Detection of Criminal Activities Through CCTV Surveillance

Richita Karira, Manav Keswani, Tanmay Chaudhary, Soumil Tawde
Computer Engineering, Vivekanand Education Society's
Institute Of Technology, India.

***Abstract*— *Hearing about the violent activities that occur on a daily basis around the world is quite overwhelming. Personal safety and social stability are seriously threatened by the violent activities. A variety of methods have been tried to curb the violent activities which includes installing surveillance systems. It will be of great significance if the surveillance systems can automatically detect violent activities and give warning or alert signals.*

The whole system can be implemented with a sequence of procedures. Firstly, the system has to identify the presence of human beings in a video frame. Then, the frames which are predicted to contain violent activities have to be extracted. The irrelevant frames are to be dropped at this stage. Finally, the trained model detects violent behavior and these frames are separately saved as images. These images are enhanced to detect faces of people involved in the activity, if possible. The enhanced images along with other necessary details such as time and location is sent as an alert to the concerned authority.

The proposed method is a deep learning based automatic detection approach that uses Convolutional Neural Network to detect violence present in a video. But, the disadvantage of using just CNN is that it requires a lot of time for computation and is less accurate.[3] Hence, a pre-trained model, MobileNetv2, which provides higher accuracy and acts as a starting point for the building of the entire model. An alert message is given to the concerned authorities using a telegram application.

Keywords*— *Criminal detection, CCTV footage analysis, Video surveillance, Machine learning in video analysis, Convolutional Neural Networks

I. INTRODUCTION

A. Overview

Violent behavior in public places is a pressing issue that necessitates urgent attention. Such acts of violence not only disrupt social harmony but also erode communities, leading to reduced productivity, diminished property values, and disruptions in essential social services. Moreover, violence poses a significant public health concern, affecting individuals across all age groups, from infants to the elderly. Recognizing and addressing violence in a timely and efficient manner is a complex challenge, particularly because it must be detected in real-time videos captured by a multitude of surveillance cameras situated in diverse locations. The primary objective of this project is to develop a robust real-time violence alert system that can reliably detect and promptly alert the relevant authorities when violent activities occur.

Furthermore, The experimental results demonstrate the superiority of the proposed MIL-based violence detection method over existing state-of-the-art approaches. The system architecture employs MobileNetV2 for efficiency and accuracy in real-time detection.

Upon the violence detection, the system triggers an alert message containing location coordinates, a timestamp, and a map link for visual reference, along with capturing photos of individuals involved in the violent activity for further analysis on Telegram using Telepot . A server-side beep sound is activated to draw immediate attention to the violent event. This comprehensive approach offers a robust solution for real-time CCTV violence detection, with potential applications in enhancing security and public safety.

B. Problem Statement

The problem at hand is the real-time detection of violent activities within video footage obtained from public surveillance cameras. The challenge lies in differentiating between regular activities and violent behavior, promptly alerting the authorities, and ensuring that the system functions with high accuracy and efficiency. This problem addresses the critical need for automated violence detection to alleviate the burden on human surveillance, allowing for faster response times and enhanced public safety.

C. Objective

An automated violence detection system leveraging deep learning methodologies, specifically MobileNetv2, is designed to analyze video frames in real-time.[17] The system efficiently identifies human presence within video frames, discerning frames indicative of violent activities while filtering out irrelevant ones. By employing advanced techniques, it enhances extracted frames to improve the identification of individuals involved in the violent behavior, ensuring precision in detection. Crucial information such as timestamp and location of the incident is collected for comprehensive analysis. Furthermore, the system incorporates a real-time alert mechanism that promptly notifies concerned authorities, utilizing its alert module. The primary objective is to achieve high accuracy and efficiency in violence detection, thereby enhancing public safety and minimizing the necessity for manual video monitoring.

D. Motivation for the Project

The motivation behind this project stems from the urgent need to enhance public safety and social stability by addressing the menace of violence in public spaces. Across the globe, violence has far-reaching consequences, and it is imperative to harness the power of technology to mitigate its impact. Existing public video surveillance systems, while valuable, are limited by the human effort required to monitor hours of footage to identify fleeting moments of violence. This project is inspired by the potential of deep learning techniques, particularly Convolutional Neural Networks (CNN), in automating the detection of violence in video streams, thus enabling swift and accurate response from law enforcement agencies.[5] By leveraging the capabilities of pre-trained models like MobileNetv2, The system aims to create an efficient and accurate violence detection system that can reduce the burden on human surveillance. [16] [17]

II. LITERATURE SURVEY

A. Survey of Existing System

Paper Name	Methodology	Data set	Limitations
"Detection of Real-world Fights in Surveillance Videos" Authors: Mauricio Perez, Alex C. Kot Published in: ICASSP 2019	The paper proposes a pipeline for fight detection in surveillance videos.The authors evaluate different feature extraction methods, including Deep Learning and Local Interest Points.	A novel and challenging dataset called CCTV Fights. It contains 1,000 videos of real fights, with 8 hours of annotated CCTV footage.The videos cover various locations. The dataset includes both CCTV footage and videos recorded from various types of	Effectiveness of existing surveillance systems in detecting fights is questionable, as they rely on continuous human supervision.
"Vision-based Fight Detection from Surveillance Cameras" Authors: Gözde Ayşe Tataroğlu, Şeymanur Hazım Kemal Ekenel Published in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019	Utilizes a combination of a Bi-LSTM network and a self-attention layer. The authors also employ Convolutional Neural Networks (CNNs) for feature extraction, specifically VGG16 and Xception architectures.	A new surveillance camera fight dataset. The dataset consists of fight scenes from surveillance camera videos available on YouTube, as well as security camera footage from various locations.	1. The results for the surveillance camera dataset were not as good as the other datasets, indicating the challenge of generalizing to diverse and real world surveillance camera footage. 2. The Peliculas dataset used in the experiments had a limited number of fight scene samples, which affected the accuracy of the model on this dataset.
"Fight Detection in Surveillance Videos" Authors: Ersin Esen, Mehmet Ali Arabaci, Medeni Soysal Published in: 11th International Workshop on Content-Based Multimedia Indexing (CBMI), 2013	A novel method for fight detection in surveillance videos using a motion feature called Motion Co-occurrence Feature (MCF). The algorithm utilizes a k-Nearest Neighbor (kNN) classifier for fight detection.	The authors evaluated their algorithm using a dataset that includes samples from existing databases (BEHAVE and CAVIAR) and videos collected from the Internet. The authors emphasize the need for more diverse datasets to improve fight detection algorithms.	The paper acknowledges the limited number of studies and datasets specifically focused on fight detection. Existing datasets from movies or sports events may not be suitable for fight detection in surveillance cameras. The authors highlight the importance of more diverse and challenging datasets to enhance the performance of fight detection algorithms.

B. Limitations of Existing System

While these approaches have made significant contributions to violence detection, they exhibit several limitations. Many of these methods involve high computational costs, making them unsuitable for real-time applications. Additionally, they may suffer from low detection rates and high false alarms in crowded scenes or occlusion-prone environments. The computational demands of feature extraction, as seen in the STIP method, may be impractical for use in surveillance and media rating systems. Moreover, sensor-network approaches require substantial computational power, which can be a hindrance to real-time deployment.

“Detection of Real-world Fights in Surveillance Videos," presented at ICASSP 2019, Mauricio Perez and Alex C. Kot propose a comprehensive pipeline for identifying fights in surveillance videos. Their approach involves evaluating various feature extraction techniques, including Deep Learning and Local Interest Points, and experimenting with different classifiers such as end-to-end CNN, LSTM, and SVM. The authors find that the Two-Stream approach, which combines spatial and temporal features, yields the best performance in fight detection. To facilitate research in this area, they introduce a novel dataset named CCTV-Fights, comprising 1,000 videos of real fights with over 8 hours of annotated CCTV footage captured in diverse locations such as public spaces, schools, malls, and residential or commercial areas. This dataset encompasses footage from different sources like mobile cameras, car dash-cams, and aerial drones, providing a more realistic and challenging testbed for surveillance system evaluation.

Authors emphasize limitations of existing surveillance systems, which heavily rely on human supervision, and highlight the scarcity of datasets containing real-world fight scenarios captured by surveillance cameras. They underscore the necessity for enhancing spatial features, leveraging sequential information, and developing early detection methods to address these challenges and improve the effectiveness of fight detection systems.

III. REQUIREMENTS

A. Functional Requirements

- Real Time analysis: The system should be able to analyze the footage captured by CCTV in real-time, enabling the detection and identification of potential criminal activities as they unfold.
- Violence detection: By utilizing machine learning algorithms, the system should have the ability to detect violent activities within the real time web cam footage, including unexpected movements, suspicious gatherings, or abnormal patterns.
- Event Notification: Whenever potential violent activities are detected, the system should promptly generate alerts or notifications to the appropriate authorities or security personnel with the server-side beep sound activated to draw immediate attention to the violent event.
- Acquisition of Live CCTV Footage: The system should possess the capability to securely acquire real-time footage from multiple CCTV cameras simultaneously.

B. *Non Functional Requirements*

- Accuracy: The system must exhibit a high degree of accuracy in accurately identifying and categorizing potential criminal activities, minimizing both false positives and false negatives.
- Speed and Efficiency: Real-time analysis of web cam footage should be conducted with minimal delay, enabling swift response and intervention when necessary.
- Reliability: The system should be robust and dependable, ensuring uninterrupted operation and minimal downtime to avoid any surveillance gaps.

C. *Software Requirements*

- Programming Language: The software solution should be implemented using the Python programming language, taking advantage of its rich libraries and frameworks for efficient development.
- Deep Learning Framework: TensorFlow should be utilized as the core deep learning framework for building and training precise machine learning models.
- Real-Time Video Processing: The software should incorporate OpenCV for real-time video processing capabilities, encompassing functions like video acquisition, frame manipulation, and feature extraction.

D. *Hardware Requirements*

- Processing Units: To ensure smooth operation of the software, it is recommended to have a minimum hardware configuration. This includes a computer or server with at least 4GB of RAM to efficiently handle the video processing tasks.
- CPU: An Intel Core i5 or an equivalent processor is recommended as a baseline for managing the computational requirements of real-time video processing and deep learning algorithms.
- Storage: It is necessary to have a minimum of 15GB of available storage space to accommodate the software application, libraries, and any additional data required for the system's operation.
- Graphics Processing Unit (GPU): Although not mandatory, the utilization of a dedicated GPU can significantly enhance the performance of deep learning algorithms and video processing tasks. The recommendation for a compatible GPU may vary depending on the system's complexity and deployment scale

IV. SYSTEM ANALYSIS & DESIGN

A. *Analysis*

The proposed surveillance system utilizes video footage from real time web cameras to monitor and identify potentially illicit or suspicious behavior in public areas. Following a structured framework, the system undergoes several preliminary steps, including feature extraction from the video and subsequent detection processes. These steps aim to demonstrate the system's efficacy in recognizing and addressing criminal activities.

Multiple security cameras contribute input video to the system. A deep learning model, specifically employing Convolutional Neural Networks (CNNs), is employed for training using the frames extracted from the video footage. Subsequently, the trained model undergoes testing using a continuous stream of video frames captured by various cameras to detect any suspicious or unlawful activities.[1]

B. *System Design*

The architecture of the system involves the processing of camera footage, including frame extraction, violence detection, image enhancement, and alert notification. The core component is the MobileNetV2 classifier for violence detection, and the system employs image enhancement using the Python Imaging Library (PIL). Additionally, an alert module is incorporated to send notifications to the authorities upon detecting violent activities in the video stream.

The methodology applied in the system relies on deep learning, specifically the use of the MobileNetV2 architecture for violence detection. This method takes advantage of spatio-temporal features to identify violent activities in real-time video streams.[3][4] Additionally, image enhancement is performed using PIL to improve the quality of the frames related to violent incidents.[17]

If violence detection occurs continuously for 20 consecutive frames, an alert is sent to the server side with the message "VIOLENCE ALERT!" along with the latitude, longitude, and a map link of the violence area. Furthermore, images of the individuals detected engaging in the violent behavior are included in the alert, providing additional context for prompt action by authorities.

By combining CNNs with the specialized MobileNetV2 architecture, the current system demonstrates a robust approach to violence detection, ensuring effective surveillance and proactive measures to maintain public safety.[1] Moreover, the incorporation of image enhancement techniques further enhances the system's ability to accurately identify and address violent incidents captured within the surveillance footage.

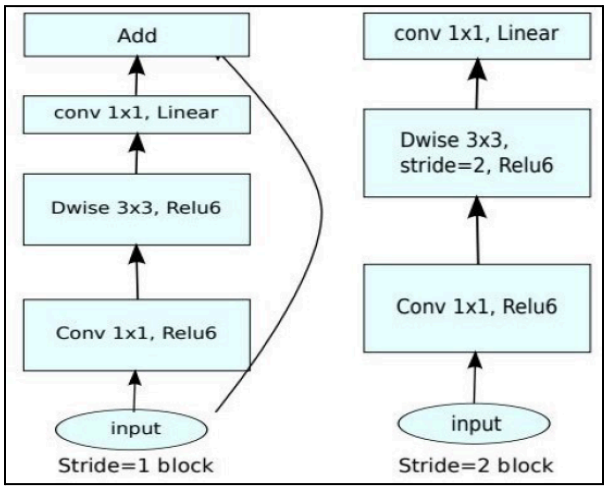

Fig 1: MobileNetv2 architecture

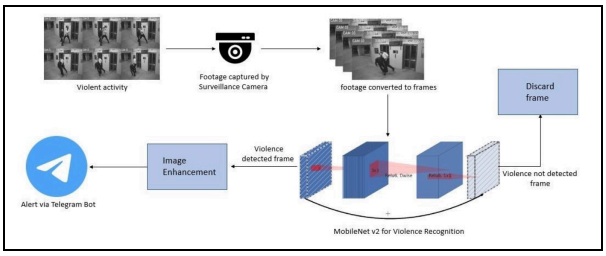

Fig 2: Algorithm / Process Design of Implementation

V. IMPLEMENTATION

A. Introduction

The implementation introduction provides an overview of the approach taken to develop the proposed system, highlighting key components and methodologies utilized. It outlines the main objectives of the implementation phase and briefly introduces the key elements involved in the system's realization. Overall, the implementation introduction serves to set the stage for the subsequent detailed explanation of the system's development and functionality.

B. Implementation

The current system implementation started by compiling a dataset of 1000 video clips, each with an average duration of 5 seconds, sourced primarily from CCTV footage. From this dataset, 350 videos depicting violent incidents and 350 videos of non-violent scenarios were selected for training purposes.

The current system incorporates a user interface (UI) for seamless interaction. Users are prompted to create an account and log in to access the system. Upon logging in, they are presented with a user-friendly interface where they can initiate the violence detection process by clicking on the "Start" button. Once activated, the system prompts users to capture footage from their CCTV, initiating real-time detection. This intuitive UI streamlines the process, enabling users to effortlessly engage with the system and contribute to the surveillance and monitoring of public spaces for violent activities.

Once video is captured through CCTV. Video processing is conducted using OpenCV to extract frames efficiently, enabling subsequent analysis and feature extraction. Concurrently, TensorFlow was integrated into the system for deep learning tasks, facilitating the incorporation of the MobileNetV2 architecture, renowned for its efficacy in violence detection within video sequences. [16]

Simultaneously, alongside TensorFlow integration, the MobileNetV2 architecture was coupled with the Python Imaging Library (PIL) for image enhancement. This combination allowed for the refinement of frame clarity and quality, enhancing the system's ability to accurately detect and classify violent behavior in real-time surveillance footage.

A critical addition to the current system is the implementation of an alert mechanism triggered by continuous violence detection. Upon detecting violence continuously for 20 consecutive frames, an alert is generated on the server side, accompanied by a distinctive beep sound for immediate attention. Furthermore, the alert notification is dispatched to a Telegram channel using the Telepot library. A Telegram bot was developed specifically for this purpose, enabling seamless communication of alerts to relevant authorities. The alert message includes a notification stating "VIOLENCE ALERT!" along with the latitude and longitude coordinates of the violence area. Additionally, a map link is provided for quick visualization of the incident location. Moreover, images capturing individuals engaging in violent behavior are included in the alert message, offering additional context for prompt action by law enforcement agencies.

In summary, the current system implementation combines video processing techniques using OpenCV, deep learning capabilities enabled by TensorFlow and MobileNetV2 architecture, and image enhancement through PIL, along with alert generation featuring a distinct beep sound and Telegram notification utilizing the Telepot library for swift communication to authorities. This comprehensive approach ensures the effective detection and timely response to violent activities in public spaces.

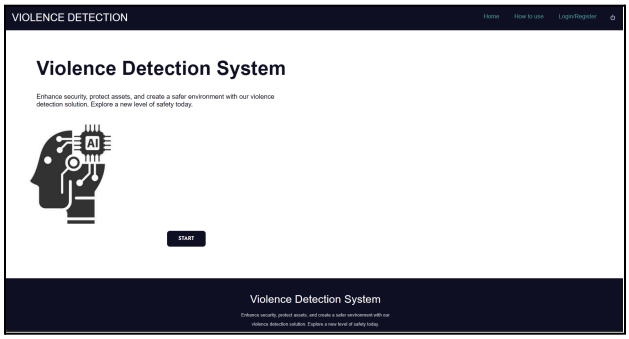

Fig 3: Homepage

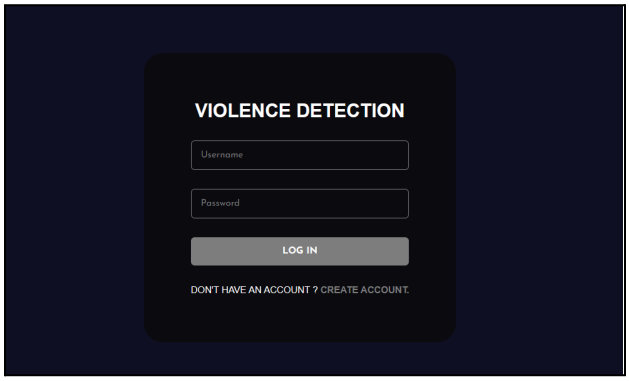

Fig 4: Login / Signup Page

C. Frameworks and Libraries Used for Implementation

Libraries play essential roles in different aspects of the violence detection system, including image and video processing, handling timestamps, interacting with external APIs (such as Telegram Bot API), visualizing data, and incorporating deep learning-based face detection algorithms.

1. NumPy: NumPy is a fundamental library for numerical computing in Python NumPy is used for various tasks such as handling arrays and matrices, performing mathematical operations efficiently, and manipulating numerical data..

2. OpenCV2: It is a powerful library for computer vision tasks. It provides a wide range of functionalities for image and video processing, reading and writing image files, image transformations (e.g., resizing, cropping), applying filters, detecting objects, and more. OpenCV2 would be essential for tasks such as extracting frames from video footage, performing image preprocessing, and implementing the violence detection algorithm.

3. datetime: The datetime library in Python provides classes and functions for working with dates and times. Datetime can be used to handle timestamps associated with video frames or alert notifications.

4. pytz: pytz is a Python library for working with time zones. It allows you to handle time zone conversions and perform operations with aware datetime objects. In the current system, pytz can be useful for ensuring consistency in timestamp information across different time zones, especially if the surveillance system operates in multiple locations with different time zones.

5. PIL (Python Imaging Library): PIL is a library for image processing tasks. It provides functionalities for opening, manipulating, and saving many different image file formats. In the proposed system, PIL can be used for tasks such as enhancing image quality, adjusting image brightness or contrast, and converting between different image formats.

6. Telepot: Telepot is a Python library for interacting with the Telegram Bot API. It allows you to create Telegram bots and send/receive messages, files, and other content through the Telegram messaging platform. Telepot can be used to implement the alert notification system, where alerts about detected violent activities are sent to relevant authorities or users via Telegram messages.

7. matplotlib: Matplotlib is a popular library for creating static, animated, and interactive visualizations in Python. Matplotlib can be used for generating visualizations or plots to analyze the performance of violence detection algorithms, visualize the distribution of detected violent activities over time or space, or display other relevant statistics or insights derived from the surveillance data.

8. MTCNN (Multi-Task Cascaded Convolutional Neural Networks): MTCNN is a deep learning-based face detection algorithm. It is commonly used for detecting and localizing human faces in images or video frames. In the current system, MTCNN can be used as part of the violence detection pipeline to detect human faces in the surveillance footage, which can provide additional context or information for analyzing and detecting violent activities involving human subjects.

9. TENSORFLOW FRAMEWORK: In the violence detection system, TensorFlow is utilized for deep learning tasks, particularly for integrating and deploying the MobileNetV2 architecture effectively. Through TensorFlow, the MobileNetv2 model can be trained on a curated dataset of violence and non-violence video clips, enabling it to learn and recognize patterns indicative of violent activities. Additionally, TensorFlow facilitates real-time inference and prediction, allowing the trained model to analyze video frames rapidly and accurately classify them as either violent or non-violent. Overall, TensorFlow plays a pivotal role in enhancing the capabilities of the surveillance system by enabling the integration of state-of-the-art deep learning techniques for violence detection

VI. RESULTS

The proposed system, utilized a dataset comprising 1000 videos, with 350 videos representing violent incidents and 350 depicting non-violent scenarios. Through rigorous training over 50 epochs, the model achieved an impressive accuracy of 96% in accurately classifying video sequences as violent or non-violent.

The achieved accuracy of 96% highlights the capability of violence detection system to reliably identify and alert authorities about potential threats or criminal activities in real-time surveillance environments. This result underscores the significance of the current system in enhancing public safety and security through advanced technological solutions.

These training and evaluation details provide insights into the model's performance, its capacity to learn from the dataset, and the achieved accuracy in detecting anomalies in videos.

```
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>
```

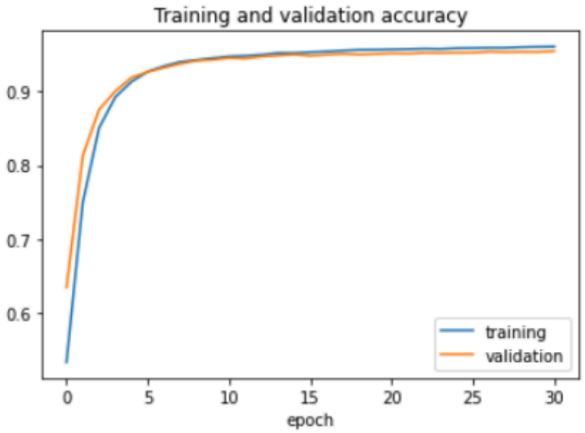

Fig 5: Accuracy v/s Epochs

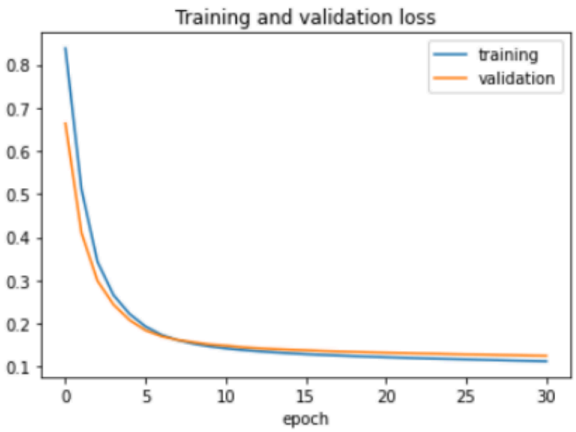

Fig 6: Loss v/s Epochs

Fig 7: Predicted Violence as TRUE

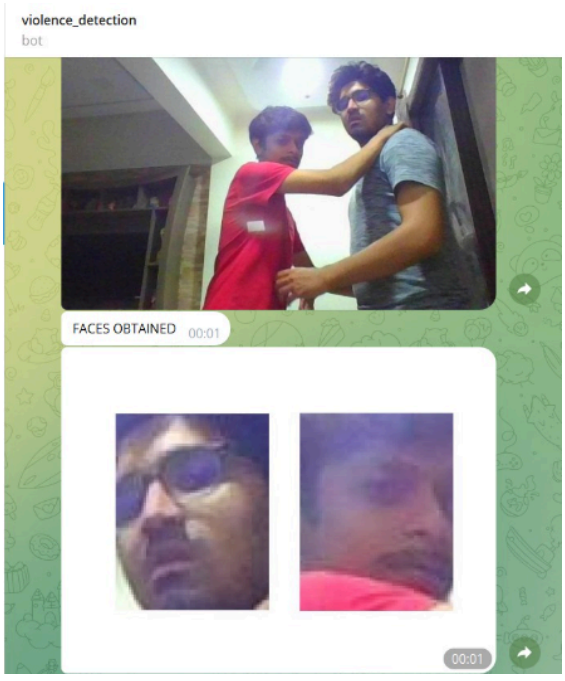

Fig 8: Notification on Telegram with Faces Detected

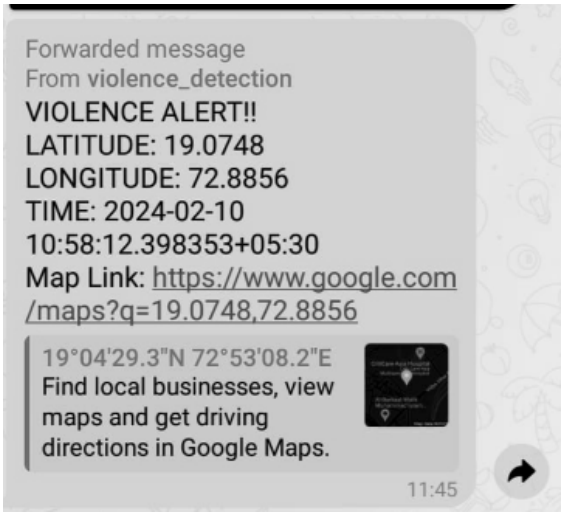

Fig 9: Notification on Telegram with Violence alert message (Including Time and Location with location link)

VII. CONCLUSION AND FUTURE SCOPE

In conclusion, the violence detection system has demonstrated significant advancements in enhancing public safety through real-time surveillance. By leveraging state-of-the-art technologies such as deep learning and computer vision, A robust system capable of accurately identifying violent activities within surveillance footage with an impressive accuracy of 96% is developed. This achievement underscores the effectiveness of the implemented approach and its potential to mitigate risks and respond promptly to security threats in public spaces.

Looking ahead, there are several avenues for future exploration and improvement. Firstly, the aim to develop a user-friendly interface for system monitoring and control, facilitating seamless interaction and management for users. Additionally, scalability remains a crucial consideration, and efforts will be directed towards ensuring the system can efficiently handle an increasing number of surveillance cameras and areas without compromising performance.

Furthermore, there is a need to enhance the system's adaptability to varied surveillance scenarios. This includes addressing challenges such as camera mobility, varying lighting conditions, and different camera placements to ensure consistent and reliable performance across diverse environments. By focusing on these areas of development, the aim is to strive to further strengthen the capabilities of the violence detection system and contribute towards creating safer communities.

VII. References

[1] 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," in IEEE Access, vol. 10, pp. 25190-25202, 2022, doi: 10.1109/ACCESS.2022.3155123.

[2] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo and A. F. Dragoni, "Deep Learning for Automatic Violence Detection: Tests on the AIRTLab Dataset," in IEEE Access, vol. 9, pp. 160580-160595, 2021, doi: 10.1109/ACCESS. 2021.3131315.

[3] 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.

[4] Ullah FUM, Ullah A, Muhammad K, Haq IU, Baik SW. Violence Detection UsingSpatiotemporal Features with 3D Convolutional Neural Network. Sensors (Basel). 2019 May 30;19(11):2472. doi: 10.3390/s19112472.

[5] Khan SU, Haq IU, Rho S, Baik SW, Lee MY. Cover the Violence: A Novel Deep-Learning-Based Approach Towards Violence-Detection in Movies. Applied Sciences.2019; 9(22):4963. https://doi.org/10.3390/app9224963J. Clerk Maxwell,A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[6] M. Ramzan et al., "A Review on State-of-the-Art Violence Detection Techniques," in IEEE Access, vol. 7, pp. 107560-107575, 2019, doi: 10.1109/ACCESS. 2019.2932114.

[7] 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.

[8]http://www.multitel.be/image/researchdevelopment/research-projects/boss.php.

[9] Unusual crowd activity dataset of the University of minnesota. http://mha.cs.umn.edu/movies/crowdactivity-all.avi.

[10] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixedlocation monitors. TPAMI, 2008.

[11] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning.

[12] B. Anti and B. Ommer. Video parsing for abnormality detection. In ICCV, 2011.

[13] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. ´ NetVLAD: CNN architecture for weakly supervised place recognition. In CVPR, 2016.

[14] C. Bergeron, J. Zaretzki, C. Breneman, and K. P. Bennett. Multiple instance ranking. In ICML, 2008.

[15] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM Comput. Surv., 2009.

[16]https://conferences.computer.org/ictapub/pdfs/ITCA2020-6EIiKprXTS23UiQ2usLpR0/114100a476/114100a476.pdf

[17] MobileNetV2 Model for Image Classification: https://ieeexplore.ieee.org/document/9422058