WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real-Time Violence Detection and Intelligent Alarm Systems

Majji Ankitha [1], Jyothsna Kondavalasa[2], Mandula Nagesh[3], Lenka Sai Viroop[4], Meesala Roshan[5], Alajangi Sai Sriram[6]

(anki.majji@gmail.com) ,
(22341A0590@gmrit.edu.in) ,
(22341A05A5@gmrit.edu.in),
(22341A0598@gmrit.edu.in),
(22341A05A8@gmrit.edu.in),
(20341A0506@gmrit.edu.in)

ABSTRACT :

In today's world, the need for efficient and proactive safety measures has become increasingly critical. This project " Real-Time Violence Detection and Intelligent Alarm Systems" designed to identify violent behaviors such as fights and assaults in real-time using video footage from CCTV cameras or uploaded files. By leveraging advanced computer vision techniques, the system enhances traditional surveillance with automated monitoring and immediate alert capabilities. The system employs deep learning techniques for analyzing action sequences, enabling precise recognition of aggressive behavior patterns. Trained on a diverse dataset of violent and non-violent scenarios, the model achieves high accuracy while minimizing false positives. When violence is detected, the system promptly sends SMS notifications to security personnel or designated individuals, accompanied by critical details to facilitate swift intervention.

This solution is scalable and versatile, suitable for various settings such as schools, malls, workplaces, and public transportation hubs. It seamlessly integrates with existing surveillance infrastructures and features a user-friendly interface for effortless configuration and monitoring. By combining the power of deep learning with real-time surveillance, this system provides a practical approach to enhancing security and reducing reliance on manual monitoring. It ensures faster responses to critical incidents, thereby fostering safer environments.

Keywords: *Violence detection, Real-Time Surveillance, Deep Learning, Computer Vision, Automated Alerts, Safety Enhancement.*

2.INTRODUCTION

Ensuring public safety has become a critical priority in today's rapidly evolving world, especially with the increasing number of violent incidents in public spaces such as schools, transportation hubs, shopping centers, and urban streets. To tackle these challenges, surveillance systems have become widespread; however, traditional methods relying on human operators to monitor video feeds are proving to be inefficient. Continuous human supervision is not only time-consuming and expensive but also prone to fatigue and human error, often resulting in missed or delayed detection of violent activities.
With advancements in Artificial Intelligence (AI) and deep learning, there is a growing shift toward intelligent surveillance systems that can automatically detect abnormal or violent behavior in real-time. These systems aim to minimize human intervention, enhance the speed and accuracy of threat detection, and enable proactive responses to potentially dangerous situations.
This project introduces a Real-Time Violence Detection and Intelligent Alarm System, which leverages deep learning models for the automatic identification of violent actions in live video streams. YOLOv5, a state-of-the-art object detection model, is used to detect and localize people within video frames quickly and accurately. To ensure that the system remains lightweight and responsive, MobileNet is used for efficient feature extraction. These features are then passed through a 3D Convolutional Neural Network (3D CNN**)** that captures spatial relationships across sequences of frames. To recognize temporal patterns and motion dynamics indicative of violence, the system incorporates a Long Short-Term Memory (LSTM) network, capable of learning time-dependent behavior patterns.
This hybrid architecture enables the system to detect violence accurately while maintaining real-time processing capabilities, making it suitable for deployment in real-world surveillance environments. Additionally, the system is designed to trigger intelligent alarms the moment a violent event is detected, allowing for immediate notification and faster intervention by security personnel or emergency responders.
By reducing reliance on manual monitoring and enhancing detection precision, this project aims to contribute significantly to public safety initiatives. It provides a foundation for next-generation surveillance systems capable of responding to threats proactively and efficiently in real time.

3.Methodology

Violence detection in real-time surveillance videos plays a crucial role in enhancing public safety and preventing criminal activities. The primary goal of this project was to develop a deep learning-based solution that can automatically detect violent scenes from video footage and raise alerts in a timely manner. Over the course of development, multiple model architectures were explored to optimize both the detection accuracy and computational efficiency. The project evolved through three distinct modeling phases: starting with a VGG16 + LSTM model, transitioning to a GRU + LSTM hybrid, and finally adopting a more robust and efficient solution using YOLOv5 + MobileNet-styled 3D-CNN + LSTM, which was finalized due to its superior performance in accuracy and real-time applicability.

3.1 Dataset Overview

For training and evaluating the violence detection system, we utilized three diverse and widely accepted datasets that simulate real-world violent and non-violent scenarios:

Hockey Fight Dataset:
This dataset contains short video clips extracted from hockey matches, labeled as either "fight" or "non-fight." The fight scenes typically involve aggressive physical contact, making this dataset ideal for initial violence detection model training due to its clear action boundaries and consistent camera angles.

- Total Clips: ~1000
- Resolution: 320×240 pixels
- Labels: Binary (Fight, Non-Fight)

fig-2: Hockey dataset sample 1

fig-3: Hockey dataset sample 2

Peliculas Dataset (Movie Fights):
This dataset comprises fight scenes extracted from various movies. Unlike the Hockey Fight dataset, it includes a mix of camera angles, lighting conditions, and cinematic effects. It adds complexity and variety to the training set, challenging the model to generalize across different visual environments.

- Total Clips: ~200
- Types: Punching, kicking, shooting, and dramatic sequences
- Labels: Binary (Violent, Non-Violent)

fig-4: Peliculas dataset sample 1

fig-5: Peliculas dataset sample 2 1

Real-Life Violence Situations Dataset:

This dataset includes real-world surveillance camera footage from public places. It presents a more realistic challenge due to variations in camera stability, frame rate, resolution, and unpredictable actions. The dataset captures diverse environmental conditions such as lighting changes, occlusions, and crowd density, making it ideal for testing the robustness of action recognition models. It also includes a wide range of human behaviors and interactions, providing valuable data for training systems to detect complex and subtle activities in real-world scenarios.

- Total Clips: ~1000
- Source: Real-life security feeds
- Labels: Binary (Violent, Non-Violent)

fig-6: RLVS dataset sample 1

fig-7: RLVS dataset sample 2

3.2 Data Preprocessing:

All datasets were preprocessed to:

1. **Frame Extraction at Fixed Intervals**

To ensure uniformity across video clips, frames are extracted at consistent time intervals. This method not only helps maintain the relevant visual information but also reduces redundancy by avoiding the extraction of unnecessary frames. By capturing frames at regular intervals, the system can focus on the essential content without being overwhelmed by repetitive data, making the video analysis more efficient and effective.

2. **Frame Resizing to Standard Dimensions**

To ensure compatibility with deep learning models and reduce computational complexity, each extracted frame is resized to a predefined resolution, such as 224x224 or 112x112 pixels. This step standardizes the input size across all frames, making them suitable for processing by neural networks. Additionally, resizing the frames helps optimize performance by reducing the amount of data the model needs to process, thus improving computational efficiency without compromising the essential features needed for accurate analysis.

3. **Pixel Value Normalization**

Adjusting pixel values to a standardized range, typically between 0 and 1 or -1 to 1, helps improve model training efficiency. This normalization process ensures that the input data has a consistent distribution, preventing large pixel values from disproportionately influencing the model's learning process. By bringing all pixel values within a standardized range, the model can converge faster and achieve better performance, as it prevents issues related to differing scales in the input data.

4. **Data Balancing for Model Fairness**

Maintaining an equal distribution of violent and non-violent clips is crucial to prevent bias in model predictions. This balanced dataset ensures that the model is trained on both categories equally, avoiding the tendency to favor one over the other. By ensuring this balance, the model's generalization capability is enhanced, leading to more accurate and fair classification. This approach helps the model make reliable predictions, even when faced with new, unseen data, ensuring that both violent and non-violent events are identified accurately.

5. **Conversion of Video Clips to Sequential Frame Data**

Transforming video clips into ordered sequences of frames is essential for making them suitable for models like LSTMs and 3D CNNs. This process enables the model to analyze the video in a structured way, capturing the temporal dependencies between frames. By organizing the frames in sequence, the model can effectively interpret motion patterns and events over time, facilitating accurate recognition of actions and behaviors in the video. This is crucial for tasks like violence detection, where understanding the sequence and flow of actions is key to making correct predictions.

3.5 PROPOSED SYSTEM:

YOLOv5 + MobileNet-styled 3D-CNN + LSTM

After experimenting with two previous models, we finalized a more advanced and powerful architecture that combines YOLOv5, a lightweight 3D-CNN inspired by MobileNet, and LSTM. This hybrid model gave us the best performance in terms of speed, accuracy, and real-life applicability.

3.5.1 Why This Model Was Chosen:

YOLOv5 provides fast and accurate object detection in each frame, helping the model focus on relevant regions of interest. A MobileNet-styled 3D-CNN, which is both efficient and lightweight, processes both spatial and temporal information from the videos, capturing space and time-based features. The use of LSTM allows the model to capture long-term changes and patterns across action sequences. This approach addressed the challenges faced in earlier models, such as slow processing, poor accuracy in noisy backgrounds, and weak pattern detection. By employing YOLOv5 for real-time object detection, the model is able to concentrate on key areas of the frames, improving both performance and robustness.

3.5.2 Working Mechanism:

YOLOv5 detects objects such as humans, weapons, or suspicious movements within the video frames. Bounding boxes are drawn around the detected objects, and only these regions are passed forward for further processing. The model's lightweight design and fast inference time make it well-suited for real-time applications. By focusing on relevant objects and reducing unnecessary data, the object detection step significantly minimized noise, ensuring that irrelevant parts of the scene did not influence the performance of the classification model.

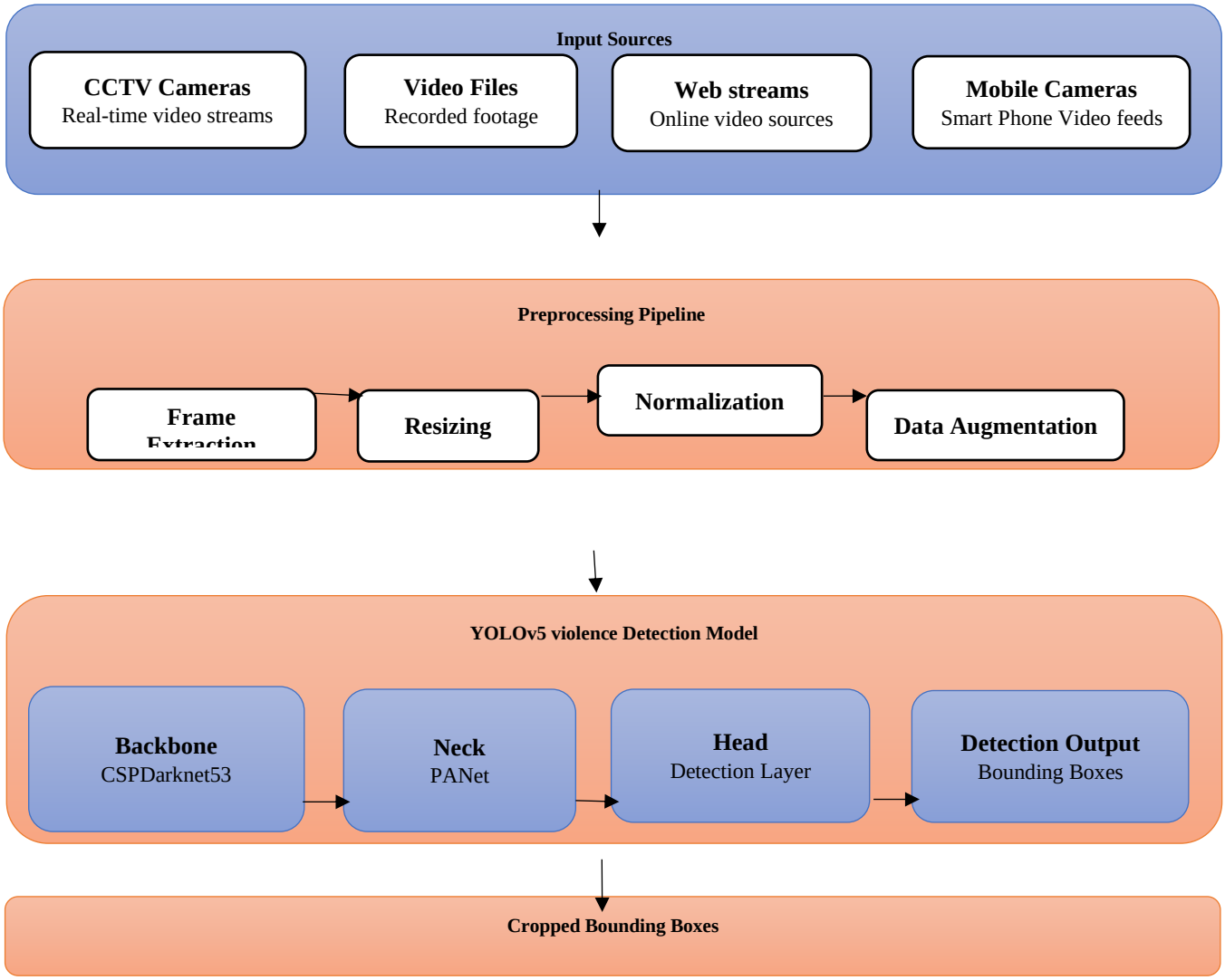

Fig-12: working of YOLO

The architecture illustrated above outlines a YOLOv5-based pipeline for violence detection in video streams. It is composed of four primary stages: Input Sources, Preprocessing Pipeline, YOLOv5 Violence Detection Model, and Cropped Bounding Boxes Output.

Input Sources:

The system accepts video input from a variety of sources including:

- CCTV Cameras: Real-time surveillance footage.
- Video Files: Pre-recorded content stored locally.
- Web Streams: Live feeds from online platforms.
- Mobile Cameras: Video captured through smartphones.

Preprocessing Pipeline:

Before feeding data into the detection model, each video source undergoes preprocessing to ensure consistency and improve model accuracy.

The video processing pipeline begins with frame extraction, where the video is divided into individual frames for analysis. These frames are then resized to meet the input requirements of the model. Next, normalization is applied to scale the pixel values to a standard range, ensuring consistent input for the model. To enhance dataset diversity and improve robustness, data augmentation techniques such as flipping and rotation are applied, allowing the model to generalize better to various variations in the video data.

YOLOv5 Violence Detection Model:

The core detection module is built on the YOLOv5 architecture, which consists of several key components. The backbone, CSPDarknet53, is responsible for extracting essential visual features from the input frames. The neck, PANet, aggregates features from various layers to improve both spatial and contextual understanding. The head, which includes the detection layer, predicts the locations of objects, class scores, and confidence values. The model is specifically trained to detect violence-related activities, such as fighting, weapon presence, and aggressive gestures. The detection output includes bounding boxes along with the corresponding class labels and confidence scores, providing precise localization and classification of violent events.

Cropped Bounding Boxes:

Finally, frames containing detected violent activities are highlighted and the relevant bounding boxes are cropped for further analysis, visualization, or alert generation.

Feature Extraction and Temporal Modeling:

Violence often involves sudden and aggressive motion, which requires modeling of both spatial and temporal features.

To achieve this, we designed a hybrid deep learning architecture consisting of:

a. MobileNet (Spatial Feature Extractor)

A lightweight CNN pre-trained on ImageNet was utilized to extract meaningful features from each video frame. MobileNet was chosen for this task because of its efficiency, compact size, and fast inference capabilities, making it well-suited for real-time applications. Each frame in the input sequence was processed through MobileNet to generate a 2D feature map, which served as a compact yet informative representation for subsequent stages of the model.

b. 3D Convolutional Neural Network (3D-CNN)

The extracted feature maps were stacked along the time dimension and fed into a 3D-CNN, allowing the model to process spatiotemporal information. The 3D convolution layer captures both motion and spatial continuity between consecutive frames, enabling the network to learn patterns of movement, such as punches, kicks, or sudden aggressive actions. This approach helps the model distinguish dynamic actions that are essential for classifying violent behaviour.

3.5.3 MOBILENET STYLED 3D CNN

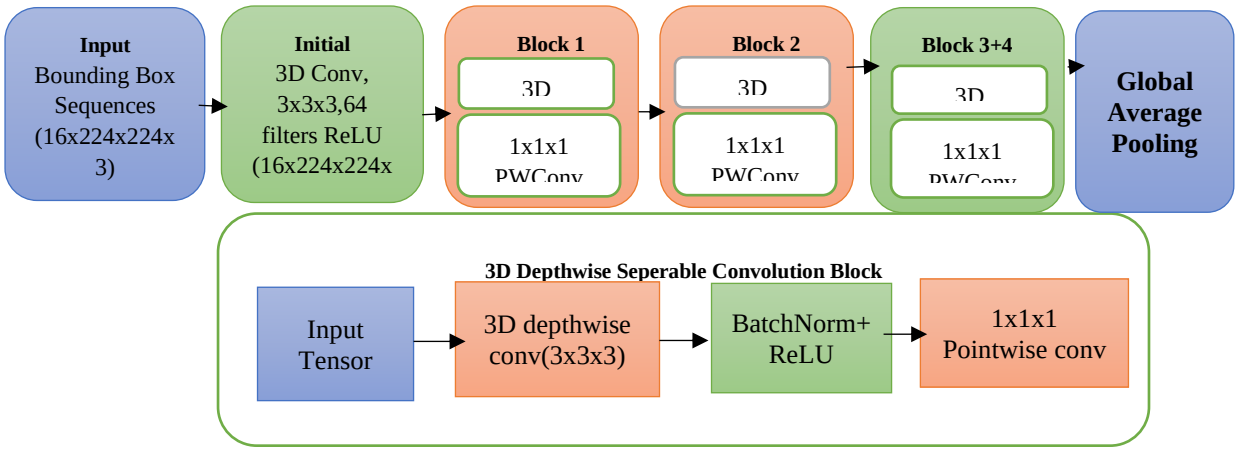

fig-13: Workflow of MobileNet styled 3dCNN

The diagram illustrates a lightweight 3D convolutional neural network (3DCNN) architecture inspired by the MobileNet design, specifically optimized for spatiotemporal feature extraction from bounding box sequences. This architecture is particularly suited for real-time violence detection tasks on edge devices due to its computational efficiency and compact structure.

1. **Input:** The model takes a sequence of bounding box crops as input with dimensions (16 × 224 × 224 × 3), representing 16 video frames of RGB image patches.
2. **Initial Layer:** A standard 3D convolutional layer with a kernel size of 3×3×3 and 64 filters is applied, followed by ReLU activation and Batch Normalization. This layer captures low-level spatiotemporal patterns.
3. **Depthwise Separable Convolution Blocks:** The architecture employs 3D Depthwise Separable Convolution Blocks across multiple stages (Blocks 1–4) to reduce computational overhead:

   The model architecture begins with Block 1, which applies a 3D depthwise convolution (DWConv) followed by a 1×1×1 pointwise convolution (PWConv), resulting in 128 feature channels. Block 2 follows the same structure but increases the number of channels to 256,

enhancing the model's capacity to capture more complex patterns. Blocks 3 and 4 consist of two sequential DWConv and PWConv stages, further expanding the feature representation to 512 channels, thereby enabling deeper and more refined feature extraction.

4. **MobileNet-Styled 3D Block Details**
   Each block is composed of three main components designed to efficiently process spatiotemporal features. First, a 3×3×3 depthwise convolution is applied independently to each channel, capturing localized patterns within the spatial and temporal dimensions. This is followed by batch normalization and a ReLU6 activation function to stabilize training and introduce non-linearity. Finally, a 1×1×1 pointwise convolution is used to combine the features across channels, enabling effective interaction between different feature maps

3.5.4 Long Short-Term Memory (LSTM)

LSTM was employed to model the temporal dependencies across sequences of video frames, allowing the system to understand the context and progression of activities over time. It processes the sequential feature vectors generated by the 3D-CNN, enabling the model to distinguish between normal and violent actions, even when they appear visually similar. The architecture consists of three stacked LSTM layers: the first two layers each contain 256 units and return sequences, while the third layer has 128 units and returns only the final state. This is followed by a fully connected layer with 1024 units and a dropout rate of 0.5 for regularization. Finally, a softmax-activated output layer is used for classifying the input sequence as violent or non-violent.

The bottom section of the diagram illustrates the internal structure of an LSTM cell, highlighting its key components and their functions. The forget gate controls which information should be discarded from the cell state, while the input gate determines what new information should be stored. The tanh layer generates candidate values to be potentially added to the cell state, which serves as the long-term memory component of the LSTM. Finally, the output gate decides which parts of the cell state should be output at each time step

The architecture illustrated above details the Long Short-Term Memory (LSTM) network used for modelling temporal dependencies in video-based violence detection. This module processes sequential feature vectors extracted from the 3D CNN and captures time-based patterns in the data.

1. **Input Features**
   The input to the LSTM network consists of feature vectors generated by the preceding 3D CNN (e.g., MobileNet-Styled 3DCNN), which encapsulate spatial and motion information across time.

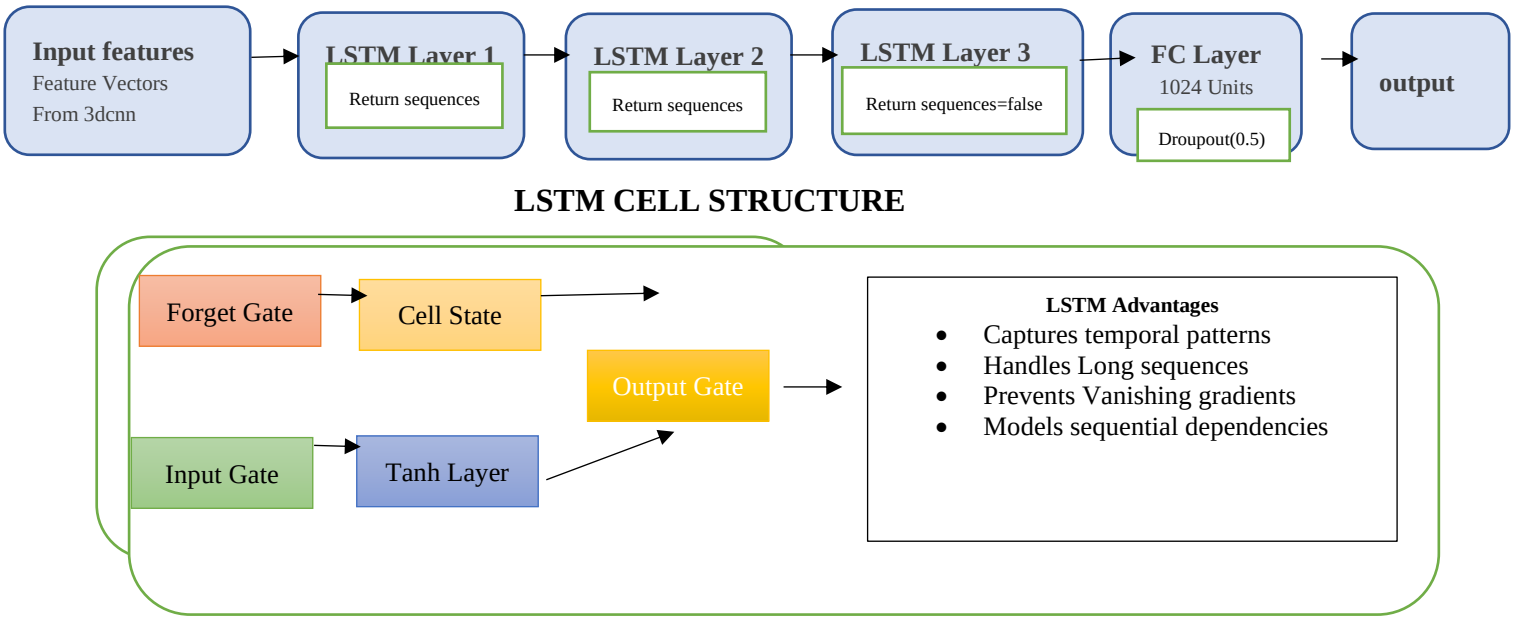

fig-14: Workflow of LSTM

2. **LSTM Layers**
   A stacked LSTM configuration is employed to handle sequential data with increasing abstraction:
   - LSTM Layer 1: Contains 256 units and returns sequences to pass the full output to the next layer.
   - LSTM Layer 2: Also has 256 units and continues passing sequential data.
   - LSTM Layer 3: Contains 128 units and only returns the final hidden state (Return Sequences=False), serving as the summarized temporal representation.
3. **Fully Connected (FC) Layer**
   A dense layer with 1024 units is used to map the LSTM output to the final classification space. A dropout layer with a dropout rate of 0.5 is included to prevent overfitting.

LSTM Cell Structure

The diagram also illustrates the internal components of an LSTM cell, highlighting how it manages information flow. The forget gate is responsible for deciding which information should be discarded from the cell state. The input gate, in combination with a tanh layer, determines what new information should be stored. The output gate then controls which part of the cell state should be output at each time step. Central to this structure is the cell state itself, which maintains memory across time steps and enables the network to capture long-term dependencies effectively.

3.6 Model Compilation and Training

3.6.1Model Details:

The model takes as input a sequence of 16 consecutive video frames and outputs a binary classification indicating whether the sequence is violent or non-violent. To train the model, Binary Cross-Entropy was used as the loss function, and the Adam optimizer was employed for efficient gradient updates. The model's performance was evaluated using standard metrics, including accuracy, precision, recall, and F1-score, to ensure a comprehensive assessment of its classification capabilities.

3.6.2 Training Process:

The training dataset was divided into training and validation sets, typically following an 80:20 ratio. To enhance the model's ability to generalize, data augmentation techniques such as horizontal flipping, rotation, and brightness adjustments were applied. The model was then trained over multiple epochs, with its performance on the validation set continuously monitored to prevent overfitting. The version of the model that achieved the best validation performance was saved for further use.

3.7Telegram Bot Integration for Alert System

Violence Detection:

When the model processes a video frame and classifies it as containing violent activity, it triggers the alert mechanism. The detection is based on features learned from datasets that include various violent and non-violent scenarios. The model achieves this by analyzing spatial and temporal patterns using deep learning techniques.

Bot Creation on Telegram:

A Telegram bot is created using Telegram's BotFather. During the setup, the developer provides the bot name and username, and an API token is generated. This token is essential for establishing communication between the system and the Telegram platform. The bot acts as a virtual assistant that sends messages to users when triggered.

Recipient Setup:

To send alerts to the right person or group, the chat ID of the recipient is identified. This can be an individual user or a group chat where security personnel or stakeholders are active. The chat ID helps in directing the alert to the appropriate destination without delays.

Telegram API Integration:

The Telegram Bot API is used to connect the alert system with Telegram. Using this API, the system can send text messages, images, or other media directly to the configured chat ID. This integration ensures seamless communication between the detection model and the notification platform.

Real-Time Alert Delivery:

Once the integration is complete, the system sends an alert as soon as violence is detected. The alert includes a custom warning message along with the previously captured frame. This immediate delivery ensures that the concerned authorities are informed without any delay.

Optional Information:

Along with the alert, additional details such as the time of detection, video name, or location (if available) can be included in the message. This extra information helps the recipients quickly assess the situation and take necessary actions more effectively.

Instant Response:

The main advantage of using a Telegram bot is its speed and reliability. Telegram delivers messages instantly, enabling quick responses from security teams. This real-time communication improves the effectiveness of the violence detection system and contributes to better safety and surveillance management.

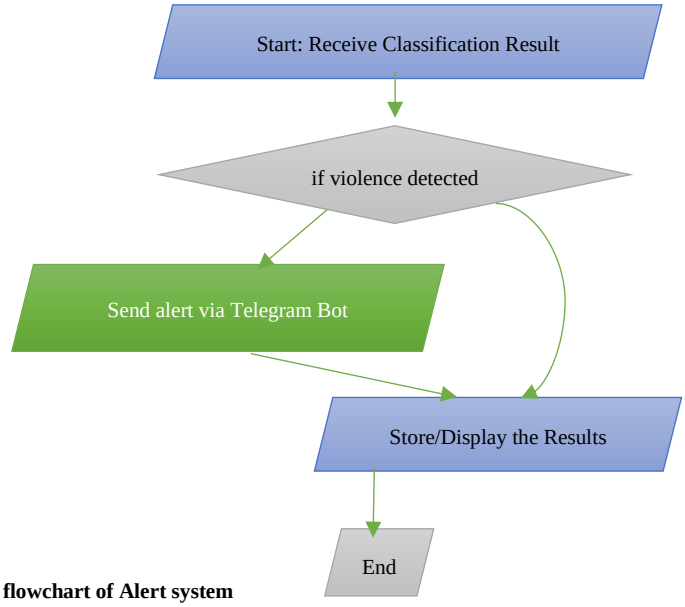

fig-15: flowchart of Alert system

4.Results & Discussions

The proposed violence detection system, which combines YOLOv5 for object detection and a deep learning model using MobileNet-styled CNN, 3D-CNN, and LSTM, was evaluated on three datasets: Hockey Fight, Movies Fight, and Real-Life Violence Situations. The model was trained using video sequences and tested on unseen samples to evaluate its effectiveness.

The system achieved an overall accuracy of 92%, with a precision of 91%, recall of 90%, and an F1-score of 90.5%, showing that it effectively distinguishes between violent and non-violent scenes.Dataset-wise performance varied slightly. The Hockey dataset performed best due to its consistent visual quality, while the Real-Life Violence dataset showed slightly lower accuracy due to poor lighting and real-world noise.. The integrated Telegram bot successfully sent alerts upon detection, making the system practical for surveillance.

Some challenges included dataset imbalance and high computational requirements. These were managed using augmentation techniques and cloud-based training with GPU support.

In conclusion, the model demonstrated strong and reliable performance, proving its potential for real-time violence detection in public safety systems.

Conclusion

This project successfully developed a real-time violence detection and intelligent alarm system capable of identifying violent behaviour from live video streams with high accuracy and speed. The proposed solution leverages a hybrid deep learning framework that integrates YOLOv5 for object detection, MobileNet for lightweight and efficient feature extraction, and a 3D Convolutional Neural Network combined with LSTM to capture both spatial and temporal features.

The system was designed with a strong emphasis on real-time performance, making it well-suited for deployment in practical surveillance environments such as public places, transport hubs, educational institutions, and smart city infrastructures. The integration of a Telegram bot for intelligent alert generation further enhances the system’s real-world usability by enabling instant notifications to relevant authorities or security personnel upon detecting violent activities.

Extensive testing on real-world datasets demonstrated the system's ability to deliver reliable and timely detection results, ensuring that potential threats can be identified and addressed without delay. Its modular design also allows for future expansion and integration with other security technologies.

In conclusion, this project highlights the potential of deep learning-based surveillance systems in enhancing public safety through intelligent, automated violence detection and rapid alert mechanisms. It serves as a foundation for building more comprehensive, scalable, and proactive security solutions.

REFERENCES

1. M. Khan, A. E. Saddik, W. Gueaieb, G. De Masi and F. Karray, "VD-Net: An Edge Vision-Based Surveillance System for Violence Detection," in IEEE Access, vol. 12, pp. 43796-43808, 2024.
2. S. Vosta and K. -C. Yow, "KianNet: A Violence Detection Model Using an Attention-Based CNN-LSTM Structure," in IEEE Access, vol. 12, pp. 2198-2209, 2024.
3. Y. Shi et al., "Caption-Guided Interpretable Video Anomaly Detection Based on Memory Similarity," in IEEE Access, vol. 12, pp. 63995-64005, 2024.
4. Mohammed, A. L. Swapnil, M. D. Peris, I. H. Nihal, R. Khan and M. A. Matin, "Multimodal Deep Learning for Violence Detection: VGGish and MobileViT Integration With Knowledge Distillation on Jetson Nano," in IEEE Open Journal of the Communications Society.
5. V. D. Huszár, V. K. Adhikarla, I. Négyesi and C. Krasznay, "Toward Fast and Accurate Violence Detection for Automated Video Surveillance Applications," in IEEE Access, vol. 11, pp. 18772-18793, 2023.
6. G. Aldehim, M. M. Asiri, M. Aljebreen, A. Mohamed, M. Assiri and S. S. Ibrahim, "Tuna Swarm Algorithm With Deep Learning Enabled Violence Detection in Smart Video Surveillance Systems," in IEEE Access, vol. 11, pp. 95104-95113, 2023.
7. M. Shoaib, A. Ullah, I. A. Abbasi, F. Algarni and A. S. Khan, "Augmenting the Robustness and Efficiency of Violence Detection Systems for Surveillance and Non-Surveillance Scenarios," in IEEE Access, vol. 11, pp. 123295-123313, 2023.
8. Abbass, M. A. B., & Kang, H. S. (2023). Violence detection enhancement by involving convolutional block attention modules into various deep learning architectures: comprehensive case study for ubi-fights dataset. *IEEE Access*, *11*, 37096-37107.
9. More, P., Patil, S., & Pattanshetti, T. (2024). Real time Violence and Weapon detection and Alert System.