VIOLENCE DETECTION SYSTEM USING DEEP LEARNING

Minor project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

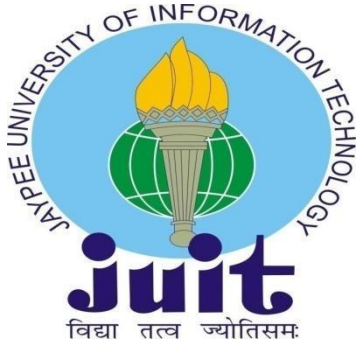
By

Aayush Sharma (211193)

Aadya Thakur (211184)

UNDER THE SUPERVISION OF

Prof. Dr. Vivek Kumar Sehgal



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,Waknaghat, 173234, Himachal Pradesh, INDIA

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 16/5/2024

Type of Document (Tick): | B.Tech./B.Sc./BBA/Other ✓

Name: Aadya Thakur, Aayush Sharma **Department:** CSE **Enrolment No** 211184, 211193

Contact No. 8627054265, 7018934066 **E-mail.** 211184@juit.solan.in, 211193@juit.solan.in

Name of the Supervisor: Prof. Dr. Vivek Sehgal

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):

Real Time Violence Alert System

REAL TIME VIOLENCE ALERT SYSTEM

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages = 45
- Total No. of Preliminary pages = 8
- Total No. of pages accommodate bibliography/references = 2

Aayush Aadya

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at .....7.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Vivek

(Signature of Guide/Supervisor)

Table of Contents

JUIT PLAGIARISM VERIFICATION REPORT **I**

LIST OF FIGURES **III**

CERTIFICATE **IV**

ACKNOWLEDGEMENT **V**

ABSTRACT **VI**

1 INTRODUCTION **1**

1.1 INTRODUCTION 1

1.2 OBJECTIVE 2

1.3 MOTIVATION 3

1.4 OPERATING ENVIRONMENT. 4

1.5 TECHNICAL REQUIREMENTS (HARDWARE). 4

1.6 DELIVERABLES/OUTCOMES 5

2 FEASIBILITY STUDY,REQUIREMENTS ANALYSIS AND DESIGN **6**

2.1 FEASIBILITY STUDY. 6

2.1.1 KEY GAPS. 12

2.1.2 PROBLEM DEFINITION. 12

2.1.3 PROBLEM ANALYSIS. 13

2.1.4 SOLUTION. 14

2.2 REQUIREMENTS 14

2.2.1 FUNCTIONAL REQUIREMENTS. 15

2.2.2 NON-FUNCTIONAL REQUIREMENTS. 15

2.3 ARCHITECTURE DIAGRAM. 16

3 IMPLEMENTATION **17**

3.1 PROBLEM STATEMENT 17

3.2 USE CASE DIAGRAM 17

3.3 DATASET INFORMATION. 18

3.4 ARCHITECTURE. 19

3.4.1 MOBILENET V2 20

3.4.2 IMAGE ENHANCEMENT 21

3.4.3 ALERT MODULE 22

3.5 IMPLEMENTATION 23

3.5.1 VIDEO TO FRAMES CONVERSION 23

3.5.2 MODEL CONSTRUCTION FOR VIOLENCE DETECTION . . 24

3.5.3 TRAINING CONFIGURATION 25

3.5.4 VIOLENCE DETECTION SYSTEM. 26

3.5.5 VIOLENCE ALERT SYSTEM. 29

4 RESULTS AND DISCUSSION **31**

4.1 DISCUSSION ON THE RESULTS ACHIEVED. 31

4.2 COMPARISON WITH CNN-LSTM ARCHITECTURE. 34

4.3 LIMITATION OF THE MINOR PROJECT. 35

4.4 FUTURE WORK. 36

REFERENCES **37**

List of Figures

No.	Title	Page No.
2.1	Architecture diagram.	16
3.1	Use Case diagram	17
3.2	The violence dataset's video clips	18
3.3	High level architecture diagram	20
3.4	Architecture of MobileNet v2	21
3.5	The Alert System's architecture diagram	22
3.6	The alert message Screenshot	23
3.7	Video to Frame Conversion with Augmentation Code	24
3.8	Model Construction for Violence Detection Code	25
3.9	Training Configuration and Callbacks for Model	26
3.10	Violence Detection System Code	28
3.11	Violence Detection System Output	28
3.12	Violence Alert System Code	30
3.13	Violence Alert System Output	30
4.1	The training set Accuracy and Error	32
4.2	The trained model Confusion matrix	32
4.3	Output frame that recognized violence	33
4.4	Output frame that did not recognize violence	33
4.5	Comparison of Training and Testing accuracy of MobileNet v2 and CNN-LSTM Models	34
4.6	CNN-LSTM Model Evaluation metrics	34

CERTIFICATE

We hereby certify that the work which is being presented in the project report titled **"REAL TIME VIOLENCE ALERT SYSTEM"** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out during the period from January 2024 to May 2024 under the supervision of **Prof. Dr. Vivek Kumar Sehgal**, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

The matter presented in this project report has not been submitted for the award of any other degree of this or any other university.

Aayush Sharma
211193
Aadya Thakur
211184

This is to certify that the above statement made by the candidates is correct and true to the best of my knowledge.

Prof. Dr. Vivek Kumar Sehgal
Professor and Head
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,

ACKNOWLEDGEMENT

Firstly, We express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

We are really grateful and wish our profound indebtedness to Supervisor **Prof. Dr. Vivek Kumar Sehgal, Professor and Head,** Department of CSE Jaypee University of Information Technology,Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of **Deep Learning** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Prof. Dr. Vivek Kumar Sehgal,** Department of CSE, for his kind help to finish our project.

We would also generously welcome each one of those individuals who have helped us straight forwardly or in a roundabout way in making this project a win. In this unique situation, We might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated our undertaking.

Finally, We must acknowledge with due respect the constant support and patience of our parents.

Aayush Sharma
211193
Aadya Thakur
211184

ABSTRACT

The pervasiveness of violent behaviours in today's world is a serious threat to both individual safety and the stability of society. The deployment of surveillance systems with automatic detection capabilities may be crucial in addressing this issue. This research uses deep learning techniques to present a comprehensive method for automatically detecting violent activities.

There are several steps in the suggested procedure. First, the algorithm detects whether or not there are people in the video frames. Then, frames that appear to depict violent acts are separated from other frames that don't belong. Finally, the chosen frames are recorded as photos after a trained model is used to identify violent behaviour within them. Wherever feasible, these photos are also improved to make it easier to identify the people involved. The resulting photographs are then sent as notifications to the appropriate authorities, together with important information like time and location.

This system's ability to detect aggression in video data effectively depends on how well it applies deep learning techniques, specifically Convolutional Neural Networks (CNNs). CNNs have many benefits, but they also have disadvantages, such as reduced accuracy and computational inefficiencies when used exclusively. To counteract this, the fundamental part of employing a pre-trained model like MobileNet expedites development and increases accuracy.

The suggested approach builds on a pre-trained MobileNet model to maximise accuracy and efficiency.

The suggested approach offers a viable option for the prompt detection and mitigation of violent behaviours, improving public safety and social well-being by fusing deep learning technology with effective alert system.

Chapter 01: INTRODUCTION

1.1 Introduction

The act of violence in public areas is a serious problem for society because it damages property values, productivity, community cohesiveness, and access to social services. Since the threat of violence affects people at every stage of life, public health and safety programmes around the world must prioritise identifying and preventing it.

It is a difficult challenge to identify violence in real time, especially with the large number of security cameras that are always watching over public spaces. Effective intervention and prevention techniques require the ability to quickly and accurately identify violent situations from live video feeds and notify the appropriate authorities in a timely manner.

Public video surveillance systems provide extensive coverage and precise information, but the laborious process of going through hours' worth of tape by hand creates a major delay in making prompt decisions. By using cutting-edge technologies to recognise violent scenarios in real-time, automated violence detection systems lessen this burden and spare authorities from having to go through copious amounts of video in order to find isolated incidents.

Recent developments in deep learning have shown to be remarkably effective at extracting spatiotemporal patterns from video data that are suggestive of violent behaviour. Deep learning techniques provide unmatched accuracy in violence detection tasks by recording both spatial and motion information over consecutive frames.

The creation of a Real-Time violence alert system using MobileNetv2, a cutting-edge deep learning model well-known for its effectiveness and accuracy in picture recognition tasks, is the main topic of this study. The suggested system analyses video frames to find violent scenes, refines the extracted frames for better analysis, and immediately notifies local law enforcement authorities of violent incidents. The alerts include important information like the time and location of the incident that was recorded.

The suggested system intends to improve public safety by enabling proactive intervention and quick response to violent situations, thereby promoting safer and more secure societies. It does this by utilising the power of deep learning and real-time surveillance technology.

The remaining report is categorized as follows , Section II presents related studies and their detailed comparisons. Section III discusses the selected approach in a detailed manner explaining the proposed architecture and the dataset, whereas section IV provides details of experimentation and discusses the evaluation of the approach. Finally, section V concludes the report and discusses future innovations that could be brought into our project.

1.2 Objective

With the ultimate goal of improving public safety and societal cohesiveness, the project's purpose is to construct an automated violence detection system employing deep learning techniques and surveillance technology. Principal goals consist of:

1. To design and implement a robust automated detection system that can swiftly locate violent activity in live video.
2. To leverage state-of-the-art deep learning techniques, particularly Convolutional Neural Networks (CNNs), to enhance the accuracy and efficiency of violence detection.
3. To integrate pre-trained models such as MobileNet to reduce computational inefficiencies and boost the overall system performance.
4. To perform enhancement of retrieved images, aiding in the identification of individuals involved in violent incidents.
5. To provide easy-to-use channels of communication, such as the Telegram app, so that the appropriate authorities are notified as soon as potentially harmful conduct is detected.
6. To evaluate the effectiveness and performance of the automated detection system through rigorous testing and validation procedures.
7. To evaluate the system's effects on social order and public safety, with an emphasis on lowering the incidence of violent conduct and facilitating early intervention.

8. To document the development process, methodologies, and outcomes comprehensively in a project report, facilitating knowledge dissemination and potential future enhancements.

1.3 Motivation

This initiative is driven by the pressing need to address the pervasive problem of violence in public areas, which seriously jeopardizes both individual safety and the welfare of society. Violence has become all too commonplace in the globe, affecting people of all ages and socioeconomic backgrounds. This alarming pattern erodes our communities' fabric and jeopardizes public safety, which lowers economic productivity and social cohesiveness.

Traditional approaches to violence prevention, such monitoring and manual intervention, are often labor-intensive, sluggish, and prone to human error. In addition, the rapid advancement of technology, particularly in the fields of artificial intelligence and deep learning, holds the potential to fundamentally alter how we perceive and respond to violent behavior.

In our specific context of living in hostels, the implementation of an automated violence detection system holds particular significance. It is not uncommon for young individuals residing in hostel environments to engage in fights, which can result in severe injuries if left unchecked. By the deployment of such a system, hostel wardens and authorities can proactively identify and intervene in these conflicts before they escalate, hence, the well-being of residents can be safeguarded all the while fostering a safer living environment for all.

Our goal is to leverage deep learning techniques and surveillance technology to construct an automated violence detection system that will improve social order and public safety by being proactive. With the use of such a system, law enforcement may quickly detect violent situations and take appropriate action in real time, lessening the damage that these episodes cause to people and communities.

Additionally, putting this system into place is in line with larger societal objectives to advance social fairness, security, and peace. We can contribute to the development of safer and more resilient communities where people may live and grow without having to worry about violence by using cutting-edge technology to combat it.

1.4 Operating Environment

Python: The language employed here is a well-known programming language that is ideal for machine learning and deep learning applications because of its vast library of pre-built modules. Tensorflow and Open-CV are two Python packages that we utilized for our project. Its simple syntax makes it widely utilized for a variety of tasks, including web development.

Google Colaboratory: It is an environment or a web application developed by Google for helping people do python related projects like Machine Learning and Deep Learning based. To help us with the tasks that require extreme amount of computational power, Google allows us to use their Graphical Processing Unit or their Tensor Processing Unit for free if we ever need them. This project does need a decent amount of computational power, hence is used here.

1.5 Technical Requirements (Hardware)

Hardware Requirements:

1) Processor: High-performance CPU with multi-core architecture for parallel processing of video frames and deep learning computations.
2) Graphics Processing Unit (GPU): NVIDIA GPU with CUDA support for accelerated training and inference tasks. A minimum of NVIDIA GeForce GTX 1060 or higher is recommended for efficient model training and real-time processing.
3) Random Access Memory (RAM): At least 16GB of RAM is recommended to handle large datasets and model operations efficiently.
4) Storage: SSD storage with sufficient capacity to store datasets, trained models, and system logs. A minimum of 500GB SSD is recommended for optimal performance.
5) Internet Connectivity: Reliable high-speed internet connection for streaming video data and communication with external services such as Telegram for alert notifications.

Software Requirements:

1) Python Environment: Python 3.x with necessary packages including TensorFlow, Keras, OpenCV, NumPy, and CUDA Toolkit for GPU acceleration.
2) Deep Learning Framework: TensorFlow or PyTorch for building and training convolutional neural networks (CNNs) for violence detection.
3) Development Tools: Integrated Development Environment (IDE) such as PyCharm

4) Version Control: Git for managing project codebase and collaboration with team members.
5) Communication Platform: Integration with Telegram API for real-time alert notifications to relevant authorities.

1.6 Deliverables/Outcomes

1) Automated Violence Detection System: Using deep learning methods and surveillance technology, this project's main objective is the creation and deployment of an automated violence detection system.
2) Software Prototype: A working software prototype for the automated violence detection system that can recognize violent behavior with accuracy from real-time video footage.
3) Algorithm Implementation: The system's ability to recognize and react to violent situations is made possible by the implementation of algorithms for frame selection, violence detection, and picture enhancement.
4) Integration with Communication Platforms: Integration of the system with communication platforms such as the Telegram app for prompt notification of relevant authorities upon detecting potentially dangerous activity.
5) Documentation and Project Report: Detailed report outlining the project's results, development process, techniques, and algorithms. This includes the project report, which will be used as a guide for further improvements and information sharing.
6) Evaluation and Testing Results: Results from rigorous testing and evaluation procedures to assess the effectiveness, accuracy, and performance of the automated violence detection system.

Chapter 02: Feasibility Study, Requirements Analysis and Design

2.1 Feasibility Study

Thanks to developments in artificial intelligence (AI) and surveillance technologies, the field of violence detection has experienced rapid expansion and innovation in recent years. A thorough comprehension of the literature influencing this developing topic is important given the junction of AI and public safety. In order to build the foundation for our suggested automated violence detection system, this section presents an overview of pertinent literature in the field of violence detection, specifically in the context of hostels. We want to uncover important insights, obstacles, and opportunities that will guide the viability and execution of our project by reviewing previous research and methodology.

[1] **The CASSANDRA2 System**: CASSANDRA2 measures environmental aggressiveness using a Dynamic Bayesian Network; physical violence is mostly detected by quickly articulating body parts. Since it is not feasible to recover a person's pose for every frame in a movie, articulation energy is captured by aggregating optical flow over foreground regions.

In order to predict violence, the system also uses auditory signals, such as yelling. However, because of various sources and reverberation, it might be difficult to identify auditory events in real-world settings. By identifying and categorizing audio events from a preset collection of sounds, CASSANDRA2 improves the system's ability to discriminate.

To guarantee a generalizable model, annotated training data is gathered from many places and used to estimate model parameters. Professional actors play a variety of situations at a train station to validate the prototype system, including typical conduct, physical aggressiveness, vandalism, and borderline instances like boisterous football supporters.

The experimental evaluation demonstrates how important it is to include contextual information and complementing audio and visual signals when estimating hostility. Audio cues are especially helpful in situations where there is an increase in hostility, as seen by increased voices before to physical attack. On the other hand, while performing a physical attack, visual cues outperform the reference system by a wide margin.

The suggested approach improves performance for each sensor modality separately and when modalities are fused, according to a report that compares its performance with findings from a prior study

[10]. The efficacy of the system in identifying instances of aggressiveness is emphasized, along with the aspects that require more refinement and contemplation for practical

[2] **Efficient Spatio-Temporal Modeling Methods for Real-Time Violence Recognition**: In order to enable prompt police reaction during crimes, the paper highlights the urgent need for trustworthy automatic surveillance systems, notably in the area of violent action detection in real time. Although tremendous progress has been made in video-based action recognition, most initiatives have focused more on performance than efficiency. While they perform well in image identification, conventional 2D CNNs have trouble storing the dynamic motion information present in video sequences. Recurrent Neural Networks (RNNs) have been used for video interpretation, however their early layers frequently lack convolution over numerous frames, which degrades performance.

The research investigates the use of 3D CNNs, such as C3D and I3D, which efficiently encode spatiotemporal information but are computationally costly, as a solution to these problems. On the other hand, techniques that combine RGB frames with optical flow characteristics have demonstrated potential, but they require a large amount of processing power. Although they concentrate on pertinent human movements, human pose estimation approaches sometimes add extra complexity and may result in performance loss.

The study suggests an effective violence detection method appropriate for on-device, real-time monitoring in response to these difficulties. By utilizing cutting-edge modules in conjunction with lightweight 2D CNN backbones like SqueezeNet, MobileNets, and EfficientNet, the system achieves performance that is similar to 3D CNNs but with less computing complexity. These modules include temporal attention modules that adjust time-wise features for enhanced identification accuracy and spatial attention mechanisms that use motion information to identify important locations. The three main components of the suggested violence detection pipeline are as follows:

1) An effective spatial attention module that improves the network's focus on moving objects by capturing significant pixel variations across successive frames.
2) A technique for encoding short-term motion data into 2D CNNs, which is essential for identifying violent, quick moves.
3) A temporal attention module that adaptively recalibrates time-wise characteristics to increase identification accuracy, modelled after Squeeze-and-Excitation (SE) blocks.

The suggested approach seeks to minimize prejudice and privacy violations while assisting

surveillance staff in quickly recognizing and responding to violent crimes by emphasizing lightweight and quick attention modules. All things considered, the report offers a viable way to improve public safety via automated video monitoring systems.

This method is inapplicable to moving cameras since violence recognition requires a series of frames taken in a stationary location. This method has a significant computational cost since it involves several computations utilizing three distinct modules to complete a predetermined set of tasks.

[3] **Space-time Interest Points** : In order to facilitate video analysis, the idea of expanding spatial interest points into the spatio-temporal domain is presented in this study. It suggests a technique to identify local structures in space-time, recording notable changes in both space and time, based on Harris and Forstner interest point operators. After these observed events are identified, event categorization and video representation are made possible by the use of scale-invariant spatio-temporal descriptors. It is shown that the method can successfully identify persons who are walking in situations with changing backdrops and occlusions.

Through the use of spatiotemporal interest points, the technique offers a reliable and effective way to analyze and understand videos. Although the work yields promising results with a high accuracy rate, the computing cost of extracting such information is prohibitively expensive for real-world applications like media rating systems and monitoring.

[4] **Violence detection in surveillance video using low-level features :** The research offers a thorough method for identifying violent scenes in videos. Video preprocessing, motion region segmentation, low-level feature extraction, feature processing, and classification/prediction are the five key stages that are introduced.

Sparse temporal sampling is used to extract frames during the video preprocessing stage in order to minimize redundancy.A new approach based on optical flow field texture analysis is suggested for motion region segmentation. This technique successfully splits moving zones, getting around problems with changing backdrops.Two types of descriptors are introduced in low-level feature extraction: Local Histogram of Oriented Gradient (LHOG) and Local Histogram of Optical Flow (LHOF). LHOF records dynamic changes whereas LHOG records appearance information, resulting in complementing spatiotemporal representations.Using a Bag of Words (BoW) model to represent extracted features is a feature processing technique. Using a late-fusion approach, the study processes the LHOF and LHOG characteristics independently before merging them for categorization.

The usefulness of the suggested method is demonstrated by experimental findings on three difficult datasets, which show that it performs better than current approaches in violence identification, especially in situations with dynamic backdrops and crowded scenes. Because of its high computing cost and multitude of parameters, this procedure is unsuitable for everyday usage and real-time situations.

[5] **Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network** : The suggested approach uses deep learning algorithms to provide an end-to-end framework for the identification of violence in surveillance footage. The system first records video feeds, which are subsequently processed by a MobileNet CNN model for human detection. By effectively removing unnecessary frames, this phase maximizes processing time. The next step uses a 3D CNN model to extract spatiotemporal characteristics from a series of frames, which allows for a more in-depth comprehension of the motion and context in the video data. After that, a Softmax classifier is given these extracted characteristics to make final predictions about the existence of violent acts.

Extensive tests were carried out on three benchmark datasets: violent crowd, violent movie, and hockey conflict to confirm the efficacy of the technique. The outcomes showed higher accuracy levels than other cutting-edge techniques, proving the stability and effectiveness of the suggested strategy.

Because of the intense occlusion and moving throng, it was difficult to detect aggression in a crowded situation. It has a high false alarm rate and a poor detection rate.

[6] **Real time violence detection in surveillance videos using Convolutional Neural Networks :** The study examines the critical need for real-time violence detection in surveillance systems in light of the drawbacks of conventional approaches that depend on human observation. It draws attention to the fundamental drawbacks of human-operated surveillance, including slow reaction times and the need for manual review of captured video, which can obstruct timely and effective intervention in emergency situations. In light of these difficulties, the research suggests a novel method that combines computer vision techniques—in particular, convolutional neural networks (CNNs)—to automatically identify violent or unusual activity in real-time video feeds.

The assessment of several CNN architectures, including well-known models like AlexNet, VGG-16, GoogleNet, and the lighter MobileNet, using a dataset made up of videos of hockey fights, is at the heart of the study. The study evaluates the accuracy, computational efficiency, and suitability of each model for real-time surveillance applications through extensive experimentation and analysis. The MobileNet model stands out as the most effective model due to its exceptional ability to accurately identify violent incidents with minimal computational overhead.

The results highlight how CNN-based methods have the potential to completely transform surveillance technology by allowing for the proactive, automated real-time detection of violent activity. Through the utilization of sophisticated machine learning algorithms, the suggested system has the potential to improve security protocols and facilitate prompt action during emergency scenarios. Furthermore, the study promotes the broad use of these technologies to support human-operated surveillance systems, reducing the drawbacks of manual monitoring and raising the efficacy of surveillance as a whole.The drawback of utilizing only CNN is that it is less accurate and takes a long time to compute.

[7] **A Sensor Network Approach for Violence Detection in Smart Cities Using Deep Learning** : The paper introduces a groundbreaking approach to automating violence detection in surveillance videos, leveraging the synergy between deep neural networks (DNNs) and MPEG flow vectors. It tackles the inherent limitations of traditional manual feature engineering methods by proposing an advanced architecture that seamlessly integrates DNNs with motion features extracted from MPEG flow vectors. This integration not only enhances detection accuracy but also ensures computational efficiency, marking a significant advancement in the field of video processing and surveillance.

The core of the suggested solution is the combination of a time domain classifier with a deep convolutional neural network (CNN), which is a revolutionary method for separating temporal and spatial processing. The system's novel design provides unparalleled precision in violence detection by allowing it to identify violent trends within the intricacies of urban settings. The study showcases the exceptional efficacy of the suggested approach, outperforming current methods and producing cutting-edge outcomes, by thorough testing and verification.

The study also emphasizes how flexible and reliable the suggested strategy is in a range of surveillance situations. The system is resilient to changes in illumination and color spectrum because it utilizes motion features that are taken from MPEG flow vectors. Because of its innate flexibility, the system can be applied more effectively in actual surveillance scenarios and still function dependably in a variety of scenarios.

Notwithstanding its innovative potential, the study admits some drawbacks with the suggested strategy. For example, the system might have trouble detecting instances of crowd violence and might have trouble accounting for camera motion. The paper outlines potential research directions to tackle these issues, such as integrating weapons recognition capabilities and broadening the system's scope to include terrorism threat detection.

In conclusion, the work offers a revolutionary approach to the vexing problem of automating the detection of violence in surveillance footage. The suggested approach, which offers previously unheard-of levels of efficiency, accuracy, and adaptability in violence detection, promises to transform urban area surveillance by utilizing cutting-edge technologies like DNNs and motion feature extraction. Similar to other methods, lots of computational power is required which is not suitable for real -time usage / environments or for daily usage

[7] **MobileNetV2: Inverted Residuals and Linear Bottlenecks:** The paper introduces MobileNetV2, an improved version of the MobileNet architecture for efficient convolutional neural networks (CNNs) designed for mobile and embedded vision applications. Compared to more conventional networks like VGG or ResNet, MobileNetV2 is intended to offer a CNN architecture that is lighter and more efficient. It seeks to attain cutting-edge performance while remaining appropriate for mobile device deployment on low-processor devices.

Using inverted residuals with linear bottlenecks is the main novelty of MobileNetV2. Non-linear activation functions, such as ReLU, are used at the bottleneck layer of classic residual networks, which might add needless computing complexity. In order to reduce computing cost and retain expressiveness, MobileNetV2 uses identity shortcuts in conjunction with linear bottlenecks.

The inverted residual with linear bottleneck is a novel building block that is introduced by MobileNetV2. This block is composed of a linear bottleneck layer and a lightweight depthwise separable convolution. In order to lower computational costs, the linear bottleneck layer reduces the number of channels to a small portion of the input channels.

To enhance the number of channels, MobileNetV2 also adds an expansion layer before to the depthwise convolution. A pointwise convolution is applied after this expansion layer to minimize dimensionality and provide a compact feature representation.

With the help of MobileNetV2's adaptable scaling mechanism, users can adjust the network's width and resolution to strike a compromise between computational efficiency and accuracy. As a result, MobileNetV2 can adjust to various deployment scenarios and resource limitations.

In-depth experimental findings showing MobileNetV2's performance on a range of benchmark datasets and tasks, such as object identification and picture classification, are presented in the study.

All things considered, MobileNetV2 makes a significant addition to the area of deep learning for resource-constrained devices by providing a highly effective and scalable CNN architecture appropriate for a variety of mobile and embedded vision applications.

2.1.1 Key Gaps

The following succinctly describes the primary gaps in the current methods for violence detection, as indicated by the limitations noted in multiple research papers:

1) Camera Mobility Limitation: Existing approaches mostly rely on static sequences of frames from fixed cameras, which makes them inappropriate for dynamic views or moving cameras. This restriction limits the use of violence detection systems in situations when camera movement is crucial.
2) High Computational Cost: The current methods for detecting violence have high computational requirements, requiring a great deal of computation over several modules with predetermined tasks. Practical applications where real-time processing and efficiency are critical, like media rating systems and surveillance, are hampered by this high computational cost.
3) Scalability and Real-time Performance: Many current techniques are not suited for real-time contexts and have scalability problems. These methods' high processing costs and reliance on several factors make them unsuitable for everyday usage and practical implementation.
4) Difficulties in Crowded Scenes: Because of crowd dynamics and occlusion, violent crime detection in crowded scenes is a challenging task. In these kinds of situations, current approaches frequently show low detection rates and high false alarm rates, underscoring the need for better algorithms that can manage complex environments.
5) Trade-offs between computation time and accuracy: Methods that only depend on convolutional neural networks (CNNs) may have extended computation times and decreased precision. One of the biggest obstacles to the development of efficient violence detection systems is continuing to strike a balance between computing efficiency and detection accuracy.
6) Real-time Suitability and Resource Constraints: A lot of the approaches that are now in use are not suitable for everyday use or real-time usage since they demand a lot of processing power. The broad implementation of violence detection systems requires balancing real-time performance with computing needs.

2.1.2 Problem Definition

The topic at hand is the widespread problem of violence in public places, which seriously jeopardizes both individual safety and the welfare of society. Violence has become all too commonplace in the globe, hurting people of all ages and backgrounds. Hostel settings in particular are known to often see resident confrontations, which can result in possible injuries and disturb the peace within the community.

Currently used to combat violence, manual monitoring and intervention approaches are frequently labor-intensive, time-consuming, and prone to human error. Furthermore, the inability of conventional methods to accurately recognize and react to violent conduct in real-time has been brought to light by the quick developments in technology.

As a result, the main issue is the requirement for a more effective and proactive way to identify and deal with instances of violence in public areas, especially in hostel settings. To lessen the effects of violent episodes on people and communities, this calls for the creation of an automated system for detecting violence that can quickly locate and notify the appropriate authorities when it occurs.

The creation of precise algorithms for violence detection, integration with current surveillance technology, and making sure privacy and ethical issues are taken into account are some of the main problems related to this issue. To guarantee the practical execution and long-term sustainability of the suggested solution, it is also necessary to thoroughly assess its scalability and economic viability.

It is imperative that this issue be resolved in order to advance community cohesiveness, public safety, and the creation of conditions in which people may live and prosper without having to worry about violence. Therefore, a crucial first step in accomplishing these broad social objectives is the creation of an automated violence detection system.

2.1.3 Problem Analysis

Detecting violence in public settings, especially in hostel contexts, is a challenging task for current approaches. Despite being frequently used, manual surveillance and intervention strategies have drawbacks that reduce their efficacy.

1) Restricted Coverage: Blind spots and restricted coverage regions are common in surveillance systems, which lead to monitoring gaps where violent acts could go unnoticed.
2) Subjective Interpretation: When watching surveillance film, human viewers may perceive activities differently, which might result in a subjective evaluation of possible threats or a delay in reacting to violent situations.
3) Dynamic Environments: Because hostels are packed and dynamic places, it can be challenging to discern between appropriate behavior and possibly aggressive behaviour. This intricacy makes it more difficult to recognise violent occurrences and take prompt action.
4) Ineffective Communication: The reaction to violent situations is sometimes delayed and made worse by ineffective or sluggish lines of communication between authorities and surveillance systems.
5) Privacy Concerns: Residents may feel uneasy with continuous monitoring and data collecting in their living areas as a result of the deployment of surveillance technology.

6) Difficulties Particular to Hostels: Hostel environments have distinct obstacles, such as a dense population, a wide range of resident demographics, and insufficient monitoring outside of regular business hours. The frequency and seriousness of violent occurrences in hostel environments are influenced by these characteristics.
7) Stakeholder Perspectives: When solving violence detection difficulties, it is important to take into account the opinions and concerns of key stakeholders, such as hostel administrators, residents, law enforcement agencies, and local communities.
8) Our goal is to get a greater understanding of the intricacies and underlying causes of violence detection in hostel situations through the implementation of a thorough issue analysis. The results of this research will guide the creation of workable solutions that tackle these issues and improve security and safety for all parties concerned.

2.1.4 Solution

The creation of an automatic violence detection system specifically designed for dorm situations is our suggested course of action. Convolutional Neural Networks (CNNs), a cutting-edge deep learning approach, will be used to precisely identify violent behaviour in video footage in order to do this. To provide thorough coverage and monitoring, this system will effortlessly interface with current surveillance technologies, including motion sensors and CCTV cameras.

In addition, we will create and put into practice complex algorithms for picture improvement, violence detection, and frame selection. These algorithms will improve the system's dependability and precision, making it possible to identify violent situations quickly and precisely.

Furthermore, we will implement strong communication procedures to instantly alert authorities and relevant parties about any violent behaviour in real-time when it is detected at the hostel. To guarantee the appropriate use of data and monitoring measures, we will give ethical norms and privacy legislation top priority during the development process.

Our goal is to greatly increase hostel surroundings' safety and security while reducing the frequency and severity of violent occurrences by putting an all-encompassing solution into place.

2.2 Requirements

The particular features and attributes necessary for the automated violence detection system's effective development and implementation are described in the requirements section. These requirements cover both non-functional elements that describe how the system should operate and functional features that outline what the system should be able to achieve. We provide a clear framework for developing,

putting into practice, and assessing the system's performance in tackling the difficulties related to violence detection in public areas by outlining these needs. This part provides a thorough grasp of the system's goals and guarantees that it satisfies our requirements.

2.2.1 Functional Requirements

1) Video Analysis: The system must be capable of analysing video streams from surveillance cameras in real time.
2) Aggression Detection: It should be able to detect various forms of aggression, like physical altercations.
3) Real-time Alerting: The system should promptly alert relevant authorities when aggressive behavior is detected.
4) Performance Optimization: Optimization techniques should be implemented to ensure efficient processing and minimize processing time per frame.
5) Scalability: The system should be scalable to accommodate a varying number of surveillance cameras and handle large volumes of data.
6) Accuracy and Reliability: It must achieve high accuracy in aggression detection while minimizing false positives and negatives.
7) Adaptability: The system should adapt to different environments and lighting conditions without compromising performance.
8) User Interface: A user-friendly interface should be provided for system configuration, monitoring, and administration.

2.2.2 Non-Functional Requirements

1) Performance: The system should have low latency and high throughput to handle real-time processing of video and audio streams.
2) Reliability: It must be reliable enough, ensuring continuous operation without system failures or downtime.
3) Scalability: It should be designed to scale horizontally to accommodate an increasing number of cameras and users.
4) Usability: The system should be intuitive and easy to use for both administrators and end-users.
5) Maintainability: The system should be easy to maintain, with modular components and clear documentation for troubleshooting and updates.

6) Interoperability: It should integrate seamlessly with existing surveillance infrastructure and communication systems.
7) Resource Efficiency: It should be resource-efficient, minimizing hardware and network bandwidth requirements while maximizing performance.

2.3 Architecture Diagram

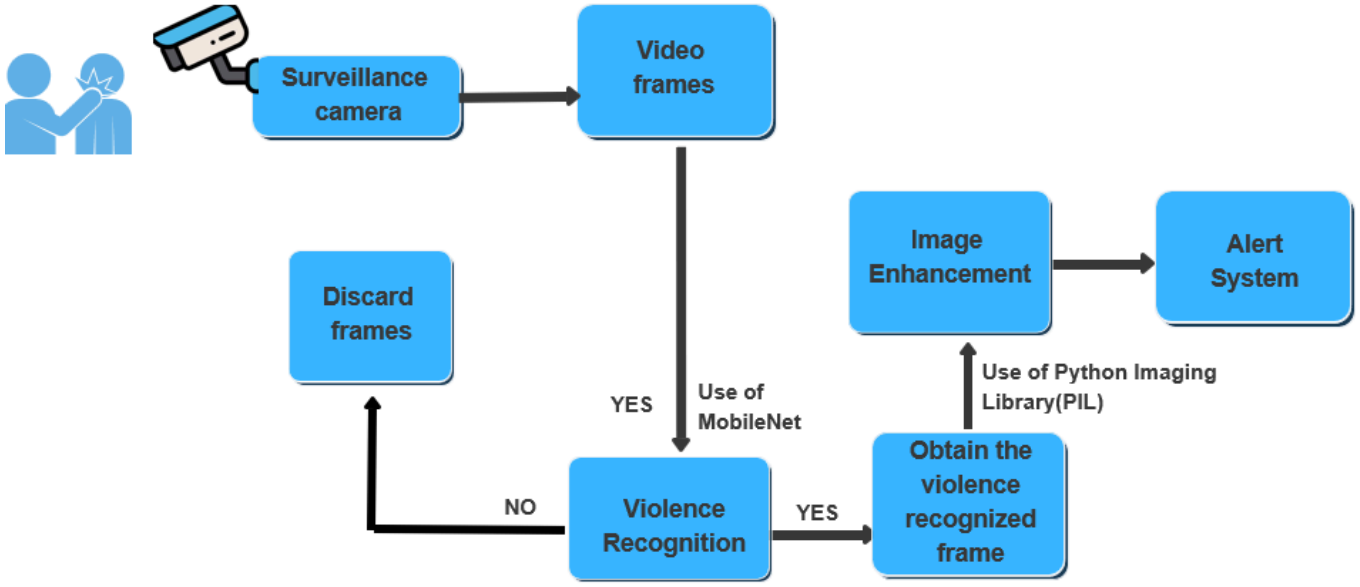


Figure 2.1: Architecture diagram

Chapter 03: IMPLEMENTATION

3.1 Problem Statement

The objective is to design and implement a real-time surveillance system capable of accurately detecting instances of violence and promptly alerting the relevant authorities to intervene.

3.2 Use Case Diagram

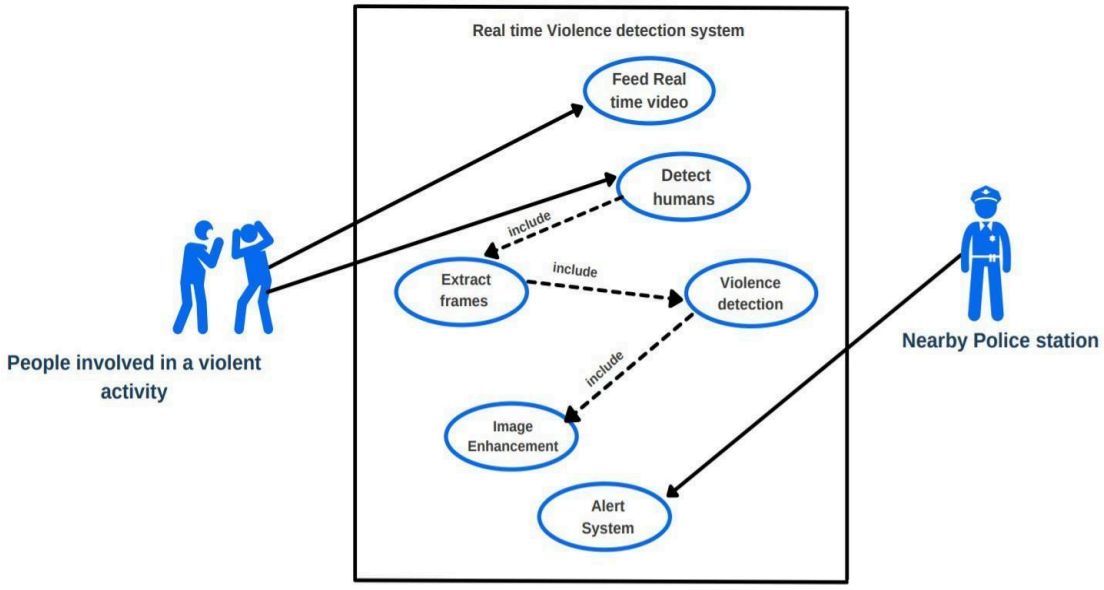


Figure 3.1: Use Case diagram

The rectangle aids in defining the suggested architecture's bounds. Everything that takes place inside the rectangle occurs inside the system. In this scenario, there are two actors: a local police station and individuals engaged in violent behaviour. Individuals engaged in violent activities are considered primary actors since a real-time system will be able to detect their behaviours. Use cases are oval-shaped representations of actions that complete specific tasks within the system. The system will attempt to identify persons whenever it receives a real-time video as input. Whenever human detection occurs, the specified use cases are also carried out. They are image enhancement, violence detection, and frames extraction. An alarm will be sent to a nearby authorities via the alert system following these three procedures.

3.3 Dataset Information

There are one thousand video clips in the dataset, divided into two categories: violent and non-violent. The majority of the five-second video clips are taken from CCTV footage, guaranteeing representative and varied samples.

350 violent and non-violent video clips from each class are chosen at random during the training procedure for each period. By ensuring that the model is adequately exposed to both classes, this balanced strategy minimises biases and maximises generalisation performance. The model can efficiently learn discriminative features suggestive of violence while capturing the features of non-violent behaviours by combining samples from both groups.

The violence detection model can be robustly trained using this dataset configuration, which allows it to categorise real-time video feeds and distinguish between violent and non-violent behaviour with good recall and precision. Furthermore, by training the model on data reflective of common surveillance scenarios, the use of CCTV footage improves the model's applicability in real-world scenarios.

Figure 3.2: The violence dataset's video clips

3.4 Architecture

The proposed architecture for the real-time violence detection and alert system consists of several key components:

1) **Frame extraction:** Individual frames are extracted from surveillance camera video recordings. Every frame shows the scene as it was at a certain point in time.
2) **MobileNet v2 Classifier:** The MobileNet v2 classifier uses the frames that were taken from the video feed as input. Pre-trained deep neural network MobileNet v2 is well known for its effectiveness and precision in picture categorization applications. In order to identify instances of violent activity, it examines the order of the input frames.
3) **Violence Detection:** The MobileNet v2 classifier recognises frames that show aggressive behaviour. A frame is eliminated if there is no evidence of violent conduct within it. Frames with violent evidence are kept for additional analysis.
4) **Frame Enhancement:** Frames identified as depicting violent activities undergo enhancement to improve clarity and facilitate better analysis. Enhancement techniques may include sharpening, contrast adjustment, and noise reduction, enhancing the visual information for accurate identification of individuals and actions involved in the altercation.
5) **Alert Generation:** Upon detecting a frame depicting violence, the system triggers an alert to notify the nearest authorities. The alert includes crucial information such as the enhanced frame, timestamp, and location of the incident. The alert is transmitted via a Telegram bot, ensuring swift communication and enabling immediate response from law enforcement agencies.

These parts come together to form a unified architecture that allows the system to identify violent behaviours in real time, improves the clarity of detected frames, and notifies authorities right away, allowing for swift action and protecting public safety.

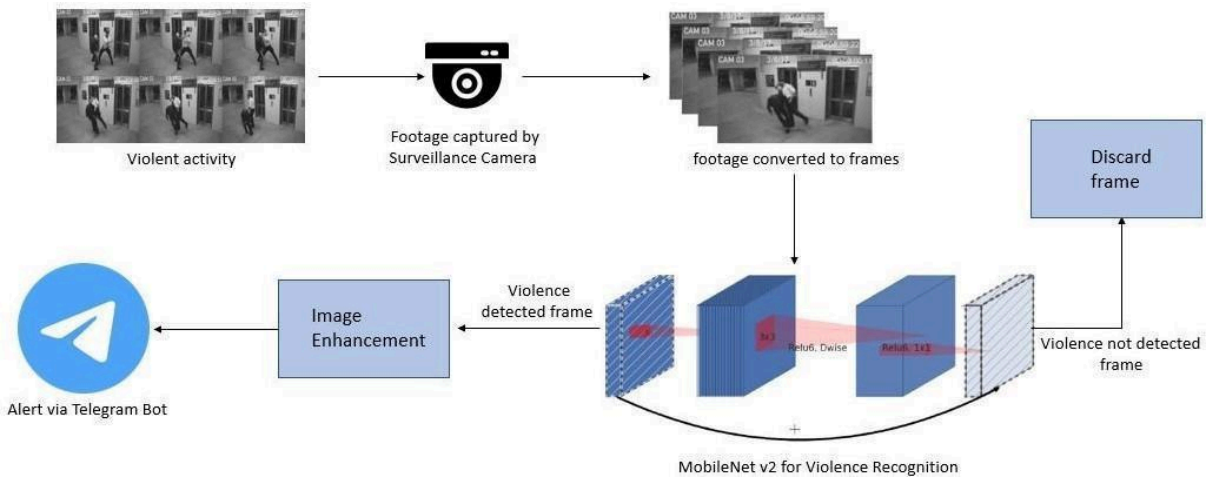


Figure 3.3: High level architecture diagram

3.4.1 MobileNet V2

The MobileNet architecture is characterized by its innovative use of depth-wise separable convolution, which decomposes traditional convolutions into two distinct stages: depth-wise convolution followed by point-wise convolution.

Two essential components comprise the MobileNet architecture: the resizing cell (with a stride of 2) and the residual cell (with a stride of 1). These cells are crucial to the structure of the network since they make it easier to extract features and reduce spatial resolution.

The notation "conv" in Figure 3.3 stands for a typical convolution operation, and the notation "dwese" is for depth-wise separable convolution. "Relu6" also denotes the Rectified Linear Unit (ReLU) activation function, limited to a maximum value of 6, while "Linear" denotes the linear activation function that is applied.

Two pivotal strategies introduced in MobileNetV2 are the linear bottleneck and inverted residual blocks. In the linear bottleneck layer, the channel dimension of the input is expanded to minimize the risk of information loss due to nonlinear operations such as ReLU activation. This approach acknowledges that information lost in some channels may be preserved in others, enhancing the network's representational capacity.

Unlike the traditional "wide-narrow-wide" arrangement seen in ordinary residual blocks, the inverted residual block employs a "narrow-wide-narrow" structure in the channel dimension. The memory footprint of the network can be decreased without sacrificing efficient information flow by creating skip links between fewer layers as opposed to bigger ones.

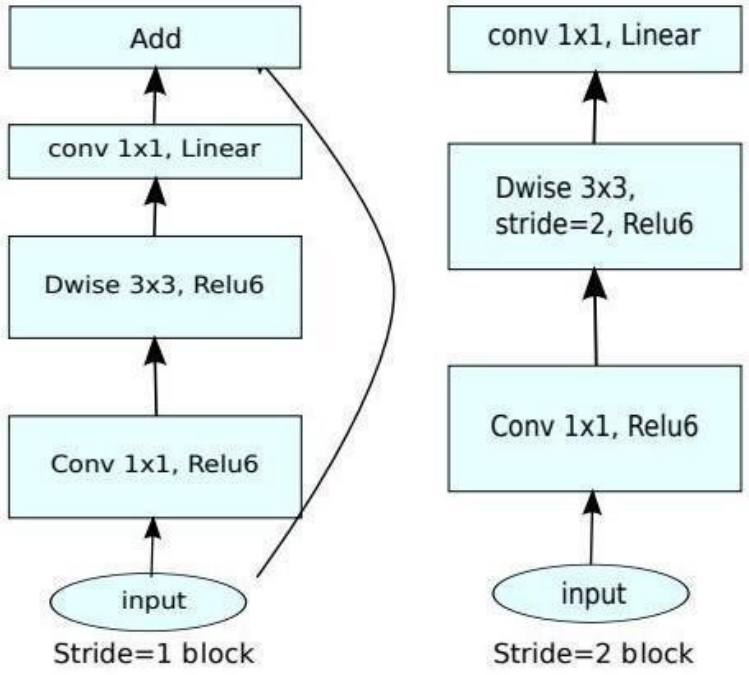


Figure 3.4: Architecture of MobileNet v2

3.4.2 Image Enhancement

We use the Python Imaging Library (PIL) for picture enhancement, which provides a large set of functions for image processing applications. PIL offers efficient image modification and support for a number of file formats.

Quick access to pixel data stored in many formats is made possible by PIL's Core Image Library, which offers a solid base for image processing operations.

The resulting output frames are improved in our system by a factor of two in terms of brightness and colour intensity. This improvement aids in enhancing the visual quality and clarity of the frames, which facilitates the identification of people and objects in the picture.

By leveraging the capabilities of PIL, we ensure efficient and effective image enhancement, enhancing the performance and accuracy of our real-time violence detection and alert system.

3.4.3 Alert Module

The alert module is essential in informing the appropriate authorities about violent incidents that are discovered. The architecture of the built alert system, which employs a particular logic to decide when to sound an alert, is depicted in Figure 3.4.

The system initializes a counter variable to one when it determines that a frame contains violence. After that, the algorithm looks at the next thirty frames to see whether any of them contain any violent scenes. Every time violence is recognised in a subsequent frame, the counter is increased. On the other hand, the counter variable is reset to zero and the system continues to check the next frames for violence if a frame is false for violence.

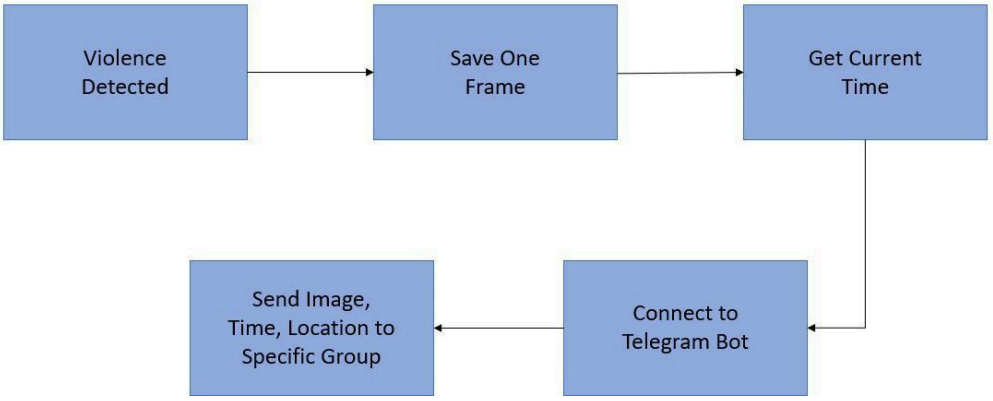


Figure 3.5: The Alert System's architecture diagram

Figure 3.5 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 3.6: Alert message Screenshot

3.5 Implementation

3.5.1 Video to Frames Conversion with Augmentation

The below code snippet implements a method called video_to_frames that makes it easier to apply picture augmentation techniques and convert a video file into a series of frames. Using OpenCV's VideoCapture function, a video capture object is initialised within the function to read the given video file. Using the imgaug library, the function iterates through each frame of the video and applies a variety of image augmentation techniques, such as rotation, horizontal flipping, zooming, and random brightness adjustment. Following a fixed-dimension resize, a list containing these enhanced frames is created. The function pulls different and augmented frames from the movie efficiently by skipping every seventh frame to reduce redundancy.

```python
import cv2
import os
import imageio
import imgaug.augmenters as iaa
import imgaug as ia
IMG_SIZE = 128
ColorChannels = 3
def video_to_frames(video):
    vidcap = cv2.VideoCapture(video)
    import math
    rate = math.floor(vidcap.get(3))
    count = 0
    ImageFrames = []
    while vidcap.isOpened():
        ID = vidcap.get(1)
        success, image = vidcap.read()
        if success:
            if (ID % 7 == 0):
                flip = iaa.Fliplr(1.0)
                zoom = iaa.Affine(scale=1.3)
                random_brightness = iaa.Multiply((1, 1.3))
                rotate = iaa.Affine(rotate=(-25, 25))
                image_aug = flip(image = image)
                image_aug = random_brightness(image = image_aug)
                image_aug = zoom(image = image_aug)
                image_aug = rotate(image = image_aug)
                rgb_img = cv2.cvtColor(image_aug, cv2.COLOR_BGR2RGB)
                resized = cv2.resize(rgb_img, (IMG_SIZE, IMG_SIZE))
                ImageFrames.append(resized)
            count += 1
        else:
            break
    vidcap.release()
    return ImageFrames
```

Figure 3.7: Videos to Frame Conversion with Augumentation Code

3.5.2 Model Construction for Violence Detection

The below code snippet describes how to build and assemble a deep learning model intended to detect violence. The MobileNetV2 architecture is used as the foundation model in the function `load_layers()`. It is specified with certain parameters, like the input shape and the exclusion of the top classification layers. The base model is then extended with a new classification head that creates binary predictions for violence detection by attaching a dense layer with a sigmoid activation function. Next, the model is compiled using the Adam optimizer with binary cross-entropy loss. For distributed training, the model instantiation takes place within a TPU strategy scope when a TensorFlow Processing Unit (TPU) is available. The summary of the constructed model provides an overview of its architecture, including layer configurations and parameter counts, facilitating insights into its structure and complexity.

```
epochs = 50

from keras import regularizers
kernel_regularizer = regularizers.l2(0.0001)

from keras.applications.mobilenet_v2 import MobileNetV2

def load_layers():
    input_tensor = Input(shape=(IMG_SIZE, IMG_SIZE, ColorChannels))
    baseModel = MobileNetV2(pooling='avg',
                            include_top=False,
                            input_tensor=input_tensor)

    headModel = baseModel.output
    headModel = Dense(1, activation="sigmoid")(headModel)
    model = Model(inputs=baseModel.input, outputs=headModel)

    for layer in baseModel.layers:
        layer.trainable = False

    print("Compiling model...")
    model.compile(loss="binary_crossentropy",
                  optimizer='adam',
                  metrics=["accuracy"])

    return model

if TPU_INIT:
    with tpu_strategy.scope():
        model = load_layers()
else:
    model = load_layers()

model.summary()
```

Figure 3.8: Model Construction for Violence Detection code

3.5.3 Training Configuration and Callbacks for Model

The code snippet demonstrates how to set up training parameters and use several callbacks to monitor and enhance a violence detection model's training process. Settings for learning rate scheduling, batch size, initial learning rate, and patience for early stopping are important elements. The code specifies custom callbacks, like `lr_callback`, which dynamically adjusts learning rates during training epochs, and `myCallback`, which terminates training early upon achieving a certain accuracy level. Standard callbacks such as `ModelCheckpoint`, `EarlyStopping`, `ReduceLROnPlateau`, and `TensorBoard` are also set up to log training metrics for visualization, avoid overfitting, save model weights, and modify learning rates. To ensure effective model training and optimization, the training loop applies the provided callbacks while iterating over epochs. After training is finished, the top-performing model weights are finally restored from the checkpoint file.

```
end_callback = myCallback()

lr_callback = LearningRateScheduler(lambda epoch: lrfn(epoch), verbose=False)

early_stopping = EarlyStopping(patience = patience, monitor='val_loss',
                               mode='min', restore_best_weights=True,
                               verbose = 1, min_delta = .00075)

PROJECT_DIR = MyDrive + '/RiskDetection'

lr_plat = ReduceLROnPlateau(patience = 2, mode = 'min')

os.system('rm -rf ./logs/')

import datetime
log_dir="logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = TensorBoard(log_dir = log_dir, write_graph=True, histogram_freq=1)

checkpoint_filepath = 'ModelWeights.h5'

model_checkpoints = ModelCheckpoint(filepath=checkpoint_filepath,
                                    save_weights_only=True,
                                    monitor='val_loss',
                                    mode='min',
                                    verbose = 1,
                                    save_best_only=True)


callbacks = [end_callback, lr_callback, model_checkpoints, tensorboard_callback, early_stopping, lr_plat]

if TPU_INIT:
    callbacks = [end_callback, lr_callback, model_checkpoints, early_stopping, lr_plat]
```

Figure 3.9: Training Configuration and Callbacks for Model

3.5.4 Violence Detection System

A real-time violence detection and alert system's operation is demonstrated by the provided code sample. The script analyses video input frame by frame after loading the trained violence detection model, applying the loaded model to each frame's prediction. To stabilise the output, forecasts are averaged over a historical series of forecasts. The frame is labelled appropriately and the output video is annotated with the detection findings if violence is identified with a probability greater than a predetermined threshold.

```
import numpy as np
import argparse
import pickle
import cv2
from google.colab.patches import cv2_imshow
import os
import time
from keras.models import load_model
from collections import deque

def print_results(video, limit=None):
        #fig=plt.figure(figsize=(16, 30))
        if not os.path.exists('output'):
            os.mkdir('output')

        print("Loading model ...")
        model = load_model('modelnew.h5')
        Q = deque(maxlen=128)
        vs = cv2.VideoCapture(video)
        writer = None
        (W, H) = (None, None)
        count = 0
        while True:
            # read the next frame from the file
            (grabbed, frame) = vs.read()

            # if the frame was not grabbed, then we have reached the end
            # of the stream
            if not grabbed:
                break

            # if the frame dimensions are empty, grab them
            if W is None or H is None:
                (H, W) = frame.shape[:2]
```

```
output = frame.copy()

frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
frame = cv2.resize(frame, (128, 128)).astype("float32")
frame = frame.reshape(128, 128, 3) / 255

# make predictions on the frame and then update the predictions
# queue
preds = model.predict(np.expand_dims(frame, axis=0))[0]
  print("preds",preds)
Q.append(preds)

# perform prediction averaging over the current history of
# previous predictions
results = np.array(Q).mean(axis=0)
i = (preds > 0.50)[0]
label = i

text_color = (0, 255, 0) # default : green

if label: # Violence prob
    text_color = (0, 0, 255) # red

else:
    text_color = (0, 255, 0)

text = "Violence: {}".format(label)
FONT = cv2.FONT_HERSHEY_SIMPLEX

cv2.putText(output, text, (35, 50), FONT,1.25, text_color, 3)

# check if the video writer is None
if writer is None:
    # initialize our video writer
    fourcc = cv2.VideoWriter_fourcc(*"MJPG")
    writer = cv2.VideoWriter("output/v_output.avi", fourcc, 30,(W, H), True)
```

```
            # write the output frame to disk
            writer.write(output)

            # show the output image
            cv2_imshow(output)
            key = cv2.waitKey(1) & 0xFF

            # if the `q` key was pressed, break from the loop
            if key == ord("q"):
                break
        # release the file pointersq
        print("[INFO] cleaning up...")
        writer.release()
        vs.release()
```

```
V_path = "V_19.mp4"
N_path = "nonv.mp4"
```

```
print_results(V_path)
```

Figure 3.10: Violence Detection System Code

Figure 3.11: Violence Detection System Output

3.5.5 Violence Alert System

This below code demonstrates a real-time violence detection system that can send out alerts when it notices violent activity. Using OpenCV, the system processes video frames in real-time for violence detection after loading a pre-trained neural network model. Based on its predictions, the model evaluates each frame to determine the likelihood of violence. The system saves the analysed video and annotates the frame appropriately if violence is identified. Furthermore, the system activates an alert mechanism after identifying a predetermined quantity of consecutive violent frames. It notifies a certain Telegram group of the occurrence, along with the incident's location and timestamp. It also includes an image of the violent action that was discovered with the alert message so that it can be examined in more detail.

```
import numpy as np
import argparse
import pickle
import cv2
from google.colab.patches import cv2_imshow
import os
import time
from keras.models import load_model
from collections import deque

def print_results(video, limit=None):
        trueCount = 0
        imageSaved = 0
        filename = 'savedImage.jpg'
        finalImage = 'finaImage.jpg'
        sendAlert = 0
        location = "Bangalore"
        #fig=plt.figure(figsize=(16, 30))

        print("Loading model ...")
        model = load_model('/home/modelnew.h5')
        Q = deque(maxlen=128)
        vs = cv2.VideoCapture(video)
        writer = None
        (W, H) = (None, None)
        count = 0
        while True:
            # read the next frame from the file
            (grabbed, frame) = vs.read()

            # if the frame was not grabbed, then we have reached the end
            # of the stream
            if not grabbed:
                break
```

```
    # check if the video writer is None
    if writer is None:
        # initialize our video writer
        fourcc = cv2.VideoWriter_fourcc(*"MJPG")
        writer = cv2.VideoWriter("recordedVideo.avi", fourcc, 30,(W, H), True)

    # write the output frame to disk
    writer.write(output)

    # show the output image
    cv2_imshow(output)

    if(trueCount == 50):
      if(imageSaved == 0):
        if(label):
          cv2.imwrite(filename, output)
          imageSaved = 1

      if(sendAlert == 0):
        timeMoment = getTime()
        bot = telepot.Bot('5309305007:AAGiCHJzRGOF6Bu3QxyMJPigG_k_MQ-NU20')
        bot.sendMessage(-1001522775837, f"VIOLENCE ALERT!! \nLOCATION: {location} \nTIME: {timeMoment}")
        bot.sendPhoto(-1001522775837, photo=open(filename, 'rb'))
        sendAlert = 1

    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
        break
# release the file pointersq
print("[INFO] cleaning up...")
writer.release()
vs.release()
```

Figure 3.12: Violence Alert System Code

Figure 3.13: Violence Alert System Output

Chapter 04: RESULTS

4.1 Discussion on the Results Achieved

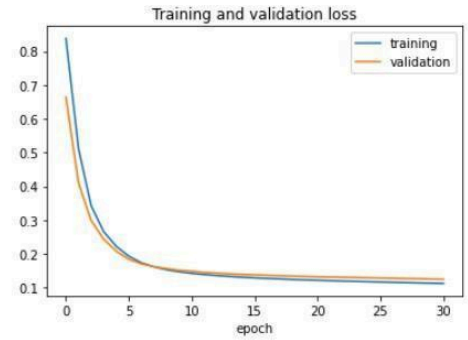
This section presents graphical representations of the MobileNet v2 model's testing and training results. The training and testing accuracy and loss obtained using a dataset of 1000 films, each lasting an average of 7 seconds, are shown in Figure 4.1. 350 films from the violence class and 350 from the non-violence class are used each epoch of instruction. Remarkably, 96% of training accuracy and 95% of testing accuracy are reached when analysing CCTV material that is not part of the dataset.

Accuracy and loss trends over several epochs are displayed in Figure 4.1's graph. Significantly, after around 5 epochs, both accuracy and loss stabilise and achieve a constant level of increment and decrement. Further information about the model's performance in other classes is shown in Figure 4.1, along with the confusion matrix that was obtained and additional evaluation parameters.

Moreover, video frames from a violent and a non-violent video clip are shown in Figures 4.3 and 4.4, respectively, to further visually demonstrate the efficacy of the model. A frame from the movie labelled as violent action is shown in Figure 4.3, illustrating the model's accuracy in identifying violent scenes. On the other hand, the model accurately classified the frame in Figure 4.4, which shows no aggressive behaviour, as false or non-violent.

Overall, these findings highlight the MobileNet v2 model's excellent accuracy and performance across training and testing datasets, which highlight the model's effectiveness in properly identifying violent events in real-time video streams.

```
Best Epochs:  31
Accuracy on train: 0.9616789817810059   Loss on train: 0.11210563778877258
Accuracy on test: 0.9577874541282654    Loss on test: 0.116333968937397
```




```
<Figure size 432x288 with 0 Axes>
```

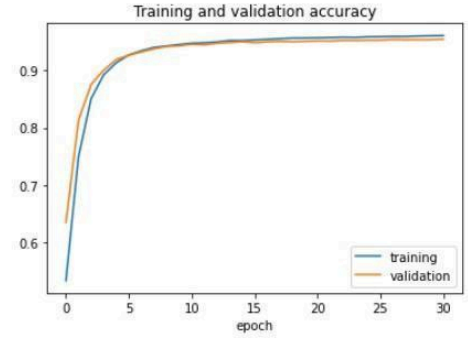
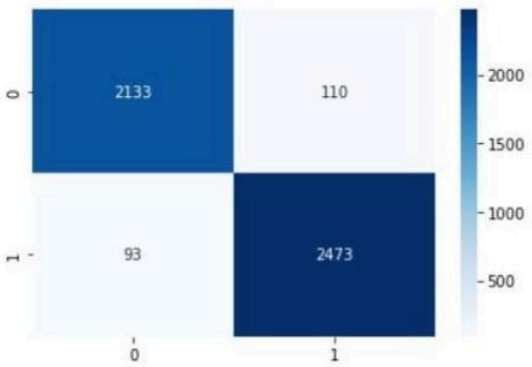


Figure 4.1: The training set's accuracy and error

```
> Correct Predictions: 4606
> Wrong Predictions: 203
```



	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 4.2: The trained model's confusion matrix

Figure 4.3: Output frame that recognized violence

Figure 4.4: Output frame that did not recognize violence

4.2 Comparison with CNN-LSTM architecture

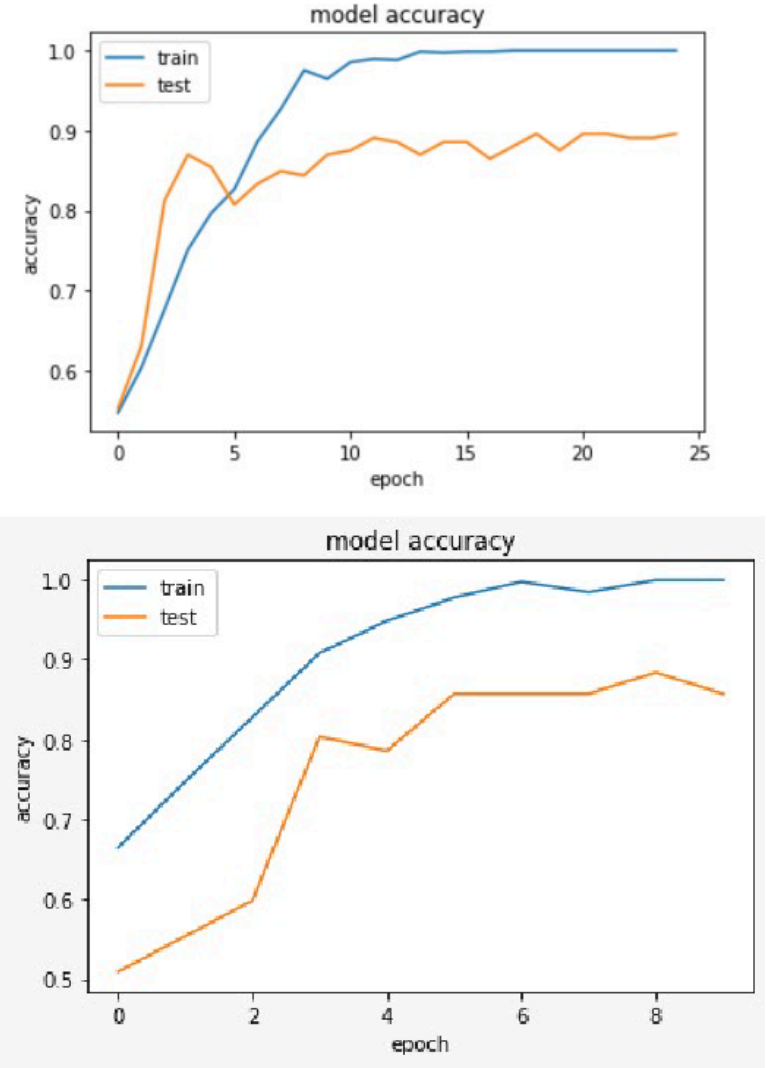


Figure 4.5: Comparison of Training and Testing accuracy of MobileNet v2 and CNN-LSTM Models

	precision	recall	f1-score	support
0	0.93	0.82	0.87	66
1	0.85	0.95	0.90	74
accuracy			0.89	140
macro avg	0.89	0.88	0.88	140
weighted avg	0.89	0.89	0.88	140

Figure 4.6: CNN-LSTM Model Evaluation Metrics

In terms of violence detection, Figure 4.1's comparison amply illustrates how much better

MobileNet v2 is than CNN-LSTM. The graphs show that, in terms of accuracy and loss among other evaluation measures, MobileNet v2 performs better than the CNN-LSTM model.

These results imply that MobileNet v2 performs more effectively and efficiently when it comes to identifying pertinent characteristics and patterns linked to violent activities in live video broadcasts. Through the use of its linear bottleneck and depth-wise separable convolution techniques, MobileNet v2 outperforms conventional CNN-LSTM models and exhibits strong performance.

The noted enhancements highlight MobileNet v2's capacity to become the cutting-edge model for applications involving real-time violence detection. Because of its high processing accuracy and efficiency, it is a good fit for surveillance systems that try to improve security and safety for the general population.

Overall, the comparison demonstrates MobileNet v2's effectiveness and its ability to establish new standards in the field of violence detection, opening the door for sophisticated and successful approaches to societal issues pertaining to violence prevention and intervention.

4.3 Limitation of the Minor Project

Although the results of the suggested real-time violence detection system are encouraging, there are a few things to keep in mind:

1) **Restricted Variation in Training Data:** The training dataset's representativeness and variety have a major impact on the model's efficacy. The model can find it difficult to generalise to new or unfamiliar situations if the dataset is predominantly composed of particular kinds of violent acts or events.
2) **Hardware Dependency:** The implementation of real-time surveillance systems necessitates a strong hardware foundation that can instantly process high-definition video streams. Inadequate computational resources could limit the scalability or performance of the system.
3) **Environmental Factors:** Variations in lighting, occlusions, or camera angles can all have an effect on how well the system performs. The accuracy and dependability of the model may be impacted by these variables introducing noise or irregularities into the input data.
4) **False Positives/Negatives:** Despite the great accuracy of the model, there remains a chance of false positives, which mistakenly label non-violent behaviours as violent, or false negatives, which fail to identify real violent incidents. These mistakes have the potential to compromise the system's efficacy and dependability.
5) **Privacy Concerns:** When it comes to the gathering and use of people's video data, the installation

of surveillance devices presents moral and privacy issues. To guarantee that the system is accepted and complies with legal requirements, it is crucial to strike a balance between individual privacy rights and public safety.

6) **Integration Challenges:** There may be technological difficulties and a need for seamless interoperability when integrating the real-time violence detection system with the current surveillance infrastructure and communication channels (such as the Telegram bot for informing authorities).
7) **Adversarial Attacks:** The system could be susceptible to hostile actors manipulating video input on purpose in order to trick the model or set off false alarms. The system's robustness against these kinds of attacks is essential to preserving its dependability and credibility.
8) **Localization Accuracy:** There may be differences in the reported incident location depending on how well the surveillance camera is located. This might have an impact on how quickly and successfully law enforcement intervenes.

To overcome these constraints, continuous research and development endeavours are necessary to augment the resilience, expandability, and moral implications of real-time violence detection systems. In addition, to guarantee the system's efficacy and societal impact, ongoing assessment and validation of its operation in real-world circumstances are crucial.

4.4 Future Work

This model could be upgraded to work in multiple cameras con nected 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] J.F.P. Kooij, M.C. Liem, J.D. Krijnders, T.C. Andringa, D.M. Gavrila, Multi-modal human aggression detection, Computer Vision and Image Understanding,Volume 144, 2016, Pages 106-120, ISSN 1077-3142.
https://doi.org/10.1016/j.cviu.2015.06.009

[2] M. -S. Kang, R. -H. Park and H. -M. Park, "Efficient Spatio-Temporal Modeling Methods for Real-Time Violence Recognition," in IEEE Access, vol. 9, pp. 76270-76285, 2021, doi: 10.1109/ACCESS.2021.3083273.
https://sci-hub.se/10.1109/access.2021.3083273

[3] Laptev and Lindeberg, "Space-time interest points," Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 2003, pp. 432-439 vol.1, doi: 10.1109/ICCV.2003.1238378. keywords: {Event detection;Computer vision;Image motion analysis;Motion analysis;Layout;Indexing;Optical computing;Spatiotemporal phenomena;Acceleration;Videoconference},
https://sci-hub.yncjkj.com/10.1109/iccv.2003.1238378

[4] Zhou P, Ding Q, Luo H, Hou X (2018) Violence detection in surveillance video using low-level features. PLoS ONE 13(10): e0203668. https://doi.org/10.1371/journal.pone.0203668

[5] Ullah, F.U.M.; Ullah, A.; Muhammad, K.; Haq, I.U.; Baik, S.W. Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network. *Sensors* **2019**, *19*, 2472. https://doi.org/10.3390/s19112472

[6] Irfanullah, Hussain, T., Iqbal, A. *et al.* Real time violence detection in surveillance videos using Convolutional Neural Networks. *Multimed Tools Appl* **81**, 38151–38173 (2022). https://doi.org/10.1007/s11042-022-13169-4

[7] Baba, M.; Gui, V.; Cernazanu, C.; Pescaru, D. A Sensor Network Approach for Violence Detection in Smart Cities Using Deep Learning. *Sensors* **2019**, *19*, 1676. https://doi.org/10.3390/s19071676

[8] J.F.P. Kooij, M.C. Liem, J.D. Krijnders, T.C. Andringa, D.M. Gavrila, Multi-modal human aggression detection, Computer Vision and Image Understanding,Volume 144, 2016, Pages 106-120, ISSN 1077-3142.
https://doi.org/10.1016/j.cviu.2015.06.009

[9] Sandler, Mark & Howard, Andrew & Zhu, Menglong & Zhmoginov, Andrey & Chen, Liang-Chieh. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 4510-4520. 10.1109/CVPR.2018.00474.
https://arxiv.org/abs/1801.04381v4

[10] W. Zajdel, J. D. Krijnders, T. Andringa and D. M. Gavrila, "CASSANDRA: audio-video sensor fusion for aggression detection," 2007 IEEE Conference on Advanced Video and Signal Based Surveillance, London, UK, 2007, pp. 200-205, doi: 10.1109/AVSS.2007.4425310. keywords: {Sensor fusion;Humans;Layout;Frequency;Speech;Surveillance;Kinetic energy;Artificial intelligence;Bayesian methods;Aggregates},

aayudh

ORIGINALITY REPORT

7%	5%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijraset.com Internet Source	2%
2	journals.plos.org Internet Source	<1%
3	www.researchgate.net Internet Source	<1%
4	Submitted to Mepco Schlenk Engineering college Student Paper	<1%
5	Fang, Xiaoyu, Yonghong Tian, Yaowei Wang, Chi Su, Teng Xu, Ziwei Xia, and Wen Gao. "Pair-wise event detection using cubic features and sequence discriminant learning", 2013 IEEE International Conference on Multimedia and Expo (ICME), 2013. Publication	<1%
6	Submitted to The University of Texas at Arlington Student Paper	<1%
7	www.siftdesk.org Internet Source	

		<1%
8	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	<1%
9	Submitted to Australasian Academy of Higher Education Student Paper	<1%
10	Submitted to Columbia University Student Paper	<1%
11	Submitted to Technological Institute of the Philippines Student Paper	<1%
12	Submitted to National College of Ireland Student Paper	<1%
13	Submitted to University of South Florida Student Paper	<1%
14	Submitted to Foundation for Liberal And Managment Education Student Paper	<1%
15	Submitted to University of Essex Student Paper	<1%
16	Submitted to Wilmington University Student Paper	<1%
17	www.intel.ai Internet Source	

		<1 %
18	Ileana Scarpino, Chiara Zucco, Mario Cannataro. "Characterization of Long COVID using text mining on narrative medicine texts", 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2021 Publication	<1 %
19	ijsrst.com Internet Source	<1 %
20	repository.unimilitar.edu.co Internet Source	<1 %
21	www.coursehero.com Internet Source	<1 %
22	dokumen.pub Internet Source	<1 %
23	pdfs.semanticscholar.org Internet Source	<1 %
24	W. A. Block, J. L. Fridley. "SIMULATION OF FOREST HARVESTING USING COMPUTER ANIMATION", Transactions of the ASAE, 1990 Publication	<1 %
25	Weishi Xu, Runjie Wang. "ALAD-YOLO:an lightweight and accurate detector for apple leaf diseases", Frontiers in Plant Science, 2023 Publication	<1 %

26	link.springer.com Internet Source	<1 %
27	www.mdpi.com Internet Source	<1 %
28	Aayush Jain, Dinesh Kumar Vishwakarma. "Deep NeuralNet For Violence Detection Using Motion Features From Dynamic Images", 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020 Publication	<1 %
29	Lecture Notes in Computer Science, 2009. Publication	<1 %
30	Lecture Notes in Computer Science, 2013. Publication	<1 %
31	Piyush Vashistha, Charul Bhatnagar, Mohd Aamir Khan. "An architecture to identify violence in video surveillance system using ViF and LBP", 2018 4th International Conference on Recent Advances in Information Technology (RAIT), 2018 Publication	<1 %
32	openaccess.hacettepe.edu.tr Internet Source	<1 %
33	pure.tue.nl Internet Source	<1 %

34	"Advances in Information and Communication", Springer Science and Business Media LLC, 2023 Publication	<1%
35	Irfanullah, Tariq Hussain, Arshad Iqbal, Bailin Yang, Altaf Hussain. "Real time violence detection in surveillance videos using Convolutional Neural Networks", Multimedia Tools and Applications, 2022 Publication	<1%
36	Dipanjan Sarkar, Raghav Bali, Tushar Sharma. "Practical Machine Learning with Python", Springer Science and Business Media LLC, 2018 Publication	<1%
37	Fath U Min Ullah, Amin Ullah, Khan Muhammad, Ijaz Ul Haq, Sung Wook Baik. "Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network", Sensors, 2019 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		