International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Study on Deep Learning Techniques for Finding Suspicious Violence Detection in a Video Surveillance

B. Venkatesh[1], Yedlla Satyam[2],Dr.N.Sandhya[3]

[1]Assistant Professor,Dept. of CSE-AIML&IoT
VNR Vignana Jyothi Institute of Engineering & Technology
Hyderabad, India.
venkycaptain@gmail.com

[2]Assistant Professor,Dept. of IT
CMR Technical Campus
Hyderabad, India
yedla.satyam@gmail.com

[3]Professor,Dept. of CSE-AIML&IoT
VNR Vignana Jyothi Institute of Engineering & Technology
Hyderabad, India.
sandhyanadela@gmail.com

Abstract—Detecting suspicious visual objects is essential to applying automatic violence detection (AVD) in video surveillance. Continuous monitoring of objects or any unusual things is a tedious task. Learning about video surveillance is an emerging research problem in AVD applications. Deep learning is an intelligent and trustworthy technique for detecting or classifying suspicious data objects. It classifies suspicious video frames by modeling specific categories of videos. The current deep models convolutional neural network (CNN), convolutional long-term and short-term memory (ConvLSTM), AlexNet, VGG-16, MobileNet, and GoogleNet, are wildly succeeded in real-time violence detection with the input of video clips. This paper presents the findings of experimental studies for deep models using classification measures to demonstrate the models' efficacy for our AVD application. Benchmarked violence (V), non-violence (NV), and weapon violence (WV) video datasets are used in the experiment to describe the model's performance while classifying suspicious videos for public safety.

Keywords-Video Surveillance, Objects Detection, Deep Learning, Model Training, Automatic Violence Detection.

I. INTRODUCTION

Video surveillance is ubiquitous for detecting descriptive or suspicious objects in a real-time public safety application [1]. The emerging visual systems must classify normal and abnormal objects by detecting motion in a dynamic video. Most organizations seeking video surveillance detection systems include banks, hospitals, shopping malls, airports, railway stations, and other crowded places [2]. Machine learning techniques [3] play a significant role in learning videos of suspected objects. Present video surveillance systems use CCTVs to ensure the security of people in public places. However, these systems support post-investigation after the crime activity happens. Thus, this paper focuses more on detecting suspicious objects dynamically during video surveillance without human intervention to prevent crimes and provide public safety in crowded places. In video surveillance, a manual investigation is initiated after the crime has happened. However, faster findings of suspicious objects are required to prevent crimes and protect the public. Thus, this paper studies the various deep learning techniques for effective video learning [4] to find suspicious videos. Continuous video learning is the most emerging task for the dynamic detection of violence in public places. Developing violence detection systems [5] is of prime importance and deep learning models efficiently implement it, which employ generalized models to recognize many suspicious activities in videos.

Prompt detection of suspicious activities is achieved using various deep-learning models and methodologies. In order to automatically extract relevant characteristics from video frame data and construct reliable representations, deep learning models are employed. The deep learning method of Convolutional Neural Networks (CNNs) [6] is used for object recognition in videos because of their ability to learn visual patterns directly from a video frame. CNN has several applications, including automatic sign identification and object detection. The CNN model's credibility can be improved through training with several architectures such as Resent, Alex net, VGG net, and dense net [7], [8], and [9]. Fig. 1 shows the typical framework of CNN of video learning for specious object detection.

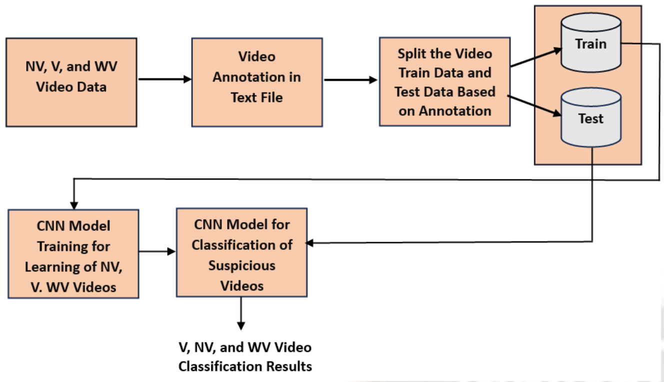

NV, V, and WV Video Data
Video Annotation in Text File
Split the Video Train Data and Test Data Based on Annotation
Train
Test
CNN Model Training for Learning of NV, V. WV Videos
CNN Model for Classification of Suspicious Videos
V, NV, and WV Video Classification Results

Figure 1. Typical Framework of CNN for SuspiciousVideo Classification Detection

This paper studies the following experimental findings of state-of-the-art video learning techniques.

1. The convolutional neural network(CNN) learning rate is determined with several performance measures for object detection.
2. Classification results are demonstrated with recent deep learning models for recognizing violence/non-violence object detection in video surveillance.
3. CNN, ConVLSTM, and other recent neural network models are experimentally studied by finding their model accuracies in finding real-time violence detection.

The following sections present the preliminaries of specious object detection, deep learning techniques, experimental study of deep learning methods, and conclusion with the future scope of the work. All Type Style and Fonts

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

II. PRELIMINARIES OF DEEP LEARNING NETWORKS

This paper describes the essential preliminaries of the learning process of video surveillance for real-time suspicious object detection. Different classes of videos are collected for deep model training. Its pre-phase step is familiar to any of the deep learning models. In the CNN model, frames are extracted initially in the training and testing video datasets, and then features are extracted from the same frames. Video annotation [10] is a critical step in the training phase of the CNN model, and it was done by adding the file name of the video file in the cache and the identical copy in the text file.

A. Video Data Pre-Processing Phase

Each video file consists of dynamic movable frames, and OpenCV extracted these frames in python. An empty list is initially defined, and the extracted frames are stored using the same library. All frames of the video need not be stored in the list. By defining the frame interval (total number of frames/sequence length), only a few frames need to be stored [11]. These collected frames maintain the frame gap or defined frame interval. These frames' size is further reduced from the original size to 224x224 or 64x64. The reduced frame size makes the training and test process faster during deep learning.

B. Frame Level Feature Extraction

After the reduction of frame sizes, either 224x224 or 64x64, features of frames are extracted using the variant of the CNN model, namely, the inception V3 model [12]. It noted that the following preliminaries need to be set up for model training of video clips: ‘weights=imagnet’, ‘top=false’, ‘pooling=avg’, and ‘input shape=size of the frame.’

C. Architectural Basics of Convolutional Neural Networks

The following are the standard terms used in this literature. The sets of parameters used in convolutional processes that can be trained are called "kernel or filter matrix" [13]. Although weight and parameter are often used interchangeably, we have tried to use the former when referring to a parameter (or kernel) outside of convolution layers. Typically, a convolutional neural network (CNN) will have three layers: a convolutional layer, a pooling layer, and a fully connected layer.Convolution and pooling, the first two layers, extract features, while the fully connected layer, the third layer, integrates the extracted features into the output, such as classification. The design of convolutional neural networks (CNNs) includes a convolution layer, a specific type of linear operation [14].

Convolution, pooling, and fully linked layers are just some of the fundamental components of the CNN architecture [15]. Multiple convolution layers, followed by a pooling layer, and then one or more fully connected layers are typical components of such an architecture. The method by which these levels transform input data into output is known as forward propagation. While this section focuses on 2D-CNN, the convolution and pooling techniques discussed are readily applicable to 3D-CNN [16].

CNN requires GPUs for model training since it is more computationally intensive and needs significantly more data to train its models because it has millions of learnable parameters to estimate.

D. Convolution Operation and Activation Functions

A common CNN design uses a convolution layer to perform feature extraction. This layer combines linear and nonlinear processes (the convolution operation and the activation function) to provide an output [17], [18], [19]. Convolution is a technique for extracting features from data by applying a small array of numbers known as the "kernel" to the input, which is also an array of numbers known as the "tensor." As a consequence of doing a feature-wise multiplication over each element of the kernel's value and the source tensor and summing the results, the output value at each position of the output tensor, also known as a feature map, can be obtained. Depending on the activation function, the output of a neural network can range from 0 to 1 or -1 to 1, respectively. There are two distinct kinds of activation functions: linear and non-

Article Received: 25 July 2023 Revised: 12 September 2023 Accepted: 30 September 2023

linear.The output of a linear activation function is also unbounded, therefore its shape is also linear. Sigmoid activation function, Tanh activation function, and ReLU (Rectified Linear Unit) activation function [20] are all examples of non-linear activation functions.The most common activation function is the Sigmoid or Logistic one. It is a value between zero and one. The slope of the sigmoid curve between any two points can be calculated to the differentiability of the sigmoid activation function. The activation function of the hyperbolic tangent, or Tanh, varies from -1 to 1.

E. Fully Connected Layers

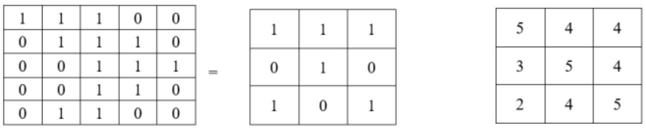
In deep learning architectures, it is common practise to flatten the feature maps produced by the final convolution or pooling layer into a 1D array of numbers (or vector), which is then connected to one or more fully connected layers [21], [22] (also called dense layers), in which each input is connected to each output by a learnable weight. Using the information recovered by the convolution layers and down-sampled by the pooling layers, a subset of fully connected layers maps the network's final outputs, such as the probabilities for each class in classification tasks. Most of the time, the number of classes determines the total number of output nodes in the final fully linked layer.

III. CONVOLUTIONAL NEURAL NETWORK (CNN), MAX AND AVERAGE POOLING, AND RECTIFIED LINEAR UNIT (RELU) MODELS FOR REDUCTION OF FRAME SIZE FOR HIGH DIMENSIONAL VIDEO REPRESENTATION

Finding distinct representations by video learning is a crucial problem. By designing model training, CNN significantly distinguishes the features of dynamic or movable objects. It aimed to convolve the inputs and process the reduced representation of high-dimensional video data. It uses the kernel or filter matrix to reduce the high-dimensional image or frames of video data. It is performed as per the following Eqn. (1).

$$Reduced_{Conv_{Matrix}} = \sum_{i=1}^{M} \sum_{j=1}^{N} K_{i,j} . F_{i+m,j+m} \quad (1)$$

Whereas K refers to the kernel or filter matrix with the size of m x m, and F refers to the original high-dimensional frame or image matrix with the size of MxN. The operator '.' is applied between the kernel and frame matrix. Convolutional neural networks, often known as CNNs or ConvNets, are a type of deep neural network that is used in deep learning. Their primary purpose is to analyze visual data. ConvNet does not involve the matrix multiplications that typically come to mind when discussing neural networks. In mathematics, convolution is a process whereby two functions are multiplied together to get a third function that represents how the first function's shape has been altered by the second. Way of computing the convolved features for the black-white image or frames of a video clip are as follows:



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

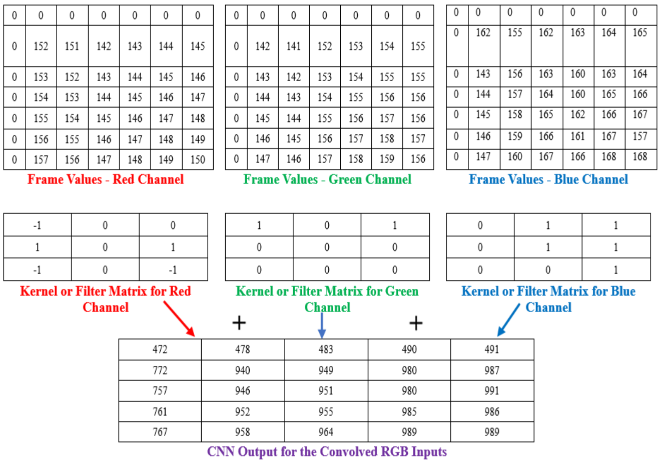
=

1	1	1
0	1	0
1	0	1

5	4	4
3	5	4
2	4	5

Image or Frame (5 x5) Kernel of Filter Matrix (3 x 3) Convolved Features

If the image is in colour, the same process is applied for R, G, B channels and make a sum of their convolved features of three R, G, B channels. The following example is illustrated for obtaining the convolved features under the three channels.



0	0	0	0	0	0	0
0	152	151	142	143	144	145
0	153	152	143	144	145	146
0	154	153	144	145	146	147
0	155	154	145	146	147	148
0	156	155	146	147	148	149
0	157	156	147	148	149	150

Frame Values - Red Channel

0	0	0	0	0	0	0
0	142	141	152	153	154	155
0	143	142	153	154	155	155
0	144	143	154	155	156	156
0	145	144	155	156	157	156
0	146	145	156	157	158	157
0	147	146	157	158	159	156

Frame Values - Green Channel

0	0	0	0	0	0	0
0	162	155	162	163	164	165
0	143	156	163	160	163	164
0	144	157	164	160	165	166
0	145	158	165	162	166	167
0	146	159	166	161	167	157
0	147	160	167	166	168	168

Frame Values - Blue Channel

-1	0	0
1	0	1
-1	0	-1

Kernel or Filter Matrix for Red Channel

1	0	1
0	0	0
0	0	0

Kernel or Filter Matrix for Green Channel

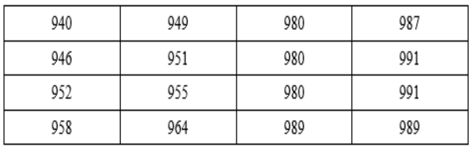
0	1	1
0	1	1
0	0	1

Kernel or Filter Matrix for Blue Channel

472	478	483	490	491
772	940	949	980	987
757	946	951	980	991
761	952	955	985	986
767	958	964	989	989

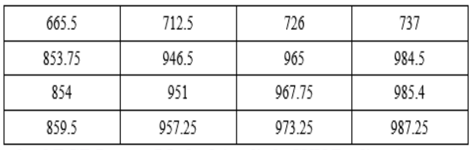
CNN Output for the Convolved RGB Inputs

Two classical types of pooling are widely used in CNN, which are max pooling and average pooling. For the size of kernel matrix on a frame values, max value is picked and placed in the result of pooling max; similarly, average of kernel regions of frame values are placed in the resulting convolved matrix when pooling average is applied. and average frame values of kernel fixed region. These results are shown in the following. Obtained convolution matrix size of (5 x 5), in which either maximum pooling (max pooling) or average pooling (avg pooling) are applied with filter matrix 'F' of size 2 x 2 and stride S=1; then, the resulting features matrix size is computed (using the formula (I-F)/S + 1), whereas I refers to either number of rows or columns of resulting convolution matrix. The resulting max pooling (or avg pooling) is 4 x 4. Obtained resulting max pooling (or avg pooling) matrices are as follows:



940	949	980	987
946	951	980	991
952	955	980	991
958	964	989	989

Output for the Convolved RGB Inputs (Method Applied: CNN Pooling Max)



665.5	712.5	726	737
853.75	946.5	965	984.5
854	951	967.75	985.4
859.5	957.25	973.25	987.25

Output for the Convolved RGB Inputs (Method Applied: CNN Pooling Average)

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

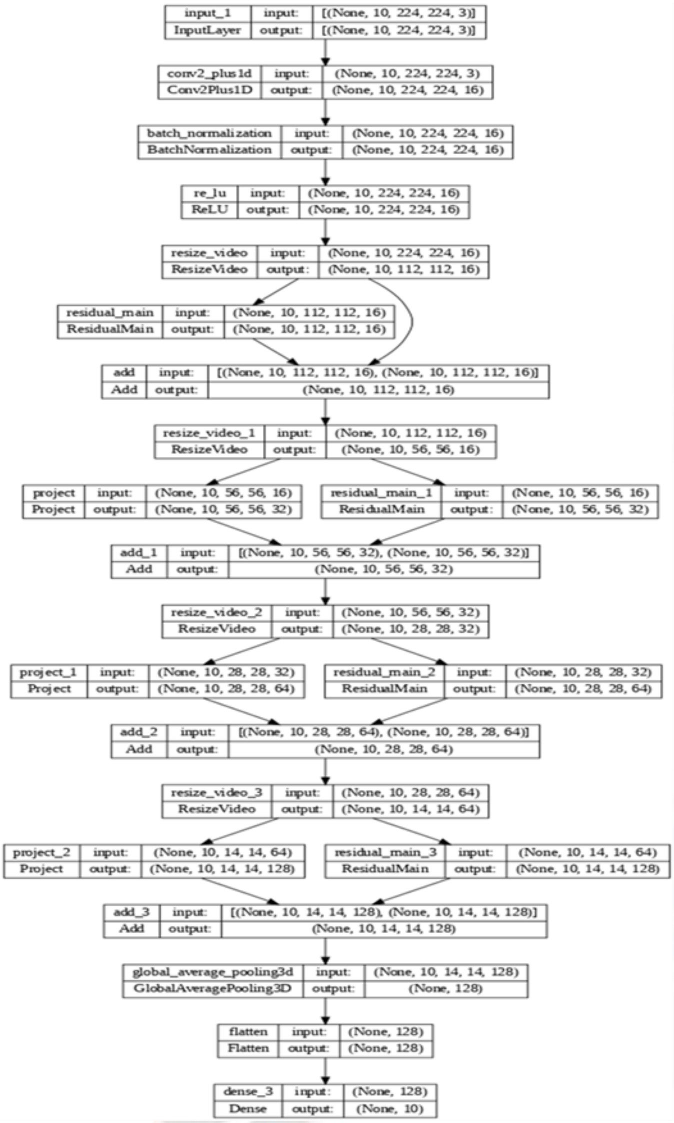

input_1 | input: | [(None, 10, 224, 224, 3)]
InputLayer | output: | [(None, 10, 224, 224, 3)]
conv2_plus1d | input: | (None, 10, 224, 224, 3)
Conv2Plus1D | output: | (None, 10, 224, 224, 16)
batch_normalization | input: | (None, 10, 224, 224, 16)
BatchNormalization | output: | (None, 10, 224, 224, 16)
re_lu | input: | (None, 10, 224, 224, 16)
ReLU | output: | (None, 10, 224, 224, 16)
resize_video | input: | (None, 10, 224, 224, 16)
ResizeVideo | output: | (None, 10, 112, 112, 16)
residual_main | input: | (None, 10, 112, 112, 16)
ResidualMain | output: | (None, 10, 112, 112, 16)
add | input: | [(None, 10, 112, 112, 16), (None, 10, 112, 112, 16)]
Add | output: | (None, 10, 112, 112, 16)
resize_video_1 | input: | (None, 10, 112, 112, 16)
ResizeVideo | output: | (None, 10, 56, 56, 16)
project | input: | (None, 10, 56, 56, 16)
Project | output: | (None, 10, 56, 56, 32)
residual_main_1 | input: | (None, 10, 56, 56, 16)
ResidualMain | output: | (None, 10, 56, 56, 32)
add_1 | input: | [(None, 10, 56, 56, 32), (None, 10, 56, 56, 32)]
Add | output: | (None, 10, 56, 56, 32)
resize_video_2 | input: | (None, 10, 56, 56, 32)
ResizeVideo | output: | (None, 10, 28, 28, 32)
project_1 | input: | (None, 10, 28, 28, 32)
Project | output: | (None, 10, 28, 28, 64)
residual_main_2 | input: | (None, 10, 28, 28, 32)
ResidualMain | output: | (None, 10, 28, 28, 64)
add_2 | input: | [(None, 10, 28, 28, 64), (None, 10, 28, 28, 64)]
Add | output: | (None, 10, 28, 28, 64)
resize_video_3 | input: | (None, 10, 28, 28, 64)
ResizeVideo | output: | (None, 10, 14, 14, 64)
project_2 | input: | (None, 10, 14, 14, 64)
Project | output: | (None, 10, 14, 14, 128)
residual_main_3 | input: | (None, 10, 14, 14, 64)
ResidualMain | output: | (None, 10, 14, 14, 128)
add_3 | input: | [(None, 10, 14, 14, 128), (None, 10, 14, 14, 128)]
Add | output: | (None, 10, 14, 14, 128)
global_average_pooling3d | input: | (None, 10, 14, 14, 128)
GlobalAveragePooling3D | output: | (None, 128)
flatten | input: | (None, 128)
Flatten | output: | (None, 128)
dense_3 | input: | (None, 128)
Dense | output: | (None, 10)

Figure 2. Convolutional Neural Network for Reduction of Dimensions of Input Data

In Fig. 2 we show the overall framework of the experimental setup. The output of the convolved inputs of the high-dimensional video frames (of the three channels) shows the reduced representation of video data without losing the frame features of the datasets. Because of this, CNNs are useful for avoiding the "Curse of Dimensionality" problem because of the exponential increase in computing needed to accomplish a machine-learning task in response to the linear growth in the dimensionality of the input. The convolutional pooling is an intermediatory result found in the hidden neural network layer. Thus, key steps of CNNs are defined as follows:

- Find the image or frame-level information of video data with three channels (R, G, and B).
- Apply the convolutions and pooling with the kernel (or filter) matrix.
- Determine the low-manifolds of high-dimensional video data for learning and classification with considerable amounts of computational time.

The activation function is essential in the CNNs. The activation function in a neural network is the mathematical operation that takes the weighted sum of the input at a node and returns the node's activation as an output. Negative values are also derived in a few of the cases in CNNs. When applying the activation function, the negative outputs are passed into the inputs of successive hidden layers. In a Rectified Linear Unit (ReLU), the positive output values are passed as the inputs of the next hidden layer. ReLU replaces the zeros if the negative outputs are received before passing the inputs of the next hidden layer. Artificial neural networks (ANN) [23], which attempt to simulate cognitive abilities in activation function, are growing in machine learning algorithms. However, as technology has advanced, a large amount of data has been generated for which ANN has fewer limits, such as processing data in batches or demanding greater parameters for computationally expensive input. Since the ANN model could not process input sequences like audio, text, and video, the Recurrent Neural Network (RNN) [24] was created to address this problem. It has been found, however, that RNNs fall to the vanishing gradient problem as their layer count increases. A function's gradient measures how much its output fluctuates in response to slight input variations. A function's slope, or gradient, is a different term for its value. A higher gradient indicates a faster learning rate for a model. However, if the slope is 0, the model will stop learning. A gradient measures all weight changes about error changes. For models to continue learning, vanishing gradients occur when gradient values are too tiny. For example, while trying to predict a word, we might need more information than just the one preceding it. The term "vanishing gradient problem" describes this situation. More than one word's worth of background may be needed when doing word prediction. The RNN design is the basis for the LSTM architecture [25], which has shown effectiveness in temporal data tasks. LSTM networks' capacity to capture long-term dependencies in the data is the key to their reliable high performance. Long-Short Term Memory (LSTM) Networks are a type of Recurrent Neural Network that is useful for dealing with the vanishing gradient problem. From the computational view, the LSTM is more efficient than RNN for long-input data cases. The LSTM is especially suited for drawing the outcomes over either video or text data's long-term data. ConvLSTM [26] is a network variation of the LSTM family that uses a convolution technique to learn to identify spatial patterns in high-dimensional input. For spatial sequence data found in video, satellite, and radar image datasets, LSTM is not a viable option due to its one-dimensional input data. Data in three dimensions works best with ConvLSTM [28].

IV. EXPERIMENTAL OBSERVATIONS OF DEEP LEARNING TECHNIQUES

This paper aims to study deep neural networks experimentally to detect suspicious crowd videos when violence happens. Detecting violence dynamically is a real challenging issue in public places. This paper uses state-of-the-art deep learning techniques to detect suspicious or abnormal activities during video surveillance efficiently. For the experimental purpose, three categories of videos are collected [27], which are violence (V), non-violence (NV), and weapon violence crowd scenes. In total, 60 high-dimensional video clips of each category are collected and pre-processed with uniform dimensions of frames. Fig. 3 shows the sample videos (NV, V, and WV) used in this experimental study.

a. Sample Non-Violence (NV) Videos
b. Sample Violence (V)Videos
c.Sample Violence (V)Videos

Figure 3. Sample Videos Used in the Experimental Study

Three specific models say, CNN, 3D CNN, and ConvLSTM, are experimentally studied to classify suspicious and non-suspicious activities during video surveillance. Large amounts (2/3rd) of NV, V, and WV high-quality video clips are used for modeling, and the remaining video clips are used for finding the testing accuracy in video classification of violence and non-violence data.

Table 1. Deep Learning Models- Accuracy, Loss, and Runtime for Detection of Violence, Non-violence, and Weapon-Violence during Video Surveillance

Epoch Number	Accuracy			Loss			Runtime (in seconds)		
	CNN	3CNN	Conv LSTM	CNN	3CNN	Conv LSTM	CNN	3CNN	Conv LSTM
1	0.348	0.777	0.376	1.454	1.416	1.326	426	144	115
2	0.404	0.963	0.432	1.201	0.837	1.152	412	82	28
3	0.460	1.000	0.416	1.105	0.670	1.175	402	87	27
4	0.415	1.000	0.472	1.046	0.524	0.984	399	81	29
5	0.449	1.000	0.528	1.040	0.435	0.939	402	91	28
6	0.471	0.963	0.624	1.035	0.354	0.791	397	87	27
7	0.460	1.000	0.720	1.045	0.309	0.736	396	88	26
8	0.404	1.000	0.648	1.073	0.273	0.762	399	87	28
9	0.539	1.000	0.696	0.968	0.236	0.713	406	79	30
10	0.550	1.000	0.760	0.920	0.209	0.636	431	88	27

Runtime comparison of three deep learning models, CNN, 3CNN, and ConvLSTMare shown in Fig. 4.

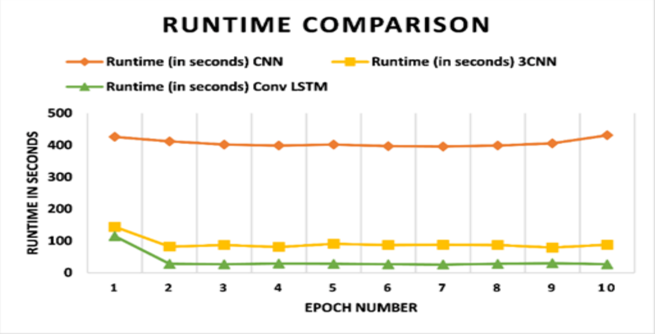


Figure4. CNN, 3CNN, and ConvLSTM Runtime Comparison

The experimental study of deep learning models observed that both 3CNN and ConvLSTM performed best when considering the accuracy and loss in the validation of testing data with the respective trained models. 3CNN achieved good accuracy and fewer loss values than ConvLSTM. However, ConvLSTM consistently improves the accuracy at every new epoch and dramatically reduces the loss value. Compared to the other two models, it is faster and achieves the optimum values of accuracy and loss. For high-quality video learning, it is required to focus more on efficient deep learning models to avoid the scalability issues concerning the runtime parameter. Thus, ConvLSTM is one of the best deep-learning models for real-time suspicious activity detection in video surveillance.

Accuracy and loss comparison of ConvLSTM shown in Fig. 5. (from Epoch 1 to Epoch. 25)

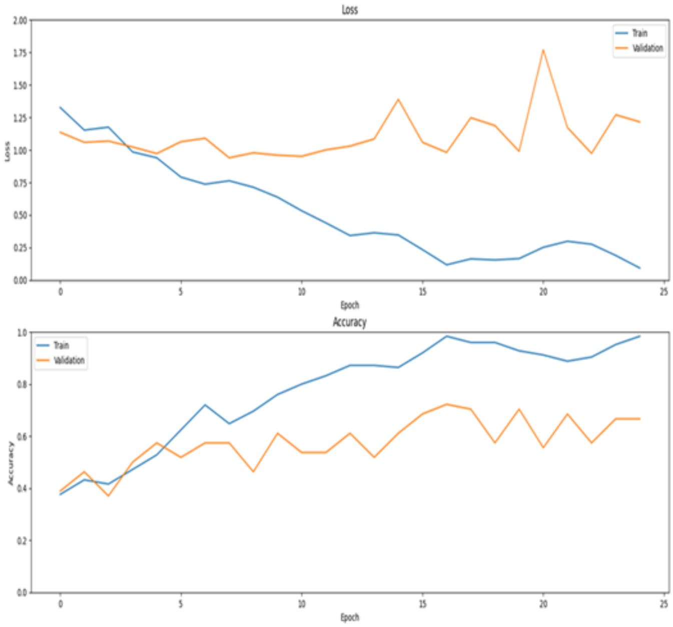


Figure5. ConvLSTM Accuracy and Loss Values for the Different Epochs

Fig.5 shows that accuracy is gradually increased, and loss is gradually reduced for every successive epoch. At every new epoch, weight values are updated in ConvLSTM. Therefore, it achieves high accuracy at the final epoch. The training data was tested for accuracy and loss at every new epoch. For both training and validation data test cases, the gradual increase of accuracy and reduction in loss values is clearly presented in the above diagram.

V. CONCLUSION WITH FUTURE SCOPE OF THE WORK

Public video surveillance and dynamic violence detection have become significant challenges. The deep learning models successfully handle this problem. This paper presents a deep learning study of the research as the experimental analysis using the three categories of the videos, say, non-violence, violence, and weapon violence. It concludes that the 3CNN and ConvLSTM achieved the best accuracy for the detection of violent activities. By studying experimental analysis, ConvLSTM is scalable and efficient considering the runtime parameter, and the experiment study shows that it is 15-25% faster than 3CNN. Thus, it is a recommended deep learning model for real-time video surveillance applications. In future work, it needs to enhance the ConVLSTM with the statistical modeling techniques for robust violence detection activities in public places.

REFERENCES

[1]. Buttar, A.M., Bano, M., Akbar, M.A. et al. Toward trustworthy human suspicious activity detection from surveillance videos using deep learning. Soft Comput (2023). https://doi.org/10.1007/s00500-023-07971-x

[2]. Mahdi MS, Mohammed AJ (2021) Detection of unusual activity in surveillance video scenes based on deep learning strategies. J Al-QadisiyahComput Sci Math 13:1

[3]. S. Soma and N. Waddenkery, "Machine-Learning Object Detection and Recognition for Surveillance System using YoloV3," 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichy, India, 2022, pp. 1-5, doi: 10.1109/ICEEICT53079.2022.9768558.

[4]. L. Abdul Saleem and E. V. Reddy, "A Survey on Deep Learning based Video Surveillance Framework," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICCCI56745.2023.10128302.

[5]. Mumtaz, A. B. Sargano and Z. Habib, "Violence Detection in Surveillance Videos with Deep Network Using Transfer Learning," 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 2018, pp. 558-563, doi: 10.1109/EECS.2018.00109.

[6]. D. B, R. K. K, R. S and R. R, "Improved Object Detection in Video Surveillance Using Deep Convolutional Neural Network Learning," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 1-8, doi: 10.1109/I-SMAC52330.2021.9640894.

[7]. S. Arya and R. Singh, "A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango leaf," 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 2019, pp. 1-6, doi: 10.1109/ICICT46931.2019.8977648.

[8]. T. -B. Xu and C. -L. Liu, "Deep Neural Network Self-Distillation Exploiting Data Representation Invariance," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 1, pp. 257-269, Jan. 2022, doi: 10.1109/TNNLS.2020.3027634.

[9]. Mehta, N.K., Prasad, S.S., Saurav, S. et al. Three-dimensional DenseNet self-attention neural network for automatic detection of student's engagement. Appl Intell 52, 13803–13823 (2022). https://doi.org/10.1007/s10489-022-03200-4.

[10]. Albiol, J. Silla, A. Albiol, J. M. Mossi and L. Sanchis, "Automatic video annotation and event detection for video surveillance," 3rd International Conference on Imaging for Crime Detection and Prevention (ICDP 2009), London, 2009, pp. 1-5, doi: 10.1049/ic.2009.0270.

[11]. D. Huszár, V. K. Adhikarla, I. Négyesi and C. Krasznay, "Toward Fast and Accurate Violence Detection for Automated Video Surveillance Applications," in IEEE Access, vol. 11, pp. 18772-18793, 2023, doi: 10.1109/ACCESS.2023.3245521.

[12]. M. T. Bhatti, M. G. Khan, M. Aslam and M. J. Fiaz, "Weapon Detection in Real-Time CCTV Videos Using Deep Learning," in IEEE Access, vol. 9, pp. 34366-34382, 2021, doi: 10.1109/ACCESS.2021.3059170.

[13]. Montesinos López, O.A., Montesinos López, A., Crossa, J. (2022). Convolutional Neural Networks. In: Multivariate Statistical Machine Learning Methods for Genomic Prediction. Springer, Cham. https://doi.org/10.1007/978-3-030-89010-0_13

[14]. R. Padmanabhan, S. Damodaran, V. Navkesh Batra and S. Gurugopinath, "A Convolutional Neural Network Architecture for Camera Model Identification with Small Datasets," 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2020, pp. 1-6, doi: 10.1109/CONECCT50063.2020.9198595.

[15]. Nirthika, R., Manivannan, S., Ramanan, A. et al. Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study. Neural Comput&Applic 34, 5321–5347 (2022). https://doi.org/10.1007/s00521-022-06953-8

[16]. Sarma, D., Kavyasree, V. & Bhuyan, M.K. Two-stream fusion model using 3D-CNN and 2D-CNN via video-frames and optical flow motion templates for hand gesture recognition. Innovations SystSoftw Eng (2022). https://doi.org/10.1007/s11334-022-00477-z

[17]. Ganivada, A., Yara, S. A novel deep convolutional encoder–decoder network: application to moving object detection in videos. Neural Comput&Applic (2023). https://doi.org/10.1007/s00521-023-08956-5

[18]. X. Wang, "Research on Video Surveillance Violence Detection Technology Based on Deep Convolution Network," 2022 International Conference on Information System, Computing and Educational Technology (ICISCET), Montreal, QC, Canada, 2022, pp. 347-350, doi: 10.1109/ICISCET56785.2022.00086.

[19]. R. Mahima, M. Maheswari, S. Roshana, E. Priyanka, N. Mohanan and N. Nandhini, "A Comparative Analysis of the Most Commonly Used Activation Functions in Deep Neural Network," 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2023, pp. 1334-1339, doi: 10.1109/ICESC57686.2023.10193390.

[20]. Urenda, J.C., Kreinovich, V. (2022). Why Rectified Linear Activation Functions? Why Max-Pooling? A Possible Explanation. In: Castillo, O., Melin, P. (eds) New Perspectives on Hybrid Intelligent System Design based on Fuzzy Logic, Neural Networks and Metaheuristics. Studies in Computational Intelligence,

vol 1050. Springer, Cham. https://doi.org/10.1007/978-3-031-08266-5_28

[21]. Ghosh, R. Determining Top Fully Connected Layer's Hidden Neuron Count for Transfer Learning, Using Knowledge Distillation: a Case Study on Chest X-Ray Classification of Pneumonia and COVID-19. J Digit Imaging 34, 1349–1358 (2021). https://doi.org/10.1007/s10278-021-00518-2

[22]. J. -N. Wu, "Compression of fully-connected layer in neural network by Kronecker product," 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), Chiang Mai, Thailand, 2016, pp. 173-179, doi: 10.1109/ICACI.2016.7449822.

[23]. Hosameldin Ahmed; Asoke K. Nandi, "Artificial Neural Networks (ANNs)," in Condition Monitoring with Vibration Signals: Compressive Sampling and Learning Algorithms for Rotating Machines , IEEE, 2019, pp.239-258, doi: 10.1002/9781119544678.ch12.

[24]. M. Kaur and A. Mohta, "A Review of Deep Learning with Recurrent Neural Network," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2019, pp. 460-465, doi: 10.1109/ICSSIT46314.2019.8987837.

[25]. Y. -J. Chung and C. -H. Shen, "Research on Deep Learning with Gesture Recognition and LSTM in Sign Language," 2022 IEEE 5th International Conference on Knowledge Innovation and Invention (ICKII), Hualien, Taiwan, 2022, pp. 193-195, doi: 10.1109/ICKII55100.2022.9983520.

[26]. S. T. Sarcar and M. A. Yousuf, "Detecting Violent Arm Movements Using CNN-LSTM," 2021 5th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2021, pp. 1-6, doi: 10.1109/EICT54103.2021.9733510.

[27]. https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset

[28]. Sánchez-Caballero, A., Fuentes-Jiménez, D. & Losada-Gutiérrez, C. Real-time human action recognition using raw depth video-based recurrent neural networks. Multimed Tools Appl 82, 16213–16235 (2023).