IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Deep Learning Based Real-Time Violence Detection System

[1]K.S.B. Ambika, [2]K. Kavya, [3]P. Padma Rani, [4]A.Sandeep, [5]M. Bharath Kumar

[1]Assistant Professor, [2]Final Year B.Tech Student, [3] Final Year B.Tech Student, [4]Final Year B.Tech Student, [5]Final Year B.Tech Student

[1]Department of CSE-AIML,

[1]Aditya college of enginerring & Technology(A), Surampalem, Andhra Pradesh, India.

Abstract: The growing use of smart surveillance systems has made people very worried about public safety, real-time monitoring, and quick responses to emergencies. Traditional monitoring systems depend a lot on people to watch over things, which is not very effective, can lead to mistakes, and can't take quick action when things go wrong. These systems also don't have features like automation, scalability, and real-time alerts. This paper describes a Real-Time Violence Detection System that uses machine learning and computer vision to automatically find violent actions in video streams. The system takes live or recorded video input, breaks it down into frames, and uses deep learning models to find patterns and decide if actions are violent or not. When the system sees violence, it sends an automated email alert to the appropriate authorities or users right away, making sure that they can respond and intervene quickly.

Index Terms - Deep Learning, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), MobileNetV2, TensorFlow & Keras.

I INTRODUCTION

The need for smart surveillance systems has grown in recent years because people are more worried about safety in both public and private places. Traditional surveillance systems rely on constant human monitoring, which is not only time-consuming but also not very good at quickly finding incidents. Fatigue, distraction, or a slow response time can cause human operators to miss important events. Thanks to improvements in deep learning and machine learning, it is now possible to use video analysis to automatically find suspicious or violent activities. Violence detection systems look at video frames to find strange behaviours like fights, attacks, or aggressive movements.

But there are still problems with current systems, such as:

Not being able to find things in real time

Ways to respond slowly

No automated alert systems

Not very accurate in complicated settings

To fix these problems, this project comes up with a Real-Time Violence Detection System that sends automatic email alerts. The system uses deep learning models to quickly find violence and sort video frames. When violence is found, an alert email is automatically sent to a list of people who have been set up to receive it. This makes sure that everyone knows about it right away and can take action.

II EXISTING & PROPOSED SYSTEM

Existing System:

The system that is already in place. Most traditional surveillance systems use CCTV cameras to watch video feeds all the time, which is done by security personnel.

These systems have a few problems:

- Human Dependency: Constant monitoring is necessary, which can make people tired and miss important events.
- Delayed Response: There is no immediate alert system when violence happens.
- Low Efficiency: It's hard to keep an eye on more than one camera at a time.
- Not automated: no smart detection of strange activities
- Not very accurate: can't tell when someone is being violent in a subtle or sudden way

In general, current systems are not very effective, are reactive instead of proactive, and don't make sure that action is taken right away in an emergency.

Proposed System:

The proposed system includes an AI-powered framework for real-time violence detection that automates surveillance and speeds up response time.

Important Features:

Detection in Real Time: Uses deep learning and machine learning models to look at live video streams and find violent actions right away

Processing frames and extracting features: We break video input up into frames, process them ahead of time, and then use trained models to analyse them.

Automatic Email Alert System: An email alert with important information is sent to the police automatically when violence is detected.

Detection with High Accuracy: Uses advanced models like CNN, LSTM, Mobile Net, and others to make classification better.

Works well and can be scaled up: Can be used in a lot of different places, like schools, parks, and offices.

Less Work for People: Reduces the need for manual monitoring

Pros:

- Emergency response that is faster
- More effective surveillance
- More safety and security
- Monitoring in real time with automation

III RELATED WORK

A lot of research has been done on intelligent surveillance systems that use machine learning and computer vision to make public spaces safer. Conventional surveillance systems depend on constant human observation of CCTV footage, which is ineffective and susceptible to human error. These systems don't have automation, the ability to detect things in real time, or ways to send alerts right away. Researchers have looked into using deep learning models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks to recognise activities in video data in order to get around these problems. These models look at both the spatial and temporal features of video frames to find violent or unusual behaviour. Multiple studies have shown that violence detection systems based on deep learning can be much more accurate than manual monitoring. People often use frame extraction, feature normalisation, and sequence modelling to figure out if an action is violent or not. Recent improvements also include the use of real-time video processing frameworks and edge computing to cut down on lag time and boost performance. Some systems have alert features like alarms or notifications, but many don't have reliable automated communication systems like email alerts.

IV METHODOLOGY

The proposed Real-Time Violence Detection System is built using a modular and structured method that makes it efficient, scalable, and able to work in real time. The system combines video processing, machine learning-based classification, and automatic alert systems into one workflow.

1. Getting video input

The system gets input from either live CCTV cameras or recorded video streams. This is the main source of data for finding violent behaviour.

2. Getting the Frame

Every so often, the input video is split into separate frames. This step makes video processing easier and lets you look at visual data in more detail.

3. Preprocessing

Frames that have been extracted go through preprocessing steps like:

- Changing the size to a set size
- Normalising the values of pixels
- Less noise

This makes sure that things are the same and makes the model work better.

4. Getting Features

Deep learning models like the following take important features out of frames:

- Convolutional Neural Networks (CNNs) for getting spatial features
- If you use pretrained models like Mobile Net or ResNet in your project,

5. Sorting

The trained model receives the processed frames and sorts the activity into one of these categories Violent or Non-Violent.

Models like LSTM can be used to find patterns over time across frames for sequence-based detection.

6. Detection in Real Time

The system keeps processing incoming frames and finds violence in real time. The system checks for violent activity if the confidence score reaches a certain level.

7. Making alerts (sending emails)

When violence is found: A notification email is sent out automatically The alert goes to a set group of people, like security staff.

The email could have information like:

Time of detection, location (if available), and a snapshot or frame of the event

8. Modularity of the system

The system is made up of separate parts, such as the Video Processing Module, the ML Model Module, and the Alert System Module.

4.1 System Architecture Overview

An architecture diagram provides a visual representation of the structure, components, and interactions of a system or application and outlines the design and layout of the system. They are used to convey complex information in a clear and concise manner about a project.

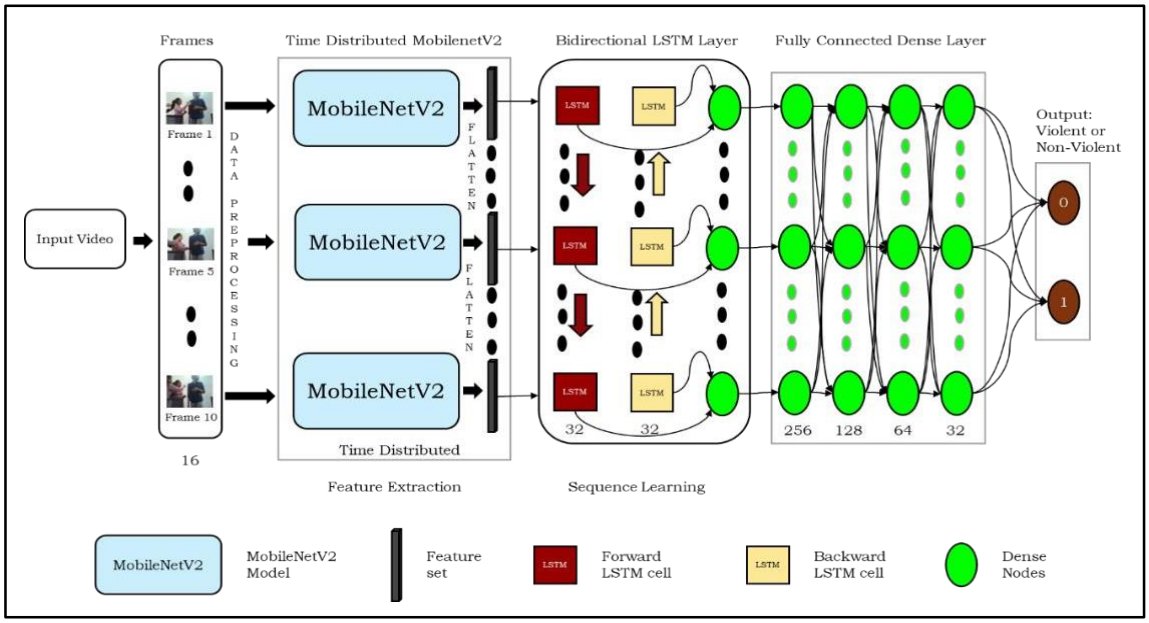

Figure 4.1 Architecture Diagram

Phase of the architecture diagram in detail.

• **Input Video**: The system starts by capturing the live video stream from the camera connected to the system. The camera feed provides a continuous stream of frames, which will be processed for violence detection. The input video may come in various formats and resolutions, depending on the camera specifications and application requirements, and is streamed to the system in real-time. Before analysis, each frame of the input video undergoes preprocessing to optimize model performance.

• **Data Preprocessing:** During real-time prediction, the "Data Preprocessing" step involves several key tasks. The live video which is broken into frames is stored as a sequence (16 frames each) which undergo the preprocessing. These clips are resized and normalized for the model. These steps ensure that the input data provided to the model during real-time prediction is properly formatted and prepared for inference.

• **Time Distributed MobileNetV2:** In the architecture diagram, the "Time Distributed MobileNetV2" layer serves as the backbone for feature extraction from the input video frames. 42 MobileNetV2 is a convolutional neural network architecture known for its efficiency and effectiveness in image classification tasks. The "Time Distributed" wrapper around MobileNetV2 indicates that the same MobileNetV2 model is applied to each frame in the sequence independently. This allows the model to capture spatial features from each frame effectively.

The output feature maps are flattened using the "Flatten" layer. This operation reshapes the spatial feature maps into a one-dimensional vector, preserving the extracted features while preparing them for further processing by subsequent layers. The feature set comprises the features from the frames produced by after Flattening. These features are fed into the neural network model for training or inference.

• **Bidirectional LSTM Layer**: The "Bidirectional LSTMs" layer represents the component responsible for capturing temporal dependencies and patterns across the sequence of feature vectors extracted by the preceding layers. LSTMs, or Long Short-Term Memory units, are a type of recurrent neural network known for their ability to model sequential data while mitigating the vanishing gradient problem.

The "Bidirectional" aspect indicates that two LSTM layers are employed: one processing the sequence in the forward direction and the other in the backward direction. This bidirectional nature allows the model to capture dependencies both from past to future and from future to past, enabling a more comprehensive understanding of the temporal dynamics in the input sequence.

Each LSTM layer consists of 32 units, representing the number of memory cells or hidden units within the layer. These units maintain a state vector that captures relevant information from previous time steps and updates it dynamically as new input is processed. By utilizing 32 units in each direction, the model 43 gains the capacity to capture complex temporal relationships and patterns in the input sequence. The forward LSTM processes the input sequence from the beginning to the end, while the backward LSTM processes it in reverse. The outputs of both LSTM layers are concatenated, effectively combining information from past and future contexts. This arrangement enables the model to leverage information from the entire input sequence, facilitating more accurate predictions.

• **Fully Connected Dense Layer:** The "Fully Connected Dense Layer" is the hidden layers in the neural network model, responsible for processing the high-level features extracted by the preceding layers and generating the final predictions. This layer consists of multiple fully connected dense layers, each with a specific number of nodes or neurons. The first dense layer contains 256 nodes, followed by subsequent layers with 128, 64, and 32 nodes, respectively. These numbers denote the dimensionality of the output space or the number of neurons in each layer. As data pass through these layers, they undergo nonlinear transformations, allowing the model to learn complex patterns and relationships present in the input features. Each neuron in the dense layers is connected to every neuron in the previous layer, forming a fully connected network. During training, the weights associated with these connections are learned through optimization algorithms such as backpropagation, adjusting them to minimize the error between the predicted and actual labels. The activation function used in each dense layer is the rectified linear unit (ReLU), a widely used activation function that introduces nonlinearity to the network by outputting the input if it is positive and zero otherwise. This activation function 44 enables the network to model complex relationships in the data more effectively.

• **Output Layer:** The "Output layer" represents the final layer of the neural network model, responsible for generating the output predictions based on the features learned by the preceding layers. This layer typically consists of a single dense layer with a number of neurons equal to the total number of classes or categories in the classification task. In the described model, the "Output layer" contains a dense layer with the number of neurons equal to the total number of classes in the dataset, which is two, for Violence and Non-Violence classes. For each input sample, the output layer computes a vector of scores or probabilities corresponding to each class, using the SoftMax activation function. The SoftMax activation

function converts the raw output scores into a probability distribution over the classes, ensuring that the predicted probabilities sum to one. This allows the model to output the likelihood of each class given the input features, enabling it to make probabilistic predictions. During inference or prediction, the class with the highest probability score is chosen as the final prediction of the model. This output layer structure, with SoftMax activation, ensures that the model produces meaningful and interpretable predictions, facilitating its use in classification tasks.

V. RESULTS AND DISCUSSION

A. Assessment of the System Workflow

We tested the proposed system by giving it both live and recorded video feeds to see if it could find violent behaviour. The workflow involved taking video, extracting frames, preprocessing, extracting features, classifying, and making alerts. The system was able to find violent actions and send automated email alerts, which showed that it was able to monitor things reliably and continuously.

B. Validation of the Dashboard Based on Roles

We tested the system dashboard with different user roles, like Admin and Security Personnel. Each user could only use the features they were allowed to:

Admin: setting up the system, managing users, and keeping an eye on alerts

User/Security: see alerts and surveillance output

This made sure that only the right people could get in and stopped people from doing things they shouldn't have.

C. How accurate the violence detection is

We used test video datasets to see how well the system could find things. The trained deep learning model successfully categorized activities into: Violence and Non-Violence

The model was good at finding clear violent actions like fights or aggressive movements.

D. Making and checking alerts

We tested the alert system by pretending to be in violent situations. When violence was found:

Alerts by email were sent right away.

Notifications were sent to the right people who had signed up for them.

This showed that the real-time alert system was reliable.

E. Security and dependability of the system

The system was put through its paces while streaming video continuously. It kept working well without crashing or slowing down. Authentication mechanisms stopped people from trying to get into system features without permission

F. Keeping track of and logging activities

The system kept records of:

Events that were found

Timestamps for alerts Responses from the system

These logs made it easier to keep an eye on how well the system was working and to look into problems.

G. Observations of Performance

We looked at the system's performance based on:

Speed of detection

Time to respond to an alert

Rate of processing frames

Most detections and alerts were finished in a matter of seconds, which made the system good for real-time use.

H. Testing for usability and user interaction

We tested the interface to see how easy it was to use and how well it worked. People could:

Watch video feeds

Get alerts

Easily understand what the system does

VI Figures and Tables:

ID	Scenario	Result
TC -01	Video input capture (live/recorded)	Pass
TC -02	Frame extraction and preprocessing	Pass
TC -03	Feature extraction using CNN	Pass
TC -04	Violence classification accuracy	Pass
TC -05	Real-time detection of violent activity	Pass
TC -06	Automatic email alert generation	Pass
TC -07	Email delivery to registered users	Pass
TC -08	System logging of detected events	Pass

Table 1: Functional Validation

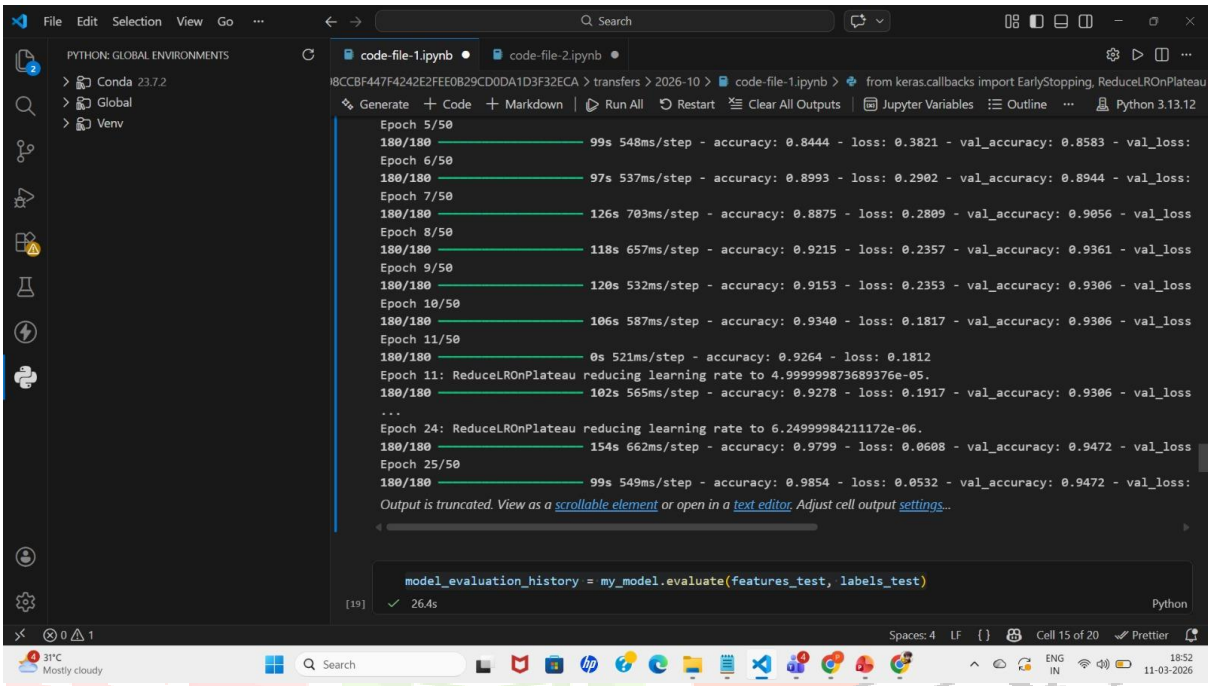

Figure 5.1 Violence classification accuracy

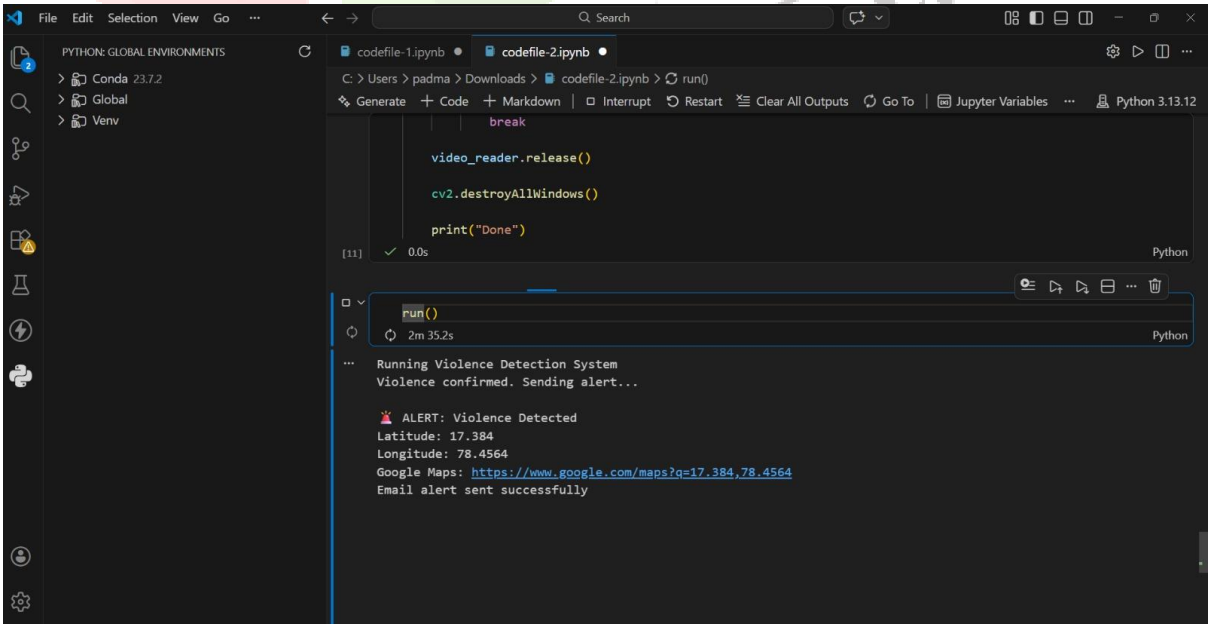

Figure 5.2: Real-time classification (Violent/Non-Violent)

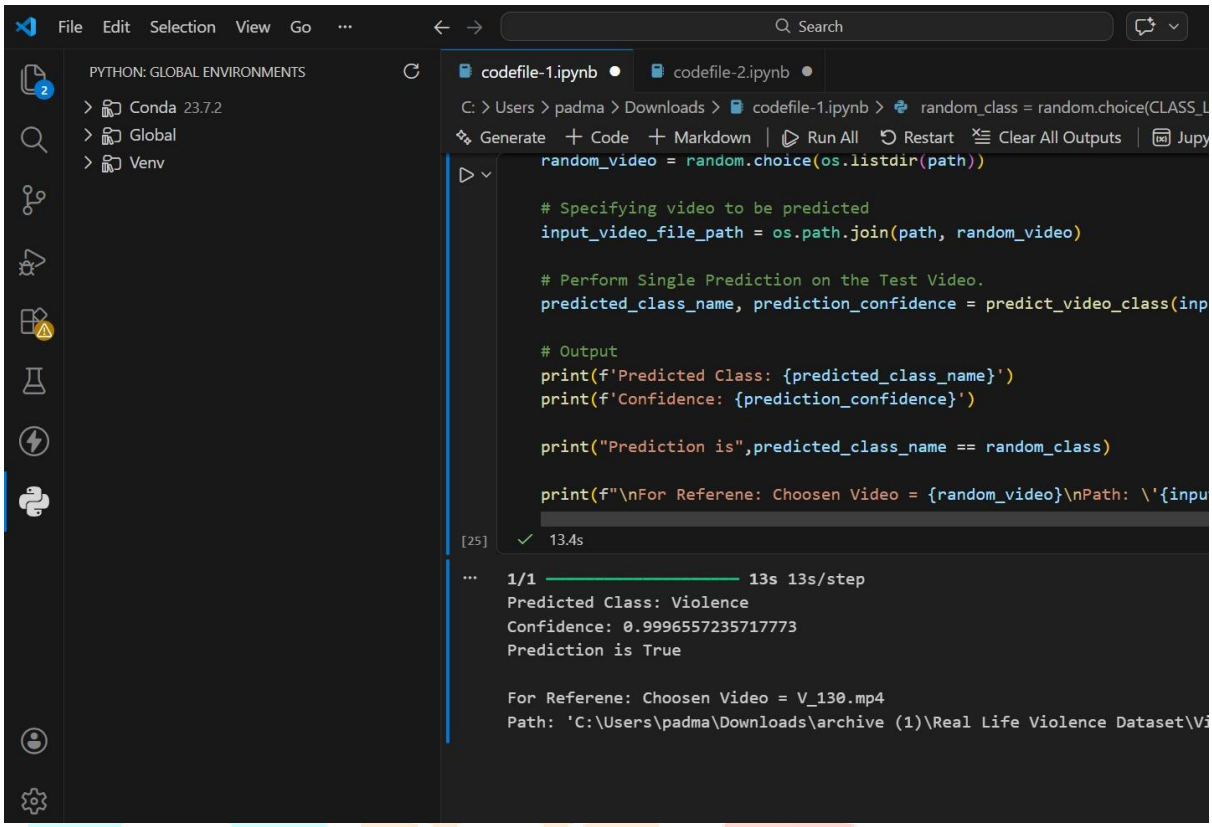

Figure 5.3: Violence detection from video frames

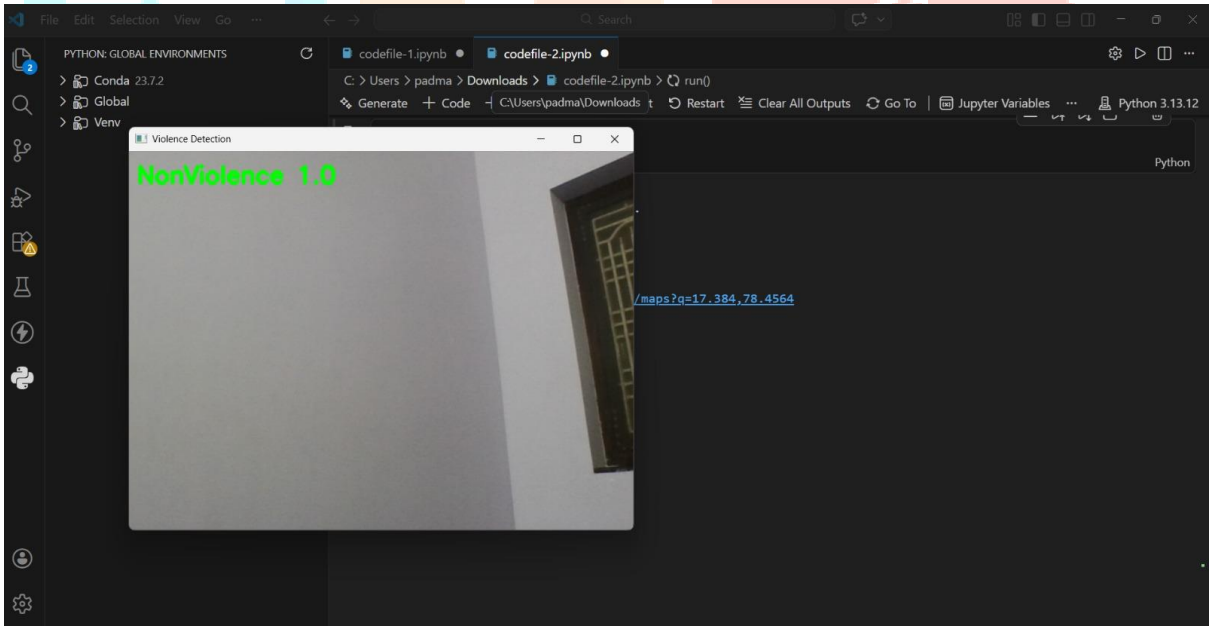

Figure 4: Video input capture (CCTV/live feed)

VII Future Scope:

The future scope of the "Violence Detection in Real-Time using Deep Learning" system lies in enhancing its accuracy, scalability, and applicability across diverse real-world scenarios. As Deep Learning technologies continue to evolve, more advanced architectures such as Transformers and 3D Convolutional Neural Networks (3D-CNNs) can be integrated into the system to improve the detection of complex and subtle violent patterns. These models can better capture both spatial and temporal information, leading to higher precision and reduced false positives. Additionally, incorporating larger and more diverse datasets can help the model generalize better across different environments, lighting conditions, and crowd densities, thereby improving overall performance .

Another significant area of future development is the integration of the system with Internet of Things (IoT) devices and smart surveillance infrastructure. By connecting cameras, sensors, and edge devices, the system can enable distributed and real-time monitoring across large-scale environments such as smart cities, transportation systems, and public safety networks. Edge computing can also be utilized to process data closer to the source, reducing latency and improving response time. This would make the system more efficient and suitable for deployment in resource-constrained environments while maintaining real-time capabilities .

The system can also be enhanced by incorporating multi-modal data analysis, combining video with audio and other sensor data to improve detection accuracy. For example, integrating sound analysis to detect screams, gunshots, or abnormal noise patterns can complement visual data and provide a more comprehensive understanding of events. Furthermore, the addition of facial recognition and person identification features can help in identifying individuals involved in violent activities, aiding law enforcement in investigation and response. However, these enhancements must be implemented with strict adherence to privacy and ethical guidelines .

Moreover, the system can be extended to detect other types of abnormal or suspicious activities beyond violence, such as theft, vandalism, or unauthorized access. This would transform it into a comprehensive intelligent surveillance solution capable of addressing a wide range of security challenges. With continuous advancements in artificial intelligence, cloud computing, and hardware acceleration, the system has the potential to become more efficient, scalable, and widely applicable across industries.

VIII Conclusion:

The “Violence Detection in Real-Time using Deep Learning” system presents an effective and intelligent solution for enhancing surveillance and public safety. By leveraging advanced technologies such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), the system is capable of analysing video streams, extracting meaningful features, and accurately classifying activities as violent or non-violent. The integration of real-time monitoring and alert generation significantly reduces the limitations of manual surveillance, enabling faster response and improved situational awareness in security-sensitive environments .

Overall, the system demonstrates the potential of Deep Learning and Computer Vision in solving real-world problems related to safety and security. It offers a scalable, reliable, and efficient approach for detecting violent activities across various environments such as public spaces, workplaces, and transportation systems. With further enhancements and advancements, this system can be expanded into a comprehensive surveillance solution, contributing to smarter security systems and a safer society .

IX References:

[1] [1] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[2] [2] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-Time Anomaly Detection and Localisation in Crowded Scenes," in IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015.

[3] [3] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in Advances in Neural Information Processing Systems (NIPS), 2014.

[4] [4]A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[5] [5] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in the International Conference on Learning Representations (ICLR), 2021.

[6] [6]H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," in IEEE International Conference on Computer Vision (ICCV), 2013.