Introduction to Deep learning

Learning to Detect Violent Videos using Convolution LSTM

Submitted To:
Gilad Katz
Department of Software
and Information Systems

Submitted By :
Lior Sidi
Yechiav Yitzchak

Contents

1 Reference

Sudhakaran, Swathikiran, and Oswald Lanz. "Learning to detect violent videos using convolution long short-term memory." In Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on, pp. 1-6. IEEE, 2017. [4]

Available at
https://ieeexplore.ieee.org/abstract/document/8078468/

2 Motivation

[**cited from the article**] Nowadays, the amount of public violence has increased dramatically. This can be a terror attack involving one or a number of persons wielding guns to a knife attack by a single person. This has resulted in the ubiquitous usage of surveillance cameras. This has helped authorities in identifying violent attacks and take the necessary steps in order to minimize the disastrous effects. But almost all the systems nowadays require manual human inspection of these videos for identifying such scenarios, which is practically infeasible and inefficient. It is in this context that the proposed study be comes relevant. Having such a practical system that can automatically monitor surveillance videos and identify the violent behavior of humans will be of immense help and assistance to the law and order establishment. In this work, we will be considering aggressive human behavior as violence rather than the presence of blood or fire.

3 Short description

This work is based on violence detection model proposed by [4] with minor modifications. The original model was implemented with Pytorch while in this work we implement it with Keras and TensorFlow as a back-end. The model incorporates pre-trained convolution Neural Network (CNN) connected to Convolutional LSTM (ConvLSTM) layer. The model takes as an inputs the raw video, converts it into frames and output a binary classification of violence or non-violence label.

Our work is available in the following Github repository:
https://github.com/liorsidi/ViolenceDetection_CNNLSTM

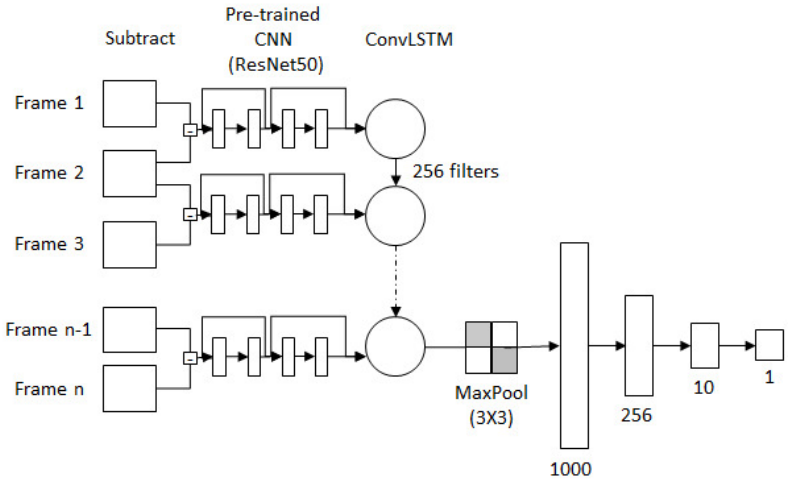
4 Architecture

4.1 Architecture structure

The network architecture is based on the architecture presented in the paper [4], In Figure 1 we describe the architecture build upon four type of layers, the first is the input layer that receive a sequence of 10 frames that are a computed difference of two adjacent frames from the original video. The second type of layers belongs to a Resnet50 CNN network that aim to classify images, the initial weights of the layers are taken form a pre-trained model on image-net, the CNN process each frame separately and during training the weights of the network are shared. The third layer is the Convolution LSTM (ConvLSTM) where each frame from the CNN enters into a ConvLSTM cell with an hidden state of 256 convolution filters of size 3. The forth type of layers process the ConvLSTM and output the binary prediction, a Max polling layer of size 2 reduces the data and chooses the most informational pixels, then the data is batch normalized and connected to a series of fully connected layer of sizes 1000, 256, 10 and finally a binary output perception with a sigmoid activation function. Between each of the fully connected layers we use RELU activation.

We use binary cross entropy as our loss function and RMSprop as an optimizer, 20% of the data is select for validation and rest 80% is selected to train. The learning rate of the network starts with value of 0.0001 and is reduced by half after 5 epochs of no improvement in the validation loss. We train the model with 50 epochs but also use early stopping in case where the network validation loss haven't improve for 15 epochs.

Figure 1: Violence detection model architecture



4.2 Modification from original paper

In this work we have made some changes from original architecture, We evaluate different CNN architectures instead of using only AlexNet. Furthermore, we use dynamic learning rate adjustments, reduced the sequence length and use one perceptron with sigmoid instead of two perceptron with softmax activation.

We train the networks on a NVIDIA GTX1080Ti GPU, the original paper GPU was NVIDIA K40 GPU which is significantly stronger and allow them to train larger sequences with 16 samples per batch while we could use only fit 2 samples per batch with up to 20 frames per sequence.

In Table 1 we summarize the modifications we made in our implementation, some of the changes are due to lack of GPU power, an algorithmic improvement we suggest or both.

Table 1: Modification from the original paper

Parameter	**Original paper**	**This project**	**Reason**
CNN architecture	AlexNet	ResNet 50	Improvement
Learning rate reducing	Fix	Dynamic	Improvement
Cross-entropy (loss)	Categorical	Binary	Improvement
Evaluation	K - Fold	Simple split	GPU
Batch size	16	2	GPU
Sequence length	20 or average	20 or 10	Both

4.3 Hyper-parameter tunning

As suggested in the original paper we evaluated the different hyper-parameters of the network based only on the "Hockey" dataset and then apply them for each dataset. We use only 20 epochs and early stopping of 5 instead of 15 as we apply in the final optimal network training. the original paper use 10 fold cross validation but because we are limited in training resources we used a simple split as follows: 80% for training and 20% for testing (where 20% of the training is used for validation).

Our tunning starts with baseline hyper-parameters which most of them presented in the original paper. We evaluate each hyper parameter separately and choose the best value for the next evaluations. We determined the order of the hyper-parameters to execute in a descending order of importances as follows: CNN architecture type, Learning rate, sequence length, augmentation usage, dropout rate and CNN network

training type (retrain or static). In Table 2 we present the different hyper parameters evaluated in each iteration.

Table 2: Hyper-Parameter tunning parameters

Parameter	**item 1**	**item 2**	**item 3**
CNN architecture	Res Net 50	Inception V3	VGG 19
Learning rate	1e-4	1e-3	
Use augmentation	True	False	
Number of frames	20	30	
Dropout	0	0.5	
Train type	CNN Retrain	CNN static	

5 Dataset

to evaluate the performance of the proposed method three standard public datasets were used, Hockey Fight Dataset [3], Movies Dataset [3] and Violent-Flows [1]. the 3 datasets captured from closed-circuited-TV, Phone or high resolution recorder, the quality, number of pixels and length varies between dataset.

- **Hockey fights:** Dataset composed of equal number of violence and non-violence action during hockey professional matches, usually Two players participating in close body interaction.
- **Movies:** This dataset consists fight sequences collected from movies, for the non-violence label - videos of general action activity gathered from movies. The dataset is made up of 123 violence and 123 non-violence videos. Unlike the Hockey dataset, this dataset varies profoundly between samples.
- **Violent-flow:** This is a crowd violence dataset as the a large number of participates taking part in the video. Most of the videos present in this dataset are collected from violent events taking place during football matches. There are 100 videos in this dataset.

Dataset Summary					
Dataset	Description	Total videos	True labels	False labels	total size
Hockey fights	hockey players	1000	500	500	214MB
Violent-Flows	big crowd videos	200	100	100	81 MB
Movies	movies clip	246	123	123	159 MB

5.1 Data preprocessing

As a preparation for the graph input few steps were taken in the dataset preparation, initially the videos were sampled to a frame by frame sequence as we were limited with computational power. The videos were sampled into a fix number of frames before given as an input to the model. For all dataset combination of augmentation methods were used and for some of the datasets, dark edges were removed from the frame as we present in Figure 3.

As the original article stated, the input to the model is a subtraction of adjacent frames, this was done in order to include a spatial movements in the input videos instead of the raw pixels from each frame. In Figure 2 we present and example of difference computation of adjacent frames where an hockey player pushes another player.

Figure 2: Frame to frame difference

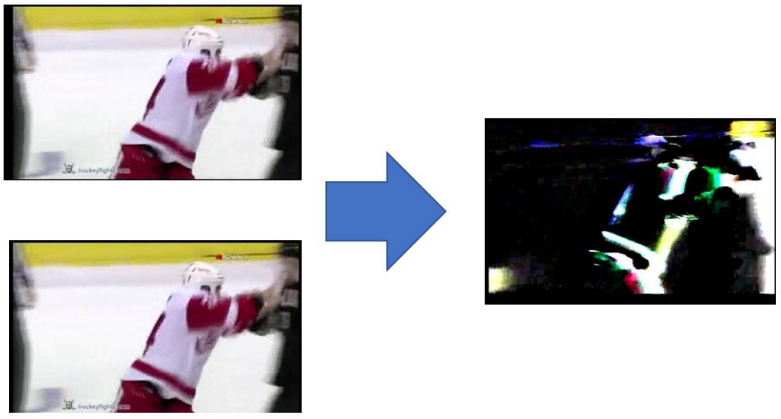


Figure 3: Dark edges removal

To enrich and enlarge the dataset we apply data augmentation with the following transformations on the frames:

- **Image cropping:** a slicing of the image, done each time with a different anchor corner was chosen (Figure 4) .
- **Image transpose:** as a complement steps to the cropping process, a transpose was done, this step was done during the fit generator process (Figure 5)

Figure 4: Cropping of the images

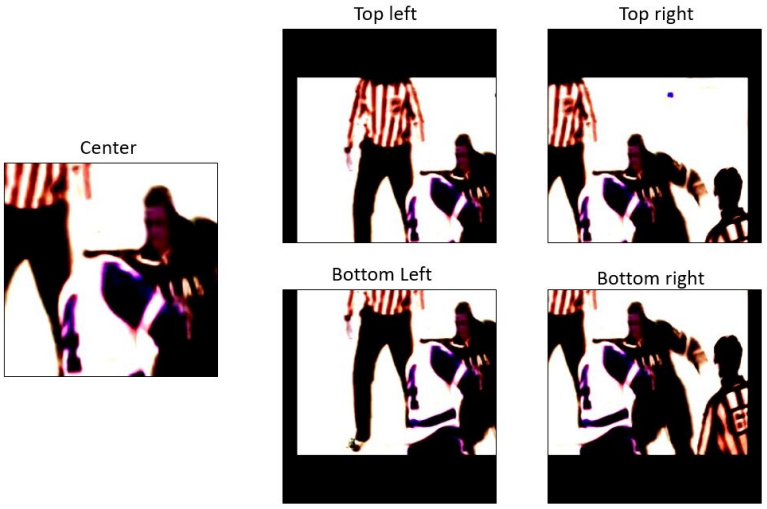
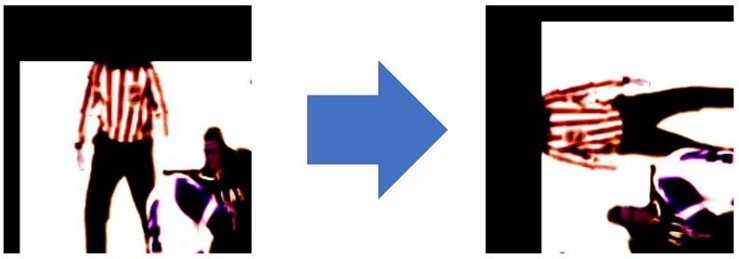


Figure 5: image transpose



6 Results

In this project we use accuracy as the metric evaluation which is in line with all the previous violence detection works. We will present the results of the hyper-tunning process and then the final model results for each of the datasets.

6.1 Hyper-parameter tunning results

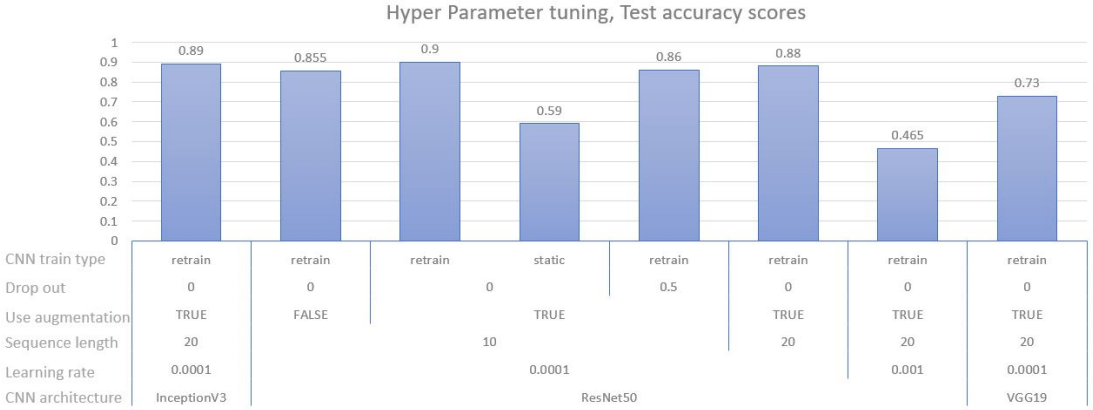
The hyper-tunning process as mentioned in section 4.3 allows us to find the best performing parameters of the network based on the "Hockey" dataset. the chosen architecture is already presented in the section 4.1. In Figure 6 we present the hyper-tunning test accuracy for each of the hyper-parameters values.

The best performing CNN is the Resnet50 with 90% accuracy, the InceptionV3 CNN was not far from the Resnet50 with 89% accuracy but the VGG19 CNN had poor results of only 79% accuracy.

The starting learning rate value had a critical effect on the network results where the 0.001 learning rate resolved with only 46% accuracy which is lower then the random classification. as already mentioned by the original paper, the learning rate of 0.0001 had far better results in all experiments.

The augmentation increases the accuracy by 4.5% and smaller length size of the sequence improve the accuracy by 2%. the dropout of 50% did no improve the model performance and results with only 86% accuracy. As expected the static CNN configuration where the CNN weights are not retrained had very poor results of 59% accuracy.

Figure 6: Hyper-parameter tunning results

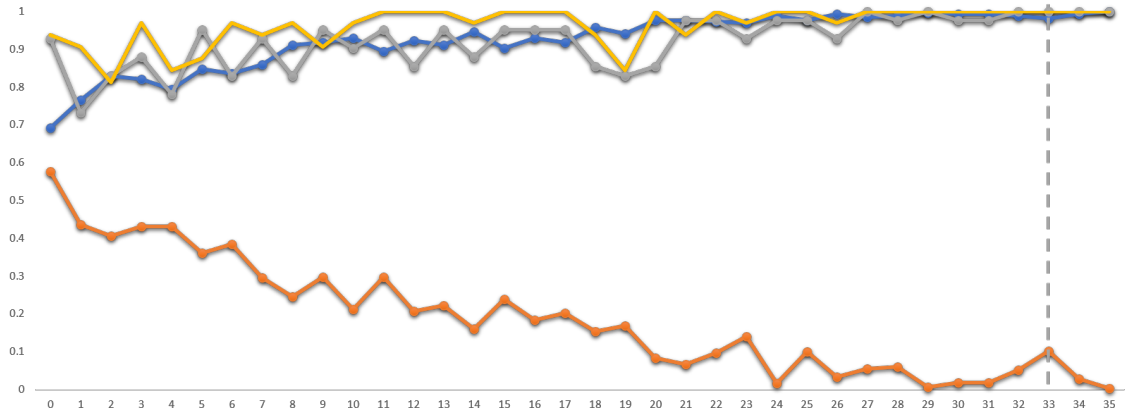


6.2 Optimized model results per dataset

The results presented bellow in Figures 7, 8 and 9 are line charts with the following series: accuracy of the train (Blue), test (Grey), validation (Yellow) along with the train loss (Orange) by number of epochs. All experiments run with 50 epochs in total, where early stopping took place for all cases.

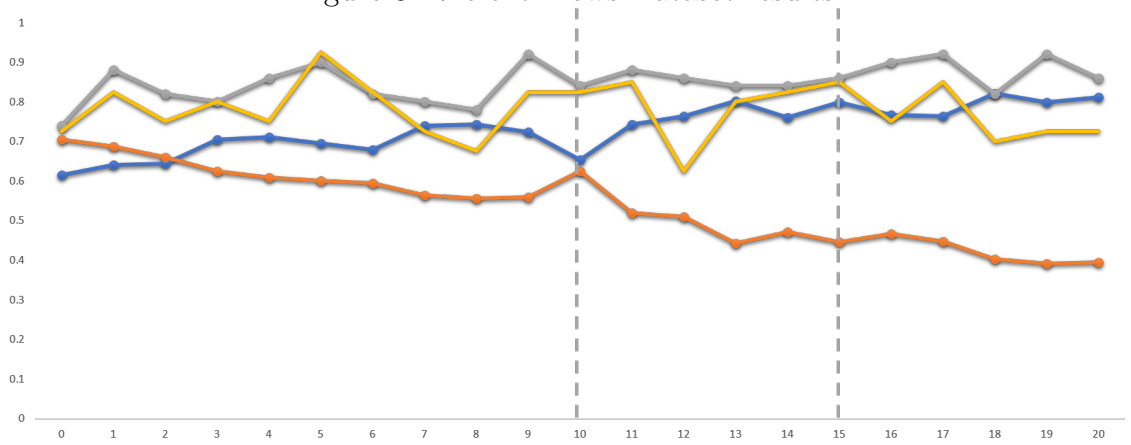
The optimized model results for the "Movies" dataset presented in Figure 7 the model learning rate was reduced one time at epoch 33 and achieved 100% in training, test and validation accuracy score.

Figure 7: Movies Dataset results



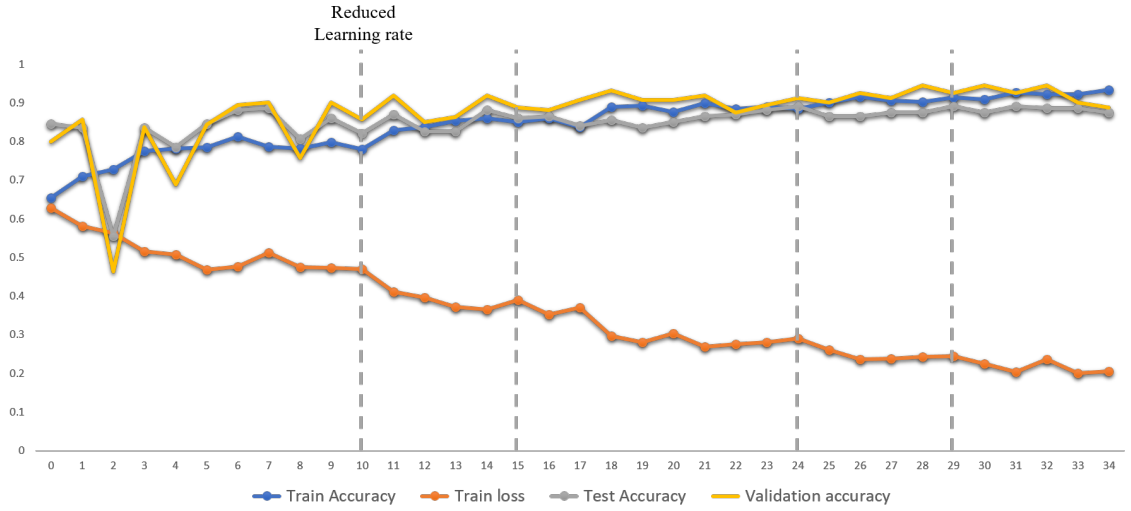
The results for the "Violent-Flow" dataset presented in Figure 8. the model has reduced the learning rate twice starting at 1e-4 at starting point to value of 2.5e-5 at the last epoch. The model test accuracy of the last epoch is 86% and the best overall accuracy from all epochs is 92%.

Figure 8: Violent-Flows Dataset results



As for the "Hockey" dataset, the optimized results presented in Figure 9, the learning rate was reduced 4 times starting at 1e-4 and ending at 5e-5 at final epoch, the early stopping has stopped the training of the model at epoch 34, reaching 87.5% for the test data accuracy in the last epoch and 89% as the best accuracy in overall epochs.

Figure 9: Hockey Dataset results



7 Results analysis

During our evaluation we came across several hyper parameters that had critical effects on the model results. The first is the CNN architecture, the advanced Resnet50 and the inceptionV3 models had significantly better results then the VGG19. this can be easily explained by the fact that both architectures had significantly better results in image classification with much less parameters while having more network [2]. The Resnet50 slightly outperform the inceptionV3 in the violence use case but on imagenet the inceptionV3 is 1.5% more accurate. We believe that the depth of the Resnet50 of 168 layers compare to inceptionV3 depth of 159 might come handy in identifying the violence activity.

We found that retraining the CNN networks improves the performance of the network dramatically, the training of the CNN on the violent data help the network to tune and find relevant patterns of violence and output them to the ConvLSTM layer.

As mentioned in the section 6 the starting learning rate had critical effect on the learning process, the lower starting learning of 0.0001 rate prove to increase the learning of the network compare to 0.001. We assume that the high learning rate cause extreme changes of the network weights and harm it's ability to converge to the right direction, the small learning rate force the network the update it's weights slowly but safely into the right direction of loss.

We believe that the dropout didn't improve the network in this experiment setup because the problem and the datasets are domain specific. The need of generalization will be critical in the future cases when the video files are more heterogeneous with different video quality, camera positioning, type of scenes (not only hockey game, movies or football games) and when more classes are available such as type of violence, amount of participants, violence tool, degree of injury etc.

The data augmentation process helped the model to deal with the small amount of labeled data, the augmentation increased the amount of samples and helped the model to find meaningful patterns the the frames.

For the optimized results analysis we first dive into the "Movies" dataset, the model reached 100% accuracy, this match the results of the original paper. We conclude that it is a relatively "easy" dataset to classify because the learning reduction only occurred once and nearly by the end of the training session.

The optimized model classifying the "Violent-Flow" dataset has reached the lowest score out of all the 3 datasets settling at 86% accuracy. the original paper has reached 94% accuracy and was the most difficult dataset to classify for the original

paper model out of all the 3 datasets, reviewing the videos and the misclassification outputs this dataset contains high variance of video's length with the highest average length of 90 frames per video and shortest length videos, with the lowest resolution camera. Furthermore, the videos contain large crowd where even in the "violent" videos most of the crowd is a spectator and doesn't intervene in the violent act. one suggestion brought up is to split the videos into smaller chunks and to agree on a bagging method to produce the final classification.

Lastly, the optimized model classifying the "Hockey" dataset has reached 87.5%. this compared to 97% in the original paper, this is the only dataset where the model reduced the learning rate 4 times. we suspect that in this dataset in particular having a long period of training the model can produce higher accuracy results, as we were limited with computing power compare to the original paper model, this could be one of the key factors explaining the 10% gap between the two models.

8 Conclusions

In this work we implemented deep learning model to predict violence in video data, We found our implementation to deal well with this task even though our GPU power was relatively low. The potential of deep learning models is high and can be used easily by law enforcements officers to identifying violence in the streets or in kindergartens.

We found the the smart data preprocessing of the video's frames play an important factor as well as some of the training parameters such as: CNN network, learning rate and data augmentation.

Looking forward to more complex violence scenarios and appliances it will take researchers to find creative solutions for data collection, advance generalization techniques and real-time optimizations.

References

[1] Tal Hassner, Yossi Itcher, and Orit 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*, pages 1–6. IEEE, 2012.

[2] Keras. Applications. [Online; accessed 21-July-2018].

[3] Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno García, and Rahul Sukthankar. Violence detection in video using computer vision techniques. In *International conference on Computer analysis of images and patterns*, pages 332–339. Springer, 2011.

[4] Swathikiran Sudhakaran and Oswald Lanz. Learning to detect violent videos using convolutional long short-term memory. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE, 2017.