Advanced CCTV Threat Detection Using 3-D CNN and LSTM

Kabilan M, Logesh S, Mathivanan R, Prasanna V B ,Praveen C, Anitha R

[1,2,3,4,5] Second Year Students
[6] Professor
Department of Computer Science and Engineering
Sri Venkateswara College of Engineering
Pennalur, Sriperumbudur 602117

Abstract-CCTV surveillance systems have become an essential tool for ensuring public safety and security. However, the number of crimes have increased as the population increased, making it more challenging to monitor criminal activities effectively using traditional CCTV systems. Authorities must rely on digital records, like CCTV footage, to find offenders and gather evidence. Thus, implementing AI into these CCTV systems can help identify crimes forthwith. In this paper, we propose an AI-based system that leverages computer vision libraries and deep learning techniques to analyze CCTV footage in real-time, identify potential criminal activity, and alert the nearest police station. This paper discuss the design, implementation, and evaluation of this system and its potential impact on public safety and security.

Keywords: Public safety; Video Analytics; Threat detection; Security; 3D-CNN;LSTM;CCTV;Alert system.

I. INTRODUCTION

Over the last few decades, the CCTV surveillance system has gained popularity. Government and various organizations are using these systems to keep an check on public safety and security. However, with the rise in crime rates, it has become challenging to monitor criminal activities effectively. Authorities must rely on digital records, like CCTV footage, to find offenders and gather evidence in order to combat crime efficiently. While CCTV footage can be useful for identifying specifics of a crime and suspects, this information is only useful after the event. Often aiding victims depends on taking quick action. As the usage of CCTV continues to grow, it is important to understand that there is no separate entity exists to monitor the activities on the large. Besides, it is a waste of human resources and also is highly fallible. Thus, implementing AI into these CCTV systems can help identify crimes forthwith. The proposed AI system uses real-time CCTV feeds to automatically identify occurrences and send notifications to the closest police station. The solution keeps a database that details the incident's or crime's type, as well as its time, place, and alert level (i.e., low, medium, high-risk alert). The proposed AI system is utilized to analyse the regions that are at higher risk of criminal activities. This system is built to analyze CCTV footage in real-time using computer vision algorithms and deep learning techniques such as Convolution Neural Network(CNN) , Long Short Term Memory(LSTM) to spot probable criminal activity based on predefined patterns and anomalies. A Web-based interface is used to send the generated alerts to the closest police station. This technology can aid law enforcement organizations in providing prompt, efficient responses to situations while also averting additional criminal activity. In addition, the system has the ability to provide reports on the detected occurrences and crimes that may be used for additional research and analysis. Overall, advanced CCTV analysis has the ability to significantly increase public safety and security by helping law enforcement officials to respond quicklyand efficiently to all possible threats and illegal actions. The rest of the paper is organised as follows: Section 2 summarizes the related work and the problem statement. Section 3 describes the system architecture model and discusses the detailed design of the system model. Section 4 describes the performance evaluation based on prototype implementation and Section 5 concludes the paper with references.

II. Related Works

A. Advance motion detection in CCTV:

The author Shubhada. P. Mone et al. has used AMD algorithm to detect the unauthorized entry in restricted area. In the first phase the object was detected using background subtraction and from frames sequences the object is extracted. The second phase detects the object using background Subtraction. The second phase also detects the suspicious activity. The advantage of the system was the algorithm works on Realtime CCTV feeds.

B. A Deep Learning Based System for the Detection of Human Violence in Video Data

The author Muhammad Shoaib et al. proposed an ensemble of Mask Region-based Convolutional Neural Networks for key-point detection scheme, and LSTM based Recurrent Neural Network is used to create a deep neural network model (Mask RCNN) for recognizing violent activities (i.e. kicking, punching, etc.) of a single person.

This research work proposes an advanced CCTV analysis model that can identify potential threats or criminal activity and alert the closest police station to them.

III System model

The architecture diagram of the proposed system is shown in Figure 1. Input from the live CCTV is taken into the system.Live video feed is taken as input using cctv and then it is converted into frames which undergoes necessary preprocessing and fed into the input layer 3D-CNN'S convolution layer where the input frames in the form of multidimensional array was convoluted and then the output of convolution layer is maxpooled and flattened and fed into a dense layer with Relu activation function. The vector from dense layer fed as input to the LSTM network which study on the sequence of input data from the consecutive frames and then the output is fed to a dense layer with softmax activation function for multiclass output and the sigmoid function for binary output, output of the final dense layer is in form of the probability and the neuron with max probability is selected as output.

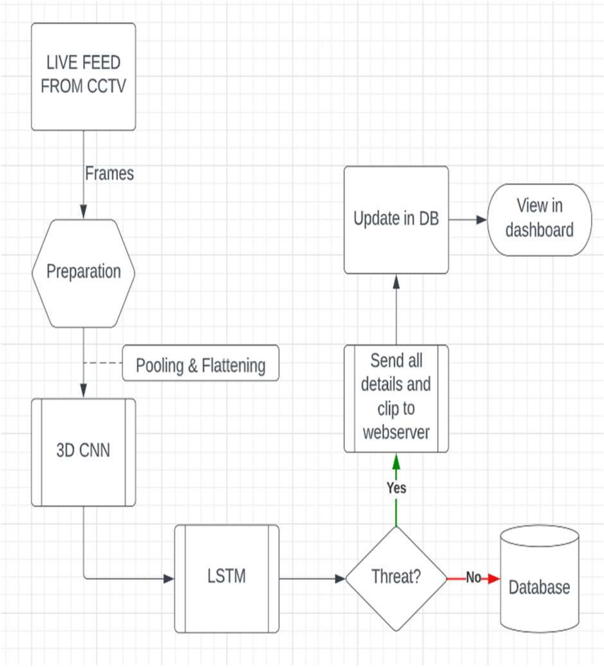


Figure 1 System Model

When the threat is detected by the proposed model, alert is send to the dashboard with video clipping, category and level of threat. By viewing the clip, the admin has to confirm whether it is threat or not. If the threat is confirmed, then it is updated to the database with clip and details of the threat along with the spot. Eventually, the emergency services such as ambulance, fire service will be sent to that spot immediately and local stations are informed. Furthermore, the crime details from database are visualized in dashboard in various forms. Different zones (red - high crime, green - low crime etc.) are assigned to areas with respect to crime rate. An additional alert is sent to concerned authorities upon change in the crime rate along with change of zone in the dashboard. The figure 2 presents the process flow diagram of the proposed model.

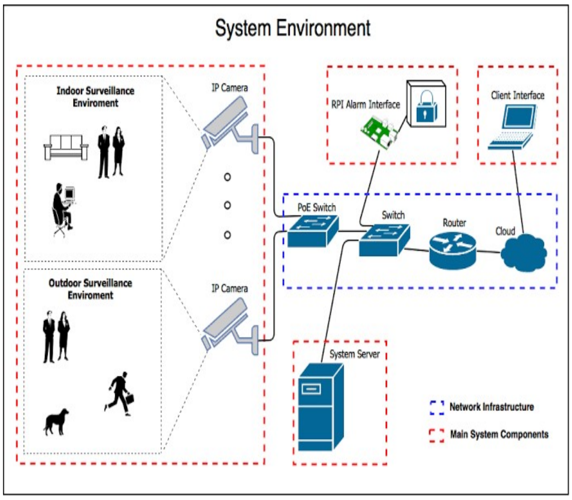


Figure 2 Process flow

IV Why 3D-CNN and LSTM?

Convolutional Neural Networks (CNNs) are commonly used for image content analysis because they are specifically designed to recognize patterns in image data. Unlike other types of neural networks that treat an image as a long vector of pixels, CNNs take into account the spatial relationship between pixels, As the input is taken from the video feed which is sequence of frames 3D- CNN is used. 3D CNNs are used for video and volumetric data analysis, where the input data has a time component or consists of 3D volumes. They are an extension of 2D CNNs commonly used for image analysis. 3D CNNs have an additional dimension compared to 2D CNNs, which represents time or depth. For example, in the case of video analysis, each frame of the video is treated as a 2D image, and the 3D CNN analyzes a sequence of frames as a 3D volume.

LSTM(Long Short-Term Memory) which is a type of recurrent neural network (RNN) architecture that is commonly used for sequence modeling and prediction. Unlike standard RNNs, which can struggle to remember long-term dependencies, LSTMs are designed to overcome the vanishing gradient problem and are able to remember information over longer periods of time. In an LSTM network, there are three main components: the input gate, the forget gate, and the output gate. The input gate controls which information from the current input should be stored in the memory cell. The forget gate controls which information from the previous memory cell should be forgotten, and the output gate controls which information from the current memory cell should be used as output. So due to these features of 3D-CNN and LSTM this combination works well for detecting the threat from the live camera feed.

V Experimental Outcomes

In figure 3 the model predicted as non - violent as there were no actions which causes threat.

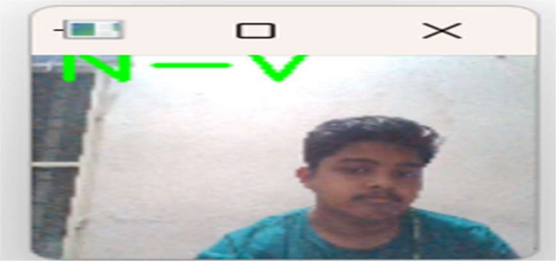


Figure 3 Input feed

In figure 4 knife was detected and it was as potential threat detected so the model predicted violent as the output

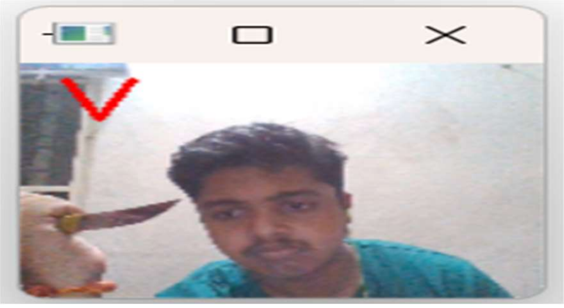
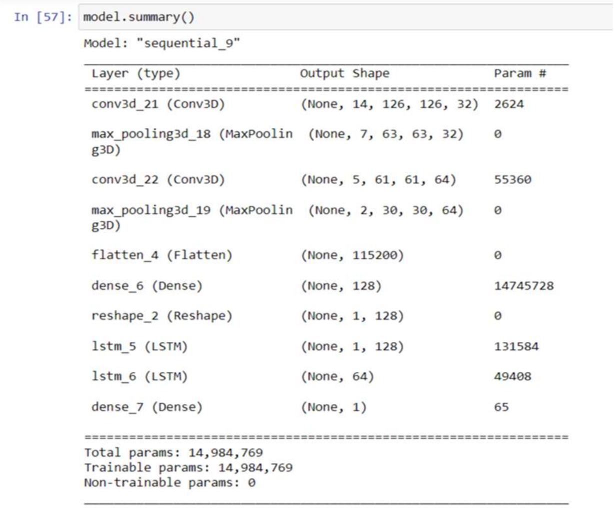


Figure 4 Crime Input

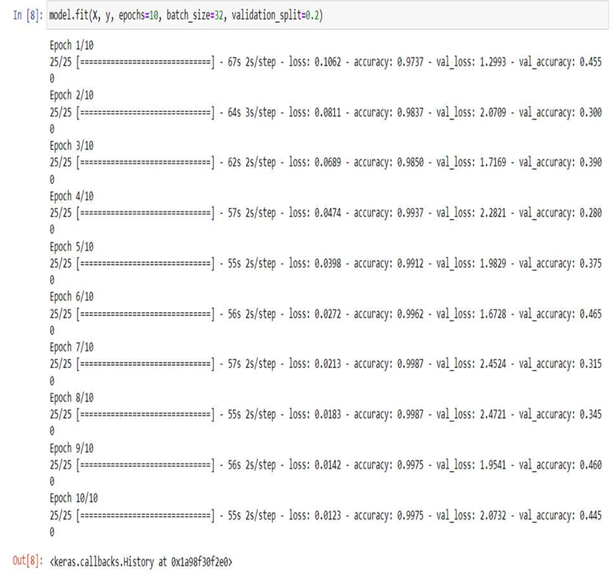
Figure 5 gives the total summary of the 3D–CNN + LSTM model, no of params , output shapes of each layer etc.



```
In [57]: model.summary()

         Model: "sequential_9"
         _________________________________________________________________
          Layer (type)                Output Shape              Param #
         =================================================================
          conv3d_21 (Conv3D)          (None, 14, 126, 126, 32)  2624

          max_pooling3d_18 (MaxPoolin  (None, 7, 63, 63, 32)    0
          g3D)

          conv3d_22 (Conv3D)          (None, 5, 61, 61, 64)     55360

          max_pooling3d_19 (MaxPoolin  (None, 2, 30, 30, 64)    0
          g3D)

          flatten_4 (Flatten)         (None, 115200)            0

          dense_6 (Dense)             (None, 128)               14745728

          reshape_2 (Reshape)         (None, 1, 128)            0

          lstm_5 (LSTM)               (None, 1, 128)            131584

          lstm_6 (LSTM)               (None, 64)                49408

          dense_7 (Dense)             (None, 1)                 65

         =================================================================
         Total params: 14,984,769
         Trainable params: 14,984,769
         Non-trainable params: 0
         _________________________________________________________________
```

Figure 5 Model Summary

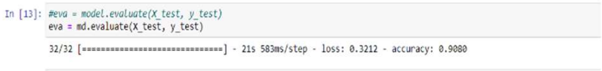
Figure 6 show the output of compilation of the model with 10 epoches along with the model accuracy and evaluation accuracy values.



```
In [8]: model.fit(X, y, epochs=10, batch_size=32, validation_split=0.2)

        Epoch 1/10
        25/25 [==============================] - 67s 2s/step - loss: 0.1062 - accuracy: 0.9737 - val_loss: 1.2993 - val_accuracy: 0.455
        0
        Epoch 2/10
        25/25 [==============================] - 64s 3s/step - loss: 0.0811 - accuracy: 0.9837 - val_loss: 2.0709 - val_accuracy: 0.300
        0
        Epoch 3/10
        25/25 [==============================] - 62s 2s/step - loss: 0.0689 - accuracy: 0.9850 - val_loss: 1.7169 - val_accuracy: 0.390
        0
        Epoch 4/10
        25/25 [==============================] - 57s 2s/step - loss: 0.0474 - accuracy: 0.9937 - val_loss: 2.2821 - val_accuracy: 0.280
        0
        Epoch 5/10
        25/25 [==============================] - 55s 2s/step - loss: 0.0398 - accuracy: 0.9912 - val_loss: 1.9829 - val_accuracy: 0.375
        0
        Epoch 6/10
        25/25 [==============================] - 56s 2s/step - loss: 0.0272 - accuracy: 0.9962 - val_loss: 1.6728 - val_accuracy: 0.465
        0
        Epoch 7/10
        25/25 [==============================] - 57s 2s/step - loss: 0.0213 - accuracy: 0.9987 - val_loss: 2.4524 - val_accuracy: 0.315
        0
        Epoch 8/10
        25/25 [==============================] - 55s 2s/step - loss: 0.0183 - accuracy: 0.9987 - val_loss: 2.4721 - val_accuracy: 0.345
        0
        Epoch 9/10
        25/25 [==============================] - 56s 2s/step - loss: 0.0142 - accuracy: 0.9975 - val_loss: 1.9541 - val_accuracy: 0.460
        0
        Epoch 10/10
        25/25 [==============================] - 55s 2s/step - loss: 0.0123 - accuracy: 0.9975 - val_loss: 2.0732 - val_accuracy: 0.445
        0

Out[8]: <keras.callbacks.History at 0x1a98f30f2e0>
```

Figure 6 Execution with 10 Epoch

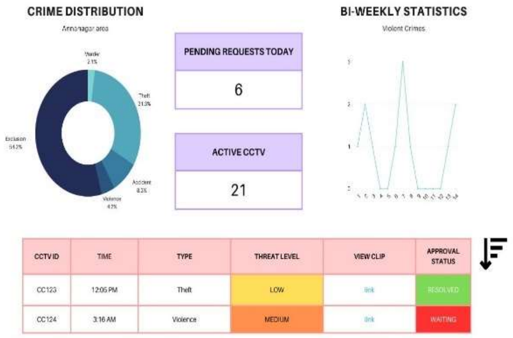
Figure 7 show the evaluation of the model on the train dataset along with its accuracy.



```
In [13]: #eva = model.evaluate(X_test, y_test)
         eva = md.evaluate(X_test, y_test)

         32/32 [==============================] - 21s 583ms/step - loss: 0.3212 - accuracy: 0.9080
```

Figure 7 Accuracy of the model

Figure 8 show the dashboard with visualization of the crime data.



PENDING REQUESTS TODAY

6

ACTIVE CCTV

21

CCTV ID	TIME	TYPE	THREAT LEVEL	VIEW CLIP	APPROVAL STATUS
CC123	12:05 PM	Theft	LOW	link	RESOLVED
CC124	3:16 AM	Violence	MEDIUM	link	WAITING

Figure 8 Dashboard

VI Conclusion

The proposed Hybrid CNN and LSTM model generates a warning message and anticipates the threat from the live camera feed. We attempt to perform threat detection and threat prevention on low quality video feeds as a further development of our proposed system.

VII References

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.

[1] H. Liu, S. Chen, and N. Kubota, “Intelligent video systems and analytics: A survey,” IEEE Transactions on Industrial Informatics, vol. 9, no. 3, pp. 1222–1233, 2013.

[2] E. B. Nievas, O. Deniz-Su ´ arez, G. B. Garc ´ ´ıa, and R. Sukthankar, “Violence detection in video using computer vision techniques,” in CAIP.

[3] L. Xu, C. Gong, J. Yang, Q. Wu, and L. Yao, “Violent video detection based on MoSIFT feature and sparse coding,” in ICASSP, pp. 3538–3542, 2014.

[4] Y. I. T. Hassner and O. Kliper-Gross, “Violent flows: Real-time detection of violent crowd behavior,” in CVPR workshops, pp. 1–6, 2012.

[5] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu, “Violence detection using oriented violent flows,” Image and Vision Computing, vol. 48-49, pp. 37–41, 2016.

[6] H. Wang, A. Klaser, C. Schmid, and C. Liu, “Dense ¨ trajectories and motion boundary descriptors for action recognition,” International Journal of Computer Vision, vol. 103, no. 1, pp. 60–79, 2013.

[7] P. Bilinski and F. Bremond, “Human violence recognition and detection in surveillance videos,” in AVSS, pp. 30–36, 2016.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in CVPR, pp. 1725–1732, 2014.

[9] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in ICCV, pp. 4489– 4497, 2015.

[10] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in CVPR, pp. 4724–4733, 2017.