Comparative Analysis of Various Deep Learning Models for Real Life Violence Detection

Soumava Das, Debshankar Dey, Sarthak Saha, and Avipriya Ghosh

[1] Heritage Institute of Technology
[2] Chowbaga Rd, Anandapur, Mundapara, Kolkata,West Bengal 700107, India

Abstract. Video Surveillance plays a crucial role in ensuring public safety in urban areas. Research works have been done in this sector to create an automated violence detection system, yet creating such a system remains challenging due to the complexity of human actions and various environmental conditions. This paper aims to conduct a comparative study of various deep learning models that have been used in the researches conducted so far with the aim to find the best model given limited environmental constraints. We implement and compared five distinct approaches: a custom-designed CNN-LSTM hybrid model, transfer learning-based approaches utilizing pre-trained VGG19 and MobileNetV2 combined with bidirectional LSTM, a 3D CNN architecture, and a ConvLSTM network. All models were evaluated using the same publicly available Real Life Violence Situations Dataset to ensure fair comparison. This comparative analysis aims at giving overview of strength and limitations of various deep learning model with relations to surveillance applications of today, promoting development of efficient autonomous systems.

Keywords: Violence Detection · CNN-LSTM · Transfer Learning · MobileNetV2 · 3D CNN · ConvLSTM.

1 Introduction

Due to growing concerns over public safety in areas such as live video surveillance, the need for violence detection has increased over the past few years. Video-based violence detection systems can monitor high-risk areas and automate alerts, thus enabling rapid response to potential threats. Video data is complicated and diverse by nature with different movements, various backgrounds, as well as background noise, but also different contexts and scenes. Deep learning models must be able to distinguish violence from non-violence scenes.

In this paper, we have critically analyzed various deep learning-based approaches in the context of violence detection. For that, we are going to use a hybrid CNN-LSTM to avail advantages from CNN-based feature extraction and LSTM layers to leverage the ability of learning the temporal features related to a violent action[6]. In addition, we make use of the benefits of pretrained CNN architectures fused with Bidirectional LSTM to leverage the benefits of transfer learning along with bidirectional knowledge in the temporal scenario[11,3].

These models are best suited for the balanced datasets with clear pattern, and the computational overhead reduces because they are pre-trained. For complex data, we have 3D CNN and ConvLSTM models[6]. The 3D CNN model captures spatiotemporal and temporal features by using 3D convolutions. They are helpful in real-time violence detection in continuous video feeds. Meanwhile, with its convolutional and temporal layers, ConvLSTM is well suited to discern fine details in video sequences and, therefore, it is effective for high-sensitive applications related to rapid motion changes[8,10].

Our research attempts to explore all the above-mentioned models of violence detection under different operational needs. In this study, we discuss the selection and performance of the above mentioned under varied environmental constraints and computational limits to try to contribute to the literature on deep learning-based violence detection.

In the article of "Violence Detection in Real Life Videos using Deep Learning", conducted by Jain et al., combined CNN and LSTM with the aim of classifying the video content either into violence or non-violence bearing in mind the capturing of the spatiotemporal features efficiently. Initial segmentation is performed using a U-Net architecture and spatial feature extraction is performed using MobileNet V 2. The LSTM model is used here for analysis of sequential information across the frames for accurate classification in this case with regard to binary cross-entropy as the loss function. For this model, Real-Life Violence Situations Dataset containing 2000 video clips was used. The results of the experiments gave an average accuracy of 95.69%[1,2].

In another paper of "Real Time CCTV Violence Detection System Using Deep Learning", Akole et al presented an approach where a MobileNetV2 and LSTM model was used. The model gave an accuracy of 82.5 % on a Hockey Dataset containing 1000 images, 75.6097% on a Movies Dataset containing 200 images and 88.46% on a custom dataset of 600 images[1,2].

Verma et al., in their paper titled "Violence Detection for Surveillance Systems using Lightweight CNN Models" put forth the idea of using a MobileNetV2 model and a ResNet50V2 model. 3 datasets were used namely Set1, Set2 and Set3, where set 1 is the real-life violence situations dataset, set 2 and set 3 stands for UBI-fights and UFC crime. The accuracies of MobileNetV2 model and the ResNet50V2 model were 94.58% and 91.83% respectively[4].

In the paper of "Human Action Recognition Using Deep Learning Methods" by Yu et al., two-Stream CNN model and a 3D CNN model were used. The accuracies of CNN+LSTM Model, two-Stream CNN model and the 3D CNN model were 89.74%, 82.37% and 86.54% respectively[6].

In another research paper "Robust Real-Time Violence Detection in Video Using CNN And LSTM", Abdali et al., experimented with Conv3d and (CNN and LSTM) models. The base model was built on top of CNN (pre-trained vgg19) as spatial feature extractor followed by LSTM cells. Each item in the base model was with the shape of (40x160x160x3). In the base model, the authors used 700 videos from the hockey dataset for the training set, and 300 videos as test set.

After training the base model on the Hockey Fight Dataset, they achieved an accuracy of 98%[2].

In the paper "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting" by Shi et al., was an extension of deep Q-network (DQN) that proposed to have two independent streams subsequent to the convolutional layers: one to estimate the state value- its importance of being in a specific state-and the other to estimate the advantage of each action-usefulness of each possible action in that state[8].

Wu et al, in their paper "Video Abnormal Event Detection Based on CNN and LSTM" proposed an improved deep learning model CNN-LSTM that was obtained by combining CNN with LSTM. The authors carried out experiments on standard dataset UMN and UCSD and achieved accuracy reaching 99.54% and 92.36% accuracy. Their method showed that their model performed better than many other models[11,3].

2 Methodology

In our study, we present a comprehensive analysis of various approaches to Violence Detection examining methods like Pretrained CNN with LSTM [1,2], 3D CNN [6], as well as a hybrid CNN and LSTM [3] and ConvLSTM[8,10].

2.1 Dataset

In our study Real Life Violence Situations Dataset [7] was used which consists of 1000 videos belonging to classes Violence and Non-Violence. The videos in this dataset are collected from YouTube videos. The violence videos in the dataset contain many real street fight situations in several environments and conditions. The non-violence videos in the dataset are collected from many different human actions like sports, eating, singing, walking, etc.

2.2 Preprocessing

In our approach, the videos were preprocessed by extracting a sequence of frames, then they were resized and normalized. Frames were captured at evenly spaced intervals based on the video's total frame count and the defined sequence length. Capturing frames at regular intervals ensures each video is represented by fixed number of frames that helps in standardizing input data which is essential for feeding consistent input shapes to the model making training stable and efficient. Then each frame was resized to specific height and width and normalized by scaling the pixel values between 0 and 1. Resizing frames to a consistent resolution reduces memory and computational load allowing for faster processing and compatibility with the neural network. Normalization stabilizes model training by ensuring that input values fall within a similar range which improves convergence and makes learning efficient. The processed frames were then added to a list and once all frames were obtained, they were organized into features

and labels corresponding to their class directories. Finally, the features and labels lists were converted to NumPy arrays and returned forming a structured dataset. Then the obtained dataset was split into training and validation dataset. Both the labels of training data and validation data were converted to one hot encoded label that is used in classification tasks to represent categorical class labels which also aids in improving training stability.

2.3 Model Architectures

In our study Real Life Violence Situations Dataset [7] was used which consists of 1000 videos belonging to classes Violence and Non-Violence. The videos in this dataset are collected from YouTube videos. The violence videos in the dataset contain many real street fight situations in several environments and conditions. The non-violence videos in the dataset are collected from many different human actions like sports, eating, singing, walking, etc.

Hybrid CNN-LSTM In our study we first used a hybrid CNN-LSTM [3,11] approach for violence video detection. CNNs are primarily used for analyzing and extracting features from spatial data especially images. CNNs are used to identify patterns, textures, shapes and complex structures within images. LSTMs (Long Short-Term Memory) are a type of Recurrent Neural Network (RNN) that is good at capturing long-term dependencies and are ideal for sequence prediction tasks.[13,15]

Our model consists of five convolutional blocks with each block containing 2D convolutional layer with ReLU activation, Batch Normalization for training stability, MaxPooling for spatial dimensionality reduction and dropout for regularization. Progressive feature map expansion includes 16, 32, 64, 64, 128 filters respectively with 3x3 convolutional kernels and dropout rates varying from 0.3 to 0.25. For the temporal processing we used two LSTM layer configuration having 128 units and 64 units with tanh activation respectively. Then for the classification part, two fully connected layers having 512 and 64 units respectively with L2 regularization (0.01) for weight decay is used. Batch Normalization and dropout (0.4) layers are also added for regularization. Then a final Dense layer with 2 neurons and softmax activation is used for the binary classification task. The model is compiled using Adam optimizer and categorical cross entropy loss function with initial learning rate of 0.001.[11]

ConvLSTM We also experimented using ConvLSTM [8,10]. (Convolutional Long Short-Term Memory) which is a specialized neural network layer that combines the spatial feature extraction capabilities of convolutional layers with the sequential data handling capabilities of LSTM layer. The ConvLSTM architecture works by maintaining the spatial information of its input while also learning temporal dependencies. The architecture of the ConvLSTM model[10] consists of four main components, Convolutional Layers, Forget gate, Input gate and Output gate. Unlike CNN-LSTM architecture where CNN layers are initially

used to extract spatial features and then flattened and passed to LSTM layer that captures temporal dependencies, ConvLSTM[10] is used to directly integrate convolutional operations with LSTM gates allowing it simultaneously to capture both spatial and temporal features.

The ConvLSTM model [8] implemented had two ConvLSTM2D layers, the first layer having 64 filters with kernel size 3x3 followed by a dropout (0.5) layer. The second ConvLSTM2D layer had 32 filters with Kernel size 3x3 followed by a Global Average Pooling Layer. Then it is followed by Dense Layer with 64 Units and Dropout (0.5) Layer. The final output player had 2 neurons with softmax activation that is useful for binary classification. The model was compiled using Adam optimizer and categorical cross entropy loss function with learning rate as 0.0001. A brief architecture of ConvLSTM model.

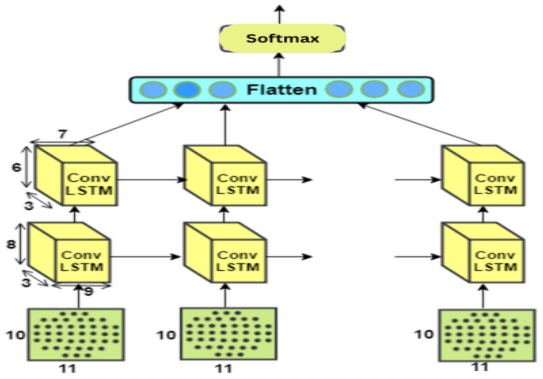


Fig. 1. ConvLSTM Architecture

VGG19-LSTM As studied previously [2], VGG19 has been used to solve this problem. VGG19 (Visual Geometry Group) is a sequential deep learning model used for feature extraction effectively. This model is used to extract the Spatial feature over a period of time. The main advantage of VGG19 is that it is a pre-trained model so transfer learning is employed to use the model to solve the problem [2][14]

Transfer Learning: It is a technique in which we start with a pre-trained model like VGG19. VGG19 has been trained on the ImageNet dataset that has more than 1.2 million images of over 1000 classes. Some of the pre-trained weights are kept frozen to prevent them from any alteration during the training process. New layers are added to the model to cater the needs of the problem. Transfer learning significantly reduces the training time and the model works well even on small datasets.[14]

Bidirectional LSTM is used to extract the temporal information. Bidirectional LSTM works in both forward and backward direction and uses both fu-

ture and past context which can take into consideration both the buildup and aftermath of a violence situation. VGG19 along with bidirectional LSTM offers an effective way to extract the spatial and temporal feature extraction over time which offers an effective way to solve the problem.[12,13]

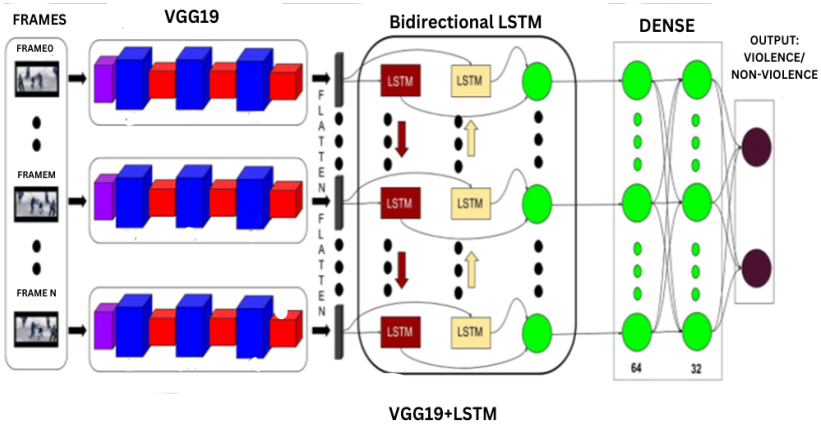


Fig. 2. VGG19 integrated with bidirectional LSTM

The model that was used for training uses 34 layers of which 19 layers belong to the VGG19 model and along with that 15 custom layers are used that include 2-time distributed layer,1 input layer,6 dropout layer,5 dense layer and 1 bidirectional LSTM layer. Of these the last 10 layers and 15 custom layers are trainable. Each dense layer uses ReLu activation function except the last dense layer that uses softmax activation function to find the probability of classification. SGD optimizer was used with 50 epochs and categorical cross entropy loss function with batch size of 8. It provided high accuracy.[11]

MobileNetV2-LSTM We also experimented using the MobileNetV2 model as referenced in [1][2]. It is also a pretrained model but it is much lighter and provides high accuracy as described in [4]. It is used to extract the spatial information over time. A bidirectional LSTM is integrated with the MobileNetV2 model to capture the temporal information for effective feature extraction.

The model that was used for training uses 172 layers of which 155 layers belong to the MobileNetV2 model and along with that 17 custom layers are used that include 2-time distributed layer,1 input layer,7 dropout layer,6 dense layer and 1 bidirectional LSTM layer. Of these layers last 20 layers of MobileNetV2 and 17 custom layers are trainable .Each dense layer uses ReLu activation function except the last dense layer that uses softmax activation function to find the probability of classification. SGD optimizer was used with 50 epochs and

categorical cross entropy loss function with batch size of 8. It provided high accuracy.[11]

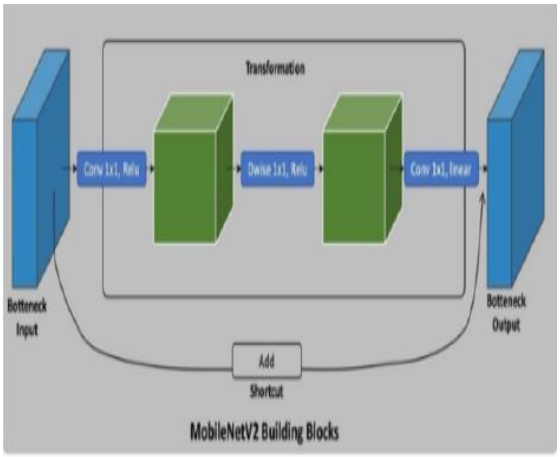


Fig. 3. MobileNetV2 Building blocks

3D CNN We finally experimented with 3D CNN [6,9] to solve this problem. A 3D CNN is a neural network architecture with multiple layers that can learn hierarchical data representations[11]. Each layer learns increasingly complex spatial features of data. 3D CNN is an extension of the 2D CNN architecture, it processes data across three spatial dimensions, width, height and depth(time). Unlike 2D CNN which uses a 2D kernel across the width and height dimension, they use a 3D kernel which moves along the depth(time) dimension. Another key difference is that in 2D CNN, each convolutional layer applies 2D filters to create feature maps by considering spatial patterns within each 2D image. These filters move in two directions (x and y axes) to detect features like edges, textures, and shapes. In contrast, 3D CNNs use 3D filters that move in all three directions (x, y, and z axes), enabling them to capture spatiotemporal relationships or volumetric patterns in the data. This means that a 3D CNN [6,9] can learn features that exist not just within a single 2D plane, but across multiple planes or time steps.[13]

The 3D CNN consists of four convolutional blocks followed by two dense layers and a classification layer, along with modern deep learning practices such as batch normalization and dropout regularization. The input dimensions are (16,120,120,3), taking 16 temporal frames. The four convolutional blocks have increasing number of filters, along with batch normalization and max pooling. The first dense layer has 512 units, the second dense layer has 256 units. Both the dense layers use batch normalization, use ReLU as activation function and

has a dropout of 0.4. The output layer has 2 units, uses Softmax as activation function and uses binary classification. For training, the model employs Adam with a learning rate of 0.001, and binary cross-entropy as loss function.

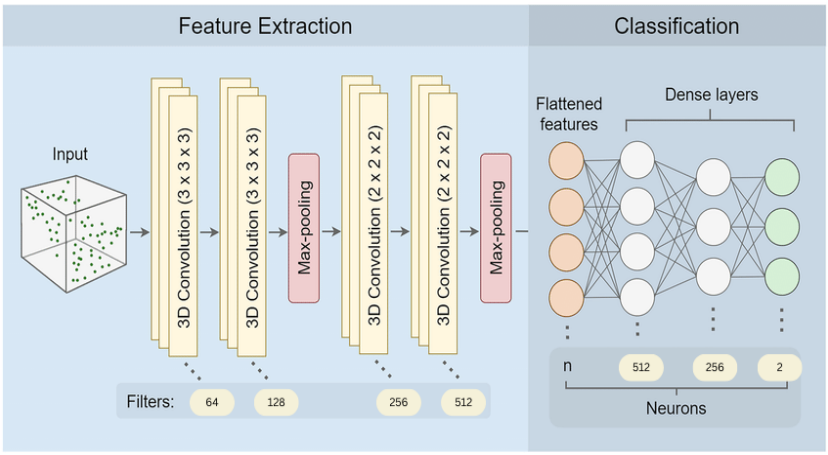


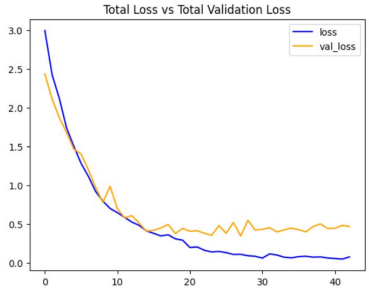
Fig. 4. 3D CNN Architecture

3 Results

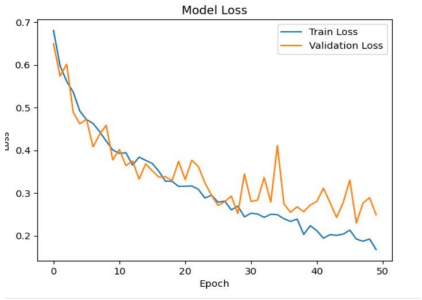
Hybrid CNN-LSTM For the hybrid CNN-LSTM Architecture[11,13,14], the model was trained with an input shape of (16, 120, 120, 3). 16 is the sequence length that represents the number of frames extracted per video. The model is trained with batch size as 8 and 50 epochs with EarlyStopping, ModelCheckpoint and ReduceLROnPlateau.The model ran for 43 out of 50 epochs. The model achieved a best training accuracy of 98.86% with loss of 0.0826 and validation accuracy of 90.71% with loss of 0.402. The model achieved a test accuracy of 85.50% and test loss of 0.49.

ConvLSTM The ConvLSTM[10] model was trained with an input shape of (16, 120, 120, 3). 16 is the sequence length that represents the number of frames extracted per video. The model is trained with batch size as 4 and 50 epochs with ModelCheckpoint. The model achieved a best training accuracy of 92.80% with loss of 0.1871 and validation accuracy of 92.24% with loss of 0.2298. The model achieved a good test accuracy of 89.58% and test loss of 0.53.

VGG19-LSTM The VGG19-LSTM[2,12] model was trained with an input shape of (16, 128, 128, 3). 16 is the sequence length that represents the number



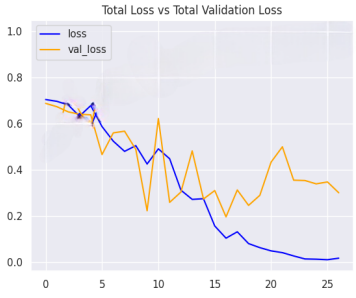
(a) Training Loss vs Validation Loss in Hybrid CNN-LSTM



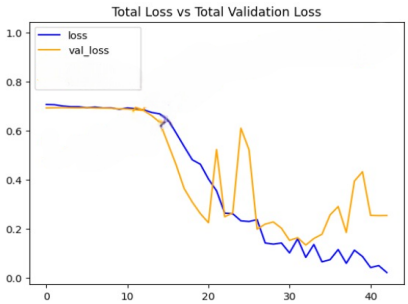
(b) Training Loss vs Validation Loss in ConvLSTM

of frames extracted per video. The model is trained with batch size as 8 and 50 epochs with EarlyStopping, ModelCheckpoint ReduceLROnPlateau. The model ran for 30 out of 50 epochs. The model achieved a best training accuracy of 96.76% with loss of 0.11 and validation accuracy of 96.02% with loss of 0.18. The model achieved a good test accuracy of 90.50% and test loss of 0.51.

MobileNetV2-LSTM The MobileNetV2-LSTM[1,2,13] model was trained with an input shape of (16, 120, 120, 3). 16 is the sequence length that represents the number of frames extracted per video. The model is trained with batch size as 8 and 50 epochs with EarlyStopping, ModelCheckpoint, ReduceLROnPlateau. The model ran for 37 out of 50 epochs. The model achieved a best training accuracy of 99.76% with loss of 0.0167 and validation accuracy of 96.11% with loss of 0.1677. The model achieved a good test accuracy of 95.05% and test loss of 0.1492.



(c) Training Loss vs Validation Loss in VGG-LSTM



(d) Training Loss vs Validation Loss in MobilNetV2-LSTM

3D CNN The 3D CNN model was trained with an input shape of (16, 120, 120, 3) with 16 as the sequence length that represents the number of frames extracted per video. The model is trained with batch size as 8 and 40 epochs with EarlyStopping, ModelCheckpoint, ReduceLROnPlateau. The model ran for 40 out of 40 epochs. The model achieved a best training accuracy of 92.46% with loss of 0.1828 and validation accuracy of 90.01% with loss of 0.2864. The model achieved a good test accuracy of 87.32

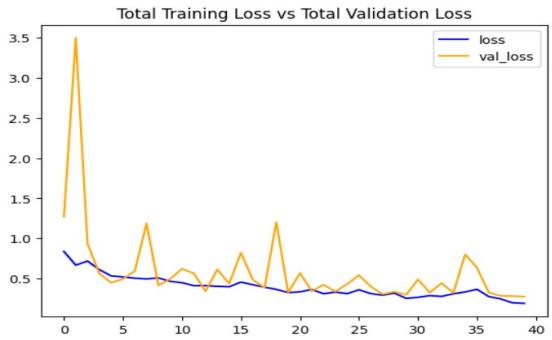


Fig. 5. Training Loss vs Validation Loss in 3D CNN

Model Name	Val Accuracy	Val Loss	Time(s)
Hybrid CNNLSTM	90.70	0.402	8.68
ConvLSTM	92.24	0.229	10.39
VGG19-LSTM	96.02	0.180	23.14
MobilNetV2-LSTM	96.11	0.167	6.12
3D CNN	90.01	0.286	10.50

Table 1. Performance Comparison of Different Models

The times shown in the table are based on testing each model using a dataset of 100 videos.

4 Conclusion

This study shows a comprehensive analysis of various models and their performance in Violence Detection. All the models were trained using GPU T4. Our study shows that based on the limited amount of computation resources and data, MobileNetV2 with LSTM[1,2,12] performed best followed by VGG19 with LSTM[3,12]. Thus, these pretrained CNN models along with LSTM are the best

approach to achieve a robust and fast model. In future, if better and varied dataset is available, even better accuracy maybe achieved.

The frames given below shows the output achieved after conducting experiments conducted on the best model MobilNetV2-LSTM [1]. The first frame is from a video which shows a fight between two person which is correctly classified as Violence. The second frame is from a video taken from a CCTV camera feed in a crowded area where there is no violence taking place and the model correctly classified it as Nonviolence.

Fig. 6. Video Classification output in MobinetV2+LSTM

References

1. P. Akole, I. Sarode, T. Raut, D. Mahadik and P. Futane, "Real Time CCTV Violence Detection System Using Deep Learning," 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2023, pp. 1-6, doi: 10.1109/ESCI56872.2023.10099886.
2. A. -M. R. Abdali and R. F. Al-Tuma, "Robust Real-Time Violence Detection in Video Using CNN And LSTM," 2019 2nd Scientific Conference of Computer Sciences (SCCS), Baghdad, Iraq, 2019, pp. 104-108, doi: 10.1109/SCCS.2019.8852616.
3. G. Wu, Z. Guo, L. Li and C. Wang, "Video Abnormal Event Detection Based on CNN and LSTM," 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), Nanjing, China, 2020, pp. 334-338, doi: 10.1109/ICSIP49896.2020.9339428.
4. N. Suba, A. Verma, P. Baviskar and S. Varma, "Violence detection for surveillance systems using lightweight CNN models," 7th International Conference on Computing in Engineering & Technology (ICCET 2022), Online Conference, 2022, pp. 23-29, doi: 10.1049/icp.2022.0587

5. B. Jain, A. Paul and P. Supraja, "Violence Detection in Real Life Videos using Deep Learning," 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2023, pp. 1-5, doi: 10.1109/ICAECT57570.2023.10117775.
6. Z. Yu and W. Q. Yan, "Human Action Recognition Using Deep Learning Methods," 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ), Wellington, New Zealand, 2020, pp. 1-6, doi:10.1109/IVCNZ51579.2020.92905946
7. M. Soliman, M. Kamal, M. Nashed, Y. Mostafa, B. Chawky, D. Khattab, " Violence Recognition from Videos using Deep Learning Techniques", Proc. 9th International Conference on Intelligent Computing and Information Systems (ICICIS'19), Cairo, pp. 79-84, 2019.
8. X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, W. C. Woo,"Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting", NIPS, 2015.
9. D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 4489-4497, doi: 10.1109/ICCV.2015.510.
10. S. Sudhakaran and O. Lanz, "Learning to detect violent videos using convolutional long short-term memory," 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 2017, pp. 1-6, doi: 10.1109/AVSS.2017.8078468.
11. Eijaz Allibhai, 'Building a Convolutional Neural Network (CNN) in Keras', https://towardsdatascienc e.com/buildinga- convolutional-neural-network-cnn-in-keras- 329fbbadc5f5, accessed 5 January, 2021.
12. Sepp Hochreiter, Jurgen Schmidhuber, "Long Short-Term Memory" Neural Computation 9(8):1735-1780, 1997
13. Ian Godfellow, "Deep Learning (Adaptive Computation and Machine Learning series)" 18 November 2016
14. Simon Haykin, "Neural Networks and Learning Machines Third Edition" Hamilton, Ontario, Canada, 2009
15. Introduction to Convolutional Neural Network - Geek for Geeks