Mini Project Report

on

Real-Time Violence Detection & Alert System

Using Machine Learning

Submitted by

Group id

Project Members

Devesh Chandak - 1032222713
Vibhor Surana - 1032222699

Under the Guidance of

Prof. Pramod Mali

Abstract

Violent Behaviour in public spaces poses a significant trouble to community safety and requires immediate intervention. Traditional surveillance systems rely on human operators, which are limited by attention span and the capability to cover multiple feeds simultaneously. This design represents a" Real- Time Violence Detection & Alert System" exercising the MobileNet v2 architecture, a convolutional neural network (CNN) optimized for real- time operations. The system is designed to analyse video streams from surveillance cameras, detect cases of violent behaviour, and send alerts to authorities with enhanced visual evidence.

Deep literacy models, particularly CNNs, have shown remarkable success in image and videotape analysis tasks due to their capability to learn complex point representations. MobileNet v2 was chosen for its effectiveness in terms of computational cost and delicacy, making it suitable for real- time processing. The system processes each videotape frame, classifying it as either violent or non-violent. Upon detecting violent conditioning, it enhances the clarity of the frames using the Python Imaging Library (PIL) and sends an alert containing the frame, timestamp, and position to a pre-configured Telegram group for immediate response.

The model was trained on a dataset of 1,000 videotape clips distributed into violent and non-violent classes. By employing data addition ways, the model was made robust to colorful scripts, similar as changes in lighting and camera angles. It achieved a delicacy of 96% on the training set and 95% on unseen footage, demonstrating its capability to generalize well to real- world surroundings.

enforcing this system posed challenges, similar as icing real- time processing and reducing false cons. still, by optimizing the MobileNet v2 armature, the system can reuse videotape aqueducts efficiently while maintaining high delicacy. This makes it a precious tool for enhancing public safety through timely discovery and intervention in violent situations. unborn advancements may include audio-visual integration and multi-camera support for broader content.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Violent incidents in public areas such as roads, parks and public transport may have serious consequences, it includes injuries, deaths, and long-term psychological impacts on people. And these events often occur quickly and are unpredictable. This makes timely intervention challenging. Traditional methods for monitoring such areas involve a human operator viewing CCTV footage. This approach has limitations due to human factors such as fatigue. Distractions and the actual amount of information that must be continuously checked

1.1 The importance of automatic violence detection

- With the increasing number of CCTV cameras in public areas. Therefore, there is an increasing demand for automated systems that can help track and detect violent behaviour. Automated violence detection systems can help:
- • **Improve public safety**: With real-time alerts to officials, the system can therefore respond quickly to events. This may reduce the severity of the consequences.
- • **Reduce human error**: Automated systems can work continuously without fatigue, reduces the chance of missing important things due to human oversight.
- • **Optimization of resource allocation**: by filtering non-violent activities, the system helps human operators focus on events and improve the efficiency of surveillance programs..

1.2 **Challenges in real-time violence detection**

Real-time detection of violence poses several challenges:

• **Variation in violent behaviour**: Violent acts can take very different forms it makes it difficult to determine characteristics of the investigation.

• **Environmental factors:** Changes in lighting, weather conditions and camera angles can affect the quality of CCTV images it complicates the detection process.

• **Computational limitations:** Real-time processing requires efficient algorithms that can quickly analyze video without sacrificing accuracy.

• Contextual differences: The system must be able to differentiate between violent behavior and non-violent activities, such as play

1.3 Deep Learning in Violence Detection

Deep learning has emerged as a powerful tool for analyzing and interpreting complex data, such as video footage . Convolutional Neural Networks (CNN) have shown great success in tasks such as image and video classification. This is due to its ability to automatically recognize and extract hierarchical features. Leveraging CNN's capabilities, the project aims to develop a system that can accurately detect violent activity in real time, alerting authorities immediately

Chapter-2
Literature Survey

Method	Description	Advantages	Disadvantages
Visual-Based Approach	Retrieves and represents visual information as local and global features. Local features include position, velocity, form, and color; global features include average speed and interactions.	Effective in representing visual characteristics and interactions.	Limited by the need for fixed camera positions; computationally expensive.
Audio-Based Approach	Uses audio data to classify violence through hierarchical techniques like Gaussian mixture models and Hidden Markov models to identify sounds such as gunshots and explosions.	Can detect specific sounds related to violence.	Limited to audio data; may not fully capture visual elements of violent incidents.
Hybrid Approach	Combines visual and audio characteristics, detecting violence through techniques like flame and blood detection, motion recording, and sound recognition.	Integrates multiple sources of information for more comprehensive detection.	May be complex to implement and computationally intensive.
Spatio-Temporal Modelling with 2D CNN	Uses Motion Saliency Map (MSM) to highlight moving objects, applies 2D CNNs for frame-grouping, and includes a Temporal Squeeze and Excitation Block for classification.	Highlights moving objects and reduces computational cost with the squeeze and excitation module.	Requires fixed camera positions; high computational cost.
Space Time Interest Point (STIP)	Detects interest points through spatial and temporal differences, using a 3D volume around each point to represent in 2D	High accuracy in detecting motion events; resistant to changes in scale and velocity.	Extremely high computational cost; not practical for real-time applications.

Violence Detection Using Low-Level Features	Segments motion regions using optical flow, extracts low-level features like LHOG and LHOF, and uses a Bag of Words (BoW) model with SVM for classification.	Uses simple low-level features and SVM for classification.	Low detection rate in crowded scenes; high false alarms; challenges with occlusions.
Spatio-Temporal Features with 3D CNN	Detects individuals using a lightweight CNN model, processes sequences of frames with 3D CNN to extract spatiotemporal features, and classifies with SoftMax.	Optimized with neural network toolkits; effective spatiotemporal feature extraction.	Computationally expensive; not suitable for real-time use.
Sensor-Network Approach	Extracts images from MPEG encoded video streams, trains DNN to recognize violent behavior.	Utilizes deep neural networks for frame recognition.	High computational power required; not practical for real-time or daily use.

Chapter 3.
Problem Statement

The main problem showed in this project is the automatic detection of violent activities in real-time CCTV footage. The challenge is to process and analyze large amounts of video data to identify short violent events detected live. The dataset used consists of 1000 video clips divided into categories of violence and nonviolence. with an average duration of 5 seconds per clip the proposed technique uses the MobileNet v2 architecture, a deep learning model. To detect violent activities in video frames. This model is optimized for real-time applications. It gives a balance between precision and computational efficiency.

3.1 Specific challenges

• Real-time processing: The system must be able to process continuous video in real time. Each frame is analysed and violent acts are identified when they occur.

• Accuracy and precision: Detection models must be able to accurately find difference between violent and non-violent behavior. Reduces false alarms both positive and negative to ensure reliable performance.

• Understanding context: The system must be able to interpret the context of detected activity. This is to avoid misunderstanding behavior that appears to be non-violent as aggressive.

3.2 Project objectives

The objective of this project is to develop a real-time surveillance system that:

• Process video data in real time: Manage continuous video streams and perform frame-by-frame analysis using deep learning models suitable for motion accuracy.

• Violent Activity Detection: Identify and classify violent and non-violent acts with high accuracy. Helps reduce false alarm rates.

• Send automatic alerts: Instant alerts to law officers or other relevant authorities, including visual evidence and information such as time and location

3.3 Proposed solutions

To address these challenges the project proposes a solution based on the MobileNet v2 architecture, a lightweight neural network designed for mobile and embedded vision applications. The system is trained on a balanced dataset of violent and non-violent video clips to recognize patterns related to violence behavior and can be detected. By detecting violent activities the system will automatically send notifications to the assigned department. Including updated frames and context information

Chapter 4
Project Requirements

1. Hardware Requirements:

- **Google Collab**: Required for model training, testing, and deployment with access to GPUs/TPUs.
- **GPU**: A dedicated GPU (e.g., Nvidia Tesla K80 or T4) to accelerate deep learning and video processing tasks.
- **Web Camera or CCTV**: To capture live video feeds for real-time violence detection.
- **Storage**: At least 5GB for datasets, pre-trained models, and result outputs.

2. Software Requirements:

- **Operating System**: Ubuntu 20.04 or similar Linux-based OS for model deployment.
- **Programming Language**: Python 3.8 or higher.

3. Libraries and Dependencies:

- **Deep Learning Frameworks**:
  - **TensorFlow** and **Keras**: For building, training, and deploying the MobileNetV2 model.
- **Video Processing**:
  - **OpenCV**: For capturing, processing, and displaying video frames.
  - **imageio**: For reading and writing video files.
- **Data Augmentation**:
  - **imgaug**: For augmenting video frames (e.g., flipping, zooming, rotating).
- **Data Handling**:
  - **NumPy**: For numerical operations and data handling.
  - **Pickle**: For saving and loading serialized data (e.g., models or preprocessed data).
- **Command-line Argument Parsing**:
  - **argparse**: For parsing command-line arguments (e.g., input video, frame skipping).
- **Time and Date**:
  - **pytz**: For managing time zones when sending alert messages with timestamps.
  - **datetime**: For handling date and time operations.

- **Visualization and Plotting**:
  - **Matplotlib**: For generating graphs (e.g., accuracy and loss).
  - **Seaborn**: For generating heatmaps (e.g., confusion matrix).
- **Alert System**:
  - **Telepot**: For sending real-time alerts (messages and images) to authorities via Telegram.
- **File Handling and System Interaction**:
  - **os**: For file system operations (e.g., creating directories, file paths).
- **Google Colab Utilities**:
  - **google.colab.patches**: For displaying images directly in Colab.
  - **google.colab.drive**: For mounting Google Drive to load the model and save output files.

4. Model Requirements:

- **MobileNetV2 Architecture**: A lightweight convolutional neural network optimized for real-time video classification tasks.
- **Model Weights**: Pre-trained weights for MobileNetV2, fine-tuned for violence detection.
- **Dataset**:
  - A dataset of 1000 videos, split into "Violent" and "Non-violent" categories. Augmented with different lighting and angles to improve model robustness.

5. Functional Requirements:

- **Real-time Video Processing**:
  - The system must process live or recorded video streams frame by frame and detect violence in real-time.
- **Violence Detection**:
  - The model should detect violent behavior in the video feed with at least 95% accuracy.
- **Alert System**:
  - Once violence is detected, the system should send a Telegram alert, including a timestamp, location, and image.
- **Image Enhancement**:
  - The system should enhance detected violent frames for better clarity (brightness, contrast, color adjustments).
- **Output Video**:
  - The processed video, with labels for detected violence, should be saved to Google Drive or a specified location.

6. Non-functional Requirements:

- **Efficiency**: The system should process 30 frames per second (FPS) to maintain real-time performance.
- **Scalability**: It should be scalable to handle multiple camera feeds.
- **Reliability**: The system should minimize false positives to avoid unnecessary alerts.

7. Security Requirements:

- **Data Privacy**: Video data should be securely stored and transmitted, particularly when sending alerts.
- **Access Control**: Only authorized users should have access to the system for managing notifications and video data.

Chapter 5

Implementation

The implementation of the real-time violence detection system is centered around the MobileNet v2 architecture, chosen for its balance between model complexity and performance. This architecture is particularly well-suited for real-time applications due to its efficient use of depth wise separable convolutions and inverted residual blocks, which reduce computational load while maintaining high accuracy.

5.1 MobileNet v2 Architecture

MobileNet v2 (Sandler et al., 2018) proposed a number of innovations that make it particularly well-suited for mobile and embedded applications:

- Depthwise Separable Convolutions**:** These convolutions split the regular convolution in two parts. Depth wise convolution that filters input channels independently and pointwise convolution that mixes the outputs into one channel. This reduces significantly the number of parameters and computational cost.
- Inverted Residual Blocks**:** Instead of stacking wide layers with residuals, we use narrow layers as another strategy to reduce computation. Classical residual blocks add inputs to output, while inverted residuals blocks connect bottleneck layers**.**
- Linear Bottlenecks**:** Linear bottlenecks expand the channel dimension of the input before applying non-linear functions. This helps in preserving information and enhancing the model's ability to learn complex features, reducing the risk of information loss.

5.1.1 Training the Model

The model was trained on a dataset of 1000 video clips (approximately 5 seconds long), containing an equal number of examples from the violent and non-violent classes. The dataset is augmented during each training epoch to improve the robustness of the learnt model in different scenarios. We use back-propagation to find the best set of weights for our model by minimizing a loss function that is high when many samples are misclassified.

5.2 Image Enhancement

After any of the frames is detected to be violent, the said frame is enhanced using the Python Imaging Library (PIL). Enhancing the frame makes it clearer and easier for the officials to evaluate the situation. The improvements include:

- Brightness Adjustment: This involves increasing the brightness in order to make the frame clearer even in dim light.
- Contrast Enhancement: The focus in this case, is on the contrast, specifically, the difference between the subject's features and the background, to better define violent actions.
- Colour Balance: This enhancement feature is geared towards increasing the colour fidelity of a given frame to the extent in which it portrays the true nature of the event.

5.2.1 Image Enhancement Techniques

The enhancement process uses PIL's built-in functions:

- ImageEnhance.Brightness**:** This function increases or decreases the brightness of the frame by a specified factor.
- ImageEnhance.Contrast: This function adjusts the contrast of the frame, emphasizing the distinction between different elements in the scene.
- ImageEnhance.Color**:** This function modifies the color balance to make the image appear more vivid and realistic.

5.3 Alert Module

The alert module is designed to notify authorities immediately upon detecting a violent activity. It uses the Telegram API to send alerts to a pre-configured Telegram group, which may include law enforcement or security personnel.

5.3.1 Alert Generation and Transmission

The alert system works as follows:

- Detection Verification**:** When violence is detected, the system initializes a counter and checks for violence in the subsequent 30 frames to ensure the detection is consistent.

- Alert Creation**:** If violence is detected in 30 consecutive frames, the system captures the current frame, enhances it, and prepares an alert message containing the frame, timestamp, and location.
- Alert Transmission**:** The alert is sent to the designated Telegram group using the Telegram API, allowing authorities to receive real-time notifications and take prompt action.

5.3.2 Integration with Surveillance Systems

The alert module can be integrated with existing surveillance systems to provide a seamless workflow. Upon receiving an alert, the system can trigger additional actions, such as:

- Storing Video Evidence: Automatically saving a short video clip of the incident for further review and analysis.
- Activating Alarms: Triggering audible or visual alarms in the vicinity to deter potential perpetrators and alert nearby individuals.

Architecture

Footage from the surveillance camera is broken down into frames. The frames are given as input to MobileNet v2 classifier for detecting violent activities in the given sequence of input frames. If no violent activity is recognized the respective frames are discarded. The violence detected frame is obtained and it is enhanced for better clarity. That frame, along with the location are sent to the nearest authorities using Telegram bot.

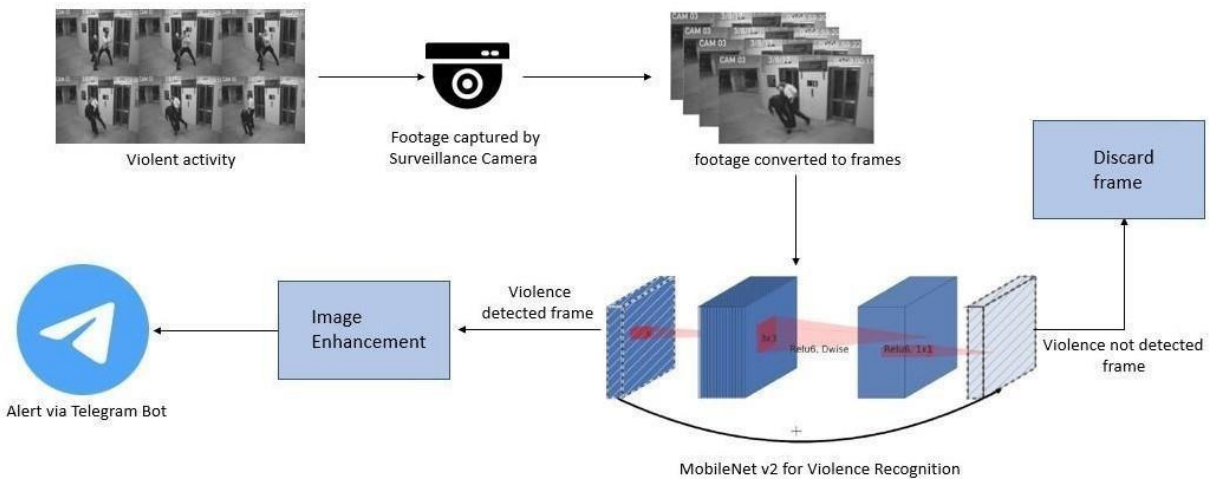


Figure 5.1: High level architecture diagram

Dataset: The dataset contains 1000 video clips which belongs to two classes, violence and non-violence respectively. The average duration of the video clips is5 seconds and majority of those videos are from CCTV footages. For training, 350 videos each from the violent and non-violent classes are taken at each epoch.

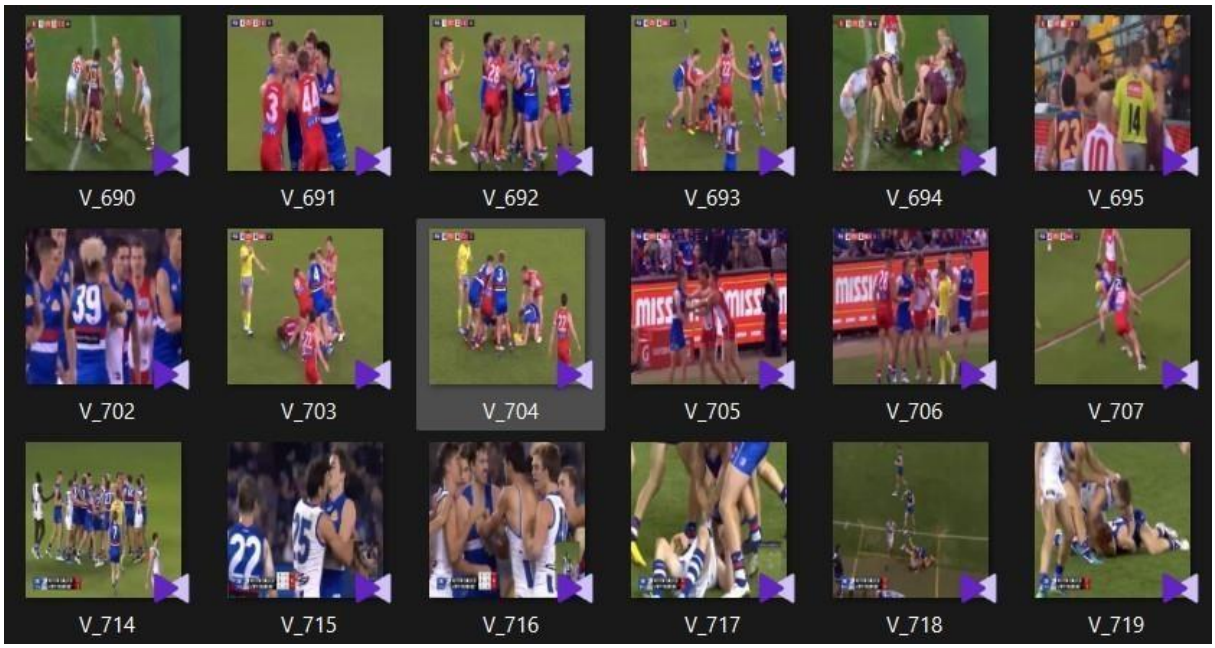


Figure 5.2: Video clips from the violence dataset

MobileNet V2: The MobileNet architecture is primarily based on depth wise separable convolution, in which factors a traditional convolution into a depth wise convolution followed by a pointwise convolution.

The module presents a residual cell (has a residual/identity connection) with stride of 1, and a resizing cell with a stride of 2. From Figure 3.3, "conv" is a normal convolution, "dwese" is a depth wise separable convolution, "Relu6" is a ReLu activation function with a magnitude limitation, and "Linear" is the use of the linear function.

The main strategies introduced in MobileNetV2 were linear bottleneck and inverted residual blocks. In the linear bottleneck layer, the channel dimension of input is expanded to reduce the risk of information loss by nonlinear functions such as ReLU. It stems from the fact that information lost in some channels might be preserved in other channels. The inverted residual block has a ("narrow" -"wide"-""narrow") structure in the channel dimension whereas a conventional residual

block has a ("wide" - "narrow"- "wide") one. Since skip connections are between narrow layers instead of wider ones, the memory footprint can be reduced.

Image Enhancement: Image Enhancement is performed on the frames that are obtained as output. This is performed using the inbuilt functions provided by the Python Imaging Library (PIL). PIL offers extensive file format support, efficient presentation, and fairly powerful image processing capabilities. The Core Image Library is designed to provide quick access to data stored in several major pixel formats. It provides a solid foundation for common image processing tools. The brightness and color of the obtained output frames is increased by a factor of 2.

Alert Module: A message is sent to the prescribed authority when an alert is triggered. The architecture of the implemented alert module is shown in figure 5.3. The first frame which gets a measure that is true- violence detection measures incur an increase of one on a counter variable that is initialized by momentarily. It then investigates the next thirty frames to see if any of them has registered a violence detection measure that is true. On an event where conservative consecutiveness is registered in connection to frames capturing violence on its detection, the counter in this instance is lifted. In the event that a frame has been marked as false for violence, the counter variable is reset to 0 and the next frame, which checks violence detection becomes active. However, if it is 30 frames with no violence, an inbuilt function of the computer picks the current time and an alert is sent to a telegram group containing higher-official members. An alert is triggered and at the same time the alert message contains an image related to the VCR activity, current time and location of the camera is placed.

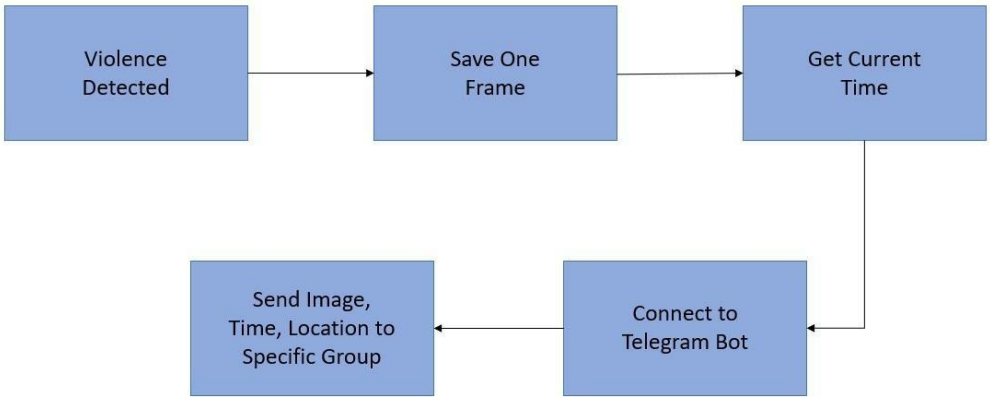


Figure 5.3: Architecture diagram of the Alert System

Figure 5.3 shows the alert message that is sent to the telegram group by the telegram bot. The concerned authorities can view the alert and take necessary actions.

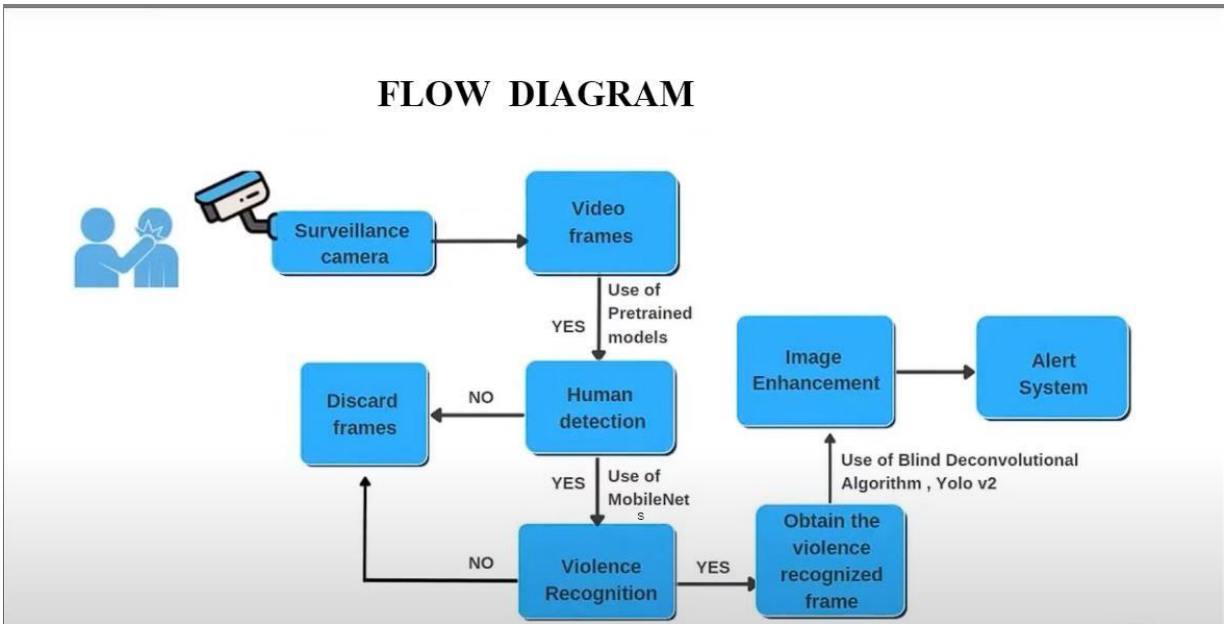


Figure 5.4: Flow diagram

The flow diagram shows how video frames from a surveillance camera are analyzed for human presence using pretrained models. If humans are detected, MobileNet checks for violent behavior. If violence is found, the frames undergo image enhancement using YOLO v2 and other techniques, and an alert system notifies authorities in real-time.

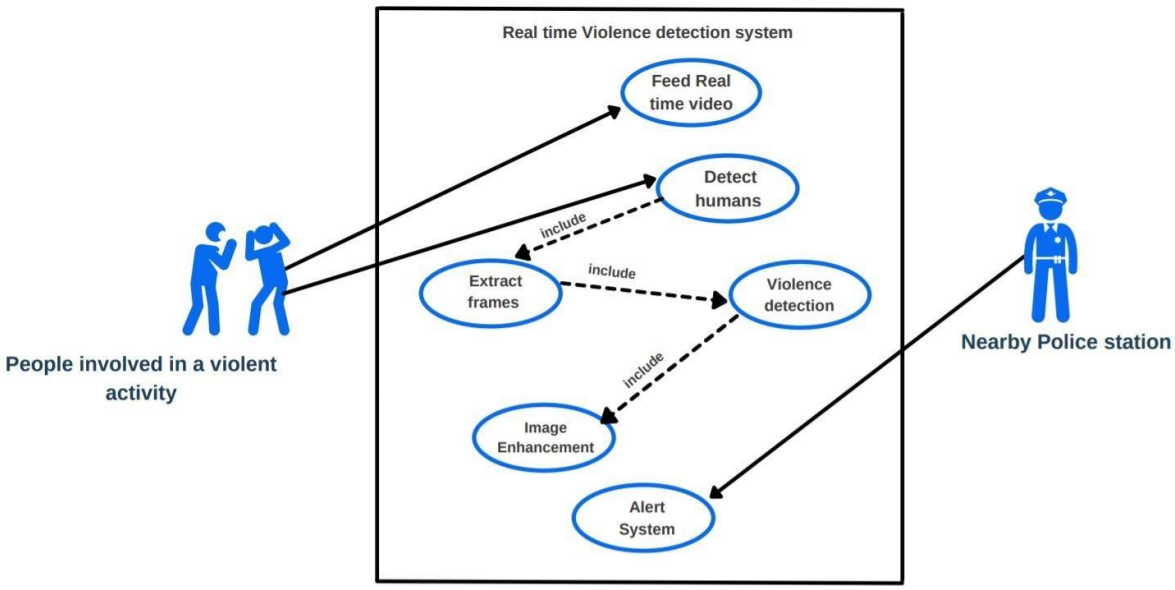


Figure 5.5: Use Case diagram

This diagram shows a real-time system where live video is processed to detect humans, followed by an analysis for violent behavior. If violence is confirmed, enhanced frames trigger an alert, which is sent directly to nearby police stations for immediate action.

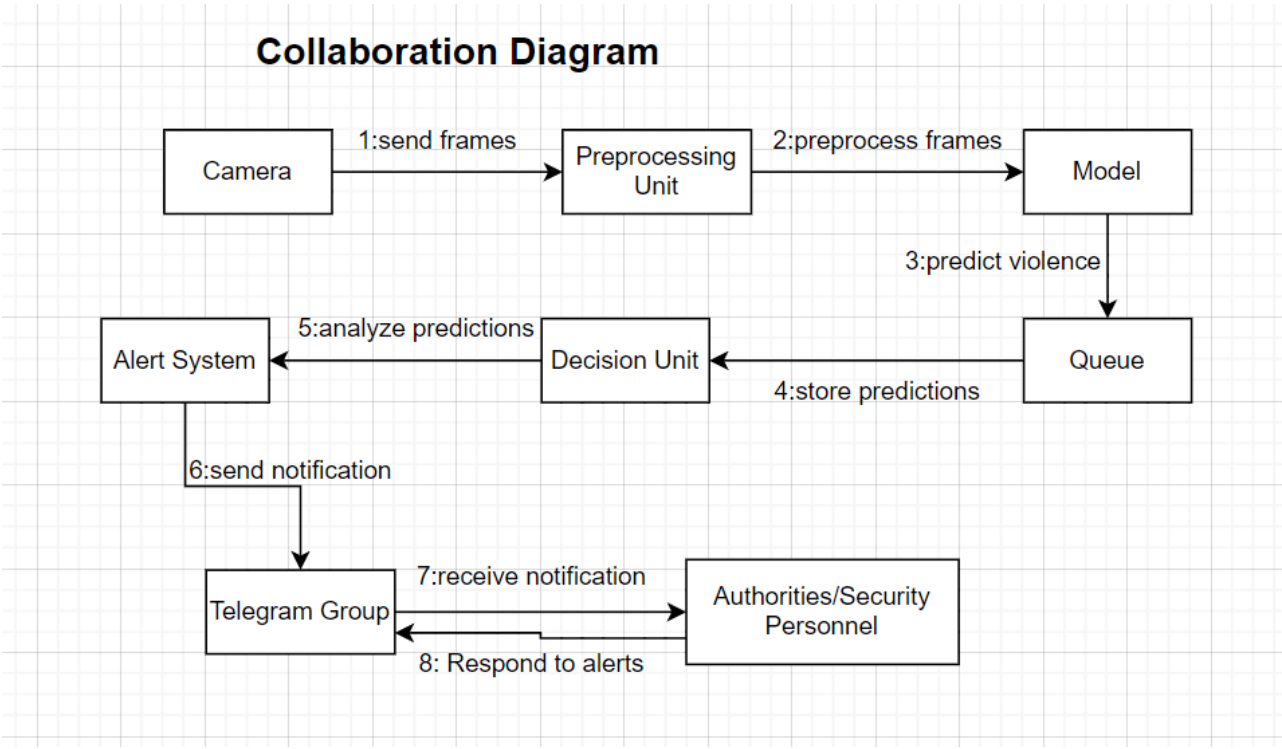
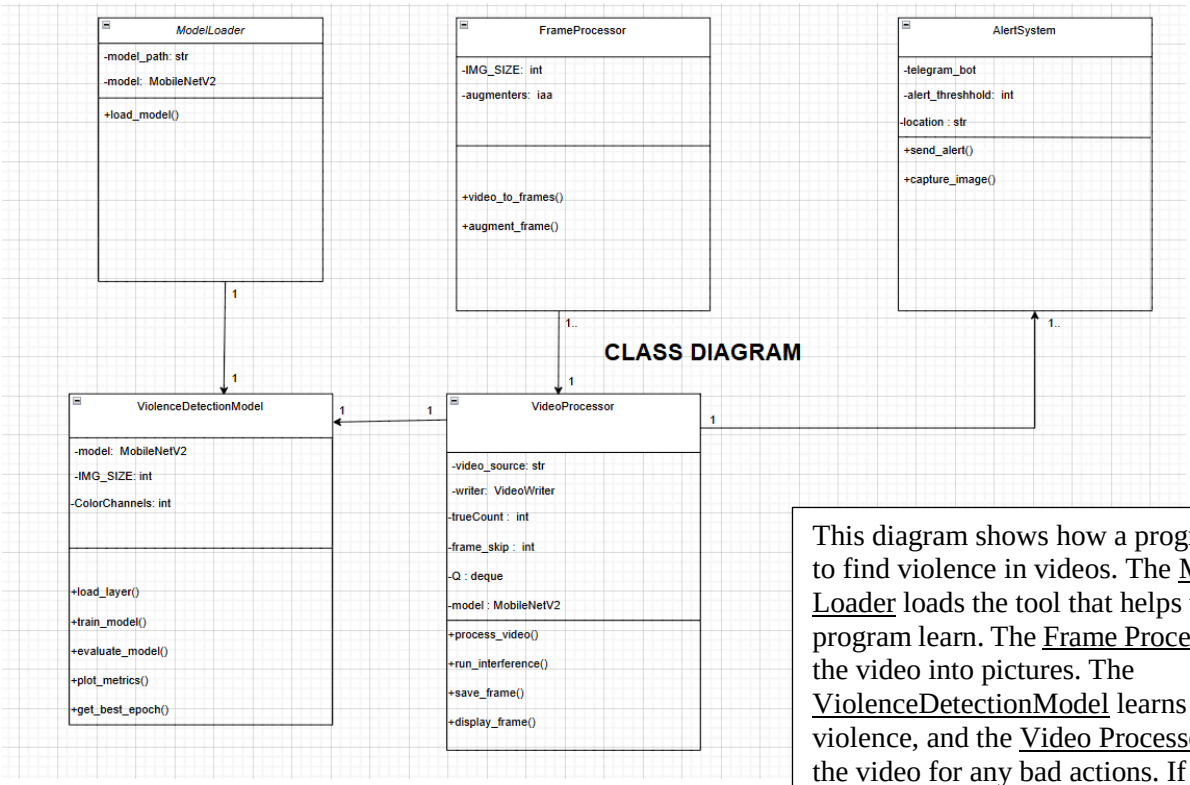


Figure 5.6: Collaboration Diagram

The collaboration diagram illustrates how frames from a camera are preprocessed and analyzed by a model to predict violence. These predictions are reviewed, and if violence is detected, alerts are sent to a Telegram group, notifying authorities for prompt response.



This diagram shows how a program works to find violence in videos. The Model Loader loads the tool that helps the program learn. The Frame Processor breaks the video into pictures. The ViolenceDetectionModel learns to spot violence, and the Video Processor checks the video for any bad actions. If it finds something, the Alert System sends a warning. All parts work together to detect violence and raise an alert.

Figure 5.7: Class Diagram

Chapter -6
Results and Discussions

In this section testing and training accuracy are displayed in the below given graphical representation. Fig. 6.1 displays the training and testing accuracy and loss for the MobileNet v2 model when a dataset containing 1000 videos of average duration 7 seconds is given as input. For each epoch 350 videos from the violence class and 350 videos from the non-violence are trained. 96% accuracy was obtained on training and a respective accuracy of 95% was obtained when a CCTV footage that was not included in the dataset was given for testing. The obtained output video frames are shown in Figure 6.3

In the graph in Figure 6.1 the accuracy and loss come to a constant level of increment and decrement after approximately 5 epochs. The obtained confusion matrix and other evaluation parameters are shown in Fig. 6.1

A video with violence is given as input to the system. Figure 6.3 shows one frame in the video that was labelled to have violent activity. Another video clip without violent activity was given as input. Figure 6.4 shows one frame of that video which is rightly labelled as false or violence

Our Model: -

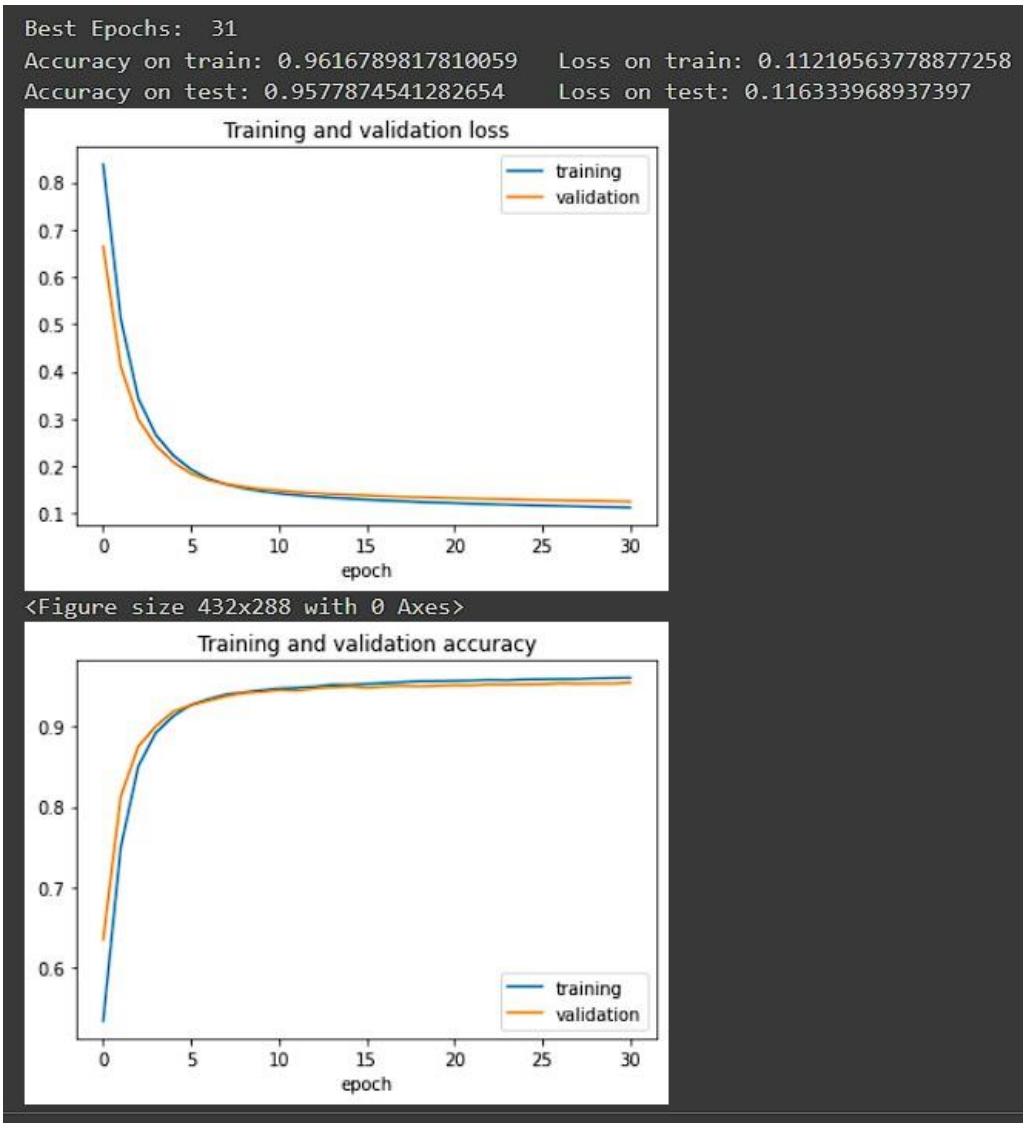
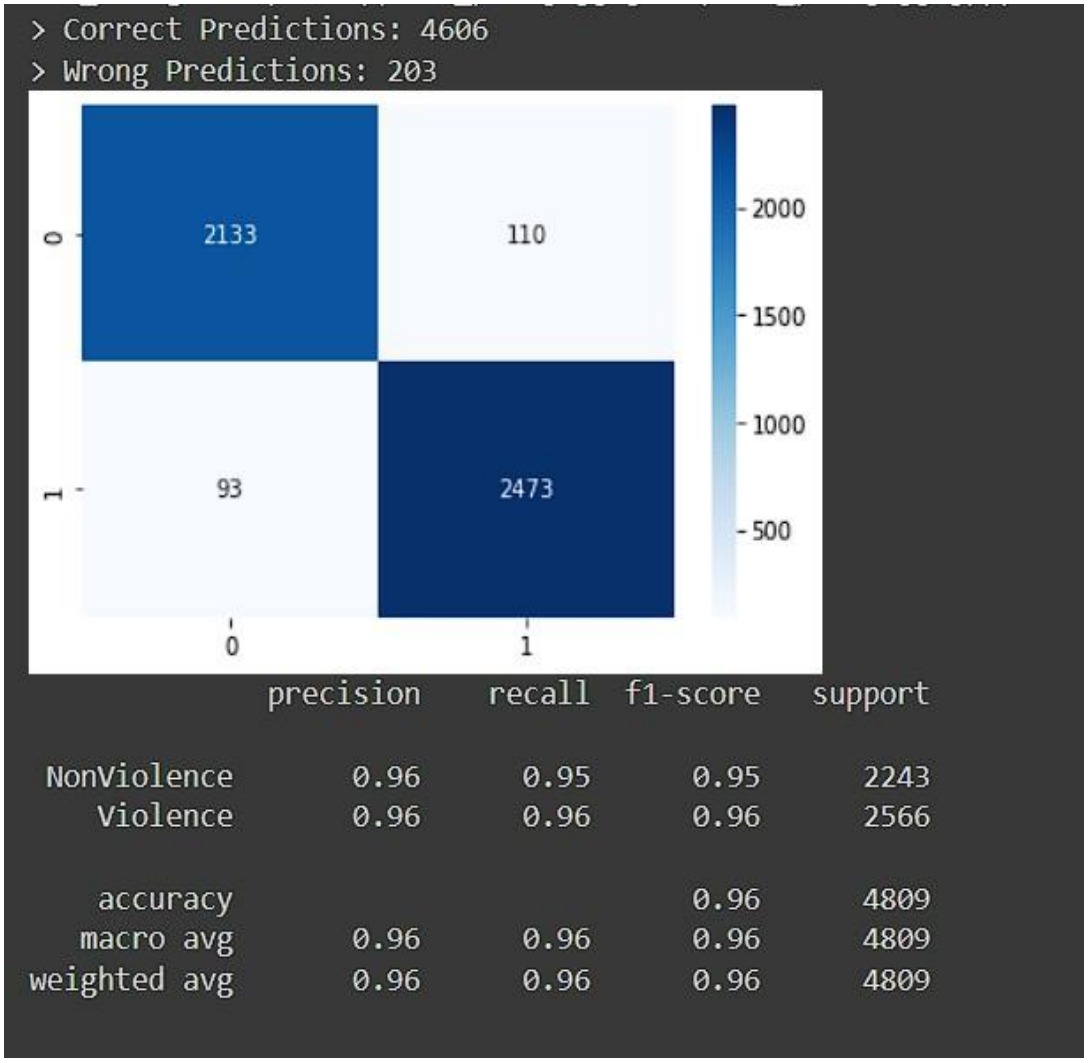


Figure 6.1: Accuracy and Error of the training set



	precision	recall	f1-score	support
NonViolence	0.96	0.95	0.95	2243
Violence	0.96	0.96	0.96	2566
accuracy			0.96	4809
macro avg	0.96	0.96	0.96	4809
weighted avg	0.96	0.96	0.96	4809

Figure 6.2: Confusion matrix of the trained model

Other models

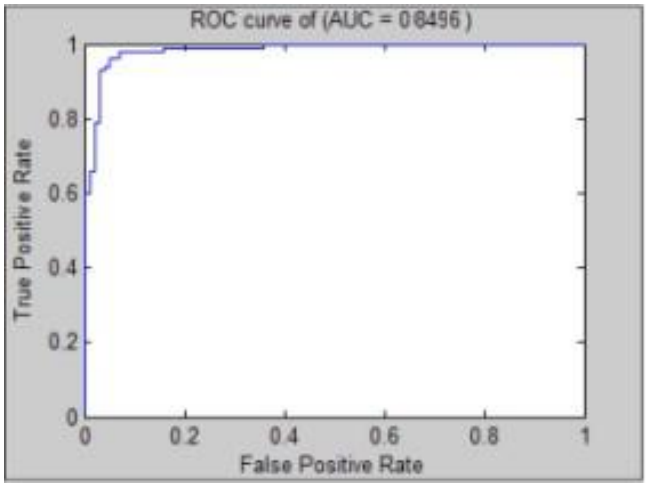


TABLE I: CNN Model IoU Results

Model	Val Loss	Val Mean IoU	Loss	Mean IoU
VGG16	0.0271	0.8446	0.0040	0.9280
VGG19	0.0263	0.8452	0.0030	0.9365
MobileNetV2	0.0462	0.7811	0.0028	0.9412

[4]

Figure 6.3: Output frame that recognized violence

Figure 6.4: Output frame that did not recognize violence

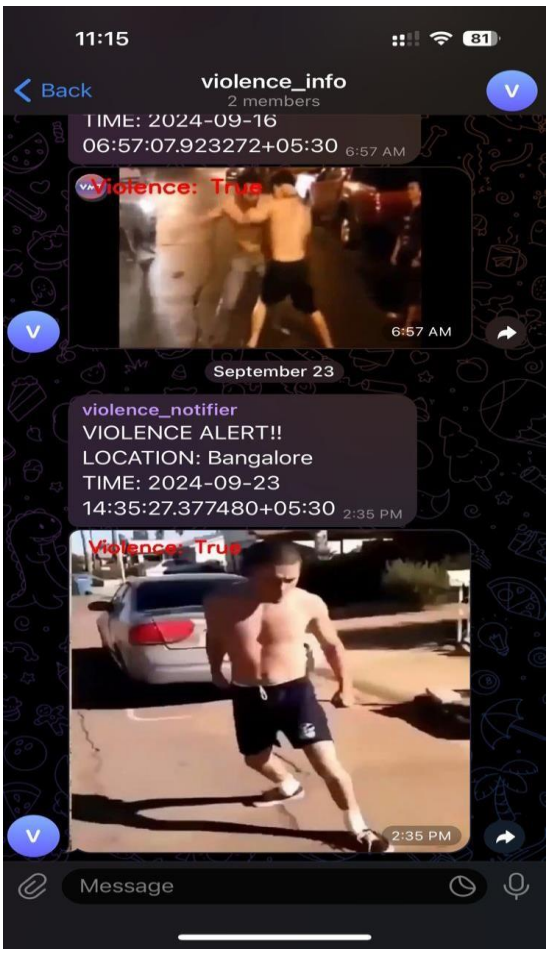


Figure 6.5: This is the alert received at telegram through telegram bot if there is violence detected in 50 frames.

Conclusion

Violence scene detection in real-time is a challenging problem due to the diverse con- tent and large variations quality. In this research, we use the MobileNet v2 model to offer an innovative and efficient technique for identifying violent events in real-time surveillance footage. The proposed network has a good recognition accuracy in typical benchmark datasets, indicating that it can learn discriminative motion saliency maps successfully. It's also computationally efficient, making it ideal for use in time-critical applications and low-end devices. Here, we had also shown the working of an alert system that is integrated with the pretrained model. In comparison to other state-of- the-art approaches, this methodology will give a far superior option.

Future Scope: This model could be upgraded to work in multiple cameras connected by a single network in a concurrent fashion. A short video of the violent activity could be incorporated along with the alert message.

References

1. S.U. Khan, I.U. Haq, S. Rho, S.W. Baik, and M.Y. Lee, "Cover the Violence: A Novel Deep-Learning-Based Approach Towards Violence Detection in Movies," *Applied Sciences*, vol. 9, no. 22, pp. 4963, Nov. 2019. doi: 10.3390/app9224963.
2. M.T. Gopalakrishna, "Violence Detection in Surveillance Video-A Survey," *International Journal of Latest Research in Engineering and Technology (IJLRET)*, NC3PS - 2016, pp. 11-17, 2016.
3. M.-S. Kang, R.-H. Park, and H.-M. Park, "Efficient Spatio-Temporal Modeling Methods for Real-Time Violence Recognition," *IEEE Access*, vol. 9, pp. 76270-76285, 2021. doi: 10.1109/ACCESS.2021.3083273.
4. F.U.M. Ullah, A. Ullah, K. Muhammad, I.U. Haq, and S.W. Baik, "Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network," *Sensors (Basel)*, vol. 19, no. 11, pp. 2472, May 2019. doi: 10.3390/s19112472.
5. A. Demir, C. C. Koçak, and M. A. Güngör, "Violence Detection: A Multi-Model Approach Towards Automated Video Surveillance and Public Safety," [Journal Name], vol. [Volume Number], pp. [Page Numbers], [Year]. doi: [DOI].
6. J.C. Vieira, A. Sartori, S.F. Stefenon, F.L. Perez, G.S. de Jesus, and V.R.Q. Leithardt, "Low-Cost CNN for Automatic Violence Recognition on Embedded System," *IEEE Access*, vol. 10, pp. 25190-25202, 2022. doi: 10.1109/ACCESS.2022.3155123.
7. M.H. Sandler, A. Zhu, M. Zhmoginov, and A. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510-4520, 2018. doi: 10.1109/CVPR.2018.00474.
8. J. Wang and Z. Xu, "Crowd Anomaly Detection for Automated Video Surveillance," *6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15)*, pp. 1-6, 2015. doi: 10.1049/ic.2015.0102.
9. P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo, and A.F. Dragoni, "Deep Learning for Automatic Violence Detection: Tests on the AIRTLab Dataset," *IEEE Access*, vol. 9, pp. 160580-160595, 2021. doi: 10.1109/ACCESS.2021.3131315.
10. M. Ramzan et al., "A Review on State-of-the-Art Violence Detection Techniques," *IEEE Access*, vol. 7, pp. 107560-107575, 2019. doi: 10.1109/ACCESS.2019.2932114.