Video Surveillance for Violence Detection Using Deep Learning

Manan Sharma and Rishabh Baghel

Abstract In order to detect violence through surveillance cameras, we provide a neural architecture which can sense violence and can be a measure to prevent any chaos. This architecture uses a pre-trained ResNet-50 model to extract features from the video frames and then feeds them further into a ConvLSTM block. We use a short-term difference of video frames to provide more robustness in order to get rid of occlusions and discrepancies. Convolutional neural networks allow us to get more concentrated spatio-temporal features in the frames, which aids the sequential nature of videos to be fed in LSTMs. The model incorporates a pre-trained convolutional neural network connected to convolutional LSTM layer. The model takes raw videos as an input, converts it into frames, and outputs a binary classification of violence or non-violence label. We have pre-processed the video frames using cropping, dark-edge removal, and other data augmentation techniques to make data get rid of unnecessary details. For evaluation of the performance of our proposed method, three standard public datasets were used, and accuracy as the metric evaluation is used.

Keywords Violence detection · Residual networks (ResNets) · Convolutional long short-term memory cells (ConvLSTM) · Deep learning

1 Introduction

Owing to increasing road rage and increasing violence in public places, it has now become a necessity to monitor and get noticed about these activities to avoid something more severe. Detecting actions in a video is actually quite harder than believed. One has to infer a situation's action from frames of data, and this is not just a question of image recognition but of action inference, which requires reasoning. The use of deep learning to solve such computer vision problems [13] is increasing. Cameras

M. Sharma (✉) · R. Baghel
Indian Institute of Information Technology Guwahati, Guwahati, India
e-mail: mananshrma858@gmail.com

R. Baghel
e-mail: baghelrishabha@gmail.com

S. Borah et al. (eds.), *Advances in Data Science and Management*,
Lecture Notes on Data Engineering and Communications Technologies 37,
https://doi.org/10.1007/978-981-15-0978-0_40

are now capable enough to replace a human reasoning power and even surpass it. Using these deep learning algorithms [2, 12] cuts the need for handcrafted features and provides an end-to-end model for successful completion of the task. Videos, of course, are sequences of images. Normally, LSTMs are used to process sequential data and convolutional neural networks to process image data, as in image classification problems shown in [8]. So, here we use a combination of convolutional neural networks and LSTMs for video classification.

Standard LSTM uses simple matrix multiplication to prioritize the input sequence and hidden previous state. In ConvLSTM, these operations are replaced by convolutions [10].

We have made a deep neural network with a ResNet50 block, a ConvLSTM block, and a fully connected block. We show how the change in action is more effective than the state of the action by feeding in the difference of the frames. We also show how videos as sequential data can be fed into recurrent networks (here ConvLSTMs), as mentioned in [14], and long-range dependencies of action can help in detecting what type of actions is being performed (here violence). To show the effectiveness of the model, we use three different datasets, namely KTH [6], Violent-Flows [7], and Hockey Dataset [6].

ConvLSTM cell is a type of LSTM cell which performs convolution operations inside the LSTM cell. The model is a special kind of RNN, capable of learning long-term dependencies. ConvLSTM replaces matrix multiplication with convolution operation at each gate in the LSTM cell. By doing so, it captures the underlying spatial features by convolution operations in multi-dimensional data, as shown in [1]. This combines the pattern recognition of ConvNets and the memory properties of pure LSTM networks. The architecture of ConvLSTM is expected to, therefore, find patterns in image sequences.

2 Network Architecture

The network contains a CNN block and a ConvLSTM block, For CNN block we have used a pre-trained ResNet50 architecture. The video frames are fed as a difference of two adjacent frames of the original video, so 20 frames are fed whose difference becomes 10 frames. These 10 frames are fed sequentially to the ResNet50. This ResNet50 is a pre-trained network on an ImageNet database [5]. The output 3-D feature maps are then fed into the ConvLSTM. The input to ConvLSTM is a 256 filters feature map, with filter size 3×3 and stride is 1. Each hidden state is of size 256 feature maps. Before feeding in the input frames, they are randomly cropped, flipped horizontally and vertically, normalized to make them centered around the mean, that is, mean zero and variance unity. The networks run for 6000 iterations. To make the final prediction, the output of LSTMs is batch normalized and fed into fully connected layers of size 1000, 256, 10, and 1, as the prediction is binary in nature. The nonlinearity used between fully connected layers is ReLu and the last

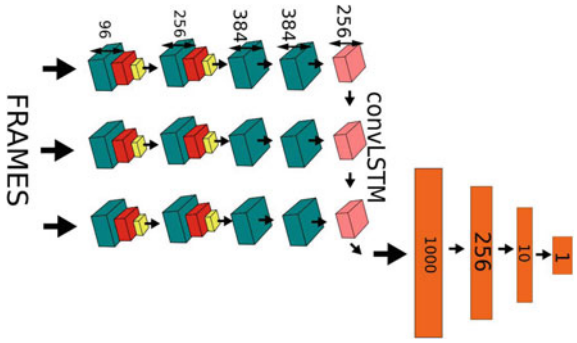

Fig. 1 Architecture of model

one neuron layer uses sigmoid with binary entropy as the loss function alongside RMSprop optimizer [3].

The reason for feeding in the difference of frames is to incorporate the changes in action rather than the actions themselves. The technique is an adaptation of optical low images for action recognition by Zisserman and Simonyan [15]. The changes in action are fed to ResNet50 which extract the features and feed them into ConvLSTM, which learns the dependency of changes on previous actions (Fig. 1).

3 Experiments

In order to determine how effective our proposed method is in classifying violence videos, three standard public datasets are used and hence classification accuracy is measured.

3.1 Experimental Settings

The whole network is implemented with Keras and TensorFlow as a back-end. The network is trained using gradient descent optimization algorithm, namely RMSprop. Since nearby frames contain overlapping information, there might be redundant computation involved while processing frames. In order to avoid these computations, frames which are extracted from each video are resized to dimension 256 $\times$ 256 during training. We evaluate different CNN architectures, and based on these architectures results are compared. Furthermore, we are using dynamic learning rate adjustments, which reduces the sequence length and one perceptron with sigmoid activation function is used. The whole network is trained on an NVIDIA GTX1080Ti GPU, and due to this reason we could fit two samples per batch, that is batch size 2, and the frames per sequence is 20. Since we do not know anything about data,

Table 1 Depicting different hyper-parameters

Parameters	Methods used
CNN architecture	ResNet50, InceptionV3, VGG19
Learning rate reducing	Dynamic
Cross entropy loss	Binary
Batch size	2
Sequence length	10 or 20
Evaluation	Simple split

so there might be some difficulty while assigning weights that would particularly work in that case. In order to overcome this problem, we can assign weights based on Gaussian distribution. So there is an algorithm called Xavier algorithm which we used for initializing weights as mentioned in [4].

In Table 1 we summarize the things that we made in our implementation.

3.2 Tuning of Hyper-Parameters

Owing to the limitation in resources, we used a simple split. For testing we have used 20% (20% of testing is also used for validation) and for training 80% of data is used. We are evaluating the different hyper-parameters of the network for the different datasets. We use only 20 epochs and early stopping of 5 as we apply in the final optimal network training. Each hyper-parameter has been evaluated separately and the best value is chosen for next evaluations. In Table 2 we have presented different hyper-parameters that are being evaluated in each iteration.

Table 2 Tuning of hyper-parameters

Parameters	Case1	Case2	Case3
Type of CNN architecture	ResNet50	InceptionV3	VGG19
Learning rate	le−4	le−3	le−2
Augmentation used	True	False	True
Number of frames	20	30	20
Dropout	0	0.5	0
Type of training	CNN retrain	CNN static	CNN retrain

3.3 Datasets

Based on our analyses, the most challenging datasets for violence detection in the literature are listed in Table 3. Different datasets represent different types of violence seen within city, street, and indoor environments.

3.4 Data Pre-processing

A fixed number of frames were sampled from the videos before given as an input to the model. For all dataset combinations augmentation methods were used and for some of the datasets, dark edges were removed from the frame as we present in Fig. 3. A subtraction of adjacent frames was provided as an input to the model; this helped in extracting changes in actions instead of a state of an action. In Fig. 2 an example is presented on difference computation of adjacent frames where a hockey player pushes another player.

Data augmentation is applied with the following transformations in order to enrich our dataset.

Image cropping: Edges of the images are removed before feeding into the network to make the pattern in the images more concrete, as shown in Fig. 4. Image transpose: As a complement steps to the cropping process, a transpose was done. This step was done during the t generator process, as shown in Fig. 5.

Table 3 Tuning of hyper-parameters

Dataset	Classes	Videos	Properties
KTH	Number of action classes = 6 and actions are walking, jogging, running, boxing, hand waving, and hand clapping	600 videos which includes black and white videos with resolution 160 × 120	Performed by 24 persons which includes 4 scenes and 6 actions
Hockey fight dataset	Actions happening in ice hockey sink	1000 videos (500 violence and 500 non-violence with resolution 720 × 576)	Non-crowded violence videos
Violent-Flows	Violent actions in crowded place	200 videos (100 violence and 100 non-violence) with resolution 320 × 240	Database of real world, footage of crowd violence

Fig. 2 Difference between frames

Fig. 3 Dark edges removal

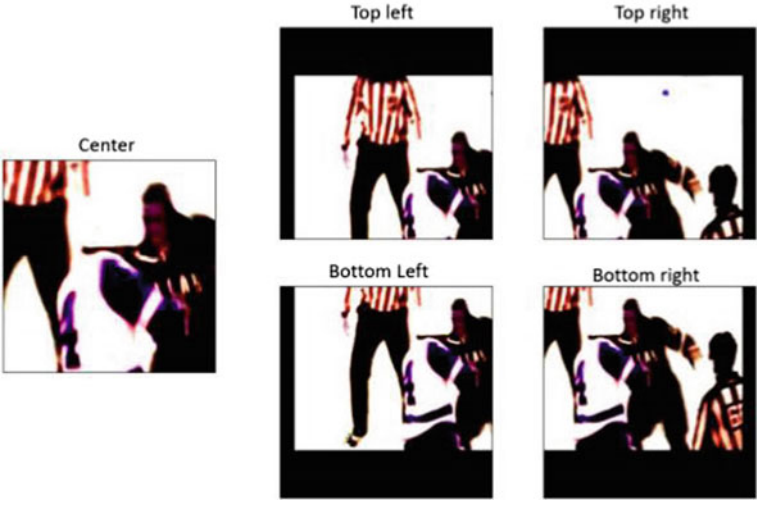

Fig. 4 Random cropping of the images

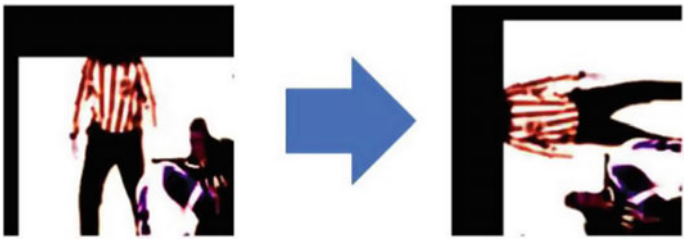

Fig. 5 Transpose of an image

4 Results

As already mentioned in Sect. 3.2 about the hyper-tuning process that allows us to find out the parameters which are performing best in the network. The test accuracy for each of the hyper-parameters values is shown in Fig. 6. Convolutional neural network (CNN) that gives good result among all the three is ResNet50 CNN with the accuracy of 89.9%; the InceptionV3 CNN almost give the same result like ResNet50 with an accuracy of 88.6% but the VGG19 CNN architecture does not perform well since it has 79.3% accuracy. It has been noted that due to augmentation the accuracy has been increased by 4.53% and also by making the length of sequence smaller, accuracy improved by 2%. As usually expected in case of static CNN configuration in which the CNN weights are not trained had very poor results of 58.9% accuracy.

The results which are presented in Figs. 7, 8 and 9 are basically depicting line charts. The accuracy of the train is depicted in blue; accuracy of the test is depicted in gray; validation is represented as yellow along with the training loss which is depicted in orange by a number of epochs. As mentioned earlier all experiments run till 50 epochs, where for all cases early stopping takes place. Figure 7 depicts the results for Hockey dataset where it is noted that learning rate was reduced to four times which is started at 1e−4 and ended at 5e−5 at final epoch. At epoch 34, due to early stopping, the training has stopped, reaching 87.7% for the test data in the last

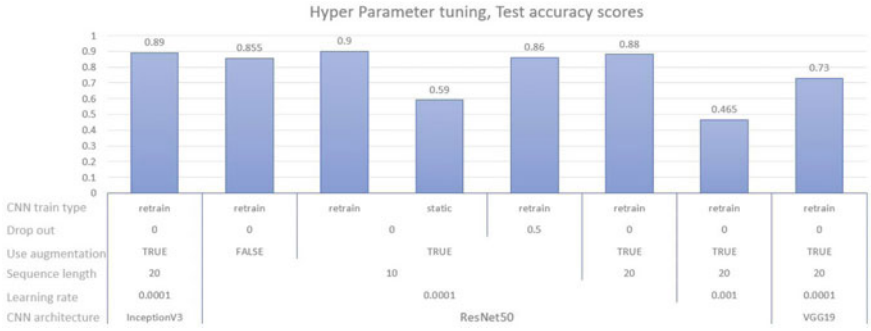

Fig. 6 Hyper-parameter tuning test accuracy scores

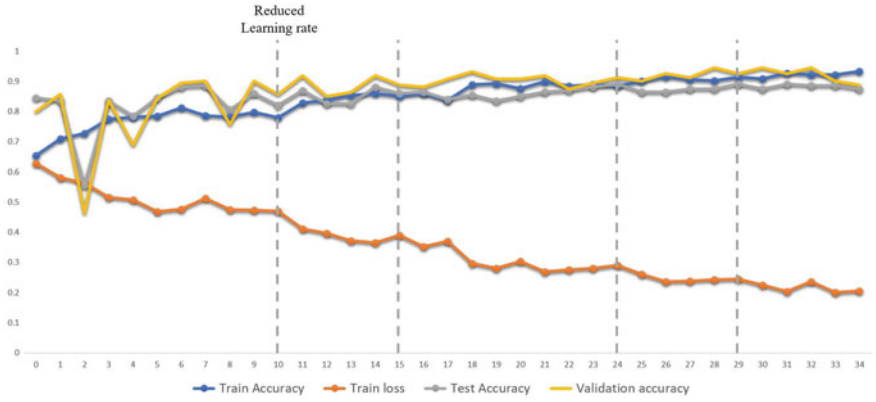

Fig. 7 Hockey dataset results

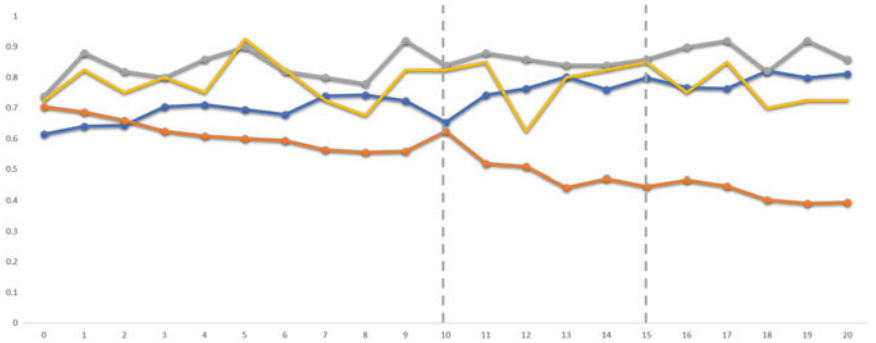

Fig. 8 Violent-Flows dataset results

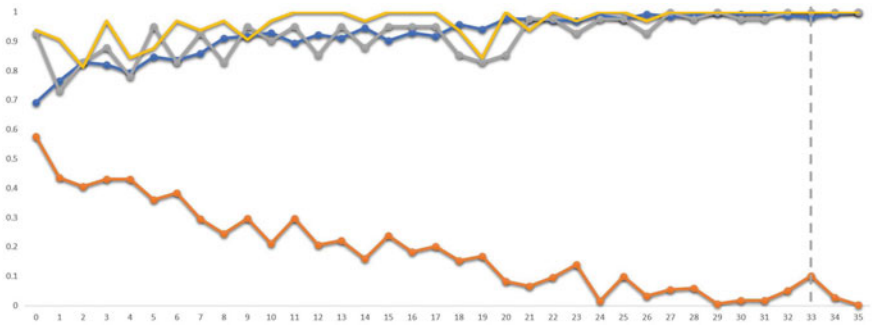

Fig. 9 KTH dataset results

epoch, and in overall epochs 89.3% is noted as best accuracy. Similarly, results for the Violent-Flows dataset are depicted in Fig. 8. Here also, it is noted that the learning rate has reduced twice, which is 1e−4 at starting epoch to a value of 2.5e−5 at the last epoch. Till the last epoch, test accuracy of the model is 86.5% and the overall accuracy is noted as 92.4% for overall epochs. The results for the KTH dataset are presented in Fig. 9. At epoch 33 the learning rate is reduced by one time and overall 100% accuracy is achieved in test, validation, and for training.

5 Discussion

For the evaluation of our architecture, we ran through several CNN architectures but we concentrated our result on three major CNN architectures, namely ResNet50, InceptionV3, and VGG19. Apart from all these we also checked out many hyper-parameter combinations. Among the three architectures, VGG19 gave the worst result. This can be explained by the fact that ResNet50 and InceptionV3 already gave better classification accuracies on ImageNet dataset. An interesting case is the better accuracy given by ResNet50 as compared to InceptionV3 in the violence detection architecture even though InceptionV3 was better in ImageNet classification [9]. InceptionV3 is 1.5% more classification accuracy on ImageNet dataset. The reason for this case can be the depth of the two architectures, where ResNet50 is deeper with 168 layers than InceptionV3 with 159 layers. Depth may be the reason for better identification of violence activities. Among the hyper-parameters, using dynamic learning rate gave better results. We got better results when we initialize the model using the starting learning rate as 0.0001 as compared to using 0.001. The higher learning rate gave a poor generalization. The reason for this can be that larger learning rates cause extreme changes in the learning process resulting in poor generalization. We started with the learning rate 0.0001, and slowly increased the learning rate which gave better accuracy. The reason for increasing the learning rate is faster learning of the optimal weights. Also, due to the domain-specific nature of the datasets, dropout did not help much.

Owing to lack of labeled datasets, certain data augmentation techniques are used to increase the number of samples and make the dataset more generalized. We took three different datasets which are KTH, Violent-Flows, and Hockey datasets. In the KTH dataset, the learning curve did not reduce until the last step. It was easier to converge compared to other datasets. It achieved 100% accuracy. In the Violent-Flows dataset, we could achieve only 80% accuracy. Owing to a large crowd [11] in videos which was not participating in violent activities, we divided the dataset into small pieces and used bagging method to make the final classification. In the Hockey dataset we achieved 87.5% accuracy, and this dataset encountered four reducing points in the learning curve.

References

1. F.D. De Souza, G.C. Chavez, E.A. do Valle Jr., A.D.A. Arajo, Violence detection in video using spatio-temporal features. in *Graphics, Patterns and Images (SIBGRAPI), 2010 23rd SIBGRAPI Conference on* (IEEE, 2010), pp. 224–230
2. P. Bilinski, F. Bremond, Human violence recognition and detection in surveillance videos. in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (IEEE, 2016), pp. 30–36
3. A. Datta, M. Shah, N.D.V. Lobo, Person-on-person violence detection in video data. in *Pattern Recognition, 2002. Proceedings. 16th Inter-national Conference on* vol. 1 (IEEE, 2002), pp. 433–438
4. J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venu-gopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 2625–2634
5. T. Giannakopoulos, A. Pikrakis, S. Theodoridis, A multi-class audio classification method with respect to violent content in movies using bayesian networks. in *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on* (IEEE, 2007), pp. 90–93
6. I.S. Gracia, O.D. Suarez, G.B. Garcia, T.K. Kim, Fast ght detection. PLoS ONE **10**(4), e0120448 (2015)
7. T. Hassner, Y. Itcher, O. Kliper-Gross, Violent flows: real-time detection of violent crowd behavior. in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on* (IEEE, 2012), pp. 1–6
8. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks. in *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105
10. J.R. Medel, A. Savakis, Anomaly detection in video using predictive convolutional long short-term memory networks (2016). arXiv preprint arXiv:1612.00390
11. S. Mohammadi, H. Kiani, A. Perina, V. Murino, Violence detection in crowded scenes using substantial derivative. in *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (IEEE, 2015), pp. 1–6
12. E.B. Nievas, O.D. Suarez, G.B. Garca, R. Sukthankar, Violence detection in video using computer vision techniques. in *International Conference on Computer Analysis of Images and Patterns* (Springer, Berlin, Heidelberg, 2011), pp. 332–339
13. P. Rota, N. Conci, N. Sebe, J.M. Rehg, Real-life violent social interaction detection. in *Image Processing (ICIP), 2015 IEEE International Conference on* (IEEE, 2015), pp. 3456–3460
14. I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks. in *Advances in Neural Information Processing Systems* (2014), pp. 3104–3112
15. K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos. in *Advances in Neural Information Processing Systems* (2014), pp. 568–576