Rate: 20–30 FPS

Recommended Requirements
● Resolution: 1080p

● Frame Rate: 30 FPS
● Good low-light performance

8.3 Network Requirements

● Internet required only for dataset download
● Local deployment supported without internet

8.4 Overall Feasibility

✔ Suitable for B.Tech final-year implementation
✔ Real-time performance achievable
✔ No specialized hardware required
