2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506142

VIOLENCE DETECTION FROM VIDEO UNDER 2D SPATIO-TEMPORAL REPRESENTATIONS

Mohamed Chelali, Camille Kurtz and Nicole Vincent

Université de Paris, LIPADE (Paris, FRANCE)
{firstname.lastname}@u-paris.fr

ABSTRACT

Action recognition in videos, especially for violence detection, is now a hot topic in computer vision. The interest of this task is related to the multiplication of videos from surveillance cameras or live television content producing complex $2D + t$ data. State-of-the-art methods rely on end-to-end learning from $3D$ neural network approaches that should be trained with a large amount of data to obtain discriminating features. To face these limitations, we present in this article a method to classify videos for violence recognition purpose, by using a classical $2D$ convolutional neural network (CNN). The strategy of the method is two-fold: (1) we start by building several $2D$ spatio-temporal representations from an input video, (2) the new representations are considered to feed the CNN to the train/test process. The classification decision of the video is carried out by aggregating the individual decisions from its different $2D$ spatio-temporal representations. An experimental study on public datasets containing violent videos highlights the interest of the presented method.

Index Terms— Video Classification, Violence Detection, Spatio-temporal Features, Convolutional Neural Network.

1. INTRODUCTION

Video analysis represents a challenging task in computer vision and pattern recognition. A video is often considered as a sequence of frames or as an image whose content evolves through time. More technically, a video is represented by a data-cube endowed with spatial and temporal dimensions. Historically, studied applications dealing with such data are action recognition, tracking and video classification [1].

Violence detection and recognition became recently a hot topic in the community. This problem can be investigated to secure public spaces under surveillance, for example in railway stations or prisons. This problem has often been viewed as a video classification task in the literature [2]. Pioneer methods were based on the extraction of hand-crafted features from the video content, to feed machine learning methods, such as SVM or Random Forests, for classification. Most efficient features were classically the Histogram Of Gradient, Histogram Of Flow, Scale-Invariant Feature Transform (SIFT) and Spatial-Temporal Interest Point. In [3], the authors proposed the concept of Motion SIFT where motion features are computed only in the neighborhood of points of interest given by SIFT. Another method designed for violence characterization relies on Violent Flow (ViF) [4], obtained by computing the magnitude of the optical flow over time. This method was improved later by considering the Orientation in ViF [5]. In [6], the improved Dense Trajectory is proposed to classify action. This method was then used in the case of violence, combined with Improved Fisher Vectors [7].

Nowadays, deep neural network (DNN) methods, which are trained in an end-to-end manner, lead to remarkable performances for video classification [8]. Mostly, used DNNs are convolutional neural networks (CNNs). When dealing with such data, CNNs have the ability to learn spatio-temporal features, generally by using $3D$ convolutions. For example the C3D [9] and I3D [10] models demonstrated outstanding results for action recognition. Others propose two-stream networks that take into account the raw values from video and optical flow in order to capture respectively the appearance and the motion [11, 12]. Feature maps from the two networks are then merged together before a dual connected layer employed to take the decision. We find also the representation flow model that captures the flow of flow by processing frame by frame [13]. Due to the large size of the videos, most of these architectures cannot consider the video as a whole, but videos are split into non-overlapping clips of few frames (e.g. 16 frames) which are inputed to the networks. The decision in all these methods is taken with a late fusion of the learned features followed by a classical fully connected layer.

Another kind of DNNs deal with $3D$ point cloud data structures [14]. As the representations are sparse sets of voxels, the methods designed for these data require large memory resources. Among these methods, PointNet++ [15] explores the metric space distances to include local features. In [16], the PointConv model is proposed. This is an adaptation of the convolution layer to $3D$ point cloud that is based on a Monte Carlo approximation of the $3D$ continuous convolutions. The authors of [17] propose the SPIL model that learns human skeleton interactions by considering the inter-relation of points. Human skeleton point clouds are extracted following [18]. SPIL treats the point cloud as a graph in order to

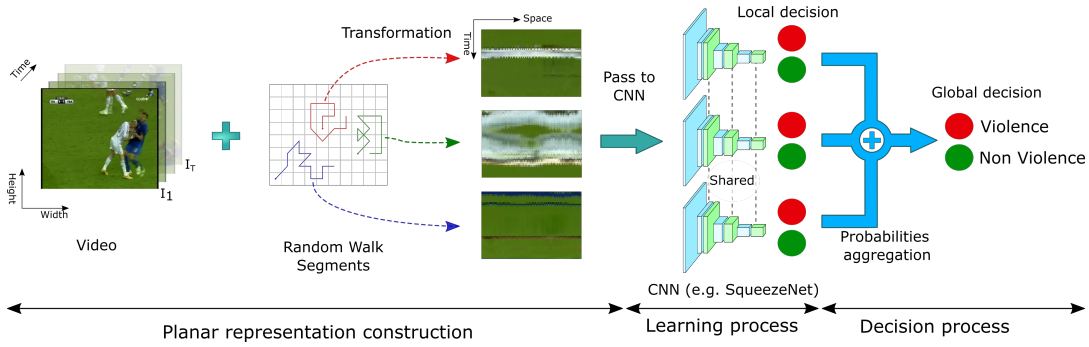


Fig. 1: Workflow of the proposed method: (left) Planar representation construction; (right) Classification process.

include local information to each point. Finally, the dynamic graph CNN (DGCNN) [19] proposes the EdgeConv layer that applies a convolution on the edges of neighboring points.

To synthesize, state-of-the-art methods lead to very good results but suffer from certain limits. On the one hand, $3D$ convolutional methods require large annotated databases to learn the weights of the associated networks. On the other hand, the methods based on point clouds require upstream calculation of points of interest, for example via a detection of persons, which can be algorithmically expensive and lead to errors. We present in this article the VDstr (Violence detection under spatio-temporal representations) method, whose methodological bases were initially proposed in [20] to analyze Image Time Series. This method is now extended and adapted to perform a video classification task by using only a $2D$ CNN. The underlying strategy is based on planar representations associated with the videos. They carry spatial and temporal information, by using $1D$ spatial structures (preserving significant spatial information), that are characterized by temporal information from video contents.

The article is organized as follows. Section 2 presents the VDstr method. An experimental study on public datasets containing violent videos is proposed in Section 3. Finally, conclusions and perspectives are provided in Section 4.

2. VDSTR: VIOLENCE DETECTION UNDER SPATIO-TEMPORAL REPRESENTATIONS

This section presents our approach (VDstr) for violence detection from videos under spatio-temporal representations. The main strategy of the method is the creation of planar representations by preserving spatial and temporal information. We first present in the next Section the global framework of the method, followed by more details for each step.

2.1. Framework

Generally, when performing video classification, we assume the use of $3D$ CNN. From another point of view, the pixels in a video can be considered as time series of color values. In our work, the VDstr method is designed to analyze videos from an intermediate point of view, between the video as a whole and the set of pixels. The main idea of the method is to consider the data no more as a 3D structure but to differentiate the three dimensions and to consider the data as a $2D + t$ structure. We replace this $2D + t$ data by a set of planar representations, noted P_{rep}, capturing both spatial and temporal information. To do so, we are using $1D$ segments generated on the spatial domain. Thus, a video will be represented by several P_{rep}. Then, a classical $2D$ CNN can be trained from these representations to learn spatio-temporal features, and in a second step a classification task is performed. Figure 1 illustrates the workflow of the method.

Section 2.2 details the video representations, while Sections 2.3 and 2.4 present the learning and decision processes.

2.2. Representing a video by planar representations

Instead of considering the whole $2D$ spatial domain of the video, we think only parts of the domain is sufficient. Most often some patches are designed as they preserve the $2D$ space topology. In order to decrease the complexity of the data, we have considered some $1D$ structures, some pieces of curves to partially preserve the notion of neighboring pixels.

The spatial domain of the video is noted $\mathcal{D} = [0, H-1] \times [0, W-1]$ where W and H represent respectively the (common) width and height of the video frames. In a curve $\mathcal{C}$, each pixel has only two neighbors, and the length L of the curve makes a link to more of less distant pixels. In our case, we have chosen to build the curves thanks to the Random Walk (RW) model, endowed with Markovian properties. We note $RW(L)$ according to the chosen L of the built curves. The first point of the curve is initialized randomly on the spatial domain $\mathcal{D}$ and for each next point, 8 directions are possible except for border pixels. Then, pixels are indexed according to their curvilinear abscissa on the $1D$ structure, leading to a string of pixels, noted $\mathcal{C}_p = (p_0, \ldots, p_{L-1})$. By doing so, spatial information are partially kept.

Finally, the planar representation P_{rep} is constructed by stacking vertically (Y axis) for each pixel p_j ($j \in [0, L-1]$),

its color intensities through the time. (Note that if the pixels p_j in the original video are characterized by multiple radiometric values (i.e., RGB), then the P_{rep} representation is a multivalued $2D$ image.) The produced image is a spatio-temporal representation in only 2 dimensions with size of $T \times L$, where T is the duration of the initial video (see on the right of left part of Figure 1).

Thanks to the randomness processes of RW, N_b curves can be built leading to different P_{rep} representations per video. N_b is a parameter of the method that enables to control the number of P_{rep} representations per video. In this context, different spatial configurations are considered ensuring a statistical representation of the variety of $2D$ configurations.

2.3. Learning process: Planar representations labeling

A CNN model is used in order to label the spatio-temporal representations P_{rep}. A $2D$ CNN is intended to learn and extract spatio-temporal features with $2D$ convolutions. Our choice comes to SqueezeNet model [21] but it can be changed with any other $2D$ CNN. SqueezeNet is a rather small network leading to the same accuracy level as AlexNet model when evaluated on the IMAGENET dataset.

The CNN model is trained with the $2D$ P_{rep} defined in each input video from the training set. The learning can use the data available but can also benefit from transfer learning strategies. Here, the computed weights on the IMAGENET dataset can be fine-tuned with our P_{rep} images.

2.4. Decision process

In general, when dealing with $3D$ CNNs, a single class probability vector is given for the video. If the video has been divided into several parts, the decision is taken by aggregating the class probabilities vectors of each part.

In our case, a second step is needed to come from the local P_{rep} representations to the video level. We generate N_b^{test} representations per video scattered in the spatial domain $\mathcal{D}$. For each P_{rep}, a local decision providing the class probability vector is computed. Then, the global decision for the video relies on the same decision strategy as when the video is split.

3. EXPERIMENTAL STUDY

The VDstr method has been evaluated on four violence datasets (Section 3.1). After data preparation and validation protocol (Section 3.2), the obtained results are described and compared to state-of-the-art methods (Section 3.3).

3.1. Datasets

Four datasets dedicated to violence detection have been used:

RWF2000: 2000 videos captured from surveillance camera collected from Youtube. Each class has 1000 videos. The videos do not share the same size but they have the same duration which is 5 seconds [12].

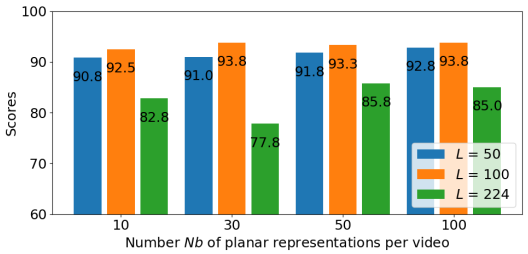


Fig. 2: Results obtained on the RWF2000 by varying the length L and number N_b^{test} of P_{rep} in the decision step.

Hockey Fights: 1000 videos collected from Hockey sport of the National league. Each class has 500 videos that have the same size with and a duration of almost 2 seconds [2].

Movies Fights: 200 videos collected from different scenes of films. It is divided into 100 violent videos and 100 non-violent videos. The videos do not have same size nor same duration [2].

Crowd Violence: 246 videos with crowd scenes. The number of videos per class is about 123. Most of the videos are extracted from football games. The videos do not share neither size nor duration [4].

3.2. Data preparation and validation protocol

Videos from these datasets are labeled at the video level. A curve is defined locally and in a violent video all parts are not necessarily violent. Then, to train the labeling model of P_{rep} more precisely, we decided to focus on the region of interest (ROI) of the violence in order to label the P_{rep} as violent or non-violent. For this, we computed dense optical flow with Gunner Farneback's method. Then, we average the obtained flow through the temporal axis of the video. Finally, the ROI is obtained by applying a binarization (Otsu method) on the mean flow to get the high motion region that we assumed to represent a violent area.

As training set of violent P_{rep}, we considered 30 P_{rep} inside the ROI of violent videos and for non-violent P_{rep}, we considered 28 P_{rep} in the non-violent videos and 2 P_{rep} outside the ROI of the violent videos. So, we get a balanced number of P_{rep} instances for each class. For the test set, N_b^{test} planar representations P_{rep} are built without using the ROI. To make a decision, we varied N_b^{test} to 10, 30, 50 or 100.

Our approach is validated by using a K-Fold cross-validation technique. Each dataset is split five times to three sets at video level: 60% (train), 20% (validation) and 20% (test). Then, the CNN is trained and evaluated five times and we indicate the average overall accuracy (OA). Except for RWF2000 dataset, as the test folder is provided in [12], we just extracted 20% from training set to get a validation set.

Table 1: Obtained accuracy when training/testing on each dataset ($L = 100$). Thickness of bold rectangles decreases from the first to the third rank.

	Method	RWF2000	Movies fights	Hockey fights	Crowd Violence
	N_b^{test}	test set	5 folds		
VDstr	10	92.5	99.5	92.2	88.2
	30	93.8	96.0	93.9	90.6
	50	93.3	98.5	93.6	89.0
	100	93.8	98.5	94.4	89.8
3D CNN	Temp. Seg. Nets [11]	81.5	94.2	91.5	81.5
	I3D [10]	83.4	95.8	93.4	83.4
	Represent. flow[13]	85.3	97.3	92.5	85.9
	Flow Gated Net [12]	87.3	n/a	98.0	88.8
	ECO[22]	83.7	96.3	94.0	84.7
Point Cloud	PointNet++ [15]	78.2	89.2	89.7	89.2
	PointConv [16]	76.8	91.3	89.7	89.2
	DGCNN [19]	80.6	92.6	90.2	87.4
	SPIL [17]	89.3	98.5	96.8	94.5

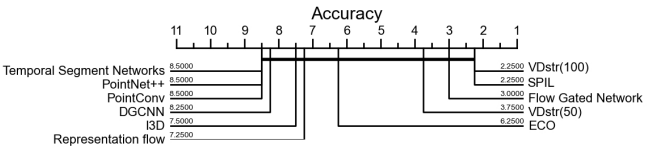


Fig. 3: Critical Difference Diagram (results from all datasets).

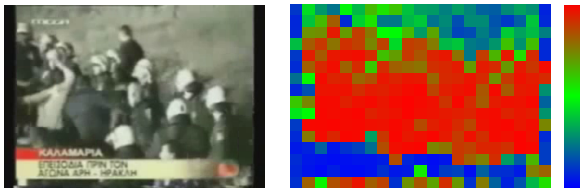
Here, the model is trained only once.

The training of SqueezeNet is performed by using *Adam* optimizer with a learning rate of 10^{-5} and default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$). We set a batch size of 32 and the training is stopped by an early stopping technique with a patience number of 10. The initial weights of the CNN are those obtained when training on IMAGENET dataset in the classification problem and we fine tune with our data.

3.3. Results and discussion

We start with a preliminary study on the RWF2000 dataset by varying the length L of the P_{rep}. The chosen lengths are 50, 100 and 224. Figure 2 illustrates the obtained scores with the different N_b^{test}. $L = 224$ provides the lowest scores, then comes $L = 50$ and the best results are those with $L = 100$. The parameter L has then an important impact on the results. If it is too long or too short, the model losses the ability to learn good features. This can be explained since a P_{rep} representation contains local information. It can be viewed as a temporal pixel enriched with spatial information. We set the length $L = 100$ for testing on the other datasets.

Table 1 provides the VDstr results, with comparisons to state-of-the-art methods (from [17]). We remark that all obtained scores with VDstr reach the 90% and stay always ranked between first and third positions by comparing with other methods. The obtained scores, on Movies Fights and



(a) (b)

Fig. 4: Classification results on a crowd video (Crowd Violence): (a) A frame example; (b) Violence probability map (colorscale: from red (high violence) to blue (no violence)).

RWF2000 datasets, outperform the others as the whole video is processed as one temporal segment. But our results are outperformed in Hockey Fights and Crowd Violence. In Crowd Violence, the length of the videos is not uniform and videos contain a lot of persons, then this increases the scene complexity. In Hockey Fights, the movement of the camera is not stable, rapid with non violence simulating movement in the background and slow when violent events occur. Besides, the scales of the scene may largely differ among the videos. To compare the methods in a more general way, we use a Critical Difference Diagram (Figure 3). From this diagram, VDstr is among the best methods. With P_{rep} of length $L = 100$, it performs as well as the SPIL method. Even when decreasing N_b^{test} in the decision making to 50, the method still ranks well. To conclude, the quality of the methods are all equivalent but are differing according to the dataset.

When applying the decision method to P_{rep} densely dispatched on the video domain, a global semantic segmentation highlighting the violence domain can be obtained. An example is provided as a heatmap in Figure 4: the colorscale ranges from from red color, indicating violent areas (probability of P_{rep} being violent) to blue (non-violence).

4. CONCLUSION

We presented in this article the VDstr method designed to classify videos according to violence content. It is suited for video with a length up to ten seconds. The cornerstone is to represent the video as several planar representations containing both spatial and temporal information. The local classification task can be performed by using any classical $2D$ CNN and the trained filters have the ability to extract directly spatio-temporal features. To achieve good performance, we benefit of a pre-trained CNN, e.g. trained on ImageNet on a similar classification problem, thank to this $2D$ P_{rep} representation, whereas it is not possible with $3D$ approaches.

As perspective, we plan to further investigate on the definition of violent P_{rep}; we can use the ROI for both violent and non violent videos, or add a third class as the background to characterize two types of non-violent zones. We can also analyze the attention of the P_{rep} on the temporal domain in order to segment more precisely the violent event.

5. REFERENCES

[1] H. Liu, S. Chen, and N. Kubota, "Intelligent video systems and analytics: A survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1222–1233, 2013.

[2] E. B. Nievas, O. Déniz-Suárez, G. B. García, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *CAIP*.

[3] L. Xu, C. Gong, J. Yang, Q. Wu, and L. Yao, "Violent video detection based on MoSIFT feature and sparse coding," in *ICASSP*, 2014, pp. 3538–3542.

[4] Y. I. T. Hassner and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *CVPR workshops*, 2012, pp. 1–6.

[5] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu, "Violence detection using oriented violent flows," *Image and Vision Computing*, vol. 48-49, pp. 37–41, 2016.

[6] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[7] P. Bilinski and F. Bremond, "Human violence recognition and detection in surveillance videos," in *AVSS*, 2016, pp. 30–36.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014, pp. 1725–1732.

[9] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015, pp. 4489–4497.

[10] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *CVPR*, 2017, pp. 4724–4733.

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016, pp. 20–36.

[12] M. Cheng, K. Cai, and M. Li, "Rwf-2000: An open large scale video database for violence detection," in *ICPR*, 2020, pp. 4183–4190.

[13] A. J. Piergiovanni and M. S. Ryoo, "Representation flow for action recognition," in *CVPR*, 2019, pp. 9945–9953.

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017, pp. 77–85.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NIPS*, 2017, pp. 5099–5108.

[16] W. Wu, Z. Qi, and F. Li, "Pointconv: Deep convolutional networks on 3d point clouds," in *CVPR*, 2019, pp. 9621–9630.

[17] Y. Su, G. Lin, J. Zhu, and Q. Wu, "Human interaction learning on 3d skeleton point clouds for video violence recognition," in *ECCV*, vol. 12349, pp. 74–90.

[18] H. Fang, S. Xie, Y. Tai, and C. Lu, "RMPE: regional multi-person pose estimation," in *ICCV*, 2017, pp. 2353–2362.

[19] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 146:1–146:12, 2019.

[20] M. Chelali, C. Kurtz, A. Puissant, and N. Vincent, "Classification of spatially enriched pixel time series with convolutional neural networks," in *ICPR*, 2020, pp. 5310–5317.

[21] F. Iandola, M. Moskewicz, and K. e. a. Ashraf, "Squeezenet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size," *arXiv*, vol. abs/1602.07360, 2016.

[22] M. Zolfaghari, K. Singh, and T. Brox, "ECO: efficient convolutional network for online video understanding," in *ECCV*, 2018, pp. 713–730.