Deep Learning Approach for Violence Detection in Urban Areas

Marius Baba[1], *Vasile* Gui[2], and *Dan* Pescaru[1]

[1]Politehnica University of Timisoara, Department of Computers and Information Technology, Bd. V. Parvan no. 2, Timisoara, Romania
[2]Politehnica University of Timisoara, Department of Electronics, Telecommunications, and Information Technology, Bd. V. Parvan no. 2, Timisoara, Romania

Abstract. Today modern cities tend to grow rapidly. The increased population density brings new challenges in term of public safety. Crime and violence are hard to be detected and managed especially in specific crowd environments like music concerts, sport events or public meetings. To overcome this issue the city administration should implement monitoring systems capable of detecting and analysing such situations. The work presented here combines two approaches that enable implementation of an efficient solution adapted for this purpose. The first one involves sensor networks that prove to be cost effective solution in a smart city environment. They can benefit on the existing surveillance infrastructure and allows rapid deployment. The second approach uses deep learning techniques. They demonstrate outstanding performances in image and actions classification based on a prior learning process. By combining these two approaches we succeed to obtain a real-time and cost-effective solution designed for urban area surveillance networks.

1 Introduction

The crowded environments characteristic to modern cities are hard to manage. An important part of this task is related with violence detection and management. To overcome this issue the city administration should either increase the density of safety ground forces or to develop an efficient monitoring system. While the first solution is requiring important financial effort, the second one could benefit on rapid development of cheap WSN & IoT technologies. This modern solution is also more convenient for long term runs. Even if it requires moderate investments at deployment, the long-term expenses are kept at a minimal level. Moreover, the built infrastructure could be used for future applications at a very convenient cost. The installation effort of a video sensor network is reduced to minimal. It requires in general a network connection and minimal software configuration. These advantages enabled spreading of sensor network monitoring systems in many cities. Some of them are engaged for capturing traffic. The others are meant for spotting abnormal pedestrian situations including vandalism, fights, and robbery.

[1] Corresponding author: mariusb008@gmail.com

Capturing violence actions in real time by surveillance system implies constant analysis applied on video streams. The main concern of the presented research is the design of a video analysis algorithm capable of capturing violent actions in generic urban areas. It uses deep learning architecture and video motion features to discover a possible violent behaviour in a typical city scenario.

In this paper we present an efficient solution for violent behavior detection in smart cities, using a Deep Neural Networks (DNNs). The paper is organized as follows. Section 2 contains a review of related work regarding behavior detection including violence detection. Both traditional and DNN based approaches are included. The architecture of the proposed violence detection system is presented in section 3, while the results of our experiments are presented in section 4. We compare two solutions for extracting motion features and demonstrate that the MPEG flow based approach yields the best results. We also include for comparison purposes results from other works concerning violence detection. Section 5 concludes this paper.

2 Violent behavior detection techniques

Violent activities can be recognized by analyzing human actions. We are interested in actions leading to suspicion about violence in surveilled urban areas. Such activities usually involve specific motion patterns and take place in relatively low time intervals. The most important information needed to recognize and characterize human physical actions and interactions is motion.

Early work on activity analysis in video concentrates on holistic representations of motion, like Motion History or Motion Energy Images [1]. A more recent trend is to use localized representations. Interest points like the Space-Time Interest Points (STIPs) or Space Time Cuboids [2] can be used to characterize image motion. The information provided by these features is further refined by generating higher level features like motion trajectories. A representative work using motion trajectories for analyzing activities in a parking lot is reported in [3]. Trajectories are generated by a combination of methods involving foreground/background segmentation [4] and a multi-hypothesis tracker [5]. Inference is obtained via random forest and support vector machine classifiers.

Another powerful feature in motion analysis is optical flow. Optic flow research has a long history [6]. Recent optic flow estimators are based on Deep Neural Networks (DNN) [7] and proves significant performance improvement. Currently DNNs also challenge state of the art in activity recognition. Frequently used architectures in activity recognition are Spatio-Temporal Convolutional Network [8] and Recurrent Networks [9]. A major advantage of DNN based solutions is that these networks discover automatically optimized features needed to solve the problem at hand. However, there are two drawbacks of this approach. One is the need of very large databases for proper training. The other one is the hardware resource needed for real time operation. Both are critical in the case of our application. A good survey of the work in activity recognition, mainly focused on DNN solutions can be found in [10].

One possible approach to capture the time domain evolution of activities is to feed the DNN with a set of consecutive frames. The simplest architecture using this idea, called early fusion is using only one spatio-temporal volume of the video stream which is slide in the time domain [11]. An alternative solution called late fusion is to input several single frames to several Convolutional Neural Networks (CNNs) which are fused at the final fully connected stage [12]. The authors also make experiments with an intermediate architecture, called slow fusion. It involves several CNNs fed with overlapping chunks of input frames. The CNNs are gradually fused in a multiresolution manner toward the final, fully connected layer. Slow fusion needs less parameters to be learned and outperforms the other version of 3D approach.

DNNs composed of long short-term memory (LSTM) and 3D CNNs are also successfully used for action recognition [13]. Descriptors generated by the CNN are feed into the LSTM network. Memory feature enables the LSTM network to learn action sequences. However, training LSTMs is challenging.

3 DNN architecture with optical flow input

The capability of DNNs to automatically discover optimized features may be very tempting for bypassing the skilled work and insight in the application specificities, needed to generate handcrafted solution. There is a price to obtain this performance. The skilled work should be replaced by the rather repetitive task of data labelling. Large labelled databases for DNNs have been built and made publicly available in many modern research fields. Unfortunately, when it comes to develop a new application this advantage is most often lost. Labelled databases containing violence scenarios in surveillance video are scarce. To our best knowledge there are only two important surveillance video databases publicly available online as BEHAVE [14] and ARENA [15]. We use both in this work.

We try to combine here the power of DNNs with the efficiency of well-chosen handcrafted motion features. To enable the system to learn from the relatively small databases available, we feed the network with a reduced set of motion features. Also, our approach uses a reduced number of convolutional layers while preserving the detection performance using a relatively small number of training samples. As a result, we obtain a real time distributed system for violence detection.

To deal with various video resolutions, every video frame is split into sub-windows of 256*x*256 pixels size. We use a focus of attention approach by selecting only one window in each step. From every window, a set of 16x16 motion vectors are then extracted representing the motion features. A block diagram of the proposed system is given in figure 1.

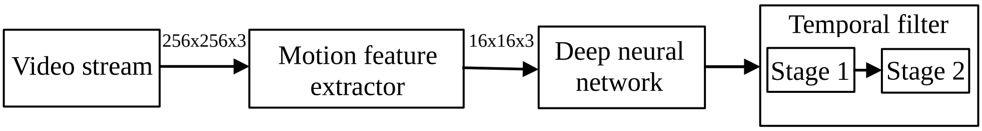

Fig. 1. Block scheme of the proposed system

3.1 Feature extraction

We extract the MPEG [16] flow motion features and use them as the input of our DNN. These features represent estimators of 16x16 image block motions which are computed and used by the image encoder to compress the video data. The usefulness of these features in activity recognition has already been demonstrated in [17][18]. In the first case MPEG flow features are used in the bag of words (BoW) approach, while in the second one the authors use spatio-temporal CNN.

For comparative purposes, two sets of motion features are used. The first one is the MPEG flow obtained directly from the MPEG stream with practically no additional processing overhead. The second one is based on the Farneback Optical Flow algorithm [19] and is illustrated in figure 2. To compare the performances of the two motion features on the same CNN, we resized the optic flow vector maps to the same 16x16 format. The 256 extracted vectors represent low level handcrafted features which are supposed to encode the motion pattern needed to discriminate fight from non-fight frames.

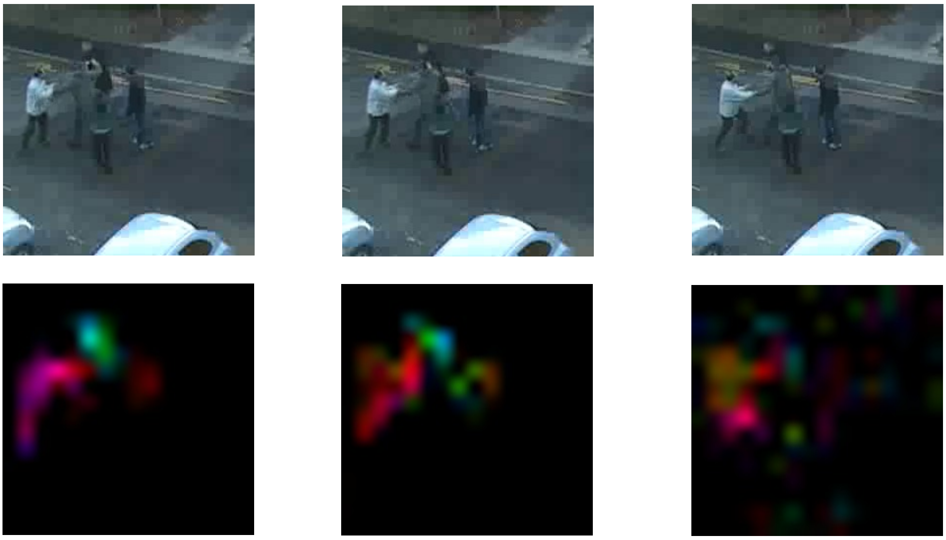

Fig 2. First row contains three frames from a fight sequence. The second row contains the color coded optical flow feature maps (rescaled).

The computational cost of the MPEG flow features is very low. The encoder provides for each frame a list of 16x16 block motions where motion is present. The list contains pairs of source and destination position vectors. Their difference represents motion vectors. Each one is encoded in our reprezentation as a color in the HSV space. Vector direction represents the hue while vector magnitude represents the value parameter. The RGB coordinates of this color is fetched as input feature to our DNN.

3.2 The CNN architecture

As we target low resources embedded hardware, we keep the network complexity to a moderate level. An additional advantage of this choice is a reduced risk or overfitting, considering the relatively small size training data set available.

The CNN architecture has a complexity of only 20418 variables (weights + biases) to be learned from the data. Given the small size of the input matrix(16x16), all convolution kernels are set to a 3x3x3 format in the space-channel domain. The first stage contains 32 such filters. The second convolution layer extends the channel dept to 64 using the same convolution kernel. Each convolution layer is followed by rectified linear unit (ReLU) and max pooling 2x2 layer. As a result, the fully connected layer is fed with 64 activation maps of size 4x4. The last fully connected layer generates two outputs corresponding to the fight and respectively the non-fight class.

The architecture of the system is composed of the following layers:

1. convolution layer - 3x3 kernel size, 3 channels, 32 output channels
2. ReLU unit
3. max pool layer - 2x2
4. convolution layer - 3x3 kernel size, 32 channels, 64 output channels
5. ReLU unit
6. max pool layer - 2x2
7. fully connected 1024 nodes
8. ReLU unit
9. fully connected 2 nodes
10. output argmax one class.

3.3 Time domain filter

To capture temporal evolution of scene activity and give a more continuous output, corresponding to human judgement and observer labelling, we use a post processing cascaded percentile filter. The binary output of stage one at frame k is given by:

$$L(k)=\begin{cases}1 \text{ for } p_k \geq Th \\ 0 \text{ for } p_k < Th\end{cases}, \tag{1}$$

where Th is a threshold and p_k is the percentage of the detected fight frames given by:

$$p_k\% = \frac{\sum_{i=k-N}^{k+N} y_i}{2N+1} * 100\% . \tag{2}$$

In equation (2), y_i represents the output of the CNN at frame *i* and *N* represents half of the window size centred at frame k. The parameter *N* was set to the value of 10, to cover a time window of about one second at a frame rate of 20 fps. One second is a reasonable time for an observer to decide whether an activity is a fight. The threshold Th was selected at the level optimizing the F1 score of our system.

The second stage filter has a similar definition with optimal parameters N=7 and Th= 15. It proved to be necessary to eliminate short bursts of false positive alarms outputted by the first stage.

4 Experimental results

Training data have been obtained from two well-known databases [14][15], containing 8765 classified frames, of which 1322 correspond to attack and the rest are frames capturing normal behaviour. Additionally, dataset augmentation technique has been applied. As the loss function, we use the cross entropy.

https://doi.org/10.1051/itmconf/20192903009

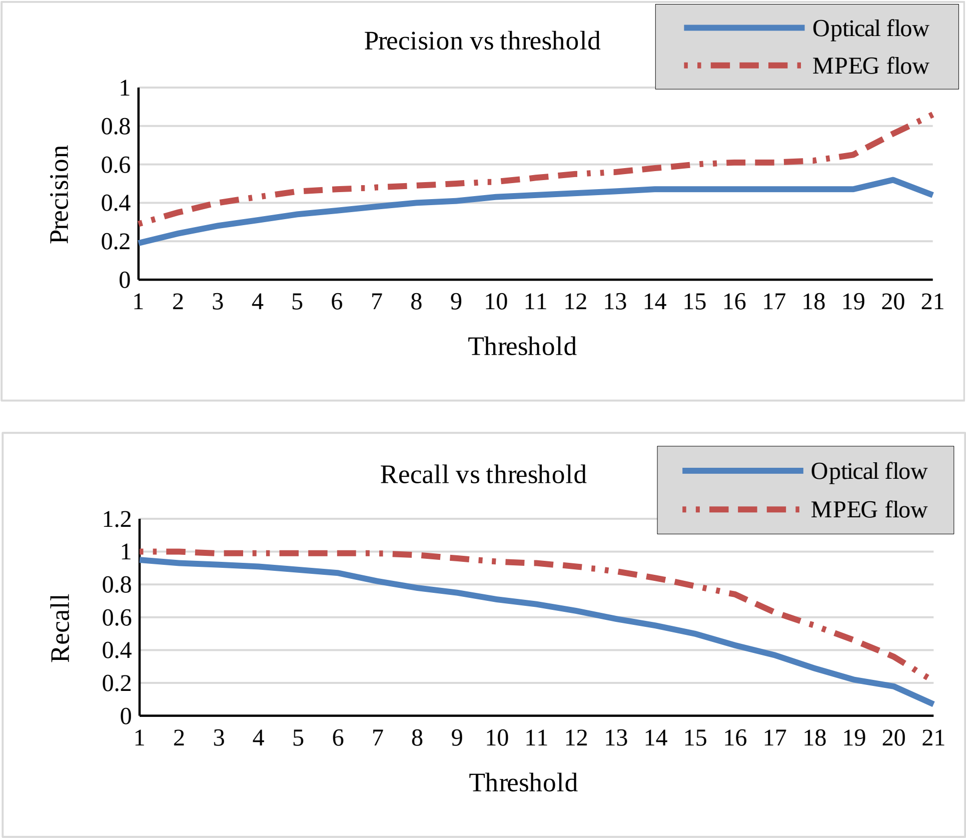

Fig. 3. Precision and recall values as a function of post processing stage one filter threshold.

Temporal filter threshold parameter was optimized using precision and recall values as a function of post processing filter threshold. Results for the first stage filter, using two different version of optical flow feature estimation are given in figure 3. Dashed-dotted curves correspond to MPEG flow, while solid curves are obtained by using Farneback Optical Flow algorithm.

Experiments show better performance for the MPEG flow features. The product of precision and recall is optimized for threshold value 13 for this classifier.

The second stage filter parameters were set to values N = 7 and p=100%, in order to optimize the performance evaluated at clip level. We obtained these parameters by testing the threshold value in the interval from 1 to 7.

To validate the performance of our method we divided the video data into 85 clips, following the procedure adopted in [20]. All clips contained a type of activity from the following categories: attack, walk, run and group activity. No empty clips were included. The class of violence was represented by attack clips, while the other activities are considered non-violent. A clip is classified as containing violence if at least one output of the second stage filter predicts this class. The results are synthetized by the confusion matrix in Table 1.

Table 1. Confusion matrix

		Actual class	
		Violence	No violence
Predicted	Violence	15	19
	No violence	0	52

Based on Table 1 we compute:

$$TPR=\frac{TP}{TP+FN}, \tag{3}$$

$$FPR=\frac{FP}{FP+TN}. \tag{4}$$

In our case the values are 100% for TPR and 26.76% for FPR. This means that no violent event is missed and only 26.76% of the non-fight clips are predicted as false alarms. The overall accuracy is 85.88% although we mentioned the first two parameters are more significant from the point of view of our application.

Zhang [20] claim slightly higher accuracy on the BEHAVE dataset, 87.17%, but they don't directly report recall data which is more important in our application. Another work using the same datasets [21] reports results in for of true positive versus false positive rate graph. From the graph the false positive rate at 100% true positive rate is very close to our result.

5. Conclusions

We propose here a DNN approach for sensor network violence detection application designed for urban area surveillance with automatic violence scene detection. We also prove that such system could benefit from existing features embedded in every MPEG video stream. It achieves state of the art performance, running on low computational embedded architecture on a sensor network node. Therefore, by combining sensor networks with deep learning approaches, we succeed to obtain a real-time and cost-effective solution designed for urban area surveillance networks.

References

1. A. F. Bobick, J. W. Davis. *The recognition of human movement using temporal templates*. IEEE Transactions on Pattern Analysis and Machine Intelligence **23.3**, pp. 257-267 (2001).
2. I. Laptev, *On space-time interest points*. IJCV **64.2-3**, pp. 107-123 (2005).
3. M. Andersson, L. Patino, G. J Burghouts, A. Flizikowski, M. Evans, D. Gustafsson, H. Petersson, K. Schutte, J. Ferryman. *Activity recognition and localization on a truck parking lot.* Advanced Video and Signal Based Surveillance (2013).

4. Z. Zivkovic. *Improved adaptive Gaussian mixture model for background subtraction*, ICPR, (2004).
5. S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*, Artech House (1999).
6. D. Fortun, P. Bouthemy, C. Kervrann, *Optical flow modeling and computation: a survey*. CVIU **134**, pp. 1-21, (2015).
7. E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, *Flownet 2.0: Evolution of optical flow estimation with deep networks*. IEEE Conference on CVPR , **Vol. 2** (2017).
8. G. W. Taylor, R. Fergus, Y. LeCun, C. Bregler, *Convolutional learning of spatio-temporal features*. ECCV, Springer, Berlin, Heidelberg (2010).
9. L. R. Medsker, L. C. Jain. *Recurrent neural networks*. Design and Applications **5** (2001).
10. S. Herath, M. Harandi, F. Porikli. *Going deeper into action recognition: A survey.* Image and vision computing **60**, pp. 4-21 (2017).
11. S. Ji, W. Xu, M. Yang, K. Yu. *3d convolutional neural networks for human action recognition.*IEEE Transactions on Pattern Analysis and Machine Intelligence **35(1),** pp. 221–231 (2013).
12. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei. *Large-scale video classification with convolutional neural networks.* CVPR, pp. 1725–1732 (2014).
13. J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell. *Long-term recurrent convolutional networks for visual recognition and description*. In Proc. IEEE Conference on CVPR, pp. 2625–2634 (2015).
14. S. Blunsden, R. B. Fisher, *The BEHAVE video dataset: ground truthed video for multi-person behavior classification.* Annals of the BMVA **4.1-12** (2010).
15. L. Patino, T. Cane, A. Vallee, J. Ferryman, *Pets 2016: Dataset and challenge*. Proceedings of the IEEE Conference on CVPR Workshops (2016).
16. MPEG standard. Retrieved on 2018, April 24 from MPEG homepage https://mpeg. chiariglione.org/.
17. V. Kantorov, I. Laptev. *Efficient feature extraction, encoding and classification for action recognition.* Proceedings of the IEEE Conference on CVPR (2014).
18. V. Gul, I. Laptev, C. Schmid, *Long-term temporal convolutions for action recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence (2017).
19. G. Farnebäck, *Fast and accurate motion estimation using orientation tensors and parametric motion models*. International Conference on Pattern Recognition, **vol. 1**, pp. 135-139 (2000).
20. T. Zhang, W. Jia, B. Yang, J. Yang, X. He,Z. Zheng, *Mowld: a robust motion image descriptor for violence detection*. Multimedia Tools and Applications **76.1**, pp. 1419-1438 (2017).
21. C. Xinyi, L. Qingshan, G. Mingchen, D. N. Metaxas, *Abnormal detection using interaction energy potentials*. CVPR (2011).