Anticipating Suspicious Actions using a Small Dataset of Action Templates

Renato Baptista, Michel Antunes, Djamila Aouada and Björn Ottersten

Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg

{*renato.baptista, djamila.aouada, bjorn.ottersten*}*@uni.lu, michel.gon.antunes@gmail.com*

Keywords: Action Templates, Alarm Generation, Skeleton.

Abstract: In this paper, we propose to detect an action as soon as possible and ideally before it is fully completed. The objective is to support the monitoring of surveillance videos for preventing criminal or terrorist attacks. For such a scenario, it is of importance to have not only high detection and recognition rates but also low time latency for the detection. Our solution consists in an adaptive sliding window approach in an online manner, which efficiently rejects irrelevant data. Furthermore, we exploit both spatial and temporal information by constructing feature vectors based on temporal blocks. For an added efficiency, only partial template actions are considered for the detection. The relationship between the template size and latency is experimentally evaluated. We show promising preliminary experimental results using Motion Capture data with a skeleton representation of the human body.

1 INTRODUCTION

Many surveillance systems are composed of cameras acquiring videos from specific locations and monitored by people (e.g. a security team) for detecting suspicious events and actions. It is a challenging task to manually monitor video feeds 24/7. As a consequence, only after criminal or terrorist attacks occur, the recorded surveillance data is actually used for analyzing what happened at that specific moment.

Nowadays, there are many visual surveillance systems that apply computer vision techniques for automatically detecting suspicious occurrences, including “human violence” recognition and detection (Bilinski and Bremond, 2016; Datta et al., 2002). General action recognition and detection is a largely investigated topic by the computer vision community, showing very promising results (Du et al., 2015; Gkioxari and Malik, 2015; Wang and Schmid, 2013; Wang et al., 2015; Papadopoulos et al., 2017). However, a major concern in security applications is not only the accurate detection and recognition of particular events or actions, but also the time latency required for achieving it. Many of the existing works are designed for action recognition (Bilen et al., 2016; Du et al., 2015; Fernando et al., 2016) and for offline action detection (Gaidon et al., 2011; Gkioxari and Malik, 2015; Jain et al., 2014; Tian et al., 2013; Wang et al., 2015; Papadopoulos et al., 2017). These methods require that the action to be recognized is completely acquired before the detection can be accomplished. However, even if the recognition accuracy is 100%, it is not recommended to use these approaches in security and surveillance applications, because an alarm can only be issued after the event has occurred. An alternative to these approaches would be an online action detection approach, where the objective is to detect an action as soon as it happens and ideally before the action is fully completed. Enabling to detect an action with low latency can be useful in many vi-

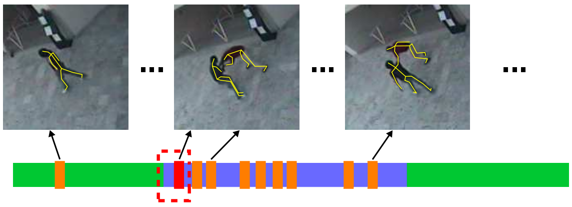


Figure 1: Alarm generation for a suspicious action during an online stream from a video surveillance camera. The green color represents a normal action and the blue color the ground truth information. The goal is to detect the suspicious action as soon as possible and ideally before it is fully completed. The orange flags represent the alarms that were generated during the video using our approach, and the red flag corresponds to the first alarm that was generated for the suspicious action. Images used were extracted from the CAVIAR dataset[1].

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIAR

deo surveillance applications.

Recently, Hoai and De la Torre (Hoai and De la Torre, 2012; Hoai and De la Torre, 2014) proposed a learning formulation based on a structured output Support Vector Machine (SVM) to recognize partial events, resulting in an early detection. Detecting an action in an online manner is not a trivial problem due to the unpredictability of real word scenarios (Geest et al., 2016; Li et al., 2016). Geest *et al.* (Geest et al., 2016) proposed a more realistic dataset for online action detection. The dataset consists of real life actions that were professionally recorded from six recent TV series (Geest et al., 2016). Li *et al.* (Li et al., 2016) presented a deep learning based architecture that allows to detect and recognize actions in an online manner. The authors show very promising results on a large dataset containing video streams. The major weakness of this approach is that, being based on deep learning methods, it requires a large amount of data for training the overall architecture. Large datasets of criminal and terrorist attacks are usually not available, because they only occur sporadically. This means that training a deep architecture with many layers and parameters is highly challenging.

In this work, the objective is to detect suspicious events and actions as soon as possible and ideally before they are fully completed. Note that, the goal of this approach is to support the video surveillance rather than being a completely automated system. We propose to use an adaptive sliding window, which efficiently rejects irrelevant data during streaming, together with a histogram based video descriptor and a nearest neighbor assignment, which have the advantage of efficient iterative computations. As the objective is to support video surveillance and security monitoring applications, one prefers to have many alerts with higher probabilities of detecting a suspicious action with very low latency, Figure 1 illustrates an example of a detection scenario using our approach. Similarly to (Li et al., 2016; Meshry et al., 2015; Sharaf et al., 2015), we use a skeleton representation of the human body as it is robust to scale, rotation and illumination changes, and it can be computed at high frame rate, allowing real-time computations (Han et al., 2017). Such a human body representation can be extracted from human pose estimation algorithms (Pishchulin et al., 2016). In contrast to (Li et al., 2016), our goal is to avoid the requirement of a large amount of annotated data. To that end, we propose to use a small dataset of suspicious actions and also to construct template actions using different percentages of the action to be detected. Furthermore, we present an analysis of the influence of using partial information of the action to be detected, showing that it is possible to achieve competitive detection results when compared to using the complete action sequence.

In summary, the contributions of this work are: 1) an efficient algorithm for detecting actions using a small dataset of template actions with low latency; and 2) an analysis of the time needed to detect an action using partial information of the action template.

The paper is organized as follows: in Section 2, we provide a brief introduction of the skeleton representation of the human body and the problem formulation for the proposed approach. Section 3 proposes the online action detection method and how to construct action templates. In Section 4, we describe and discuss the experimental results and Section 5 concludes the paper.

2 BACKGROUND & PROBLEM FORMULATION

In this section, we introduce the skeleton human body representation that is used throughout the paper. Let us assume that a human action video is represented by the spatial positions of the body joints (Antunes et al., 2016; Baptista et al., 2017a; Baptista et al., 2017b; Vemulapalli et al., 2014). A skeleton $\mathsf{S} = [\mathbf{j}_1, \cdots, \mathbf{j}_N]$ is defined using N joints, and each joint is represented by its 3D coordinates $\mathbf{j} = [j_x, j_y, j_z]^\mathsf{T}$, where $j_{x,y,z} \in \mathbb{R}^3$ and T denotes the matrix transpose. A human skeleton sequence is represented by $\mathsf{H} = \{\mathsf{S}_1, \cdots, \mathsf{S}_F\}$, where F is the total number of frames. In order to normalize each skeleton in a way that the size of each body part is in correspondence, a spatial registration is done by transforming each skeleton S, such that the world coordinate system is placed at the hip center and rotated in such a manner that the projection of the vector from the left hip to the right hip is parallel to the x-axis (Vemulapalli et al., 2014). Figure 2 shows an example of the normalized skeleton with respect to the world coordinate system placed at the hip center and the enumerated skeleton joints. Each skeleton S is then represented by the 3D normalized coordinates of N joints as a vector of size $3N$. In this work, we adopt the same approach as in (Chu et al., 2012), and a sequence H is represented using a bag of temporal words model (Sivic and Zisserman, 2003; Yuan et al., 2011). In this model, the codebook is built by using k-means in order to group similar feature vectors. Each skeleton S from H is discretized into histograms according to the k-entry dictionary. Then, the resulting representation of the sequence H is defined by the feature vector $\varphi(\mathsf{H})$, which is the cumulative summation of all the individual histograms of the sequence H.

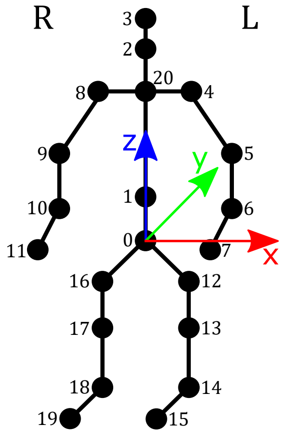


Figure 2: Representation of the normalized skeleton with respect to the world coordinate system placed at the hip center (joint number 0). L and R stand for the left and right side of the human body, respectively.

Considering an action template A_T of length T, and a subsequence of an input human action $\mathsf{H}_s^t \subset \mathsf{H}$ starting from frame s to the current frame t, i.e., $\mathsf{H}_s^t = \{\mathsf{S}_s, \cdots, \mathsf{S}_t\}$ such that $1 < s,t < F$, the objective is to estimate the starting point $\hat{s}$ of the action of interest such as (Chu et al., 2012):

$$\hat{s} = \underset{s}{\operatorname{argmin}}\ d(\varphi(\mathsf{H}_s^t), \varphi(\mathsf{A}_T)) \quad \text{s.t.} \quad t - s \geq L, \quad (1)$$

where L is the minimum length of the interval of interest and d is a distance function measuring dissimilarity between histograms. In (Chu et al., 2012), equation (1) is solved considering the full sequence H, while in the problem at hand, the objective is to find a solution in an online manner. To that end, we propose to follow an efficient adaptive sliding window strategy combined with: 1) adding temporal information to the spatial feature representation $\varphi(\cdot)$; and 2) decreasing the amount of data required from the action template. The proposed approach is detailed in Section 3.

3 PROPOSED APPROACH

We herein describe the proposed method for online action detection and how action templates are defined.

First of all, the temporal information is added by aggregating a consecutive number b of skeletons S together as a new $(b \times 3N)$-dimensional vector, where b defines the number of considered consecutive frames, also known as the *temporal block size*. Then, the feature representation of the sequence H using b is represented by $\varphi_b(\mathsf{H})$.

To solve (1), we compare the values of the distance function d for the two intervals $\mathsf{w}_1 = [s, t+1]$, and $\mathsf{w}_2 = [s+1, t+1]$ corresponding to the two subsequences H_s^{t+1} and H_{s+1}^{t+1}, respectively. We denote by d_{w_1} and d_{w_2} the resulting distances with respect to the template action A_T. The distance function d is the Euclidean distance between histograms. The minimum of the two, $d_{min} = \min(d_{\mathsf{w}_1}, d_{\mathsf{w}_2})$ is saved until a new minimum is found, and the start point is accordingly updated. This means that, if d_{min} is obtained from w_2, the start point will be increased by one for the next time instant $t+2$. In this case, the method rejects irrelevant data while it searches for the best start point of the action of interest. The minimal distance vector $\mathbf{d}_{min}$ is then the stored values of d_{min} over time, i.e., $\mathbf{d}_{min} = [d_{min}]$. As a temporal function, $\mathbf{d}_{min}$ starts to decrease as soon as the action of interest occurs, which means that in that interval the number of generated alarms increases significantly. An action is considered detected when $\mathbf{d}_{min}$ decreases by a number of τ consecutive blocks. For every time that the threshold τ is met, an alarm is generated. Figure 3 illustrates the relation of the distance function over time for the two temporal windows w_1 and w_2, and $\mathbf{d}_{min}$. As shown in Figure 3, the action is detected as soon as $\mathbf{d}_{min}$ decreases for a consecutive number of blocks τ. Note that, in this example, $\mathbf{d}_{min}$ is decreasing during the ground truth interval, which means that while the action is happening, a relative number of alarms are generated for each time that the threshold τ is met. Then, the alarm of interest is the first alarm that is generated within the ground truth interval.

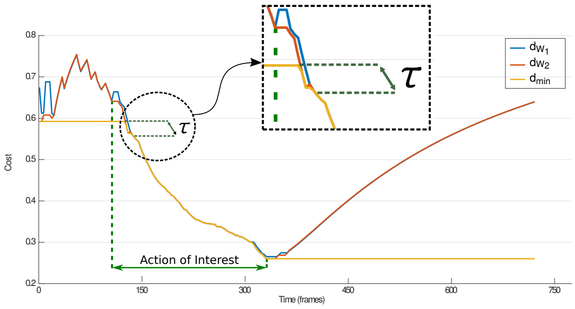


Figure 3: Representation of the minimal distance $\mathbf{d}_{min}$ over time. The blue and red lines represent the cost for each temporal window w_1 and w_2, respectively. The yellow line represents the smallest distance between the distances corresponding to the two temporal windows w_1 and w_2 over time. τ defines the threshold to validate a detection.

As the objective is to detect an action as soon as possible with low latency, using subsequences of the template action can be advantageous to detect an action before it is fully completed since less data is used. Therefore, we propose to use partial action templates from the full action A_T. We define subsequences $\mathsf{A}_{pT} \subset \mathsf{A}_T$, for $0 < p \leq 1$. Similarly to the aggregated feature vectors φ_b defined for H, for each subsequence A_{pT}, a histogram is created by accumulating individual blocks, where each block of size b is assigned according to the k-entry codebook. Then, the representation of an action template $\varphi_b(\mathsf{A}_{pT})$ is the average of all histograms of the same suspicious action.

4 EXPERIMENTAL RESULTS

In this section, we evaluate the detection and the time to detection performance on the public CMU-Mocap dataset [3]. From this dataset we select a set of actions from different subjects that fall into two different groups, the normal and the suspicious actions. The actions "walking", "standing" and "looking around" are considered as normal actions, and the actions "running", "punching" and "kicking" are considered as suspicious actions. Each sequence from the dataset was recorded at 120 frames per second (fps). To make it more realistic for surveillance purposes, we downsample each sequence by a factor of 4, resulting in a 30 fps sequence. Figure 4 shows an example of the skeleton representation of human body while a suspicious action is happening and the corresponding alarm generation.

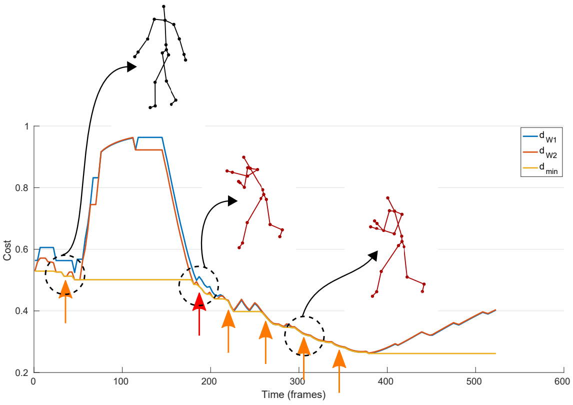


Figure 4: Skeleton representation of the human body while a suspicious action is happening, in this case: "Punching". The orange arrows represent the alarms that were generated over time and the red arrow is the alarm of interest which is the first alarm generated within the ground truth.

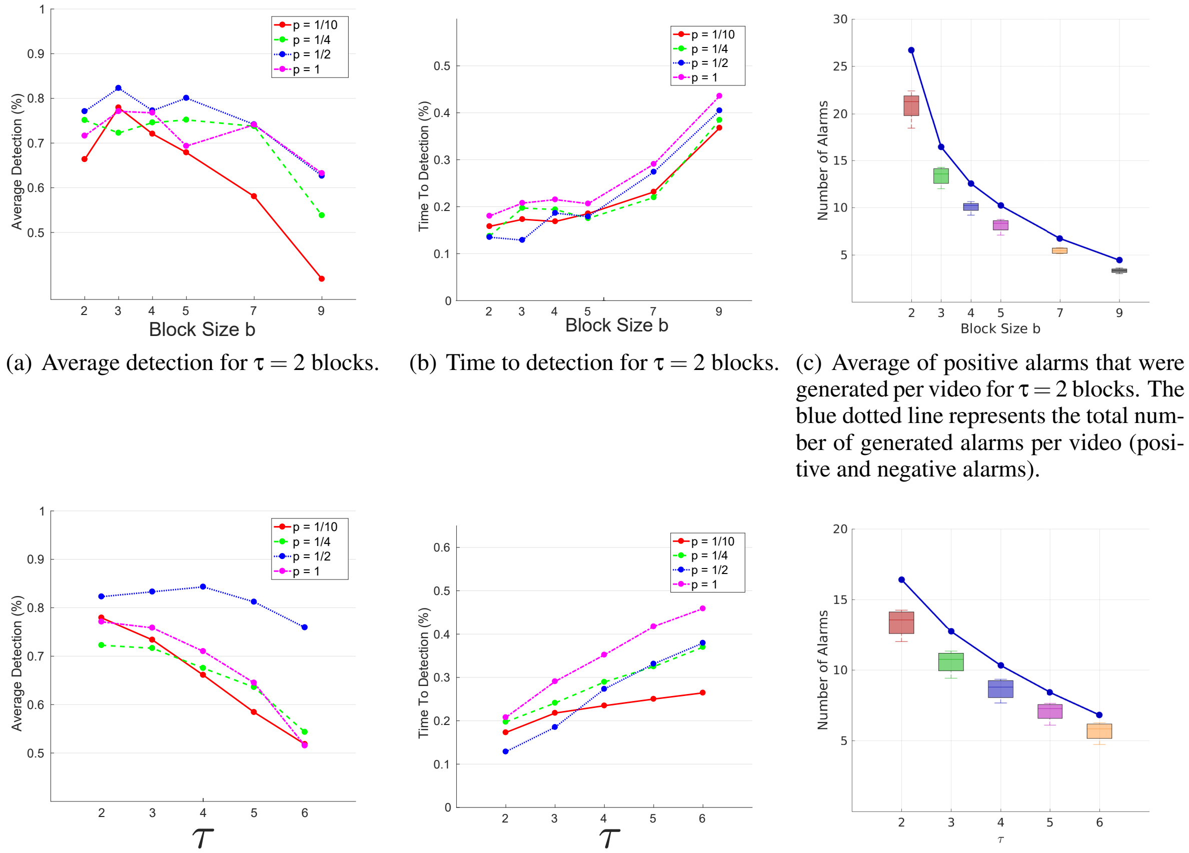
4.1 Blocks vs. Threshold

In order to evaluate the detection and the time needed to detect an action, we simulate an input video stream by randomly concatenating normal actions before and after the suspicious action (action of interest). First, we start by evaluating the relation between the block size b and the detection accuracy. Moreover, an evaluation of the time needed to detection is also done in order to see how much $X\%$ of the action was needed to complete a detection. We determine a correct detection if the minimal distance function $\mathbf{d}_{min}$ decreases by τ consecutive blocks and if the resulting detection is within the ground truth interval. A range of different values for the parameters were tested and for the best detection accuracy with the lowest latency. We fix $b = 3$ frames and $\tau = 2$ for the following experiments. Figure 5(a) and 5(d) illustrate the detection accuracy using different lengths of the subsequence of the template action. Figure 5(b) and 5(e) show an evaluation of the time needed to detect an action for the different subsequences of the template action, where the lowest latency obtained is around 12%. In these experiments, it is shown that the proposed approach can be applied on a wide range of applications for video surveillance. For applications that require fast detection, the value of the threshold τ should be lower and on the other hand, if the application requires a more precise detection and the latency is not a constraint, the threshold value should be higher. For example, in a multi screen surveillance monitoring system, when detection occurs, an alarm is flagged for the specific screen where the action is happening. This will get the attention of the security guard, to then analyse the action and make a decision. Such applications allow a less sensitive system, where a false alarm can be generated without compromising the security, since it only alerts the security guard to look at the camera where the alarm was generated. Figure 5(c) and 5(f) show the average number of alarms that were generated per video. Furthermore, it also shows the number of positive alarms per video, where each color represents the number of correct detections per video for the different b and τ. Note that, using $b = 2$ until $b = 5$, the latency is between 12% and 22% and the detection accuracy is between 68% and 81%. Considering this and depending on the application, these parameters can be tuned with respect to the desired number of generated alarms per video. Increasing the number of temporal blocks will reduce the number of alarms that are generated per video.

4.2 Impact of the Size of the Template Action

In order to evaluate the impact of the size of the subsequences of the template action, we use the following percentages of the initial part of the action $p = \frac{1}{10}, \frac{1}{4}, \frac{1}{2}$ and the full action. We propose these subsequence lengths in order to understand the relations-

[3] http://mocap.cs.cmu.edu



(a) Average detection for $\tau = 2$ blocks.

(b) Time to detection for $\tau = 2$ blocks.

(c) Average of positive alarms that were generated per video for $\tau = 2$ blocks. The blue dotted line represents the total number of generated alarms per video (positive and negative alarms).

(d) Average detection for $b = 3$ frames.

(e) Time to detection for $b = 3$ frames.

(f) Average of positive alarms that were generated per video for $b = 3$ frames. The blue dotted line represents the total number of generated alarms per video (positive and negative alarms).

Figure 5: In the first row of images we fixed $\tau = 2$ blocks and computed the average detection, the time to detection and the average number of alarms and also the number of positive alarms that were generated per video for the different temporal blocks b, respectively. In the second row we proceed with the same experiments, fixing the block size $b = 3$ frames for different τ blocks.

hip between the size of the template action and the time needed to detect the action. Figure 6 shows the average detection accuracy for the different subsequences of the template action. The best detection accuracy obtained is for $b = 3$ frames and for $p = \frac{1}{2}$. This means that using half of the action as a subsequence of the template action, we achieve competitive detection results with low latency. This setup can be advantageous for applications where the action needs to be detected with low latency. In addition, using only a percentage of the initial part of an action can decrease the time needed to detect an action, resulting in a faster detection. Table 1 shows the detection accuracy for each suspicious action separately. Note that, for the action "Punching" the detection is higher due to the fact that this action is more discriminative for the upper limbs when compared with the other two suspicious actions. One possible way to increase the detection accuracy for all actions would be a more robust representation, such as the relative position of the joints or a Fourier Temporal Pyramid (Vemulapalli et al., 2014) representation. We did not implement these representations to avoid the complexity of the descriptors in order to have a better understanding of the performance and characteristics of the proposed method.

Table 1: Detection accuracy (%) for the following scenario: $b = 3$ frames and $\tau = 2$ blocks, for $p = \frac{1}{10}, \frac{1}{4}, \frac{1}{2}$ and 1.

p Actions	$\frac{1}{10}$	$\frac{1}{4}$	$\frac{1}{2}$	1
Running	61.23	44.95	70.82	**79.87**
Kicking	76.7	81.73	**84.17**	60.76
Punching	**95.83**	90.08	91.89	90.67

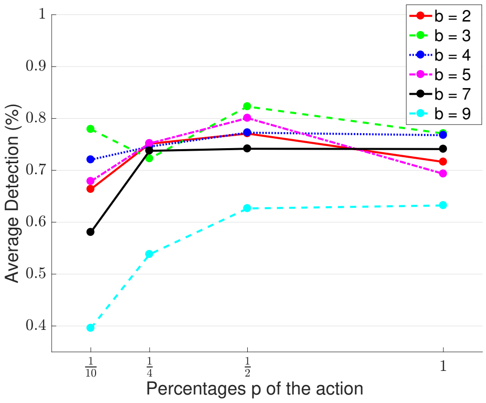


Figure 6: Average detection for the different subsequences with length p of the template action for $\tau = 2$.

5 CONCLUSIONS

In this paper, we proposed an online method to detect suspicious actions with low latency. This method is based on an adaptive sliding window which efficiently rejects irrelevant data during streaming. We explored the feature representation of a subsequence using the spatial and temporal information of the video stream. Furthermore, we evaluated the relationship between the size of the template action and latency, where we conclude that using half of the action as a template action, the detection accuracy and the time needed to detect the action achieve competitive and promising results compared to using the full action as a template. We also observed that tuning the parameters, the method can be used for different setups of video surveillance. Next, we intend to use real surveillance videos coupled with a robust human pose detection approach, e.g. (Pishchulin et al., 2016).

ACKNOWLEDGEMENTS

This work has been partially funded by the National Research Fund (FNR), Luxembourg, under the CORE project C15/IS 10415355/3D-ACT/Björn Ottersten. This work was also supported by the European Union‘s Horizon 2020 research and innovation project STARR under grant agreement No.689947.

REFERENCES

Antunes, M., Baptista, R., Demisse, G., Aouada, D., and Ottersten, B. (2016). Visual and human-interpretable feedback for assisting physical activity. In *European Conference on Computer Vision (ECCV) Workshop on Assistive Computer Vision and Robotics Amsterdam,*.

Baptista, R., Antunes, M., Aouada, D., and Ottersten, B. (2017a). Video-based feedback for assisting physical activity. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*.

Baptista, R., Antunes, M., Shabayek, A. E. R., Aouada, D., and Ottersten, B. (2017b). Flexible feedback system for posture monitoring and correction. In *IEEE International Conference on Image Information Processing (ICIIP)*.

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., and Gould, S. (2016). Dynamic image networks for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bilinski, P. and Bremond, F. (2016). Human violence recognition and detection in surveillance videos. In *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, pages 30–36. IEEE.

Chu, W.-S., Zhou, F., and De la Torre, F. (2012). Unsupervised temporal commonality discovery. In *European Conference on Computer Vision*, pages 373–387. Springer Berlin Heidelberg.

Datta, A., Shah, M., and Da Vitoria Lobo, N. (2002). Person-on-person violence detection in video data. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1 - Volume 1*, ICPR '02, pages 10433–, Washington, DC, USA. IEEE Computer Society.

Du, Y., Wang, W., and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., and Tuytelaars, T. (2016). Rank pooling for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gaidon, A., Harchaoui, Z., and Schmid, C. (2011). Actom Sequence Models for Efficient Action Detection. In *CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition*, pages 3201–3208, Colorado Springs, United States. IEEE.

Geest, R. D., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., and Tuytelaars, T. (2016). Online action detection. *CoRR*, abs/1604.06506.

Gkioxari, G. and Malik, J. (2015). Finding action tubes.

Han, F., Reily, B., Hoff, W., and Zhang, H. (2017). Space-time representation of people based on 3d skeletal data: A review. *Computer Vision and Image Understanding*, pages –.

Hoai, M. and De la Torre, F. (2012). Max-margin early event detectors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Hoai, M. and De la Torre, F. (2014). Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202.

Jain, M., van Gemert, J. C., Jégou, H., Bouthemy, P., and Snoek, C. G. M. (2014). Action localization by tubelets from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., and Liu, J. (2016). Online human action detection using joint classification-regression recurrent neural networks. *European Conference on Computer Vision*.

Meshry, M., Hussein, M. E., and Torki, M. (2015). Action detection from skeletal data using effecient linear search. *CoRR*, abs/1502.01228.

Papadopoulos, K., Antunes, M., Aouada, D., and Ottersten, B. (2017). Enhanced trajectory-based action recognition using human pose. In *IEEE International Conference on Image Processing (ICIP)*.

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., and Schiele, B. (2016). Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4929–4937.

Sharaf, A., Torki, M., Hussein, M. E., and El-Saban, M. (2015). Real-time multi-scale action detection from 3d skeleton data. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 998–1005. IEEE.

Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, volume 2, pages 1470–1477.

Tian, Y., Sukthankar, R., and Shah, M. (2013). Spatiotemporal deformable part models for action detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 2642–2649, Washington, DC, USA. IEEE Computer Society.

Vemulapalli, R., Arrate, F., and Chellappa, R. (2014). Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia.

Wang, Z., Wang, L., Du, W., and Qiao, Y. (2015). Exploring fisher vector and deep networks for action spotting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Yuan, J., Liu, Z., and Wu, Y. (2011). Discriminative video pattern search for efficient action detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1728–1743.