Automatic Detection of Riots Using Deep Learning

Mayur Jadhav and V. A. Chakkarwar

November 18, 2019

"Automatic Detection of Riots Using Deep Learning"

Mayur K. Jadhav[1] and V. A. Chakkarwar[1]

[1]Department of Computer Science & Engineering,
Government College of Engineering, Aurangabad.
mj8055x@gmail.com

Abstract. Riot like situation styles serious consequence on societal and individual security and a simple primary cautioning of any fierce, vehement, vicious, force-full or violent movement could significantly condense these dangers. At present, there be located oodles of video surveillance kit applied in civic places, such as bus-stations, highways, airports, Signal-squares, crossings, and railway_stations. It is of vivacious importance to examine the detrimental anomalous fillings from enormous quantities of surveillance-video data. For the thought of real-world application, this effort focuses on the challenging task of detecting violent situations in videotapes & aims to proposition a fangled way that could automatically distinguish violent behaviours by resources of computer vision methods. For our system, the main objective will be to detect furious activities seamlessly from video streams or recorded video-clips. We take a proportion of videotapes & we train those precise sequences as violent/non-violent situations & once we have a model ready, we deploy for example on intelligent surveillance camera, any action which is close to this precise entity would be classified as violent situation and we can direct an alarm/warning back to the control-room for further necessary steps with highest possible accuracy.

Keywords: Neural Networks, Deep Learning, Image Processing, Violence, Riots, Surveillance System, CCTV monitoring, Activity detection, Convolutional_Neural_Network, Video, Recurrent_Neural_Network, Long_Short_Term_storage, Support_Vector_Machine, Random Forest, improvised Dense Trajectories, K-Nearest Neighbors, Spacetime volume, Space-Time Interest Points, Motion SIFT, Kernel Density Estimation.

1 Introduction

We stand alive in a digital universe, encircled by electronic devices all over. These devices are designed to assist humans in carrying various tasks easily and efficiently. Surveillance cameras are broadly used and existing all over the world with persistent supervision by humans to check for any anomalies, the main problem ascends with the human part of this, with humanoid supervision we may gain human error along with manipulation possibilities & also the need of a particular experienced human-being in the first place. According to a survey done by British_Security lndustry_Authority (BSIA) [2], the total quantity of CCTV cameras in India could be as high as one for every 51 people. With these numbers ever increasing, the human workforce is clearly inadequate to analyse these videos. Even though CCTVs are very useful for analysing a scene after an event has happened, they are rarely used to detect or predict events. Most of the surveillance videos can be sub-divided into 2 categories:

1. Involving humans - for e.g. classrooms, footpaths, hallways, shops, road crossings etc.
2. Not involving humans - for e.g. highways, parking lots, industries

In this work, we will solely focus on videos involving humans. Our system proposes the detection of violence in a scene gained from surveillance videotape, as these videos do not comprise of any audio tracks the system only can rely on visual features. The idea is to detect crowd-based violence & with crowd arises the issue of too much motion & hence we terminate the use of high-level motion features & analysis & as an alternative, we dive into changes observed in low level features for classification. Short frame-sequences are used to classify the videotapes two ways using deep_learning model. We have used Convolutional_Neural_Network (CNN), Recurrent_Neural_Network (RNN) along with Long Short-Term-storage (LSTM) in different combinations and also various other techniques which eventually made our unique system validate its action detection techniques with super efficiency. The videotapes for experiments are obtained from an annotated public database used in a similar project as ours T. Hassner & Kliper-Gross [2012] as well as from other social media resources such as YouTube for local videos.

2 Past related work

Violence detection is subtask of action recognition can be frame-based or interest-point based, in situation of motion-based interest-points the tricky problematic state arises when there are too few interest-points or like in the cases of crowds too much motion bag of words approach fails immensely. The frame-based method is efficient but uses a search-based approach which is not practical (Too slow) for real time detection. Liu et al. [2009] Dollar et al. [2005] Boiman & Irani Boiman & Irani [2005] proposed an approach that involved categorizing videos as violent by analyzing sudden changes in videos. Hendel et al Hendel et al. [2011] defined a probabilistic method to detect sudden changes by using space-time tubes containing an object moving in the scene. This method is known to under-perform with crowd videos. Another approach is to use dynamic features produced by a stochastic process which are stationary in space & time but crowds are not stationary but recently local_binary_pattern (LBP) have confirmed to be fairly effective & efficient. Crook et al. [2008] Zhao & Pietikainen [2007] Hassner, Yossi & Klipper-Gross T. Hassner & Kliper-Gross [2012] proposed a unique method for riots detection using their unique feature descriptor called ViF (Violent_Flows). They classified surveillance clips as violent/non-violent using ViF-descriptors & Support-Vector-Machines (SVM). In our opinion, their hard-work is by far the best when it comes to making predictions in real time & we strategy to originate motivation, incentive & inspiration from their efforts in our project. Most Recently, some deep_learning_based methods have been discovered in order to recognize actions & activities [30, 19, 18, 25]. Deng et al. projected a deep model [18] to capture distinct actions, pairwise interactions, & group activities. In one more work [19], Deng et al. first estimate the distinct & scene activities which are complementary refined by means of some efficient message_passing algorithm under an outlined_framework of a re_current neural network. In [30], the authors projected a two-staged LSTM model where the first stage captures distinct temporal dynamics trailed by scene activity acknowledgement based on combined discrete information. Furthermost of the current approaches attention on scene activity acknowledgement & overlook the fact that numerous groups with diverse actions are present in the videos. Group level information can be employed for high-level claims such as irregular activity detection & is significant to understand the scene in its completeness. We shape upon our group detector and detect group-activities as well, sideways with scene activities.

3 The Dataset

Creating a good dataset for group and scene activity is a challenging job, since annotations have to be done at various levels. The dataset that is used is an annotated dataset that is a mixture of surveillance data & other in the wild videos acquired from YouTube.

Fig. 1. Non-Violent local Video dataset screens

The complete number of videos is about 1230 with half of them annotated as violent & other as non-violent as seen in Figure 1. The tiniest video is of 1 second & the longest is of 6.52 seconds with an average duration of 3.6 seconds vindicating our method to work with a short numeral of frames. The videotapes are fragmented into 5 dissimilar groupings each exhibiting some type of crowd situation whether a sporting or other social gathering with many people with half displaying acts of violence & the other half displaying normal behaviour. As our idea is to detect riot like behaviour in crowds & perform actions to stop it through surveillance cameras & other forms of monitoring. A rather in-depth motive is to understand crowd behaviour from image data analysis. The actual data is in the video & our model is fed with images that are the frames extracted from the video data with almost a total image_count of about 220000 images with 120000 non_violent & 100000 violent marked images in separate folders marked as labels before pre-processing kicks-in. The ratio of violent to non- violent data points is about 6:5 which is least mildly biased towards the violent data. The training: testing split considered is well thought out & is 80:20.

4 Video & Camera

We have also tested the system with live video input from usb-camera. There are numerous settings which can vary across surveillance video-camera model, the total number of cameras, video resolution, camera motion, the location of recording, proximity to the scene, crowd density & presence of objects like cars to list a few. Before going any further, it's critical to mention all the types of videos we analysed along with all settings, hence only the settings with best output is considered in this research paper.

5 Approach

5.1 Preprocessing

The initialisation part of this approach is to use the video data set as several images. The reason behind it is that the extreme features that are used, are not temporal which that is, the features are a lot suitable for images also the other reason is of data, we begin with a modest number of videos, but converting them to images would let us work with a very rigorous dataset & help the model generalize better. To achieve this OpenCV was used with python scripting & each video was converted into many frames. An advantage that comes along with using images instead of videos is the discrepancy in the length of the videos which if used would have needed normalization that is converting each video to the same length as CNN requires a consistent size of the feature vector. The second step is to select our features from the several images obtained, after contemplating with histogram of orientations which are some spatial features using descriptors like SIFT, but after experimentation we came to the conclusion that using orientation-based features will surely give bad results as the data at hand contain drastic actions which would rattle the descriptors & the number of interest points may be several to very few in number. Thus, we eventually ended up by selecting the extreme intensities of the images as one of our features for the neuraI_network. The sequential next step is to pre_process the data that is extracted for the convolutional_neuraI_network being used further all down the process.Initially, here we used a super_vectorized version of almost all images are of a particular size (320_x_240) also which was humongously large along with our tiny dataset & the network when the training dropped into the difficulties of some memory issues on a machine with hardly enough or fairly good specifications. Thus, because of it, we made the decision of skewing our data by hand, by converting each image used to a size of (224_x_224). This was initially achieved in Matlab & the images saved for further processing. But later on we also implemented it on the go with some efficient video clipping python programs developed for this purpose, intentionally. Also the machine was upgraded to 32Gb Memory later on.

5.2 Architecture Explanation and work-flow

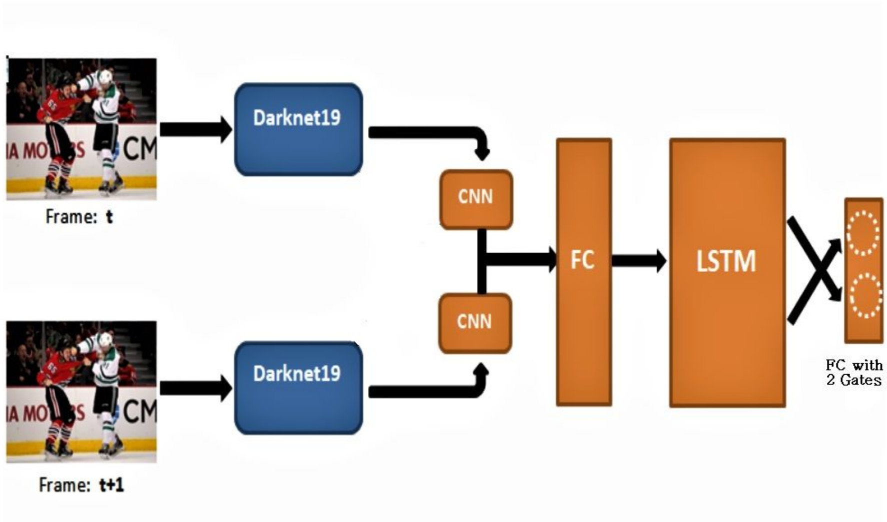

Fig. 2. Proposed Architecture

The proposed architecture of the network is displayed in Figure 2 It has been displayed that the local temporary features that could be obtained from the optical_flow are also important in addition to adding the LSTM (which is supposed to extract universal temporary features) after the CNN[74]. It has also been reported that the virtue of optical_flow is due to its appearance invariance as well as its accuracy at limitations and at small displacements[73]. Therefore, in this work, by taking two video_frames as input, the effect of optical_flow should be mimicked. Before finalizing on this architecture, we experimented with adding more FC layers, but more layers resulted in drop of training accuracy. The pre-trained CNN processes the 2 input frames. The first neural network is a convolutional neural network aimed at extracting high-level image features and reducing input complexity. It uses 16 filters of 2 x 2 size. The output of these filters was forced to be kept the same as the input by padding the borders before convolution. Output from this convolutional_layer was passed through relu_activation & into the max-pooling layer We are using a pre-trained DarkNet model trained on the large visual detection challenge ImageNet dataset. The two frame outputs of the pre-trained model's bottom layer are combined in the very last channel and then fed into the extra-additional CNN (labeled in our Figure 1 by orange color).

Since the output results from the bottom most layer are considered to be the required low-level features, eventually by comparing the 2 frames feature map, the additional CNN should learn both the local motion features and the appearance invariant features. The two frame outputs from the pre-trained network's top layer are also concatenated and fed into the other additional CNN to compare the two

frames ' high-level features.
In order to learn the universal temporary features, the outputs from the two additional CNN are then concatenated and passed to a fully joined layer and the LSTM cell. Lastly, the LSTM cell outputs are classified by a fully joined layer containing two neurons representing the two categories (riots and non-riots), respectively. The blue_colored_layers are pre-trained on the ImageNet dataset and also frozen during its training. On the video-clip dataset, the layers marked by the light-orange color are trained. Due to its exact-accuracy on ImageNet and the mentioned real_time efficiency, Darknet19[32] implements the pre-trained model. Since the Darknet19 already contains 19 convolution layers, the additional CNN is implemented by the residual_layers[29] to ignore the degradation difficulty.
If we did not use any max-pooling layer, the training accuracy would increase but testing accuracy would go down. We used one batch normalizer between the first pooling layer & the second convolutional layer. Batch normalizer makes sure that the input weights & bias to the next layer have 0 mean & unit variance. The primary use of batch normalization is to speed up training process by squashing the range of possible values for weights & bias to a normalized range. This however introduces noise & lowers training accuracy. Normalization can help reduce over-fitting. In our case, our training accuracy was already over 99% & we could do with some normalization to reduce over-fitting. Testing accuracy with & without batch normalization had a difference of about 2% with the model lacking batch normalization having lower accuracy of the two. Finally, we used dropout layer with a dropout value of about 0.5. Dropout works by randomly switching certain proportion of neurons on & the rest off by multiplying by either a 1 or a 0. This process is known to introduce multiplicative noise in training phase. It's again used to combat over-fitting & helps improve testing accuracy. Leaky_Rectified_Linear_Unit (i.e. Leaky_ReLu) The standard equations for our LSTM model are as follows:

$$i_t = \sigma(w_x^i * I_t + w_h^i * h_{t-1} + b^i) \tag{1}$$

$$f_t = \sigma(w_x^f * I_t + w_h^f * h_{t-1} + b^f) \tag{2}$$

$$\tilde{c}_t = tanh(w_x^c * I_t + w_h^c * h_{t-1} + b^c) \tag{3}$$

$$c_t = \tilde{c}_t \odot i_t + c_{t-1} \odot f_t \tag{4}$$

$$o_t = \sigma(w_x^o * I_t + w_h^o * h_{t-1} + b^o) \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

In the above equations, '*' represents convolution operation & '⊙' represents the Hadamard product. The hidden state ht, the memory cell ct & the gate activations it, ft & ot are all 3D tensors in the case of LSTM.

5.3 Experiments

There are 7 different versions for this model & the gained results for the successful experiments are mentioned here, for our versions 1 & 2 the resulting testing accuracy is very bad hence their confusion matrix is not discussed. For version three, the number of epochs was set to 30 & a drop rate of 0.2 was used with no batch normalization which gave a classification rate of 78% on testing data. The true positives for violence data is far fewer than the non-violence positives, with great accuracy achieved for non-violent testing data.
The next version which is version no. 4 gave us extremely good satisfying results with a super classification rate of 82.75% where the selected number of regular epochs were equal to 100 only with a dropout rate of 0.5 & no batch normalization implemented. The violence data in this case gave extremely good results while the results for non-violence data fell down a bit.
Version 4, 5 and 6 gave satisfactory results for the task at hand but we wanted to experiment with batch normalization, CNN+LSTM, CNN+RNN & thus implemented that for version 7 giving us the best results thus far.

6 Results

At the end of the research, the most accurate, convenient and efficient options were used from the variety of options available in each part. Bermejo et. al proposed the dataset for hockey. The model with best result was version 5 which gave a classification rate of 98.52% which will move around the surrounding neighbourhood of values based on the data sequence selected as a random shuffle is carried out to achieve a more grounded result. The training accuracy came to about 98.8% which may suggest the model to be over fitting but as the number of data samples are less comparatively over fitting seemed necessary, while using even more larger data set, overfitting will be unnecessary. Here, we can say that for the training portion the model over fits with zero false positives for violent_data & nominal false negatives for some of the non-violent data. The final result can be seen in Figure 3

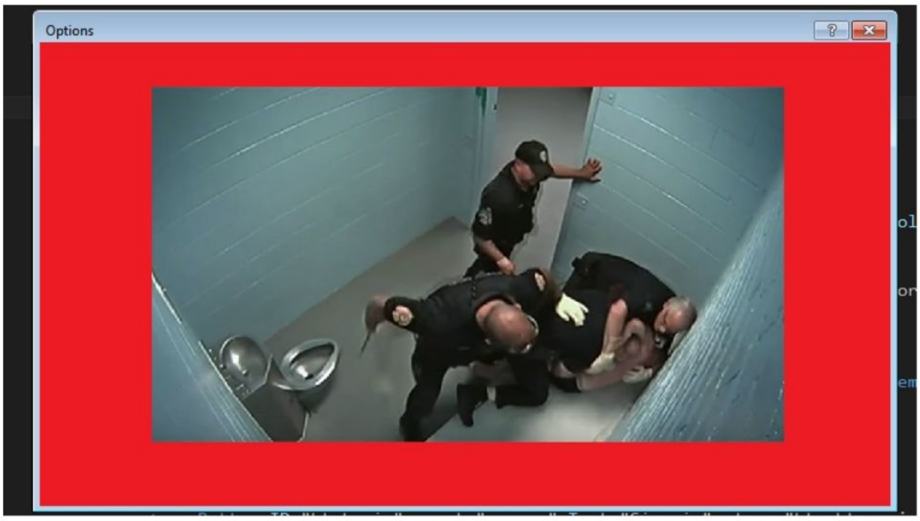

Fig. 3. Output screen of analysed video

Table 1. Comparison Between Two Proposed Models.

Model	Accuracy with Violent-Flows dataset	Accuracy with our datased
MoSIFT+KDE+Sparse coading	89.05±3.26%	91.33%
Three streams+LSTM	93.92%	96.29%
CNN only	92%	94%
Proposed Model CNN+LSTM	97±1.33%	98±0.55%

In table 6.1, we could see that CNN model gives less accuracy than CNN + LSTM. CNN only considers the latest input while proposed model considers the latest input along with the previously received inputs. Because of its internal storage, it could memorize previous inputs.
RNN also handles sequential data and has a short term storage. However, as we are using LSTM, it has a Long_Short_Term storage. Because of LSTM, training takes less time and also has high accuracy. Furthermore it solves the difficulty of gradients disappearing.

7. Conclusion & future work

Crowd visual analysis is an interesting & newly emerging technical field of computer_vision & with increasing amounts of surveillance cameras set up all over the world, detection of crowd behaviour using this type of data is very crucial. The task accomplished here surpasses many research projects in the same domain, but as most of this system are modelled for real-time feedback this result may be not comparable. In conclusion, we can positively say that a high precision riot detection system was implemented using deep learning concepts like convolutional neural network on video data. The proposed network uses pre-trained model on ImageNet (Hybrid Darknet19) dataset which also extracts universal and local temporary features. CNN is efficiently used for frame level feature extraction. The basic idea here, is to extend these outputs of the experiment conducted further & achieve a more realistic & better accuracy by tweaking hyperparameters & also by experimenting with our network layer architecture. In terms of future work, there is scope to expand the model to incorporate functionalities with real-time data with implementations of spatial-temporal features to achieve a more functional system. Also, one other thing that can also be usually done is to develop the model into a windows or IOS system for law enforcement departments with real-time machine learning based monitoring of large crowds specifically it would prove very useful for countries with huge populations like India or China. Lastly, future research can be invested in expanding the domain of action detection from crowds & extend it to more diverse actions other than just violence & non-violence detection. In conclusion, we can positively say that a high precision violence & non-violence system was implemented using deep learning concepts like convolutional neuraI network on video data. We created a system with a high accuracy in detecting furious activities from pre-recorded video-clips as well as from live input from usb-camera. To detect riots in real-time frame by frame, we needed higher processing speed. With further advancement & research going about in the field of crowd behaviour analysis the system will only get better. In future, we plan to design an online front-end application where we could upload video-clips to detect furious activities. Furthermore, we are planning to take our research into next step by detecting suspicious task in real-time. We will try to connect this prototype with cctv monitoring cameras and a hardware device with alarm so that it could detect suspicious task or criminal task. The moment the system detects suspicious or criminal task it could activate an alarm or alert the police or guards.

References

1. C. C. Aggarwal. “A human-computer interactive method for projected clustering”.In: IEEE Transactions on Knowledge and Data Engineering 16.4 (2004), pp. 448–460. issn: 1041-4347. doi: 10.1109/TKDE.2004.1269669.
2. David Barrett. One surveillance camera for every 11 people in Britain, says CCTV survey. 2013. url: http://www.telegraph.co.uk/technology/10172298/Onesurveillance-camera-for-every-11-people-in-Britain-says-CCTV-survey.html.
3. Loris Bazzani et al. “Analyzing Groups: A Social Signaling Perspective”. In: Video Analytics for Business Intelligence. Ed. by Caifeng Shan et al. Berlin, Heidelberg:

Springer Berlin H
4. C. Chen, A. Heili, and J. M. Odobez. “A joint estimation of head and body orientation cues in surveillance video”. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). 2011, pp. 860–867. doi: 10.1109/ ICCVW.2011.6130342.
5. W. Choi and S. Savarese. “A Unified Framework for Multi-Target Tracking and Collective Activity Recognition”. In: ECCV. 2012.
6. W. Choi and S. Savarese. “A Unified Framework for Multi-Target Tracking and Collective Activity Recognition”. In: ECCV. 2012. url: http://www-personal.umich.edu/~wgchoi/eccv12/wongun_eccv12.html.
7. Wongun Choi, Khuram Shahid, and Silvio Savarese. “Learning context for collective activity recognition”. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE. 2011, pp. 3273–3280.
8. Wongun Choi, Khuram Shahid, and Silvio Savarese. “What are they doing?: Collective activity classification using spatio-temporal relationship among people”. In: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. IEEE. 2009, pp. 1282–1289.
9. Fran¸cois Chollet. Keras. 2015. url: https://github.com/fchollet/keras.
10. Marco Cristani et al. “Social interaction discovery by statistical analysis of Formations”. In: Proceedings of the British Machine Vision Conference. BMVA Press, 2011, pp. 23.1–23.12. isbn: 1-901725-43-X.
11. N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05).
12. Zhiwei Deng et al. “Deep structured models for group activity recognition”. In: arXiv preprint arXiv:1506.04191 (2015).
13. Zhiwei Deng et al. “Structure inference machines: Recurrent neuraI networks for analyzing relations in group activity recognition”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 4772–4781.
14. Piotr Doll´ar. Piotr’s Computer Vision Matlab Toolbox (PMT). https://github.com/pdollar/toolbox.
15. Piotr Doll´ar et al. “Pedestrian Detection: An Evaluation of the State of the Art”.In: PAMI 34 (2012).
16. Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: Kdd. Vol. 96. 34. 1996, pp. 226–231.
17. W. Ge, R. T. Collins, and R. B. Ruback. “Vision-Based Analysis of Small Groups in Pedestrian Crowds”. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 34.5 (2012),
18. Hossein Hajimirsadeghi et al. “Visual recognition by counting instances: A multiinstance cardinality potential kernel”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, pp. 2596–2605.
19. Hossein Hajimirsadeghi et al. “Visual recognition by counting instances: A multiinstance cardinality potential kernel”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, pp. 2596–2605.
20. David Hall and Pietro Perona. “Fine-Grained Classification of Pedestrians in Video: Benchmark and State of the Art”. In: CoRR abs/1605.06177 (2016).
21. Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: CoRR abs/1512.03385 (2015). url: http://arxiv.org/abs/1512.03385.
22. Kurt Hornik. “Approximation Capabilities of Multilayer Feedforward Networks”. In: neuraI Netw. 4.2 (Mar. 1991), pp. 251–257. issn: 0893-6080. doi:

10.1016/0893-6080(91)90009-T.
23. Jan Hendrik Hosang et al. “Taking a Deeper Look at Pedestrians”. In: CoRR abs/1501.05790 (2015). url: http://arxiv.org/abs/1501.05790.
24. Mostafa S Ibrahim et al. “A hierarchical deep temporal model for group activity recognition”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 1971–1980.
25. O. Boiman and M. Irani. Detecting irregularities in images and in video. In Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1, volume 1, pages 462–469 Vol. 1, Oct 2005. doi: 10.1109/ICCV.2005.70.
26. P.A. Crook, V. Kellokumpu, G. Zhao, and M. Pietikainen. Human activity recognition using a dynamic texture based method. In Proceedings of the British Machine Vision Conference, pages 88.1–88.10. BMVA Press, 2008.
27. P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pages 65–72, Oct 2005.
28. [34]Avishai Hendel, Daphna Weinshall, and Shmuel Peleg. Identifying Surprising Events in Videos Using Bayesian Topic Models, pages 448–459. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
29. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks”, in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
30. Y. Itcher T. Hassner and O. Kliper-Gross. Violent flows: Real-time detection of violent crowd behavior. In 3rd IEEE International Workshop on Socially Intelligent Surveillance and Monitoring (SISM) at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 2012.
31. G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6):915–928, June 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1110.
32. J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger”, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.