2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 11-14, 2020. Toronto, Canada

Violence Detection in Videos using Deep Recurrent and Convolutional Neural Networks

Abdarahmane Traoré and Moulay A. Akhloufi, *Senior Member IEEE*
Perception, Robotics, and Intelligent Machines Research Group (PRIME)
Department of Computer Science, Université de Moncton
Moncton, NB, Canada
{eat4651, moulay.akhloufi}@umoncton.ca

***Abstract*—Violence and abnormal behavior detection research have known an increase of interest in recent years, due mainly to a rise in crimes in large cities worldwide. In this work, we propose a deep learning architecture for violence detection, which combines both recurrent neural networks (RNNs) and 2-dimensional convolutional neural networks (2D CNN). In addition to video frames, we use optical flow computed using the captured sequences. CNN extracts spatial characteristics in each frame, while RNN extracts temporal characteristics. The use of optical flow allows to encode the movements in the scenes. The proposed approaches reach the same level as state-of-the-art techniques and sometimes surpass them. The techniques were validated on three databases achieving very interesting results.**

***Index Terms*—CNN, GRU, Optical Flow, Abnormal behavior detection, Violence detection, Video classification.**

I. Introduction

With a growing population and expanding cities, we are facing an unprecedented rise of criminality [23]. Monitoring systems where a human watches multiple screens to detect violence, theft, or other abnormal behaviors are becoming obsolete. It is hard for persons to focus for a long time to detect violence when they must monitor large crowds. The developed computer vision methods are less effective because of the large volume of data that must be processed. Indeed characteristics resulting from those methods are extracted manually, processed, and then classified by an algorithm. This extraction takes time and becomes almost impossible to perform with a large dataset. With the progress in artificial intelligence, several methods have been developed to detect violence. In fact, it is possible to train convolution neural networks (CNN) to extract spatial and temporal features from a video to classify its content.

In this work, we introduce end-to-end deep learning methods using RGB frames and an optical flow with a CNN-LSTM network to detect violent scenes in videos. The architectures presented reach the same level as modern techniques and sometimes surpassed them. Our approaches have been tested on three public databases to validate their performances.

Thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number RGPIN-2018-06233.

II. Related works

Many methods of violence detection have been proposed in recent years. We can classify these techniques into two categories, classical machine learning and deep learning.

A. Machine learning approaches

Machine learning methods are based on algorithms such as k-nearest neighbors (KNNs) [3], support vector machines (SVMs) [4], or random forests. In addition to the popular descriptors like MoSIFT [37] or VIF [15] to extract features, various extraction methods have been developed to detect violence. For the detection of violence, some approaches use motion blobs. In [14], Gracia *et al.* assume that violence scenes have a position and a shape. The method consists of computing the difference between two consecutive frames then binarize the result to obtain the number of motion blobs. The largest motion blobs are marked on a fight sequence and a no fight sequence. Only k motion blobs are selected. The k blobs are categorized by parameters such as centroid, area, or perimeter between blobs. The selected k blobs are then classified using SVMs, KNNs, and random forests. Motion blobs outperform methods like MoSIFT, SIFT, VIF, and LMP in term of accuracy on popular datasets. In [25], Senst *et al.* proposed a specialized method based on the Lagrangian theory to automatically detect violent scenes. A new spatio-temporal model based on the Lagrangian direction fields was used to capture the features. This new model exploits motion background compensation, appearances, and long-term motion information. The experiment was conducted on three databases, Hockey dataset, Movies dataset, and violent crowd dataset. The new Lagrangian model trained with an SVM performs better than methods such as ViF and HOG + BoW. In [12], Gao *et al.* proposed Oriented Violent Flows (OViF), a statistical motion orientations that takes full advantage of the motion magnitude change information. AdaBoost is used to choose the features that are then trained by an SVM. Tests have been performed on Hockey dataset and Violent Flow, and the results are superior to those of baselines such as

LTP and ViF. In [36], Xia *et al.* present a method that uses a bi-channels CNN and an SVM. First, bi-channels CNN is used to extract two types of features. The first one represents the features of appearance, and the second one represents the difference between adjacent frames. The two features are then classified using SVM. The methods were tested on two databases, Hockey, and Violent crowd datasets. The results are superior to methods like HOG, HOF, MoSIFT, SIFT.

B. Deep learning approaches

Methods of violence scenes detection with Deep Learning are generally based on Deep Convolutional Neural Networks (DCNN).

One approach to detect violence scenes is to use 3D CNN. These algorithms are computationally expensive but are accurate [10], [20], [28], [35]. For example, Song *et al.* [28] propose the use of 3D CNN to extract both spatial and temporal features. The CNN is based on C3D introduced by Tran *et al.* [33]. The CNN consists of eight 3D convolutional layers and five 3D pooling layers. They have also introduced a new frame sampling method based on the gray centroid to reduce frame redundancy caused by uniform sampling. This 3D CNN method has been validated on violence flow, Hockey dataset, and Movies dataset. In [35], Ullah *et al.* propose a three-stage deep learning approach for violence detection. First, a detection algorithm is used to track people in surveillance images, and its output is feed to a 3D CNN where spatio-temporal features are extracted. The results of this 3D CNN are then sent to a softmax classifier for violence detection. The 3D CNN is based on C3D [33] and is composed of eight convolutional and four pooling layers. It has been tested on Violent Flow, Hockey dataset, and Movies dataset. It ranks above the classical methods but is not the best of the Deep Learning methods.

There are also other algorithms that combine two types of neural networks (NN). For example, a 2D time distributed CNN in order to extract the spatio-temporal characteristics and an RNN to refine these characteristics. This end-to-end approach is computationally efficient and gives interesting results, as mentioned in [2]. As features extractor Simonyan *et al.* use VGG19 [26] pre-trained on ImageNet. The extracted features are fed to an LSTM whose output will be passed to a time distributed fully connected layer in order to detect violence. The approach was tested on Violent Flow, Hockey dataset, and Movies dataset with results close to the state-of-the-art. Most algorithms which are based on CNN and RNN use feature extraction followed by a Long Short-Term Memory (LSTM) [16]. There are three types of features extraction techniques, first the standard method that feeds a whole frame to a CNN such as in [30], second a more sophisticated approach such as in [11] that divides a video frame into patches and extracts characteristics from each patch using a CNN. This method of patch division allows to avoid the loss of discriminatory elements caused by differences in scale and location of persons in the frames. Finally, techniques that use CNNs which can provide multiscale features. In [11], Ditsanthia *et al.* use a method called multiscale convolutional feature extraction to manage videos with different scales, and feed the LSTMs with these multiscale extracted characteristics.

Some methods use other types of data in addition to RGB frames. Most often, it is optical flow. The optical flow makes it possible to encode the movement. Adding this information to the RGB frames allows to get better performance. To combine RGB and optical flow two subnetworks are trained, one on RGB frames and the other on optical flow, their outputs are then combined for classification, this approach is used in [38].

Finally, we have the methods that use special LSTMs called ConvLSTMs. These are LSTMs whose matrix operations have been replaced by convolutions. ConvLSTM enable to capture the temporal and spatial characteristics [22], [29]. In [29] Sudhakaran *et al.* use this special convolution to classify actions. Features are first extracted by a CNN, aggregated using a ConvLSTM and then classified.

III. Proposed Method

We use 2D CNN, distributed in time to capture temporal and spatial characteristics. Capturing the temporal and spatial features is relevant because the spatial aspect allows focusing on the identification of spatial patterns of an image, while the temporal aspect deals with how these spatial characteristics evolve over time. For example, the action of punching is a well-defined sequence of actions in time captured by both spatial and temporal features. We combine this 2D CNN with a bi-directional RNN (GRU or LSTM) to improve our detections. We also use optical flow to encode movement between frames for better performance. The optical flow is extracted using a deep CNN model. The network is composed of two identical specialized blocks (figure 1), one for the RGB frames, and the other for the optical flow. The characteristics from these two networks are then added together, refined by an RNN, and then classified using a fully connected layer with sigmoid activation.

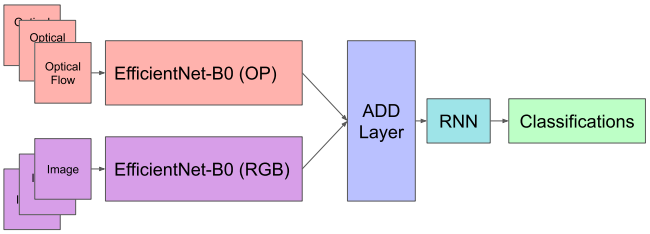

Fig. 1. Proposed architecture pipeline

A. Convolutional Neural Network

As a convolutional neural network, we have selected EfficientNet [32], which is characterized by its compound scaling principle and its efficiency during inference. There are eight versions of EfficientNet (B0-B7), the network consists of MBBLOCKS [17] from MobileNets which are associated with a squeeze and excitation blocks [18]. Figure 2 illustrates the MBCONV (MBBLOCK + squeeze and excitation block) used

by EfficientNet. We used EfficientNet B0 (figure 3) pre-trained on ImageNet [8].

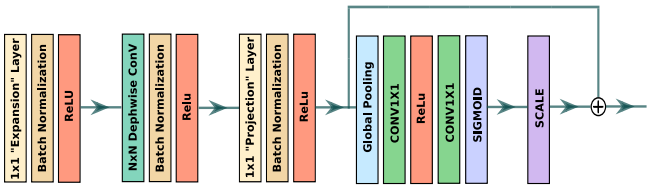

Fig. 2. MBCONV block of EfficientNet

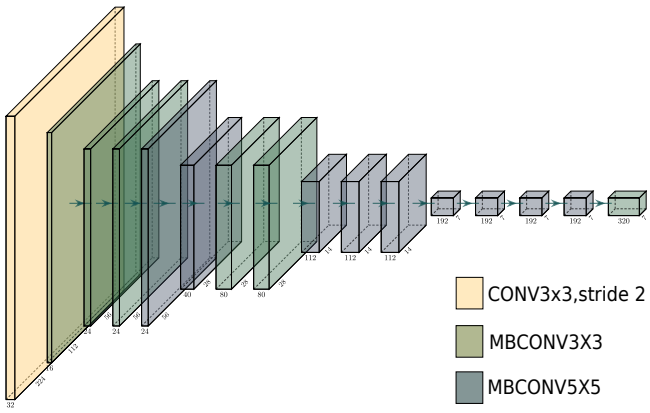

Fig. 3. EfficientNetB0 used to capture spatial features

B. Long Short-Term Memory

The first RNN used to get a temporal representation is LSTM. This RNN has three gates: input gate, forget gate, and output gate. The input gate controls the amount of information that enters in the cell, the forget gate controls the flow of information that remains in the cell and the output gate controls the information that will be used to calculate the output activation of the LSTM unit.

Equation 1 illustrates the LSTM used in this work [13]. f_t represent the forget gate, i_t the input gate, o_t the output gate. x_t is the input vector to the LSTM. c_t and $\tilde{c}_t$ are respectively the cell state vector and the cell input activation vector. h_t is the hidden state vector, W and U are weight matrices that will be learned during the training and b is the bias. σ_g is a sigmoid activation function, and σ_h is a hyperbolic tangent. Furthermore, we use our LSTM in a bidirectional mode to consider not only the preceding sequences but also the following sequences, which helps in getting a better performance.

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\ \tilde{c}_t &= \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\ h_t &= o_t \circ \sigma_h(c_t) \end{aligned} \tag{1}$$

C. Gated Recurrent Unit

The GRU's option is made to prevent the LSTM's problem of a vanishing gradient. For some tasks, GRU is more robust to noise and outperforms the LSTM [6]. The GRUs are less computationally intensive because unlike the LSTM which has three gates, they have only two.

Equation 2 provides the GRU functions used in this work [9]. Z_t, update gate determines what information to retain or drop, the r_t reset gate determines how much past knowledge to forget, and h_t is the output gate. W, U, and b are matrix and vector parameters, σ_g is a sigmoid activation function, and σ_h is a hyperbolic tangent. The input vector is x_t. We use GRU in bidirectional mode to consider both following and preceding information.

$$\begin{aligned} Z_t &= \sigma_g(W_z x_t + U_z h_{h-1} + b_z) \\ r_t &= \sigma_g(W_r x_t + U_z h_{h-1} + b_z) \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + \\ & U_h(t_t \circ h_{t-1}) + b_h) \end{aligned} \tag{2}$$

D. Optical Flow

Optical flow is a method of perceiving movement in a sequence of images (videos) that can be computed in different ways, in our case, we opted for the use of PWC-Net [31]. PWC-Net is an approach which was proposed in order to obtain a smaller network than FLowNet2 [19] and is also more efficient in term of accuracy by adding domain knowledge into the design of the network (see figure 4). To compute the optical flow, PWC-Net first uses a learnable features pyramid to counteract the variations of shadows and brightness in the raw image. Then a warping operation is performed to capture large motions. After this warping operation, the features are passed to a layer that computes the cost volume. Cost volume is a more discriminating representation of the optical flow than the image. The representation from the cost volume layer is then processed by CNN to estimate the flow. Since warping and cost volume have no learnable parameters, it reduces the size of the model. At the end of the pipeline, a post-processing of the context information is done by using a network to refine the optical flow. We used PWC-Net pre-trained on MPI Sintel dataset [5] to extract the optical flow for our datasets.

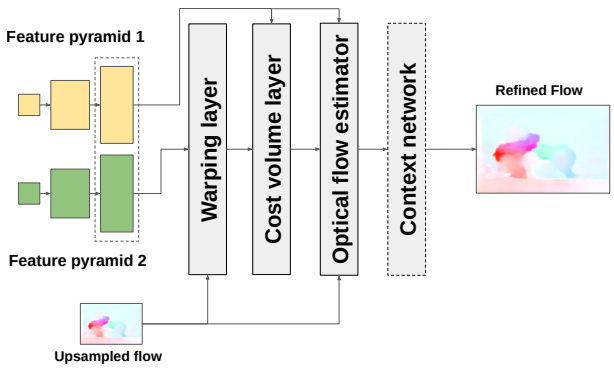

Fig. 4. PWC-NET architecture

IV. Experiments and Results

We used three datasets to test our network: Hockey dataset [24], Violent Flow dataset [15], and Real Life Violence Situations Dataset [21]. We used the accuracy metric to measure the performance of our networks.

A. Datasets

Datasets were randomly divided into two groups, training (80%) and validation/testing (20%).

1) Hockey Dataset: Hockey Fight Dataset [24] is a 2000 videos Dataset with 1000 fight and 1000 no fight videos of Hockey (figure 5). Clips last approximately 2 seconds and consist of approximately 41 frames with a resolution of 360x288. The details of the sequences are quite similar. We have resized the clips frames to 128x128.

Fig. 5. Frames from Hockey Dataset, on the left we have the first frame, in the middle we have the second frame, on the right we have the optical flow computed using the first and the second frame.

2) Violent Flow Dataset: Violent Flow Dataset [15] is a dataset containing real-world video recordings of crowd violence (see figure 6). There are 246 videos in the dataset. The shortest clip length is 1.04 seconds, the longest clip is 6.52 seconds and the average length of the clip is 3.60 seconds. We have also resized the frames to 128x128.

Fig. 6. Frames from Violent Flow, on the left we have the first frame, in the middle we have the second frame, on the right we have the optical flow computed using the first and the second frame.

3) Real Life Violence Situations Dataset: Real Life Violence Situations Dataset [21] is a dataset of violence video clips from various situations of real life (see figure 7). It contains 1000 fight and 1000 no fight sequences. We resized the images to 128x128.

Fig. 7. Frames from Real Life Violence Situations Dataset, on the left we have the first frame, in the middle we have the second frame, on the right we have the optical flow computed using the first and the second frame.

B. Parameters and sampling

We used Keras [7] with the TensorFlow [1] backend to set up our networks. Having different lengths of videos, we decided to choose a fixed number of 12 frames for each dataset. To choose these frames, we made a uniform sampling that allowed at the same time to avoid unnecessary computation of the network caused by redundant frames. For the computation of optical flow, we used RGB frames that we have sampled and the frames that follow them. For example, for the computation of the optical flow for frame 6, we used frame 6 and frame 7. To go further in our experimentation, we have also made jumps of 2 and then 3 frames from the current RGB frame, for example, for frame 6, we have also calculated the optical flow with frame 8 and frame 9. Our EfficientNet-B0 was pre-trained on ImageNet and then combined with RNNs, followed by 3 fully connected layers. We named this architecture ValdNet. There is three versions of ValdNet. ValdNet1, ValdNet2 and ValdNet3, the number in the name indicates the interval between frames when calculating the optical flow. Each version of ValdNet can have a GRU version and an LSTM version.

ValdNet was trained on all the datasets using rmsprop as optimizer with a batch size of 4. The learning rate was set to 0.001. The networks were trained for 50 epochs.

C. Results and discussions

Tables I, II and III present our results on the 3 datasets, Hockey, Violent Flow and Real Life Violence Situations. The tables are composed of classic methods (Machine Learning) and Deep Learning methods. The classical methods have different inputs such as bag of words or histogram bins.

On Hockey dataset, we obtained an accuracy of 99% with ValdNet1 (GRU), results close to [28] with 99.62%. On Violent Flow, we obtained an accuracy of 93.53% with ValdNet3 (LSTM), lower than our previous work with 95.00% based on VGG+GRU without optical flow [34]. In the visualization of the optical flow for Violent Flow (figure 6), we can see its difficulty to correctly detect movements, which limits its contribution to the classification of violence. On Real Life Violence Situations Dataset, we surpassed the result of our previous work [34] with all versions of ValdNet. Our highest

result is 96.74% of accuracy, 6.24% better than our previous experiment, and the proposed method in [27], which obtained 86.39%.

TABLE I
HOCKEY DATASET VIOLENCE DETECTION, COMPARISON WITH OTHER METHODS (W REPRESENT WORDS, O IS OPTICAL FLOW, H REPRESENT HISTOGRAM BINS, F ARE FRAMES, C IS A CXC WINDOW, "?" MEANS NOT SPECIFIED AND "-" MEANS NOT USED)

Method	Inputs	Sampling	Accuracy (%)
AdaBoost + SVM [12]	20 (H)	-	87.50
MoWLD + Sparse Coding [28]	1000 (W)	-	93.70
MoSIFT + KDE + Sparce Coding [28]	500 (W)	-	94.30
LaSift [25]	500 (W)	-	94.42
MoWLD + KDE + Sparce Coding [28]	500 (W)	-	94.90
MoIWLD + KDE + SRC [28]	1800 (W)	-	96.80
3D-CNN [28]	16 (F)	Uniform	91.00
Bi-channels CNN [36]	?	Uniform	95.90
3D ConvNet [35]	16 (F)	Intervall of 8	96.00
ValdNet2 (GRU)	12 (F+O)	Uniform	96.00
FightNet [39]	25 (F)	Random	97.00
VGG19+LSTM [2]	40 (F)	?	97.00
ValdNet3 (GRU)	12 (F+O)	Uniform	97.00
ConvLSTM [29]	20 (F)	Custom	97.10
ValdNet1 (LSTM)	12 (F+O)	Uniform	98.00
VGG16 + GRU [34]	10 (F)	Uniform	98.00
ValdNet2 (LSTM)	12 (F+O)	Uniform	98.00
ValdNet3 (LSTM)	12 (F+O)	Uniform	98.00
3D ConvNet [28]	16 (F)	Uniform	98.96
ValdNet1 (GRU)	12 (F+O)	Uniform	**99.00**
3D ConvNet [28]	32 (F)	Uniform	**99.62**

TABLE II
VIOLENT FLOW DATASET VIOLENCE DETECTION, COMPARISON WITH OTHER METHODS (W, H, F, O, C, "?" AND "-" ARE THE SAME AS IN TABLE I)

Method	Inputs	Sampling	Accuracy (%)
MoWLD + BoW [28]	500 (W)	-	82.56
RVD [28]	4x4 (C) /5 frames	-	82.79
MoWLD + Sparse Coding [28]	500 (W)	-	86.39
VGG19+LSTM [2]	40 (F)	?	85.71
AdaBoost + SVM [12]	20 (H)	-	88.00
MMoSIFT + KDE + Sparce Coding [28]	500 (W)	-	89.05
MMoWLD + KDE + Sparce Coding [28]	500 (W)	-	89.78
ValdNet2 (GRU)	12	Uniform	91.66
ValdNet2 (LSTM)	12	Uniform	91.66
ValdNet1 (LSTM)	12	Uniform	91.66
LaSift [25]	500 (W)	-	93.12
MoWLD + KDE +SRC [28]	1800 (W)	-	93.19
Bi-channels CNN [36]	?	Uniform	93.25
3D ConvNet [28]	?	Uniform	93.50
ValdNet1 (GRU)	12 (F+O)	Uniform	93.75
ValdNet3 (GRU)	12 (F+O)	Uniform	93.75
ValdNet3 (LSTM)	12 (F+O)	Uniform	**93.75**
ConvLSTM [29]	20 (F)	?	94.57
VGG16 + GRU [34]	10 (F)	Uniform	**95.50**

TABLE III
VIOLENT FLOW DATASET VIOLENCE DETECTION, COMPARISON WITH OTHER METHODS (W, H, F, O, C, "?" AND "-" ARE THE SAME AS IN TABLE I)

Method	Inputs	Sampling	Accuracy (%)
VGG16 + LSTM [27]	? (F)	-	86.39
VGG16 + GRU [34]	10 (F)	Uniform	90.50
ValdNet3 (LSTM)	12 (F + O)	Uniform	93.75
ValdNet2 (LSTM)	12 (F + O)	Uniform	93.99
ValdNet1 (GRU)	12 (F + O)	Uniform	94.74
ValdNet3 (GRU)	12 (F + O)	Uniform	95.49
ValdNet1 (LSTM)	12 (F + O)	Uniform	96.24
ValdNet2 (GRU)	12 (F + O)	Uniform	**96.74**

V. CONCLUSION AND FUTURE WORKS

We used 2D time distributed CNN to capture spatio-temporal features, and we refined them using RNN. Also, we used two specialized sub-networks, one for RGB images and the other for optical flow, which outputs we have summed to encode the motion from the optical flow into our features. These combinations have allowed increasing our results in the proposed dataset. We achieved an accuracy of 99%, 93.75%, and 96.74% respectively on Hockey dataset, Violent Flow, and Real Life Violence Situations Dataset. We are second on Hockey dataset behind [28] by 0.62 difference. On Violent Flow, the presence of several people seems to limit the performance of the optical flow. The low resolution of the optical flow does not allow to clearly determine what people are doing in the scene (figure 6), our previous work without optical flow is still the best performing [34]. Real Life Violence Situations Dataset is a fairly new database, so there is no other test for the moment except the baseline [27], which we outperformed by more than 10%. For our future work, we will introduce other datasets in order to benchmark more databases and techniques. We will also benchmark the performance in terms of flops and speed of inference of the best performing techniques.

REFERENCES

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org

[2] Abdali, A.M.R., Al-Tuma, R.F.: Robust Real-Time Violence Detection in Video Using CNN And LSTM. In: 2019 2nd Scientific Conference of Computer Sciences (SCCS). pp. 104–108 (Mar 2019)

[3] Altman, N.S.: An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician **46**(3), 175–185 (Aug 1992). https://doi.org/10.1080/00031305.1992.10475879, https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879

[4] Bernhard Schölkopf, A.J.S.: IEEE Xplore Book Abstract - Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (2002), https://ieeexplore.ieee.org/book/6267332

[5] Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: A. Fitzgibbon et al. (Eds.) (ed.) European Conf. on Computer Vision (ECCV). pp. 611–625. Part IV, LNCS 7577, Springer-Verlag (Oct 2012)

[6] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078 [cs, stat] (Sep 2014), http://arxiv.org/abs/1406.1078, arXiv: 1406.1078

[7] Chollet, F., et al.: Keras. https://github.com/fchollet/keras (2015)

[8] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)

[9] Dey, R., Salem, F.M.: Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks. arXiv:1701.05923 [cs, stat] (Jan 2017), http://arxiv.org/abs/1701.05923, arXiv: 1701.05923

[10] Ding, C., Fan, S., Zhu, M., Feng, W., Jia, B.: Violence Detection in Video by Using 3D Convolutional Neural Networks. In: ISVC (2014)

[11] Ditsanthia, E., Pipanmaekaporn, L., Kamonsantiroj, S.: Video Representation Learning for CCTV-Based Violence Detection. In: 2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON). pp. 1–5 (Dec 2018)

[12] Gao, Y., Liu, H., Sun, X., Wang, C., Liu, Y.: Violence detection using Oriented VIolent Flows. Image and Vision Computing **48-49**, 37–41 (Apr 2016). https://doi.org/10.1016/j.imavis.2016.01.006, https://linkinghub.elsevier.com/retrieve/pii/S0262885616300063

[13] Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to Forget: Continual Prediction with LSTM. Neural Computation **12**(10), 2451–2471 (Oct 2000). https://doi.org/10.1162/089976600300015015, http://www.mitpressjournals.org/doi/10.1162/089976600300015015

[14] Gracia, I.S., Suarez, O.D., Garcia, G.B., Kim, T.K.: Fast Fight Detection. PLOS ONE **10**(4), e0120448 (Apr 2015). https://doi.org/10.1371/journal.pone.0120448, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0120448

[15] Hassner, T., Itcher, Y., Kliper-Gross, O.: Violent flows: Real-time detection of violent crowd behavior. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–6 (Jun 2012)

[16] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, https://doi.org/10.1162/neco.1997.9.8.1735

[17] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861 [cs] (Apr 2017), http://arxiv.org/abs/1704.04861, arXiv: 1704.04861

[18] Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-Excitation Networks. arXiv:1709.01507 [cs] (May 2019), http://arxiv.org/abs/1709.01507, arXiv: 1709.01507

[19] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. arXiv:1612.01925 [cs] (Dec 2016), http://arxiv.org/abs/1612.01925, arXiv: 1612.01925

[20] Li, C., Zhu, L., Zhu, D., Chen, J., Pan, Z., Li, X., Wang, B.: End-to-end Multiplayer Violence Detection Based on Deep 3d CNN. In: Proceedings of the 2018 VII International Conference on Network, Communication and Computing. pp. 227–230. ICNCC 2018, ACM, New York, NY, USA (2018), event-place: Taipei City, Taiwan

[21] Mohamed, E., Mohamad, H., Massih, M.A.E.: Real Life Violence Situations Dataset. https://kaggle.com/mohamedmustafa/real-life-violence-situations-dataset

[22] Morales, G., Salazar-Reque, I., Telles, J., Díaz, D.: Detecting Violent Robberies in CCTV Videos Using Deep Learning. In: MacIntyre, J., Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) Artificial Intelligence Applications and Innovations, vol. 559, pp. 282–291. Springer International Publishing, Cham (2019)

[23] MT, S.: Increasing Crimes vs. Population Density in Megacities. Sociology and Criminology-Open Access **4**(1), 1–2 (2016)

[24] Nievas, E.B., Suarez, O.D., Garcia, G.B., Sukthankar, R.: Hockey fight detection dataset. In: Computer Analysis of Images and Patterns. pp. 332–339. Springer (2011)

[25] Senst, T., Eiselein, V., Kuhn, A., Sikora, T.: Crowd Violence Detection Using Global Motion-Compensated Lagrangian Features and Scale-Sensitive Video-Level Representation. IEEE Transactions on Information Forensics and Security **12**(12), 2945–2956 (Dec 2017). https://doi.org/10.1109/TIFS.2017.2725820

[26] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs] (Apr 2015), http://arxiv.org/abs/1409.1556, arXiv: 1409.1556

[27] Soliman, M.M., Kamal, M.H., El-Massih Nashed, M.A., Mostafa, Y.M., Chawky, B.S., Khattab, D.: Violence Recognition from Videos using Deep Learning Techniques. In: 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS). pp. 80–85 (Dec 2019). https://doi.org/10.1109/ICICIS46948.2019.9014714

[28] Song, W., Zhang, D., Zhao, X., Yu, J., Zheng, R., Wang, A.: A Novel Violent Video Detection Scheme Based on Modified 3d Convolutional Neural Networks. IEEE Access **7**, 39172–39179 (2019)

[29] Sudhakaran, S., Lanz, O.: Learning to detect violent videos using convolutional long short-term memory. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6 (Aug 2017)

[30] Sumon, S.A., Shahria, M.T., Goni, M.R., Hasan, N., Almarufuzzaman, A.M., Rahman, R.M.: Violent Crowd Flow Detection Using Deep Learning. In: Nguyen, N.T., Gaol, F.L., Hong, T.P., Trawiński, B. (eds.) Intelligent Information and Database Systems, vol. 11431, pp. 613–625. Springer International Publishing, Cham (2019)

[31] Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. arXiv:1709.02371 [cs] (Jun 2018), http://arxiv.org/abs/1709.02371, arXiv: 1709.02371

[32] Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv:1905.11946 [cs, stat] (May 2019), http://arxiv.org/abs/1905.11946, arXiv: 1905.11946

[33] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning Spatiotemporal Features with 3D Convolutional Networks. arXiv:1412.0767 [cs] (Oct 2015), http://arxiv.org/abs/1412.0767, arXiv: 1412.0767

[34] Traoré, A., Akhloufi, M.A.: 2D Bidirectional Gated Recurrent Unit Convolutional Neural Networks for End-to-End Violence Detection in Videos. In: Campilho, A., Karray, F., Wang, Z. (eds.) Image Analysis and Recognition. pp. 152–160. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978.3-030-50347-5_14

[35] Ullah, F.U.M., Ullah, A., Muhammad, K., Haq, I.U., Baik, S.W.: Violence detection using spatiotemporal features with 3D convolutional neural network. Sensors (Basel, Switzerland) **19**(11), 2472 (May 2019)

[36] Xia, Q., Zhang, P., Wang, J., Tian, M., Fei, C.: Real Time Violence Detection Based on Deep Spatio-Temporal Features. In: Zhou, J., Wang, Y., Sun, Z., Jia, Z., Feng, J., Shan, S., Ubul, K., Guo, Z. (eds.) Biometric Recognition. pp. 157–165. Lecture Notes in Computer Science, Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-97909-0_17

[37] Xu, L., Gong, C., Yang, J., Wu, Q., Yao, L.: Violent video detection based on MoSIFT feature and sparse coding. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 3538–3542 (2014)

[38] Xu, X., Wu, X., Wang, G., Wang, H.: Violent Video Classification Based on Spatial-Temporal Cues Using Deep Learning. In: 2018 11th International Symposium on Computational Intelligence and Design (ISCID). vol. 01, pp. 319–322 (Dec 2018)

[39] Zhou, P., Ding, Q., Luo, H., Hou, X.: Violent interaction detection in video based on deep learning. Journal of Physics: Conference Series **844**, 012044 (06 2017)