Vietnam Journal of Computer Science
Vol. 7, No. 1 (2020) 1–22
© The Author(s)
DOI: 10.1142/S2196888820500013

World Scientific
www.worldscientific.com

Violence Detection by Pretrained Modules with Different Deep Learning Approaches

Shakil Ahmed Sumon*, Raihan Goni†, Niyaz Bin Hashem‡,
Tanzil Shahria§ and Rashedur M. Rahman¶

Department of Electrical and Computer Engineering
North South University
Dhaka 1229, Bangladesh
*shakil.sumon@northsouth.edu
†raihan.goni@northsouth.edu
‡niyaz.hashem@northsouth.edu
§tanzil.shahria@northsouth.edu
¶rashedur.rahman@northsouth.edu

Received 23 May 2019
Accepted 13 September 2019
Published 25 October 2019

In this paper, we have explored different strategies to find out the saliency of the features from different pretrained models in detecting violence in videos. A dataset has been created which consists of violent and non-violent videos of different settings. Three ImageNet models; VGG16, VGG19, ResNet50 are being used to extract features from the frames of the videos. In one of the experiments, the extracted features have been feed into a fully connected network which detects violence in frame level. Moreover, in another experiment, we have fed the extracted features of 30 frames to a long short-term memory (LSTM) network at a time. Furthermore, we have applied attention to the features extracted from the frames through spatial transformer network which also enables transformations like rotation, translation and scale. Along with these models, we have designed a custom convolutional neural network (CNN) as a feature extractor and a pretrained model which is initially trained on a movie violence dataset. In the end, the features extracted from the ResNet50 pretrained model proved to be more salient towards detecting violence. These ResNet50 features, in combination with LSTM provide an accuracy of 97.06% which is better than the other models we have experimented with.

Keywords: Long short-term memory; convolutional neural network; spatial transformer network; transfer learning.

1. Introduction

In different regions of the world, including Bangladesh, violence is taking over humanity. The political and economic structures of these regions have failed to stop

¶Corresponding author.

violence effectively rather in some places, they demand and encourage violence. Violence is spreading all over the world in the name of religion, race and nationality and there is hardly any tool available to prevent it. However, recent advanced technology has taken some efforts to detect violence from live video feed.

The violence we are talking about is of large scale like fights in a football stadium between two fan groups or riots in the streets between supporters of political parties. Moreover, irrespective of developing and developed countries, violence on minority groups are on the rise as well. There are horrible consequences of these kinds of violence including people being died and properties being destroyed. However, the loss of lives and properties can be minimized if violence can be detected right when it happens.

Violence in other parts of the world involves heavy weapons like machine guns, grenades, etc. whereas violence in Bangladesh is quite distinct as riots between different communal and religious groups dominates here. This violence hardly incriminates any deadly weapons rather most of the time are fist fights or incorporates bamboo sticks and heavy metal sticks. According to universal approximation theory, in order to detect these violent crowd flows, the dataset needs to be of similar distribution.[1] Hence, we felt the urge of creating a dataset on Bangladesh context. In our previous work,[2] we have explored different deep learning techniques to detect violence in videos which were collected from YouTube. The videos which were collected were mainly of Bangladesh context containing political precessions, violent protests, cricket match galleries, etc. However, we have applied CNN, LSTM and CONVLSTM in our collected dataset and have found some exciting insights. Additionally, we have leveraged transfer learning by retraining a model with our collected videos which was initially trained on a movie violence dataset.[3] We have experienced that the transfer learning model has acted better on this dataset than the other models applied.

This study dives deep into transfer learning and explores the potentials of extracting features from pretrained models like VGG-16, VGG-19 and ResNet-50. Additionally, in one of the experiments of this study, we have leveraged the remembering and forgetting nature of the LSTM network by feeding 30 frames of a video at a time.

Moreover, in some of our experiments, we have applied attention on the features extracted by the pretrained models from the videos. We have experimented with a special kind of attention mechanism, spatial transformer network; which does not need the assistance of sequence models to implement its encoder-decoder architecture.[4] The spatial transformer is invariance to translations and rotations and attends the frames spatially.[5]

This study extends our previous work[1] in following areas:

(1) Instead of traditional CNNs, it extracts features from frames by pretrained network like VGG-16, VGG-19 and ResNet-50.

(2) It combines transfer learning and LSTM network by giving the extracted features of a particular timestamp from a video to a LSTM network to classify whether it contains violence or not.
(3) It applies attention to the features extracted by the pretrained models from the frames of the videos spatially.

2. Related Work

In the detection of violent videos, the importance of temporal information is huge but 2D CNN provides only the spatial information. Thus, researchers have introduced a 3D CNN model instead of a 2D CNN.[6] This 3D CNN model consists of nine layers and it has been trained using the hockey dataset.[7] Since only one node has been used as output, it gives one of two values: true or false. The activation and cost function used here are sigmoid and stochastic gradient descent, respectively. This architecture has achieved an accuracy of 91%.

A different approach has been taken by some researchers in regards to violence prediction.[8] A new model has been built which took four different types of features which includes audio features, attribute features, trajectory-based motion features and spatial-temporal interest points (STIP). STIP algorithm has been used to locate the interest points in both temporal and spatial dimensions. Audio from each of the videos has been taken as one of the features. Classification is done using the support vector machine (SVM). A maximum of 68.2% accuracy has been accomplished from this model.

A new architectural model has been proposed in Ref. 9 which uses convolutional long short-term Memory (convLSTM). This model has been constructed using a series of layers (convolutional and pooling) to extract features. A total of 256 filters has been used with Rectified Linear Unit (ReLU) being the activation function. Difference between adjoining frames in the input layer has been taken to identify changes in videos. The model has been trained and tested using three different open datasets namely movie dataset, hockey flight dataset and violence-flow dataset.[3,7] Several pre-processing techniques has been applied on the datasets before training. Using the same data, an accuracy of 94.6% was achieved by LSTM while an appreciably higher accuracy of 97.1% has been accomplished using convLSTM.

The work in Ref. 10 based their approach on a completely new algorithm known as the Motion Weber Local Descriptor (MoWLD). This is capable of finding both temporal and spatial information from the interest points of the videos. The researchers have rebuilt the WLD histograms and oriented them by collecting the WLD histograms from the adjacent regions. Multi-scale optical flow has been used to adopt WLD features while reduction of data dimension has been obtained using the Kernel Density Estimation (KDE). Furthermore, max-pooling technique has been applied in order to get more compact features representation. The model has been trained and tested using three different sources of datasets. The BEHAVE dataset has obtained the highest accuracy rate of 94.9% followed by the hockey fight dataset with an appreciable accuracy of 91.9% and finally the crowd violence dataset with a satisfactory accuracy rate of 89.78%.

To extract information from raw video, a model was presented in Ref. 11, with the idea of multi-stream. They used three streams and those are acceleration, temporal and spatial streams. They calculated velocity of movement in the acceleration stream. 2D CNN is used in the spatial stream to collect information about relation between image and violence. They also used few pretrained models for temporal streams and sent the output to LSTM.

Some authors focused on the real-time violent detection problems, worked on that and obtained some fascinating results. In Ref. 12, surveillance video cameras are used to get real-time unique data. Depending on the changes of magnitudes of flow vector, they made some statistic by using descriptor named violent flow. Then, they collected the statistics from their dataset for short frames. By using this technique, they got accuracy of 82.9%.

Other researchers of same fields tried to build a model in a different way. To classify important context, like violence, they used CENTRIST-based features.[13] The whole process starts with preprocessing followed by feature extraction. After that, for classification, they normalized the data and then applied feature reduction. They used two different datasets and those are violent flow dataset and Hockey Fights Dataset. From the first dataset, they obtained an accuracy of 91.46% and from the second dataset they obtained an accuracy of 92.79%.

A group of researchers tried different approaches for this kind of problem. Some concepts of fluid mechanics are used in Ref. 14. Authors analyzed and calculate the rate of change of fluid property of a particular video. Two histograms from two optic flows is created and for the final descriptor those histograms have been concatenated. To train and test the model, four different datasets is used. By using this model, accuracy gain is 95%.

A new technique to detect violence is introduced in Ref. 15. The model used local optical flow technique and spatio-temporal features. For designing the model, they combined Harris 3D spatio-temporal interest point detector and the optical flow technique. But they obtained a very fluctuated result and their finest accuracy was 69.43%.

A model to improve the security systems by detecting robbery is introduced in Ref. 16. At first, motion region is being classified. Then, they evaluate the optical flow and the flow of energy since energy flow tends to be high in motion area. Their dataset was very small. In that dataset, there was only 13 videos from which then used 8 videos to train the model. By using the whole dataset, the obtained an accuracy of 76.9%.

Here in Ref. 17, authors tried to avoid existing methods which are vision based because they cannot adapt new dataset so easily. They proposed a series of work starts with framing each video into RGB images followed by computing optical flow fields and obtaining acceleration fields. Then they trained the FightNet with different kind of inputs. They also made a huge dataset containing 2,314 videos of two categories. They obtained a great accuracy also which is around 97%.

On the other hand, some researchers started to work on aggressive behavior detection which is not well studied, as, only action recognition usually focused on detecting simple actions.[18] Basically, they used a well-known framework, which is Bag-of-Words and two other action descriptors, STIP and MoSIFT in their model. They also made big dataset containing 1,000 videos of two categories, fights and non-fights. Applying the model, they obtained an accuracy near 90%.

Existing models and frameworks are mainly focused on detection of different kind of actions. But, in this paper, researchers focused on localizing the violence also. They proposed a Gaussian Model which will extract the violence regions.[19] After that, by performing the densely sampling violence is detected. They obtained different types of results on different dataset but the best accuracy they obtained were 86.59%.

Surveillance cameras are very popular nowadays. Keeping that in mind, in Ref. 20, researchers tried to make an improved violence detector by following two steps. Firstly, they designed a feature extraction method which is focused on motion magnitude changes. Then, they tried to adopt features and multi-classifier combination. They applied their models on two public datasets and obtained a great result. They also figure out that, a great accuracy can be obtained by applying ViF and OViF together which is 94.84%.

Nitish *et al.* have presented a thorough analysis on unsupervised video representation learning using LSTM autoencoder.[21] The initial model is a very straight forward LSTM Encoder-Decoder model. The first LSTM layer learns video representation while the second LSTM layer produces exactly the same representation in the output layer but in the reversed order. The latter LSTM autoencoder model does the same thing in addition to predicting the future frames (13 frames). The decoder is conditioned based on whether it should generate frames from the last frames of videos or it should not generate any frames at all. Finally, both of the previous models are combined to create a composite model which reconstructs the input as well as predicts the future frames. Their predictive model acquires an accuracy rate of 75.8% on UFC-101 dataset. After combining the flow model with the predictions from RGB, accuracy raises to 84.3%.

In Ref. 22, several features were extracted. CNN-based features are captured by training AlexNet (on ImageNet). Two-stream CNN features namely spatial stream and temporal stream are collected. Spatial stream is obtained from CNN model trained on ImageNet while for the temporal stream, a CNN model is trained on stacked optical flows. Output of the FC is considered as temporal features. Two-stream CNN features have been used as input of LSTM, and averaged output of each time-steps is considered as features. Finally, some local features such as HOG, HOF, motion boundary histograms (MBH), and trajectory shape (TrajShape) is also used as features. All of the features have been run on the SVM classifier. Violence learning accuracy for CNN-violence features is 27%, conventional features is 16.5%, two-stream CNN is 29.5%, and all features combined resulted in an accuracy rate of 29.6%.

LSTM is used for video classification in Ref. 23. Initially, different types of features are collected using Bag of Word (BoW) and SIFT-based approaches. These feature descriptors are then passed to the LSTM-RNN model to take best benefits of temporal evolution for classification. The model accomplishes an appreciable accuracy of 92% on the soccer dataset.

3. Dataset

The dataset we have used in this paper is a modified and extended version of our previous work. The goal of the work is to detect violence on Bangladeshi crowd, that is why the dataset is of Bangladesh context. The dataset contains videos of street fighting, political riots, fights in football field which we labeled as violent videos. It also contains videos of peaceful political precessions, celebration in the gallery after winning cricket matches, street marching for environmental causes which we labeled as non-violent.

This dataset had been collected from different video sharing website like YouTube and social networking platforms like Facebook and Twitter. As the platforms are different, the videos are of varying resolutions and lengths. 110 videos for each class had been collected that gives us 220 videos in total. We have edited the collected videos as in the non-violent parts of the violent videos are been cut off and vice versa.

We have split the collected dataset into training and testing part. The training and testing part have 90 and 20 videos per class, respectively.

In our knowledge, this is the first of this kind of dataset in Bangladesh context. The dataset has been developed with a view to inspiring research in similar field hence the dataset will be made available for researchers after publishing this paper.

Additionally, a dataset has been used to pretrain the model.[3] The dataset contains violent and non-violent scenes from different Hollywood movies.

4. Methodology

As the videos are of different resolutions, we have resized them to be 28 × 28 pixels for our CNN architecture. We have down-sampled the frame to make the training faster. A total of 30 frames have been extracted per second from the videos.

Convolutional neural network (CNN) has a layered architecture. Typically, it has an input layer, couple of convolution and pooling layers in combination, followed by an output layer. The CNN acts as a feature extractor. The features are then being passed into couple of fully connected layers which in combination act as a classifier. However, our proposed CNN architecture has two convolution layers, three densely connected layers which are followed by a sigmoid layer. The convolution layers have 32 filters and the dense layers has 10 nodes on each of them. The filters which are used in the convolution layers has the size of 3×3. A batch normalization layer has been introduced after each of the convolution and fully connected layers. Rectified linear unit (ReLU) has been used as activation function in all the convolution and

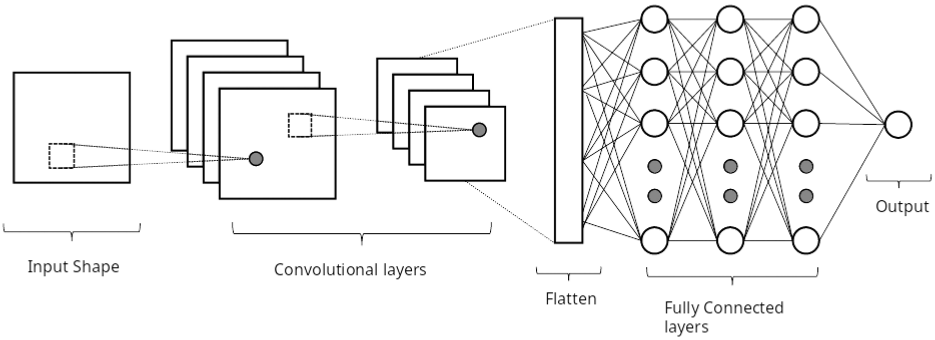


Fig. 1. A high-level overview of CNN.

fully connected layers. As violent crowd flow detection is a binary classification problem, we have used sigmoid as the activation function of the output layer. Figure 1 shows a high-level view of the CNN architecture we have used.

The CNN model prevents overfitting by applying regularization techniques like kernel regularizers and dropout of 0.5. The cost function of this model is binary cross entropy and the optimizer is "adam".

Moreover, a pretrained model which we have trained initially with a violent movie dataset has been retrained on our collected videos. The model which has been used for pretraining has similar architectures like the CNN model. The model has learnt features from the movie dataset which then proved to be helpful in classifying violent and non-violent videos of Bangladesh context.

However, we have extracted features from our videos using three pretrained ImageNet models, VGG16, VGG19 and ResNet50.[24,25] ImageNet is a dataset containing roughly 15 million high-resolution human-labeled images belonging to approximately 22,000 classes. An annual competition was launched based on a portion of this dataset in 2010. The three above-mentioned models had done exceptionally well in the competition in different years. Fortunately, the creators of the architectures have made their pretrained models publicly available.

VGG16 is a simple deep CNN architecture. It has a deep stacked layered CNN followed by two fully connected layers which have 4,096 neurons on each of them. The output layer has a softmax classifier. VGG19 has similar architectures except it is deeper than VGG16. ResNet50 is a deeper model than VGG19 but it proposes a very interesting architecture consisting residual modules. Interestingly, despite of being deeper than the VGG models, the model size of ResNet50 is much smaller than them because it uses global average pooling rather than fully connected layers.

These pretrained models demand that the images feed to them are the size of 224 × 224 pixels. That is why, we have resized our frames before feeding them to the models for extracting features. These features are then being feed into three fully connected layers for classification which are being followed by a sigmoid output layer. The fully connected layers have 64 neurons on each of them. Figure 2 shows the

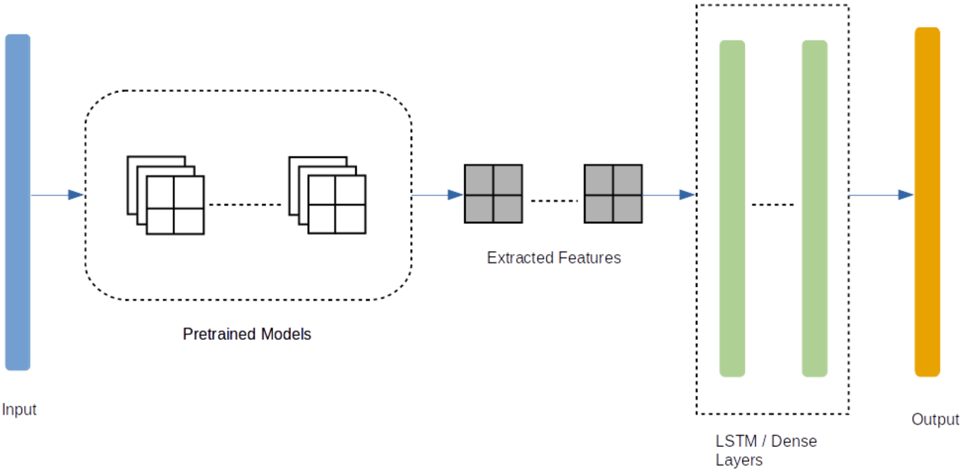


Fig. 2. A workflow of the pretrained feature extraction and feeding to dense layers.

workflow of our proposed networks. The fully connected layers incorporate ReLu as their activation function. The optimizer for the models is adam and the loss is binary cross entropy.

Moreover, we have designed three other models where the extracted features are fed, instead of into fully connected layers, to an LSTM network as sequences. Here, we redefine the video classification as a sequence problem. We have extracted 30 frames per second from the videos. Features have been extracted from these 30 frames by the pretrained models and then these frames are being fed into the LSTM layer at a time. The model has been designed with one LSTM layer with 50 LSTM units. The LSTM layer traditionally uses the hyperbolic tangent function as the activation. The last layer is the sigmoid classifier. The optimizer and the loss function used in this model is adam and binary cross entropy respectively. Figure 3 gives a high-level view of an LSTM network.

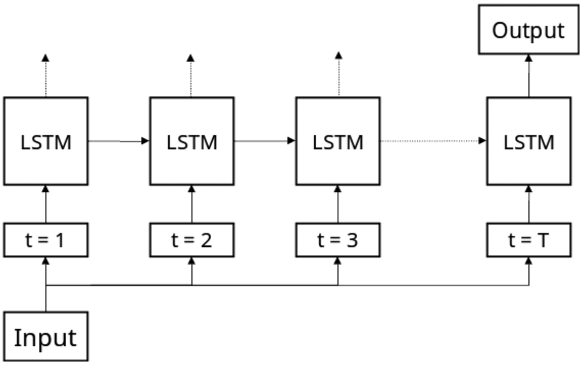


Fig. 3. A high-level overview of LSTM network.

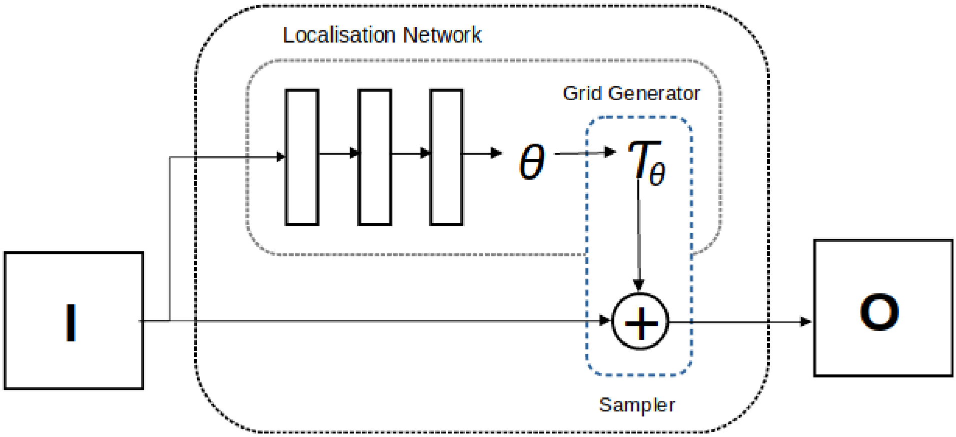


Fig. 4. A high-level overview of spatial transformer network.

However, attention mechanism is proving to be very fruitful natural language processing tasks such as neural machine translation.[26] When we enable attention in a model, we give the model ability to look at a specific portion of the input which is very crucial in producing the desired output. Transformer network is an approach of applying attention without the explicit need of implementing encoder-decoder architecture with sequence models like RNN, LSTM, GRU, etc. Transformer network, instead implements encoder-decoder architecture with stacked self-attention and point-wise fully connected layers.

However, spatial transformer network not just applies attention, it transforms the attended region of the inputs in such ways that simplifies inference in later layers.[5]

As shown in Fig. 4, the spatial transformer network consists of three main parts, the localization network, the grid generator and the sampler. The localization network takes an input feature map and gives a transformation conditioned on that particular feature map. The grid generator gives a set of points in the feature map where it should be sampled to construct the predicted transformation. The sampler takes the feature map and the sampling grid as inputs and generates the output map sampled from the feature map at the grid points.

We integrate the spatial transformer network with a CNN module. The CNN module is capable of having multiple spatial transformation. The transformations applied here are translation, scale and rotation.

5. Experimental Result and Evaluation

We have used Keras deep learning library with TensorFlow backend to implement our desired models. The training has been done on a NVIDIA 1060 GPU.

We have trained all the models with a common training set and tested them all with the same 10 videos. We let the model run for couple of epochs and reported their accuracy metrics for training and testing phase in Table 1.

Table 1. CNN model performance with respect to accuracy.

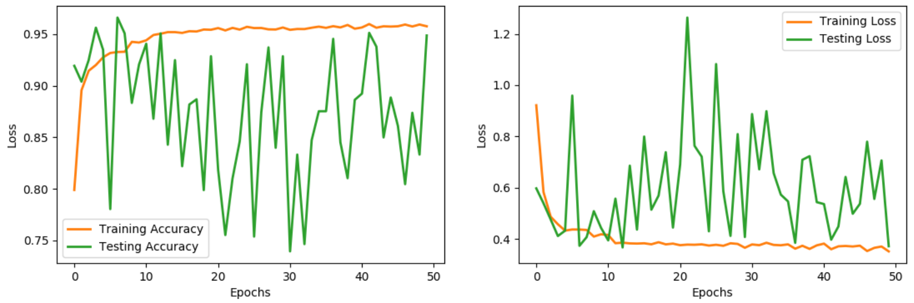
		Accuracy	
		Training	Testing
Epochs	10	94.25	92.90
	20	94.23	93.93
	50	95.57	94.86
	100	95.72	94.47
	200	94.23	91.53
	500	94.59	92.27

We can infer from this table that the accuracy of CNN model is not converging to any extent after 50 epochs. As a result, we have taken 50 epochs as the standard for the CNN-based model. However, Figs. 5(a) and 5(b) show the epochs vs. accuracy and epochs vs. loss of training and testing phase of the CNN model, respectively. The difference between the training and testing accuracy tells us that the model overfits to some extent. The fluctuating testing accuracy and testing loss are signs that the model is not generalizing well in the given task. However, increasing the training dataset might solve the problem of not generalizing. We will test this hypothesis in our future work.

Additionally, we have trained a model with a movie violence dataset and have used its learned features in our task at hand. We have taken two approaches while using the model trained on the violent movie scenes dataset:

(1) We have frozen all the layers of the pretrained model i.e. the model has not been retrained on our collected dataset. We have judged its performance entirely on previously learned features.
(2) The layers are not been frozen i.e. the model is retrained on our collected violence dataset.

Table 2 shows the accuracy metrics of both approaches of the pretrained model.



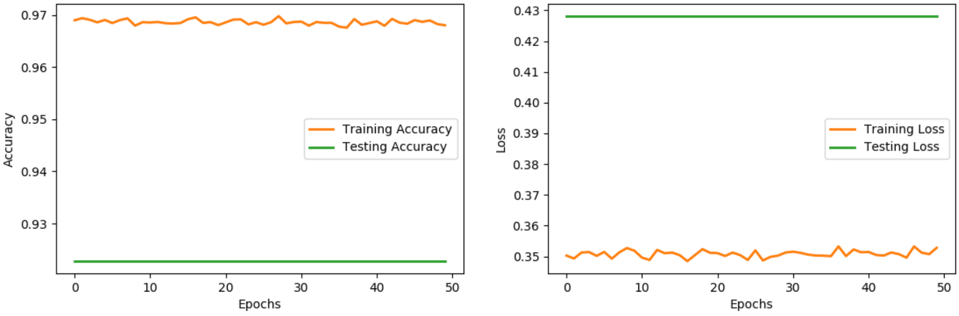
(a)

(b)

Fig. 5. CNN models': (a) Accuracy graph (b) Loss graph.

Table 2. Transfer learning models' accuracy.

	Training	Testing
Freezing the layers	96.85	92.27
Without freezing the layers	95.67	95.70

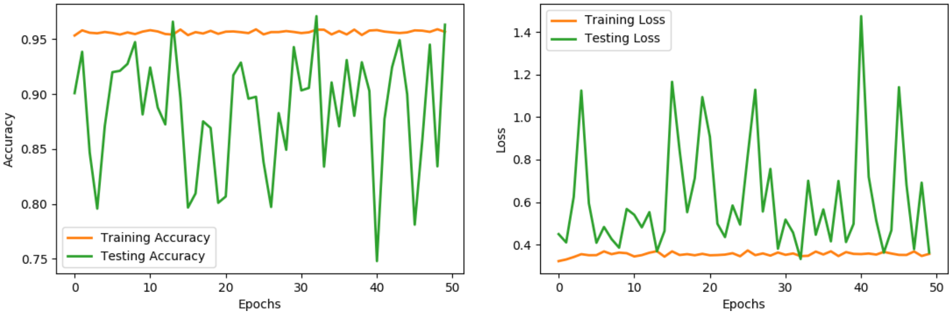


(a) (b)

Fig. 6. Graph of the pretrained model without retraining (a) Epochs vs. Accuracy graph (b) Epochs vs. Loss.

We can see from Fig. 6 that the model which was not restrained have given us almost flat graphs of accuracy and loss. This has happened because the model has made predictions on the basis of a fixed set of learned features. On the other hand, after retraining the model with our custom dataset, the fluctuations in testing accuracy and loss have re-emerged which has been shown in Fig. 7. The newly learned features have introduced some sort of chaos which prevents generalization on our dataset.

Moreover, we have extracted features from the frames of the videos using three famous ImageNet models, VGG16, VGG19 and ResNet50. The extracted features

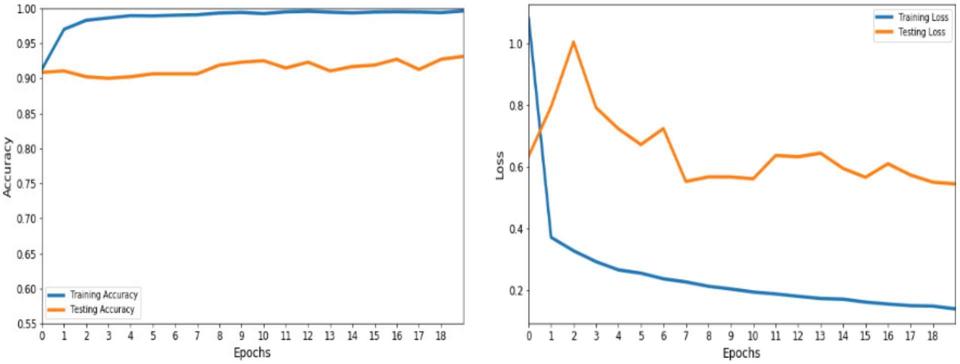


(a) (b)

Fig. 7. Graph of the pretrained model with retraining: (a) Epochs vs. Accuracy graph (b) Epochs vs. Loss.

Table 3. Accuracies of VGG16, VGG19 and ResNet50.

	Training	Testing
VGG16	99.60	92.34
VGG19	99.33	93.50
ResNet50	99.79	97.06



(a)

(b)

Fig. 8. Graph of the VGG16 + FCN model: (a) Epochs vs. Accuracy (b) Epochs vs. Loss.

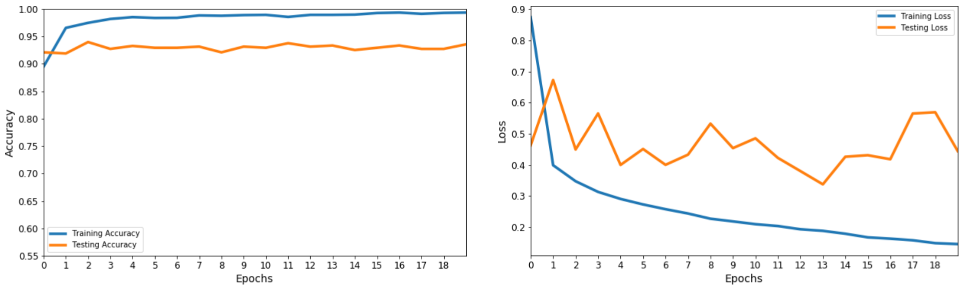
are being fed into a fully connected layer. Three different models have been trained on the features extracted by three different pretrained models. Table 3 reports the accuracy of the models.

In these cases, training epochs have been set to 20 as we have found empirically that 20 is the optimum number of epochs for these models. However, Fig. 8 shows us that the VGG16 + FCN (fully connected network) model is a classic case of overfitting. The model has not been generalized well enough to show consistency in the testing phase.

From Fig. 9, we can infer that VGG19 + FCN model is also guilty of overfitting but the testing accuracy metric of this model seems consistent to some extent over time.

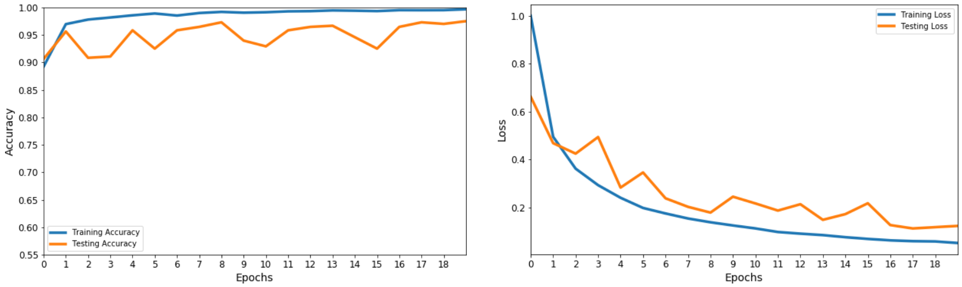
Figure 10 shows the epochs vs. accuracy and epochs vs. loss graph, respectively of the ResNet50 + FCN model. This is not different than the other two except the gap between training and testing accuracy is somewhat reduced. The ResNet50 + FCN model overfits less and the minimization of the loss over time seems consistent and promising than the other two models.

Additionally, to leverage the internal memory structure of LSTM network, instead of feeding the features extracted from the pretrained models to a fully connected network, this time we have to feed them into an LSTM network. Table 5 reports the accuracy of the models.



(a) (b)

Fig. 9. Graph of the VGG19 + FCN model: (a) Epochs vs. Accuracy (b) Epochs vs. Loss.



(a) (b)

Fig. 10. Graph of the ResNet50 + FCN model: (a) Epochs vs. Accuracy and (b) Epochs vs. Loss.

We can see from Table 4 that unlike the pretrained network + FCN models, accuracies are almost similar for these LSTM-based models. Although ResNet50 + LSTM is on top of the leaderboard, the difference with other two are not quite significant.

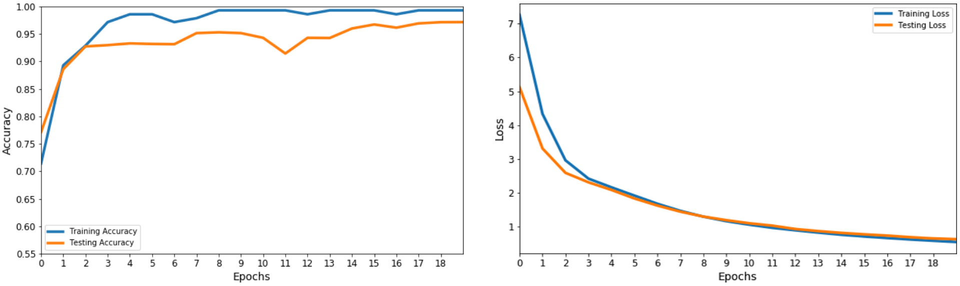
Here, by applying LSTM to the features extracted by VGG models, we have enabled an architecture which can remember the information of previous frames. So, now the model can take more informed decision about each frame and ultimately, about the video. The ResNet50 architecture, on the other hand, does take into account the features from the previous layers. The detailed discussion is provided later in this section.

Table 4. Accuracies of VGG16, VGG19 and ResNet50 with LSTM.

	Training	Testing
VGG16 + LSTM	99.29	95.52
VGG19 + LSTM	99.98	96.88
ResNet50 + LSTM	99.29	97.06

Table 5. Accuracies of different pretrained models with spatial transformer model.

	Training	Testing
VGG16 + spatial transformer	99.85	93.25
VGG19 + spatial transformer	99.43	89.50
ResNet50 + spatial transformer	99.77	94.86

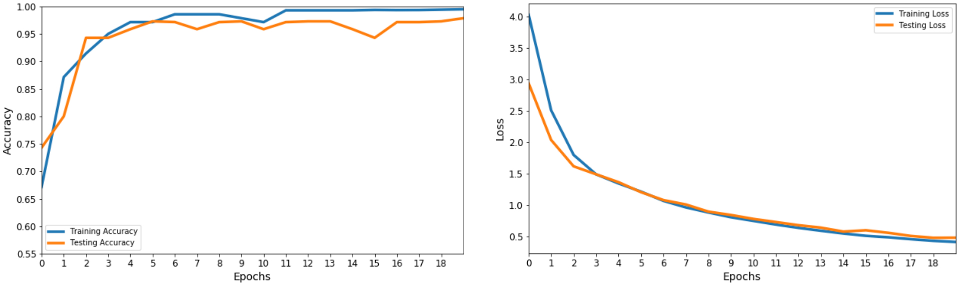


(a) (b)

Fig. 11. Graph of the VGG16 + LSTM model: (a) Epochs vs. Accuracy (b) Epochs vs. Loss.

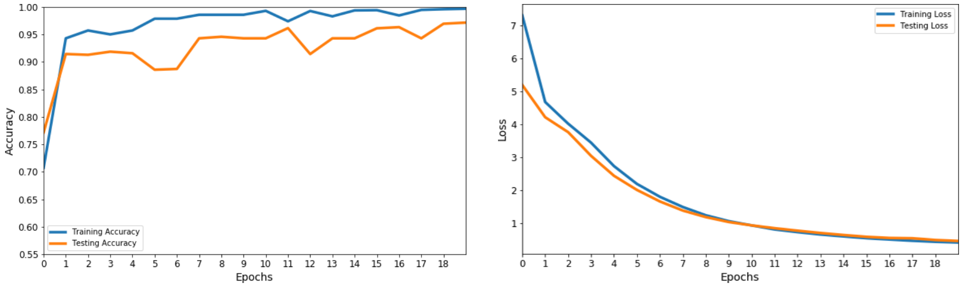
However, in Fig. 11(a), we can see that the accuracy of the VGG16 + LSTM model is still fluctuating, and overfitting still persists. The loss in Fig. 11(b), nevertheless, is decreasing monotonically over the time period. Figure 12 shows that the training and testing accuracy and loss of VGG19 + LSTM model is quite consistent and less fluctuating. On the other hand, the accuracy of ResNet50 which is shown in Fig. 13(a) in contrast to the other two LSTM models fluctuates a lot. The loss is decreasing and compatible with the other two models.

The performance of ResNet50-based model is higher as usual but surprisingly, VGG19 along with spatial transformer have failed to deliver. Attention applied to



(a) (b)

Fig. 12. Graph of the VGG19 + LSTM model: (a) Epochs vs. Accuracy (b) Epochs vs. Loss.

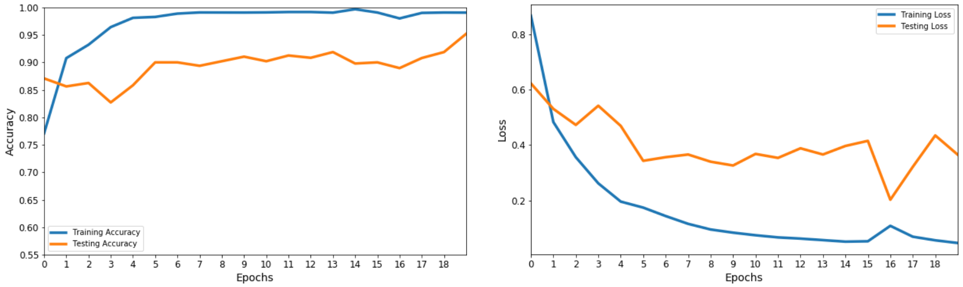


(a) (b)

Fig. 13. Graph of the ResNet50 + LSTM model: (a) Epochs vs. Accuracy (b) Epochs vs. Loss.

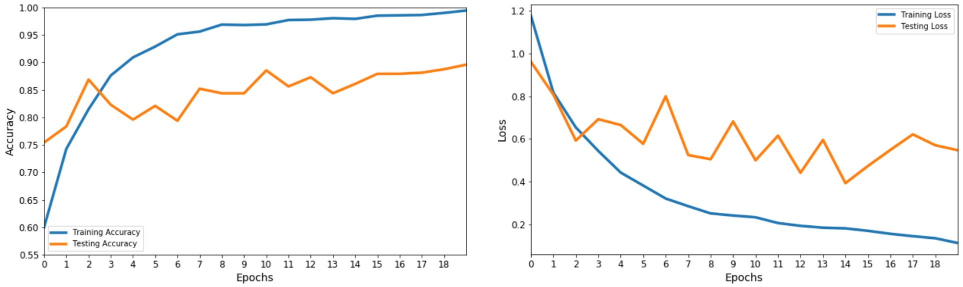
the features extracted from VGG16 has a moderate performance in light with the other models.

However, from the epochs vs. accuracy and epochs vs. loss graphs of the spatial transformer-based models which are been shown in Figs. 14, 15, 16 and 17, we have seen that these models are very prone to overfitting. The testing accuracy of all the



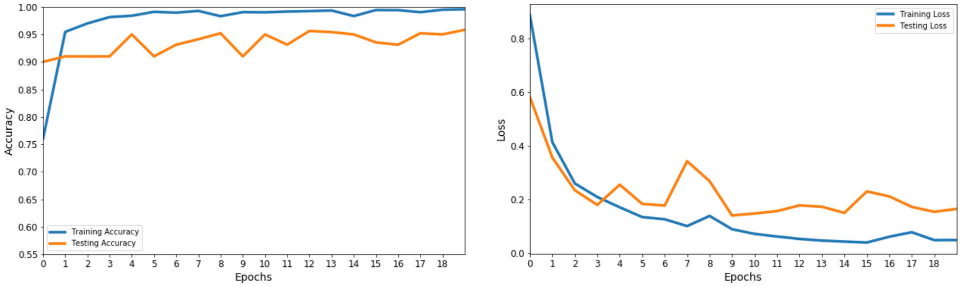
(a) (b)

Fig. 14. Graph of the VGG16 + spatial transformer model: (a) Epochs vs. Accuracy (b) Epochs vs. Loss.



(a) (b)

Fig. 15. Graph of the VGG19 + spatial transformer model: (a) Epochs vs. Accuracy (b) Epochs vs. Loss.



(a) (b)

Fig. 16. Graph of the ResNet50 + spatial transformer model: (a) Epochs vs. Accuracy (b) Epochs vs. Loss.

models are fluctuating and the loss, except for the ResNet50 associated model, are not decreasing consistently.

Nevertheless, after evaluating the accuracies and graph of all the experimented models, we can say that, the features extracted from the ResNet50 pretrained model seem to be more salient. ResNet50 has a unique architecture which contributes to its usefulness.

ResNet50 introduces deep residual connections which actually means "short-cut" connections in between layers.[27] As shown in Fig. 17, the residual network, instead of having a fixed underlying mapping, is designed to have a residual mapping. This tweak in the traditional neural network architecture enables the residual network to not just have information from its previous layer but from its ancestors as well.

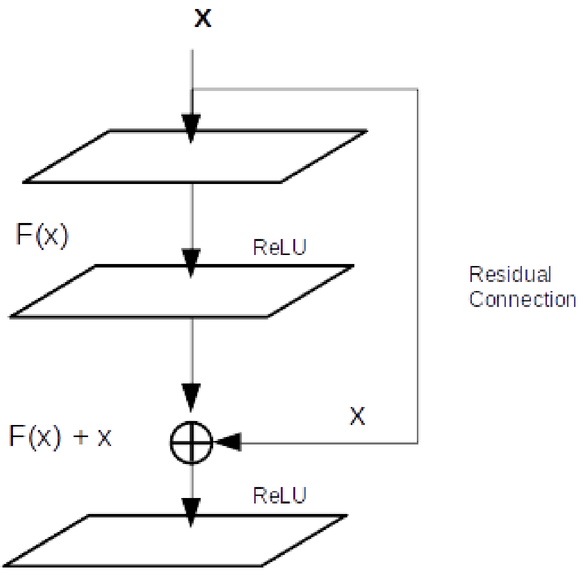


Fig. 17. A block diagram of a residual network.

Table 6. Classification report of the non-violent class.

	Precision	Recall	F1-score
CNN	0.96	0.94	0.95
Transfer with freezing	0.89	0.98	0.93
Transfer without freezing	0.95	0.99	0.97
VGG16 + FCN	0.91	0.96	0.93
VGG19 + FCN	0.90	0.97	0.93
ResNet50 + FCN	0.98	0.94	0.97
VGG16 + LSTM	0.87	1.00	0.93
VGG19 + LSTM	0.93	1.00	0.97
ResNet50 + LSTM	1.00	0.94	0.97
VGG16 + attention	0.85	0.96	0.90
VGG19 + attention	0.86	0.93	0.89
ResNet50 + attention	0.91	1.00	0.95

Moreover, this network is subsequently deeper, so they can extract more salient features from a given image. These are the reasons why ResNet50-based models have done better on this dataset than the other models.

Furthermore, the classification report of the non-violent and violent class has been reported in Tables 6 and 7. The overall classification report has been reported in Table 8. The classification reports indicate that the ResNet50, VGG-based models and our custom pretrained model with all of its layers unfreeze did better in detecting non-violent videos. On the other hand, ResNet50 + FCN and ResNet50 + LSTM models outperform all other in the task of detecting violent videos. The more noticeable features extracted by the ResNet50 pretrained models helped the fully connected layers and LSTM network to predict better.

After evaluating the accuracies and graph of all the experimented models, we can say that the features extracted from the ResNet50 pretrained model seem to be more salient. ResNet50 has a unique architecture which contributes to its such usefulness. ResNet50 introduces deep residual connections which actually means "short-cut"

Table 7. Classification report of the violent class.

	Precision	Recall	F1-score
CNN	0.93	0.95	0.94
Transfer with freezing	0.97	0.86	0.91
Transfer without freezing	0.97	0.92	0.94
VGG16 + FCN	0.95	0.87	0.90
VGG19 + FCN	0.94	0.89	0.91
ResNet50 + FCN	0.94	0.98	0.98
VGG16 + LSTM	1.00	0.91	0.93
VGG19 + LSTM	1.00	0.95	0.98
ResNet50 + LSTM	0.95	1.00	0.98
VGG16 + attention	0.96	0.87	0.90
VGG19 + attention	0.93	0.87	0.89
ResNet50 + attention	1.00	0.92	0.96

Table 8. Overall classification report of the models.

	Precision	Recall	F1-score
CNN	0.95	0.95	0.95
Transfer with freezing	0.93	0.92	0.92
Transfer without freezing	0.96	0.95	0.95
VGG16 + FCN	0.93	0.92	0.92
VGG19 + FCN	0.92	0.93	0.92
ResNet50 + FCN	0.96	0.96	0.98
VGG16 + LSTM	0.95	0.94	0.94
VGG19 + LSTM	0.97	0.97	0.97
ResNet50 + LSTM	0.97	0.97	0.97
VGG16 + attention	0.91	0.91	0.91
VGG19 + attention	0.90	0.90	0.90
ResNet50 + attention	0.96	0.96	0.96

connections in between layers. The residual network, instead of having a fixed underlying mapping, is designed to have a residual mapping. This change in the traditional neural network architecture enables the residual network to not just have information from its previous layer but from its ancestors as well. Moreover, this network is subsequently deeper, so they can extract more salient features from a given image. These are the reasons why ResNet50-based models have done better on this dataset than the other models.

Furthermore, when applied with LSTM and fully connected network, features extracted from VGG19 proved to be better than the attributes extracted from VGG16. This phenomenon can be explained by the variance in the depth of these two networks. VGG19 is deeper than VGG16 which enables VGG19 to extract lower level features from a frame than VGG16. These lower lever features contribute to better classification.

However, Figs. 18 and 19 show some predictions on the violent and non-violent video frames. We have used ResNet50 + FCN model to generate predictions on the frame. The model has done overwhelmingly well on predicting the non-violent video frames as it has predicted all the non-violent video frames correctly. The model nonetheless, has missed one violent video frame and predicted it as non-violent.

Figure 20 shows the execution time of the experimented pretrained models. The time is shown in seconds. It includes the time taken for feature extraction and for training. These elapsed times, nevertheless, are not universal. These times we have reported here are based on our environmental setup. The models were being trained for 20 epochs each. From the figure, we cannot make any clear verdict about which model consumes less time but we can infer that the ResNet50-based models have taken more time than the other corresponding models of any particular approach. Though the differences are not significant, we still can make a judgement call on why the ResNet50-based models take more time. ResNet50 has more layers than the other two models we have experimented with. So, the extra time ResNet50-based models take is for feature extraction, not for training. It is also evident that LSTM models

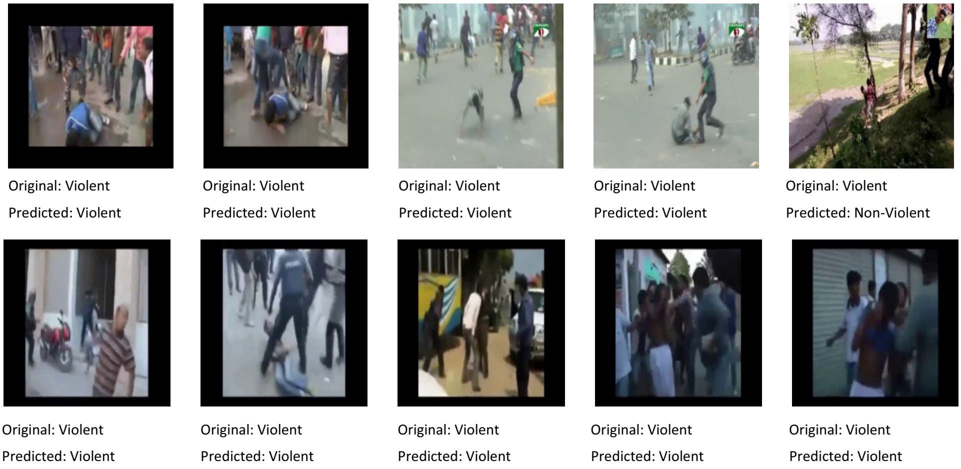


Fig. 18. Prediction on violent video frames.

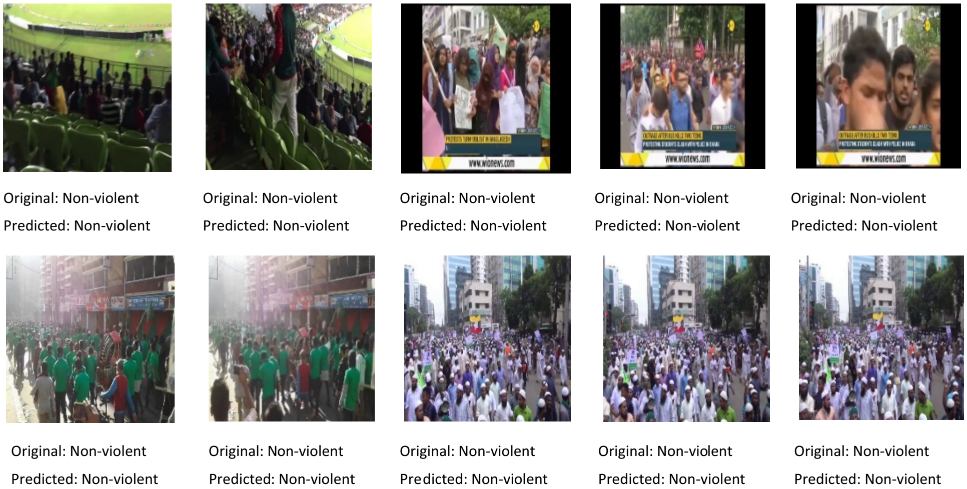


Fig. 19. Prediction on non-violent video frames.

take less time than the other approaches. It is because we did not train the models end to end. At first, we extracted the features and saved them in a NumPy file. After that, while training, we fed these NumPy files to the model. In other models, we had to extract features frame by frame, that means one frame at a time. But, in LSTM we extracted features from 30 frames at a time. This accelerated the process as python takes arrays to its backend and operates the calculations with some low-level faster programming language and then returns the result.

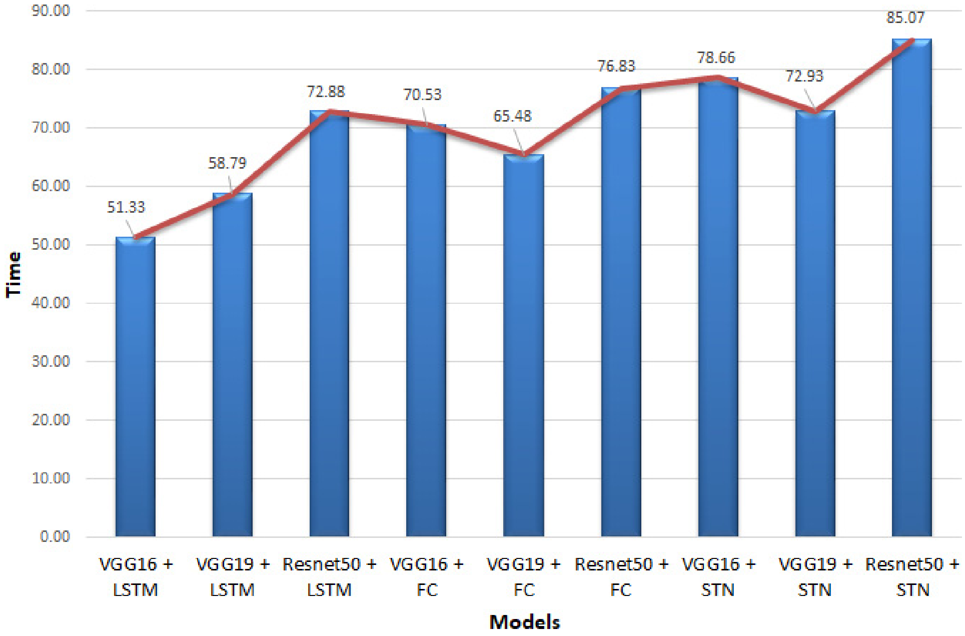


Fig. 20. Execution time of the pretrained models.

6. Conclusion

This study explores and dives deep into leveraging the potential of extracting salient features from the frames which then have been used in detecting violence in the videos. We have experimented with three pretrained ImageNet models, VGG16, VGG19 and ResNet50. The extracted features from each of the frames have been fed into a fully connected network. Moreover, in another experiment, extracted features from 30 frames at a time have been given to an LSTM network as an input sequence. Furthermore, a spatial transformer network has been designed to apply transformation and attention to the extracted features. Along with these models, a custom pretrained model which was initially trained on a Hollywood movie violence dataset has been retrained on our collected videos. We have constructed a CNN model as well to compare the saliency of the extracted features with other pre-trained models. However, the features extracted by the ResNet50 pre-trained model proved be more salient than the other models as classification on these features provided more accurate results.

The journey does not end here though. We have plan to deploy these models on different devices like CCTV and unmanned aerial vehicles (UAV). We plan of pruning the models in order to make them deployable on devices which have low internal memory units. Moreover, in our future study, we want to detect actions which trigger violence on Bangladesh and Global context.

References

1. S. A. Sumon, M. T. Shahria, M. R. Goni, N. Hasan, A. M. Almarufuzzaman and R. M. Rahman, Violent crowd flow detection using deep learning, *Asian Conf. on Intelligent Information and Database Systems* (Springer, Cham, 2019), pp. 613–625.
2. C.-H. Demarty, C. Penet, M. Soleymani and G. Gravier, VSD, a public dataset for the detection of violent scenes in movies: Design, annotation, analysis and evaluation, *Multimedia Tools and Applications*, Vol. 74, No. 17 (Springer Verlag, 2014), pp. 7379–7404.
3. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez and I. Polosukhin, Attention is all you need, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2017), pp. 5998–6008.
4. M. Jaderberg, K. Simonyan and A. Zisserman, Spatial transformer networks, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2015), pp. 2017–2025.
5. C. Ding, S. Fan, M. Zhu, W. Feng and B. Jia, Violence detection in video by using 3D convolutional neural networks, *International Symp. Visual Computing* (Springer, Cham, 2014), pp. 551–558.
6. E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García and R. Sukthankar, Violence detection in video using computer vision techniques, in *CAIP 2011*, eds., P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano and W. Kropatsch, LNCS, Vol. 6855 (Springer, Heidelberg, 2011), pp. 332–339.
7. I. Dai, J. Tu, Z. Shi, Y. G. Jiang and X. Xue, Violent scenes detection using motion features and part-level attributes, in *MediaEval Workshop* (Barcelona, Catalunya, Spain, 18–19 October, 2013).
8. S. Mohammadi, H. Kiani, A. Perina and V. Murino, Violence detection in crowded scenes using substantial derivative, *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (IEEE, 2015), pp. 1–6.
9. T. Zhang, W. Jia, B. Yang, J. Yang, X. He and Z. Zheng, Mowld: A robust motion image descriptor for violence detection, *Multim. Tools Appl.* **76**(1) (2017) 1419–1438.
10. Z. Dong, J. Qin and Y. Wang, Multi-stream deep networks for person to person violence detection in videos, *Chinese Conf. Pattern Recognition* (Springer, Singapore, 2016), pp. 517–531.
11. T. Hassner, Y. Itcher and O. Kliper-Gross, Violent flows: Real-time detection of violent crowd behavior, *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conf.* (IEEE, 2012), pp. 1–6.
12. F. De Souza and H. Pedrini, Detection of violent events in video sequences based on census transform histogram, *Graphics, Patterns and Images (SIBGRAPI), 2017 30th SIBGRAPI Conf.* (IEEE, 2017), pp. 323–329.
13. S. Mohammadi, H. Kiani, A. Perina and V. Murino, Violence detection in crowded scenes using substantial derivative, *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (IEEE, 2015), pp. 1–6.
14. Y. Lyu and Y. Yang, Violence detection algorithm based on local spatiotemporal features and optical flow, *2015 Int. Conf. Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)* (IEEE, 2015), pp. 307–311.
15. Y. Xu and J. Wen, Detecting robbery and violent scenarios, *Robot, Vision and Signal Processing (RVSP), 2013 Second Int. Conf.* (IEEE, 2013), pp. 25–30.
16. P. Zhou, Q. Ding, H. Luo and X. Hou, Violent interaction detection in video based on deep learning, Journal of Physics: Conference Series, Vol. 844, No. 1 (IOP Publishing, 2017), p. 012044.
17. E. B. Nievas, O. D. Suarez, G. B. García and R. Sukthankar, Violence detection in video using computer vision techniques, *International Conference on Computer Analysis of Images and Patterns* (Springer, Berlin, Heidelberg, 2011), pp. 332–339.

18. T. Zhang, Z. Yang, W. Jia, B. Yang, J. Yang and X. He, A new method for violence detection in surveillance scenes, *Multim. Tools Appl.* **75**(12) (2016) 7327–7349.
19. Y. Gao, H. Liu, X. Sun, C. Wang and Y. Liu, Violence detection using oriented violent flows, *Image Vis. Comput.* **48** (2016) 37–41.
20. N. Srivastava, E. Mansimov and R. Salakhudinov, Unsupervised learning of video representations using LSTMS, *Int. Conf. Machine Learning* (2015), pp. 843–852.
21. Q. Dai, R. W. Zhao, Z. Wu, X. Wang, Z. Gu, W. Wu and Y. G. Jiang, Fudan-Huawei at MediaEval 2015: Detecting violent scenes and affective impact in movies with deep learning, *MediaEval 2015 Workshop* (Wurzen, Germany, September 14–15, 2015).
22. M. M. Baccouche, Action classification in soccer videos with long short-term memory recurrent neural networks, *Int. Conf. on Artificial Neural Networks* (Springer, Berlin, Heidelberg, 2010), pp. 154–159.
23. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
24. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2016), pp. 770–778.
25. M. T. Luong, H. Pham and C. D. Manning, Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:1508.04025 (2015).
26. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2016), pp. 770–778.
27. G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control Signals Syst.* **2**(4) (1989) 303–314.