International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2025 www.irjet.net p-ISSN: 2395-0072

HUMAN VIOLENT ACTIVITY RECOGNIZATION

Prof. S. V. Patil[1], Shubham Lanke[2], Shubham Mukhekar[3], Amol Marathe[4], Manav Chauhan[5]

*[*1]Assistant Professor, Department Of Computer Engineering, Sinhgad College Of Engineering, Pune, India.*
*[*2,3,4],5Department Of Computer Engineering, Sinhgad College Of Engineering, Pune, India.*
DOI : https://www.doi.org/10.56726/IRJMETS48058

Abstract - *The conventional approaches to violence detection primarily focus on predicting body part or joint locations from images or videos. However, these methods face limitations, such as difficulty estimating human poses due to low-resolution and noisy data from depth sensors, especially in real-time settings like CCTV footage. Human violence detection has been a key problem in computer vision for over last many years due to its numerous applications, such as video surveillance, human-computer interaction, and behavior analysis. This project aims to address these challenges by using neural networks to detect violent human activity from real-time surveillance footage. The primary features of this system include: 1. Real-time Violence Detection 2. Violence Activity Monitoring 3. Automated Alerts 4. Categorization of Violence and Non-Violence Activities Our intelligent video surveillance system can be applied in public areas like airports, malls, schools, and parking lots to monitor human activities and prevent incidents such as theft, accidents, and vandalism. The system provides real-time monitoring and can automatically generate alerts when unusual or violent activities are detected, helping enhance public safety.*

Key Words: Violence Detection, Real-time Monitoring, Violence Activity Recognition, Neural Networks, Intelligent Surveillance.

1.INTRODUCTION

The project focuses on developing a real-time violence detection system for public places such as malls, airports, and railway stations, utilizing deep learning, neural networks, and computer vision. To understand the entire project, familiarity with deep learning, neural networks, computer vision, video analysis, and desktop application development is essential. Deep learning is used to enable the model to learn from extensive datasets and recognize patterns in video frames, with neural networks helping to identify human body parts and detect violent movements. The project leverages these techniques to analyze real-time CCTV footage and trigger alerts upon detecting violent activities. The application has potential uses beyond security, such as monitoring patient movements in healthcare facilities to prevent injuries. Computer vision plays a vital role in detecting and classifying human poses within video data, introducing temporal *complexity that requires advanced techniques to track movements over time. By analyzing video frames, the system can efficiently detect violent actions in real time, essential for its deployment in diverse public spaces. Furthermore, unsupervised learning techniques are incorporated to handle violent behavior detection without the need for extensive labeled datasets. The system is designed as a cross-platform application, compatible with both mobile and desktop devices, allowing administrators to receive alerts and manage incidents remotely. Its adaptability also allows it to run on low-cost hardware, including embedded systems and mobile devices, making it user-friendly and accessible. In terms of motivation, violence detection has been a challenging area in computer vision, with applications in surveillance, behavior analysis, and human-computer interaction. The project aims to improve real-time surveillance by overcoming limitations related to low-cost sensors through the use of deep learning and CNNs, thereby enhancing public safety in high-risk environments. The main objectives of the project are to detect violent activities in real-time and provide an easy-to-use interface for administrators to receive timely alerts.*

2. LITERATURE REVIEW

In the paper Real-Time Violence Detection and Localization in Crowded Scenes by Mohammad Sabokrou and Mahmood Fathy, a method is proposed for real-time detection and localization of violent activities in crowded areas. The approach segments each video into non-overlapping cubic patches, using two types of descriptors—local and global—to capture various video properties. These descriptors analyze structural similarities between adjacent patches to differentiate normal activities from anomalies. By using cost-effective Gaussian classifiers and learning HUMAN VIOLENT ACTIVITY RECOGNIZATION Shubham Lanke, Shubham Mukhekar, Amol Marathe, Manav Chauhan Pranali D. Dahiwal B.E. Computer Engineering, Sinhgad College of Engineering, Pune, India features in an unsupervised manner with a sparse autoencoder, the system can efficiently identify violent actions. Experimental results demonstrate the algorithm's effectiveness and comparable accuracy to state-of-the-art methods, with improved time efficiency, especially on the UCSD ped2 and UMN benchmark datasets. However, a notable limitation is that scalability to larger real-world environments may affect performance [1]. In the paper Learning Temporal Regularity in Video Sequences by Mahmudul Hasan and Jonghyun Choi, the authors explore methods to identify meaningful activities in long, cluttered video sequences. This challenge is heightened by the subjective nature of what constitutes 'meaningful' activity. To address this, they propose a generative model that learns regular motion patterns, or 'regularity,' using multiple data

sources with minimal supervision. Two autoencoder-based methods are introduced: the first uses traditional handcrafted spatio-temporal local features and learns a fully connected autoencoder on them, while the second uses a fully convolutional feed-forward autoencoder, allowing end-to-end learning of both local features and classifiers. Limitations include that handcrafted spatio-temporal features may not capture all complex motions, and the model's performance may vary depending on dataset size and quality [2]. The paper Violence Detection in Video Using Predictive Convolutional Long ShortTerm Memory Networks by Jefferson Ryan Medel focuses on automating the detection of anomalous events in long video sequences, a challenging task due to the ambiguity of defining such events. The authors propose an end-to-end trainable model utilizing composite Convolutional Long Short-Term Memory (Conv-LSTM) networks to predict the progression of a video sequence from a limited set of input frames. The model calculates regularity scores based on reconstruction errors, with abnormal sequences yielding lower scores as they deviate from expected predictions. Conv-LSTM networks in this approach leverage 'conditioning' to learn more meaningful representations, and the model selection is based on reconstruction and prediction accuracy. Evaluated both qualitatively and quantitatively, the Conv-LSTM model demonstrates competitive performance on violence detection datasets. However, a limitation is that minimal supervision may reduce accuracy for ambiguous events, and reconstruction errors may not always effectively differentiate regular from abnormal sequences [3]. In the paper Abnormal Event Detection in Videos using Spatiotemporal Autoencoder by Jeffe Yong Shean Chong, an efficient method for anomaly detection in videos is proposed, utilizing a spatiotemporal autoencoder architecture. Unlike traditional convolutional neural networks (CNNs), which excel in object detection and recognition tasks but rely on labeled data for supervised learning, this spatiotemporal architecture is tailored for violence detection in videos, including crowded scenes. The model consists of two main components: one for spatial feature extraction and another for learning the temporal evolution of these features. Evaluated on the Avenue, Subway, and UCSD benchmarks, the model demonstrates accuracy comparable to state-of-the-art methods, with a high processing speed of up to 140 frames per second (fps). However, high-speed detection may compromise accuracy in complex scenes, and the model may not generalize well across diverse environments and scenarios [4]. In the paper Unrolled Optimization with Deep Priors by Steven Diamond and Vincent Sitzmann, the authors introduce a framework aimed at addressing inverse problems in computational imaging, sensing, and low-level computer vision. This "unrolled optimization with deep priors" approach focuses on extracting latent images from measurements using a known physical image formation model. Traditionally, such tasks have relied on handcrafted priors combined with iterative optimization methods. In contrast, this framework integrates the knowledge of image formation directly into deep networks, inspired by classical iterative techniques, resulting in enhanced performance for imaging challenges like denoising, deblurring, and compressed sensing for MRI. Experimental results show that this approach outperforms state-of-the-art methods in these domains. However, the method is computationally demanding, which could limit its applicability in resource-constrained environments, and its design is specific to imaging problems, restricting its generalizability to other fields [5].

3. METHODOLOGY

3.1 System Architecture:

Expanding on the architecture, the system begins with the video input, which is typically a live feed from CCTV cameras. The preprocessing step includes several essential operations, such as resizing the video frames, noise reduction, and possibly color correction, which collectively enhance the quality of data fed into the model. These adjustments are crucial as they help in removing unnecessary artifacts and ensuring that the deep learning model receives clearer images, improving its ability to detect details related to violent activities. Once preprocessing is complete, segmentation is applied to the frames. Segmentation isolates significant portions of the frame, such as human figures or objects that might be involved in a violent act. This focused approach saves computational resources, as the model doesn't have to process the entire frame but only specific regions where human activity is detected. By isolating these areas, the system can more effectively recognize patterns associated with violent behavior. Following segmentation, the feature extraction phase comes into play.

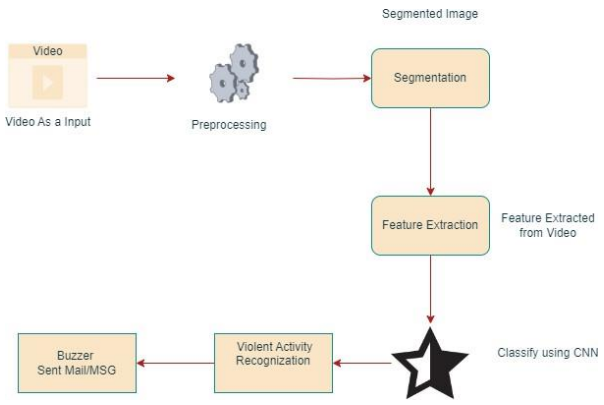

Fig 1: High-level System Architecture

This step involves identifying specific characteristics or "features" from the segmented regions, such as the posture of a person, movement speed, and joint orientation. These extracted features are fed into the CNN model, which has been trained on large datasets containing both violent and non-violent activities. Through this training, the CNN learns to differentiate between aggressive actions (like punching,

kicking) and nonaggressive ones (like walking or talking). The CNN applies these learned patterns to classify the behavior observed in the video as either violent or non-violent. Upon detecting violent behavior, the system triggers an alert mechanism. This alert system could include various methods, such as sounding a buzzer in the surveillance control room, sending SMS or email notifications to security personnel, or even logging the incident with a timestamp for later review. The alerts are intended to provide real-time feedback, enabling a rapid response to potentially dangerous situations, which is crucial in preventing escalations and ensuring public safety. The system's design allows it to operate across different environments and can be integrated into existing surveillance infrastructure, making it versatile for use in crowded places like malls and airports, as well as in quieter spaces like hospital wards for monitoring patient activity.

3.1 Process Flow:

1. **Video as Input**: The system begins by taking video footage as input, typically from a CCTV feed in public spaces like malls, airports, or railway stations. This video input serves as the raw data for the detection pipeline.

2. **Preprocessing**: Before analysis, the video goes through a preprocessing stage. This step might include resizing frames, noise reduction, and adjusting lighting or contrast to ensure consistent quality and improve detection accuracy. Preprocessing prepares the frames for further analysis by normalizing and enhancing them.

3. **Segmentation**: After preprocessing, the frames are passed through a segmentation step. Segmentation involves dividing the frame into meaningful sections to isolate regions of interest, such as human figures. This step is crucial as it helps in focusing on parts of the image that are relevant for identifying violent activities, excluding irrelevant background information.

4. **Feature Extraction**: Once the frames are segmented, the system performs feature extraction to capture essential characteristics of the movement within the frame. These features could include details about human poses, body movements, and spatial relations, which help in distinguishing violent activities from normal ones. Feature extraction creates a compact representation of the frame data, making it easier for the model to process.

5. **Classification using CNN**: The extracted features are then fed into a Convolutional Neural Network (CNN) model. CNNs are particularly effective in analyzing spatial data like images and video frames, allowing them to classify movements. The CNN model examines the features and determines if there is any violent activity. This classification step is where the system decides if the frame sequences indicate a violent or non-violent event.

6. **Violent Activity Recognition**: If the CNN model detects violent activity, the system marks it as recognized violent behavior. This recognition step indicates that the system has identified patterns associated with violent actions, such as punching or aggressive movements.

7. **Buzzer/Alert Notification (Sent Mail/MSG)**: Once violent activity is detected, the system triggers an alert. This alert can be in the form of a buzzer sound,a notification sent via email, or an SMS message to security personnel. This real-time notification allows authorities to take immediate action and potentially prevent further escalation. In summary, this pipeline enables real-time violence detection in public spaces by segmenting video frames, extracting features, and using a CNN model for classification. The final step is an alert system to notify relevant authorities instantly. This solution is highly relevant for enhancing security in public environments.

3.2 Implementation

3.2.1 Designing the Environment for Violent Activity Detection The primary objective of developing the violent activity detection system is to create a real-time surveillance application that assists security personnel in identifying potential violent threats. The system is designed to accept video uploads, process these videos frame by frame, and detect violent actions like fighting, aggression, or other criminal activities. The user interface is built using Tkinter, providing a clear and simple way to upload videos and manage the process. The application's design emphasizes accessibility and ease of use, ensuring that it can be operated by users with minimal technical experience. In this phase, the focus was on creating a system that can easily handle video uploads and provide real-time notifications upon detecting potential violent actions based on the analysis of these frames. The system uses OpenCV for video processing, converting the video into individual frames for further analysis. The primary alert mechanism involves notifying the administrator when violent activities are detected based on the analysis of these frames.

3.3.2 Development Environment and Setup

The project was developed using Python, which allowed easy integration of libraries like Tkinter for the GUI and OpenCV for video processing. Python's flexibility was key in ensuring that the system could process video data efficiently while maintaining a user-friendly interface.

The environment setup began by installing necessary dependencies such as OpenCV, Tkinter, and SQLite for data storage.The user interface (UI) is created using Tkinter, and the core logic is separated into distinct functions to handle tasks like uploading videos, extracting frames from the video, and notifying users about successful uploads. The application's backend is configured to store data in an SQLite database, where details like the uploaded video's file name and extracted frames are recorded. The database ensures that every video processed is logged, providing a system that can track uploads and associated data, making it easy for users to monitor their activity. To handle video uploads, the askopenfilename() method is used, which opens a file dialog allowing the user to choose the video. After the video is selected, OpenCV is used to extract each frame from the video and save it in a newly created directory called "data." This ensures that the frames can later be analyzed or stored for future reference.

3.3.3 Violent Activity Detection Implementation

The detection mechanism is currently focused on extracting video frames and preparing them for future analysis. When the video is uploaded, each frame is saved as a JPEG file in the "data" directory. OpenCV's VideoCapture() method is used to open the video file and read each frame. The frames are then saved sequentially, and the extraction continues until the entire video has been processed. The system currently saves each frame for future analysis, and in the future, machine learning models can be integrated to detect violent activities in real-time. Once frames are extracted, the application provides an alert to the user through a message box (ms.showinfo()) to inform them that the video has been successfully uploaded and processed. This helps keep the user informed throughout the process. The next logical step would be to introduce activity recognition algorithms that could analyze the frames and identify violent activities, such as fighting or aggression, based on a trained model. For now, the violent activity detection part of the system is in the preparation phase. Future work would involve integrating a deep learning model, such as a convolutional neural network (CNN), to classify the activities in the extracted frames and trigger alerts accordingly.

3.3.4 User Interaction and Experience

The user interface (UI) has been designed with simplicity in mind. Upon opening the application, users are presented with a clean interface featuring a prominent button to upload videos. This button allows the user to choose a video file from their local storage, and once selected, the system processes the video and extracts frames. The progress of the video upload and frame extraction is handled in the background, and users are notified once the task is complete via a pop-up message. In addition to the "Upload Video" button, an "Exit" button is provided to allow users to close the application gracefully. This button ensures a smooth termination of the program, freeing up any resources that may have been allocated during the session. The system also ensures that the uploaded videos and extracted frames are stored in an organized manner, making it easier for users to track which videos have been processed. The feedback system, while currently limited to popup messages, can be extended in the future to include audio or visual alerts on the UI to notify users about violent activities detected during video analysis.

3.3.5 User Data Management via Database Integration

The system integrates SQLite to manage data about the videos uploaded by the user. Each uploaded video is recorded in the database, along with important metadata such as the video's file name, the time of upload, and the status of the video processing. This ensures that the application keeps a log of all activities, and users can track the progress of their video uploads and frame extractions. In the current version, the database schema records the video file name, the timestamp of upload, and the number of frames extracted. Future enhancements may include logging the results of violent activity detection, allowing users to view past detections and track patterns in violent actions over time. The integration of the database allows for scalability, meaning that as the system grows and more features are added (such as real-time threat detection), user data will be stored and accessed efficiently, providing a seamless experience for users. This database can also be extended to support features like storing alerts and results from the analysis, creating a complete system for managing security footage and user interactions. This ensures that the application keeps a log of all activities, and users can track the progress of their video uploads and frame extractions. In the current version, the database schema records the video file name, the timestamp of upload.

4. RESULT

In this architecture, preprocessing is a crucial step where the video quality is improved and unnecessary information is filtered out, ensuring that only relevant data is passed forward. This may involve resizing, noise reduction, and frame selection to make the system more efficient and accurate. The segmentation phase splits the video frames into smaller regions to focus on specific parts of the frame, such as human figures or suspicious objects. By isolating these areas, the system can more accurately identify movements or actions that may indicate violent behavior. Feature extraction is performed to capture the essential characteristics of each segment, such as motion patterns,

shapes, and spatial relationships. These features are vital for training the CNN model to distinguish between normal and violent activities effectively. The CNN model, trained on a dataset of violent and non-violent actions, uses these features to classify actions with high precision. When violent activity is detected, the system's alert mechanism is activated. This includes sounding an alarm, which can serve as an immediate warning, and sending notifications via email or SMS to security personnel. This multi-layered alert system ensures that authorities are promptly informed, allowing for a rapid response to potential threats.

5. CONCLUSION

A system designed to process real-time CCTV footage for violence detection can revolutionize public safety by providing automated surveillance and reducing the need for human monitoring. This technology leverages advanced deep learning algorithms to detect violent human behavior, such as aggressive movements, fights, or potential threats, in public spaces like malls, airports, and railway stations. In addition to improving security, violence detection systems can be integrated with law enforcement to enable faster response times in critical situations. By identifying threats early, the system can send real-time alerts to security personnel, ensuring immediate action. Moreover, this system can assist in preventing crime in highrisk areas, thereby enhancing overall public safety. Research in related fields, such as human activity tracking and pose estimation, can further enhance the system's effectiveness. For example, integrating activity tracking can help the system monitor specific individuals over time, flagging violent patterns and behaviors before violence occurs. Combining this with predictive analytics can also allow the system to anticipate potential threats based on prior events. The application of violence detection systems extends beyond public security. In healthcare, it can be used to monitor patients with psychological conditions prone to aggressive behavior, improving care and safety in hospitals. In educational institutions, the system can help identify bullying or other violent activities, creating safer learning environments. The growing advancements in Artificial Intelligence (AI), Internet of Things (IoT), and Neural Networks will continue to push the boundaries of such systems. Future iterations may include enhanced accuracy in lowlight or crowded environments, reduced false positives, and integration with facial recognition for identifying individuals involved in violent incidents. This kind of technology could eventually serve as a cornerstone for smart cities, enabling a more secure and efficient urban environment with minimal human interventions.

6. REFERENCES

[1] Amrita 1 Eralda Nishani, Betim Cico : "Computer Vision Approaches based on Deep Learning and Neural Networks" Deep Neural Networks for Video Analysis of Human Pose Estimation- 2017 6th MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO), 11-15 JUNE 2017, BAR, MONTENEGRO

[2] Naimat Ullah Khan , Wanggen Wan : "A Review of Human Pose Estimation from Single Image"- 978-1-5386-5195-7/18/ 2018 IEEE

[3] Qiuhui Chen, Chongyang Zhang, Weiwei Liu, and Dan Wang, "Surveillance Human Pose Dataset And Performance Evaluation For Coarse-Grained Pose Estimation", Athens 2018.

[4] Baole Ai, Yu Zhou, Yao Yu : "Human Pose Estimation using Deep Structure Guided Learning"- 978- 1-5090-4822-9/17 2017 IEEE DOI 10.1109/WACV.2017.141

[5] Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh The Robotics Institute, Carnegie Mellon University "Real time Multiperson 2D Pose Estimation using part affinity fields" - 1063-6919/17 2017 IEEE DOI 10.1109/CVPR.2017.143

[6] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 1251- 1258.

[7] W. Song, D. Zhang, X. Zhao, J. Yu, R. Zheng, and A. Wang, "A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks," IEEE Access, vol. 7, pp. 39172– 39179, 2019.

[8] A. S. Keçeli and A. Kaya, "Violent activity detection with transfer learning method," in Electronics Letters, vol. 53, no. 15, pp. 1047–1048, 2017. [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in neural information processing systems, pp. 1–9, 2012.

[10] LecunY., Bottou L., Bengio Y et al., "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[12] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," arXiv: 1801.04264, 2018.