Action Localization in Video using a Graph-based Feature Representation

Iveel Jargalsaikhan, Suzanne Little and Noel E O'Connor
Insight Centre for Data Analytics,
Dublin City University, Ireland
iveel.jargalsaikhan2@mail.dcu.ie

Abstract

We propose a new framework for human action localization in video sequences. The option to not only detect but also localize actions in surveillance video is crucial to improving system's ability to manage high volumes of CCTV. In the approach, the action localization task is formulated the maximum-path finding problem in the directed spatio-temporal video-graph. The graph is constructed on the top of frame and temporal-based low-level features. To localize actions in the video-graph, we apply a maximum-path algorithm to find the path in the graph that is considered to be the localized action in the video. The proposed approach achieves competitive performance with the J-HMDB and the UCF-Sports dataset.

1. Introduction

Understanding of human action in video sequences is useful for a variety of applications such as detecting relevant activities, summarizing and indexing video sequences, organizing a digital video library according to the relevant actions, etc. In security applications, CCTV footage can be analysed in order to index actions of interest and enable queries relating to actions such as anti-social or criminal behaviour or to monitor crowd volume or aggression. However it remains a challenging problem for computer vision to robustly recognize action due to cluttered backgrounds, camera motion, occlusion, view point changes and the geometric and photometric variances of objects.

Recent methods [13, 25, 7] for action recognition mostly focus on action classification rather than action localisation. Mostly the top-performing classification approaches in the action modelling process [25, 16, 22] explicitly or implicitly use the background information, i.e., the region where the action is not performed. This significantly contributes to the classification performance [19, 5] but prevents the identification of the region where the action is taking place. However, the action localisation task requires the classified action to be localised both spatially and temporally. The temporal localisation can be efficiently [18] detected using the typical classification method coupled with the sliding window technique. However, the spatial localisation is complicated for the classification-based methods due to the above mentioned use of the background. This paper aims to address the localisation problem emphasising the spatial localisation of the action. In particular, we propose an action localisation framework based on a directed video-graph and present two major contributions as follows,

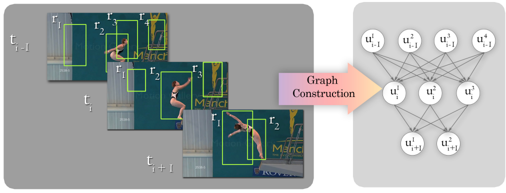

Figure 1. A directed video graph construction process

First, we propose a new directed video graph suited for the localisation task. In the graph, the node describes a candidate action region and its connectivity (edge) describes the similarity with the adjacent region regarding cues such as region colour, motion and region geometry. The discriminative score of each node is calculated using a late fusion technique based on the corresponding local and regional features. The late fusion provides a means to integrate a variety of features of different type and dimensions (local, global and regional). Also, it makes the video graph representation sufficiently flexible to combine a richer set of features that has a potential to increase the performance. Secondly, this paper presents the application of the maximum-path finding algorithm to identify the localised action. This method has been successfully adopted in an object tracking problem [4] where it showed its effectiveness. We propose that action localisation can be understood as semantic concept tracking over time. Therefore we investigate whether this approach can be extended to the challenge of action localisation. The proposed approach is evaluated using two benchmark action datasets, namely J-HMDB and UCF-Sports. In the literature [20, 12, 27], these datasets have been extensively used for action localization. The remainder of this paper is organized as follows: Section 3 presents a overview of the proposed framework followed the video graph construction (Section 3.1) and the maximum-path (MPF) formulation on the video graph (Section 4) and the experimental result (Section 5).

2. Related Work

Action localisation is becoming crucial for effective analysis of the *realistic* video capture scenario that consists of videos captured in complex settings that have significant background clutter or contain multiple actors or actions. The earlier works [20] propose to directly use a classifier on the action localisation task using 3D sliding window or similar technique. The main advantage is that the mid-level representation may not be necessary. However, the sliding window approach substantially increases the computational complexity when an input video has a long duration or high-resolution.

In other approaches, the localisation is primarily based on the action proposal [8] inspired by the success of the region proposal methods for the object localisation task in 2D images. For instance, the *objectness* technique [1] for object localization is extended to video by [4] and *selective search* [21] is modified into spatio-temporal tubelets in [27]. This class of method overcomes the short-coming of classifier-based approach by investigating the selected part or region of the video rather than the entire video. The region proposal approach is computationally efficient and obtains good results for action localisation in comparison to the other methods. We adopt this strategy in the development of the video graph.

Recently the approaches based on features from convolutional neural networks [11] [14] have achieved significant progress in the object detection and image classification task. In particular, the approaches based on regional convolutional neural networks (RCNN) [17] are the state-of-the-art that have produced best results with a high-margin of difference compared to competing approaches for the object localisation task. Gkioxari et al [8] first applied the region-based convolutional feature (R-CNN) in the action localisation task and achieved promising results. However, the action detection is frame-based and can not take into account the temporal dynamics of the action which is an important cue in any action recognition system. More recently, Weinzaepfela et al. [27] introduced a method to overcome this weakness by fusing the region-based feature (R-CNN) with a track descriptor, that is similar to the trajectory feature used in our approach, and achieved further improvement. This shows that combining the frame-level descriptor, such as R-CNN, with local temporal features (motion trajectories) that are complementary to each other and improves the performance. Our framework embeds both the region-based convolutional features (R-CNN) and the local trajectory features to obtain the discriminative graph model.

In an approach similar to our work, [24] introduces an action localisation framework based on action proposals from dense trajectories features. However our proposed framework differs in several key aspects: first, we develop the effective graph structure that is capable of integrating the different feature types i.e., local trajectory and RCNN features. Furthermore, the additional cues such as local, motion and region geometry are captured as the graph edges. Finally, the localisation is performed by maximising the path score in a video graph.

3. Proposed Framework

In the localisation framework, given a video, we first apply a region proposal technique at the frame level. This step produces the candidate action regions that form the basis for constructing the video graph. In the video graph, the node represents the region along with its corresponding features and the edge describes the similarity with its adjacent region. To assign the discriminative node score, support vector machines (SVMs) classifier is built with training videos for each type of feature (local and regional) and integrated using a late fusion method (details in Section 3.1.3). Finally, the maximum-path finding algorithm is used to find the maximum scoring path in the video graph of a test video. The regions associated with the maximum path is considered as the localised action. Next, we describe construction of the video graph.

3.1. Video Graph Construction

Given a video sequence $V = \{I_1, I_2, .., I_n\}$, where I_k is a static frame at the time instance k, we construct the corresponding video graph $G(V, E)$. As shown in Figure 1, the node u_k^j describes the action candidate region regions defined by a rectangular region $r_k^j = (x_k^j, y_k^j, h_k^j, w_k^j)$ in the static frame I_k. There are various ways to acquire candidate regions such as dense sampling [6] that subdivides the frame into fixed grids at different scales. However, it has an implication to substantially increase the number of candidate regions whereby the computational complexity increases. Consequently, the alternative strategy is to use the region proposal method that efficiently identifies the likely object regions using only texture and edge information. Although any object proposal can be used in our framework, the selective-search method [21] is used in the experiment due to the availability of its implementation [1]. The region proposal is applied on the video frames to generate approximately 2000 candidate action regions per frame. Furthermore, we filter the candidate regions where there is no significant motion according to the method [8]. This signifi-

[1] http //koen.me/research/selectivesearch/

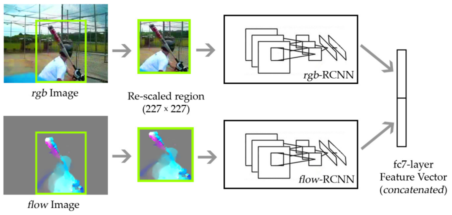

Figure 2. The regional feature calculation process

cantly reduces the number of a region by 85% with a loss of only 4% (action positive regions).

Once the node region is determined, the next important step is feature extraction process. The recently [27, 8] successfully used region-based neural network (RCNN) features for action localisation, are adopted for describing the node region. The RCCN is shown to be highly discriminative as well as able to describe the region with arbitrary size. However, it does not capture the temporal dynamics of the action beyond two consecutive frames. Thus, the local dense trajectory feature is extracted from the node region to complement the RCCN feature. Next, we discuss how the features are extracted in detail.

3.1.1 Regional feature

Gkioxari et al [8] introduced RCNN features that operate separately on the image and optical flow. We use the same set of RCNN features i.e., rgb-RCNN and $flow$-RCNN. Given a region re-scaled to the dimension of 227×227, the rgb-RCNN operates on a three-channel of the colour image. It captures the static appearance of the actor/scene. For $flow$-CNN feature extraction, the flow image is first formed by transforming the dense optical flow into a 3-channel image (the flow x & y component and its amplitude) followed by the re-scaling and convolutional process. The $flow$-RCNN captures the motion pattern of the action. In the experiment, the pre-trained RCNN network[2] is used to compute the rgb-RCNN and the $flow$-RCNN features from the video frame region associated with the graph nodes as show in Figure 2. We use the concatenation of the $fc7 - layer$ (4096 dimension) features of rgb-RCNN and $flow$-RCNN network. We refer to the concatenated vector (9192 dimension) as v_i^f for the node u_i with the corresponding region r_i.

3.1.2 Local feature

Although the proposed approach is not constrained by the type of local features, we adopt the feature/descriptor described in the work [25]. In particular, we use the dense trajectory [25] that extracts the motion trajectories. In the experiment, the trajectory length is set short $L = 15$ frames to avoid the drifting trajectory problem. We apply the feature extraction for the entire video. Then the video graph node u_i is associated with the local features located in its region r_k. For each feature, four descriptors (TRAJ, HOG, HOF, MBH) are calculated and concatenated to form a single vector.

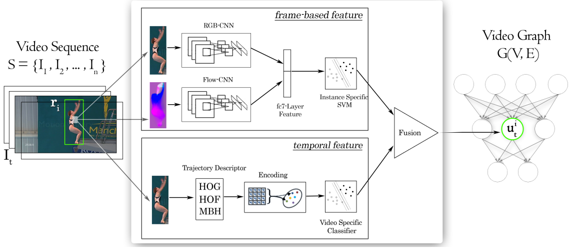

Figure 3. The procedure of calculating the node score of the action graph. We use two different feature types: *local* and *region-based*. Each feature is aggregated into final node score using late-fusion method.

3.1.3 Classifier training and node discriminative score

Since we use two sets of features, two separate classifiers (regional and local) are trained. For the regional feature, we train SVM classifiers for each action class $c \in C$, where ground truth regions are considered as positive examples and regions that overlap by factor of less than 0.3 times the area with the ground truth as negative. During training, the hard-negative mining technique is used. This strategy has shown significant improvement compared to traditional training [17] in the object localisation task.

For training a classifier for local features, we use the Bag-of-Features (BoF) model with the re-formulated scoring function introduced in the work [6] . In the experiment, the one-against-rest strategy is used to produce a binary classifier for each action class c. Once the SVM classifier is learned, the discriminative score for node u_i is calculated as follows:

Regional Classifier: Given a region r_i of node u_i with the extracted regional feature vector v_i^f and the trained classifiers for action class. Each node u_i in the video graph is assigned with a discriminative score for action class c:

$$score_c^f(u_i) = \beta_c + w_c' \cdot v_i^f \qquad (1)$$

where the discriminative score is the estimate of a likelihood that action c is performed within the region r_i of the node u_i and w_c, β_c are learned bias and support vector of the trained regional SVM classifier for action c.

Local classifier: As we formulated the localisation as the maximum path, the discriminative score should be able to be combined additively to give the cumulative score for traversing the path in the video graph. The additivity

[2] https://github.com/gkioxari/ActionTubes

requirement on the classifier property is applicable here. Therefore, we use the linear (additive) SVM classifier for training. In particular, for each training video, we compute the BoF encoding with K visual words. A training video with N local features is described by the set $S = \{(\boldsymbol{x}_i, v_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i = (x_i, y_i, t_i)$ refers to the local feature position in space and time, and v_i is the associated local descriptor. Let $h(S)$ be function maps feature set S into K-dimensional BoF coded vector.

The one-against-rest strategy is to learn a linear SVM for each action class $c \in C$. The resulting score function can be re-formulated as a sum over the contribution from each feature and this formulation is used calculate the discriminative score for node u_i ,

$$score_c^t(u_i) = \beta_c + \sum_{j=1}^{K} w_c^j h^j(S(r)) = \beta_c + \sum_{i \in r_i} w_c^{c_i} \quad (2)$$

where $h^j(S)$ denotes the j-th bin count for histogram $h(S)$. The j-th word is associated with a weight $w^j = \sum_i \alpha h^j(S_i)$ and w_c, β_c are learned bias and support vector of the learned SVM classifier for action c.

Late-Fusion: To calculate the final discriminative score for a given node u_i, we use the fusion technique to combine the respective scores as follows:

$$score_c(u_i) = \alpha \cdot score_c^r(u_i) + (1 - \alpha) \cdot score_c^l(u_i) \quad (3)$$

where α is a scalar. In the experiment, we use this parameter to investigate the respective feature type contribution to localisation performance.

3.2. Edge weight

The edge $e(u_i, u_j)$ represents the similarity between given nodes u_i, u_j. In the proposed video graph, the edge is formed between temporally adjacent nodes as shown in Figure 4 and the edge direction is used to enforce the path to flow in time. The action localisation can be understood as semantic concept tracking over time. In tracking methods [2][3], the authors use the color, motion cues for successful object tracking. A rich set of cues is crucial for the accurate registration of the object over different frames. Therefore we propose to combine multiple cues (colour, descriptor and geometric) to determine the edge weight:

$$e(u_i, u_j) = \begin{cases} f_c(r_i, r_j) + f_g(r_i, r_j) + f_d(r_i, r_j), & \text{if adj}(r_i, r_j) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where r_i, r_j are the corresponding region for the node u_i and u_j, respectively and *adj*(r_i, r_j) implies the temporally adjacent nodes and the term f_c, f_g and f_d are defined as follows:

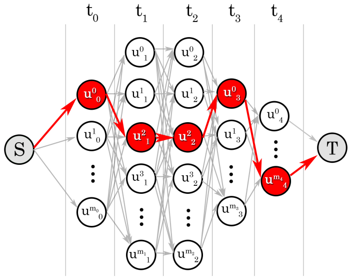

Figure 4. The maximum path in the graph considered to be the localized action. In the experiment, we use the Boykov-Kolmogorov method to calculate the maximum flow between node S, T .

- **Color Similarity Term** (f_c): Many colour descriptors have been proposed in the literature. In the experiment, we use the region-based color descriptor proposed by Van et al.[23] due the availability of its implementation[3]. The color descriptor (108 dimension) is extracted from the region r_k for each color hannel (RGB) and concatenated to create the combined descriptor c_k. Then color similarity term $f_c(r_i, r_j)$ is defined as a cosine measure between $cosine(c_i, c_j)$ where c_k is concatenated color descriptor extracted from a region r_k of the node u_k. The cosine similarity is selected due to it has positive space, where the outcome is neatly bounded in a range of $[0, 1]$.

- **Descriptor Similarity Term** (f_d): This term is based on the assumption that the features extracted from the same actor/action should resemble similarity. In the experiment, we use the regional feature to determine the descriptor similarity term as follows: $f_c(r_i, r_j) = cosine(v_i, v_j)$ where v_j v_i is the regional features extracted at the region r_i and r_j respectively.

- **Geometric Similarity Term** (f_g): This term encourages the spatial coherence between the node regions. In other words, the term scores high if the spatial extent significantly overlaps. The geometric similarity is defined as intersection of over union measure, $f_g(r_i, r_j) = IOU(r_i, r_j)$ i.e the full overlap between the regions gives a score of 1.

4. Action localization in the video-graph

Assuming a video sequence is mapped into directed video-graph $V(G, E)$ as discussed in Section 3.1. We now describe how to localize action in the graph. Given a path p, the score $M_c(p)$ is defined as:

$$M_c(p) = \sum_{i \in p} score_c(u_i) + \lambda \sum_{(i,j) \in p} e(u_i, u_j) \quad (5)$$

[3] http://lear.inrialpes.fr/people/vandeweijer/color_descriptors.html

where c is the action class and λ is a scalar. The edge weight $e(u_i, u_j)$ scores high if the corresponding node regions r_i, r_j overlap and agree in terms of color and regional feature. To localize the action, the problem becomes to find the optimal path $p*$ with highest accumulated score:

$$(p^*, c^*) = \arg\max_{c \in C} \arg \max_{p \in path(G)} M_c(p) \tag{6}$$

where $p^* = [u_1, u_2, ..., u_t]$ is the trajectory that maximizes the video graph with action class c^*. Finally the corresponding regions $[r_1, r_2, ..., r_t]$ will be considered as the localised action in the video sequence. The Maximum path problem is efficiently solved using dynamic programming. In the experiment, we have used Boykov-Kolmogorov algorithm to find the maximum flow in the graph by adding zero-weighted source S and terminal T node as shown in Figure 4.

5. Evaluation

5.1. Datasets

We evaluate our approach on two widely used datasets, namely UCF Sports [15] and J-HMDB [10]. On UCF sports we compare against other techniques and show substantial improvement from state-of-the-art approaches. We present an ablation study of our CNN-based approach and show results on action classification using our action tubes on JHMDB, which is a substantially larger dataset than UCF Sports. The UCF Sports dataset consists of 150 videos with 10 different actions. There are on average 10.3 videos per action for training, and 4.7 for testing 1 . J-HMDB contains about 900 videos of 21 different actions. The videos are extracted from the larger HMDB dataset [24], consisting of 51 actions.To date, UCF Sports has been widely used by scientists for evaluation purposes.

5.2. Experimental Protocol

To quantify our results, we report AUC curves for the UCF-Sports dataset, a metric commonly used by other approaches. A number of recent methods have used AP metrics, and we have compared our method performance against these reported methods for both the J-HMDB and UCF-Sports dataset.

5.3. Results

5.3.1 UCF Sports

In Figure 5 we plot the average AUC (Area Under Curve) for different values of σ (IOU parameter). The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold setting. We plot the curves as produced by the recent state-of-the-art approaches, Jain et al. [9] , Wang et al. [26], Tian et al. [20], Lan et al. [12], Action Tube [8] and SMTH [27]. Our approach outperforms most of these techniques, showing the most improvement for high values of overlap. In particular, the proposed method achieves the competitive performance with the recent state-of-the-art work [27] only falling short by a slight margin. For comparison with the state-of-the-art methods, as shown in column 2 at Table 2, our method achieves competitive performance of MAP = 88.7 % with IOU parameter σ = 0.5.

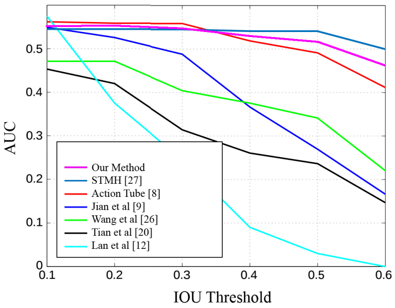

Figure 5. AUC for varying IoU thresholds for UCF-Sports Dataset

action class	local	region	combined
brush_hair	79.1%	59.8%	84.9%
catch	27.8%	11.6%	33.6%
clap	57.3%	20.1%	60.3%
climb_stairs	21.8%	23.0%	63.9%
golf	92.3%	29.7%	95.7%
jump	14.4%	9.0%	14.5%
kick_ball	14.5%	3.7%	15.5%
pick	42.4%	14.4%	53.9%
pour	92.3%	70.3%	97.1%
pullup	89.8%	52.1%	96.2%
push	63.3%	31.6%	55.8%
run	37.9%	13.7%	42.9%
shoot_ball	23.0%	19.9%	26.5%
shoot_bow	79.1%	31.0%	79.6%
shoot_gun	25.7%	19.6%	48.5%
sit	40.0%	30.5%	50.2%
stand	39.1%	32.3%	42.6%
swing_baseball	79.5%	9.6%	81.3%
throw	25.9%	11.1%	28.5%
walk	70.7%	33.7%	77.8%
wave	37.0%	23.9%	50.0%
MAP	26.2%	50.1%	57.1%

Table 1. The performance by action class for J-HMDB dataset (Split 1)

5.3.2 J-HMDB dataset

First, we report the performance of the 21 actions of the J-HMDB dataset. Table 1 presents the result by the differ-

J-HMDB ($\sigma = 0.5$)		UCF-Sports ($\sigma = 0.5$)	
Action Tube[8]	53.3 %	Action Tube [8]	75.8 %
STMH [27]	60.7 %	STMH [27]	90.5 %
Our method	56.3 %	*Our method*	88.7 %

Table 2. Comparison of the method with the state-of-the-art methods

ent combination of features used: local (TRAJ, HOG, HOF, MBH), regional (flow RCNN + RGB RCNN) and fused (local + regional). It is apparent that the fused approach consistently outperforms the individual features. The regional feature performs significantly better for almost all actions in comparison with the local counterpart. It proves the highly discriminative nature of the convolutional feature. Regarding MAP, feature fusing (57.1%) shows the improvement of 7%, 31 % in comparison to using regional (50.1%) and local feature (26.2%) alone.

For comparison with the-state-of-art methods, recently two methods have evaluated their system MAP performance averaged over all three splits with IOU parameter $\sigma = 0.5$. As shown at column 1 of Table 2, our method achieves competitive MAP performance of 56.30 %.

6. Conclusion

We propose a novel video graph-based framework for human action localisation from video sequences. The ability to not only detect but also localize actions in surveillance video is crucial to improving surveillance system's capacity to manage high volumes of CCTV. The proposed approach can effectively accommodate different types of feature using the late fusion method. Also, the additional cues such as colour, motion and the geometrical information are captured within the graph representation. We perform the action localisation by maximising the score associated with the node and the edge in the video graph. The proposed approach achieves competitive performance with J-HMDB and UCF-Sports dataset.

Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under grant number SFI/12/RC/2289 and the technical support from NVIDIA corporation.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11):2189–2202, 2012.

[2] J. G. Allen, R. Y. Xu, and J. S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Visual information processing*, 2004.

[3] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33(8):1619–1632, 2011.

[4] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI, IEEE Transactions on*, 33(9):1806–1819, 2011.

[5] L. Cao, Z. Liu, and T. S. Huang. Cross-dataset action detection. In *CVPR (CVPR), 2010 IEEE conference on*, pages 1998–2005. IEEE, 2010.

[6] C.-Y. Chen and K. Grauman. Efficient activity detection with max-subgraph search. In *CVPR 2012*.

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2005.

[8] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, pages 759–768, 2015.

[9] M. Jain, J. Gemert, H. Jégou, P. Bouthemy, and C. Snoek. Action localization with tubelets from motion. In *Conference on CVPR*, pages 740–747, 2014.

[10] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. Black. Towards understanding action recognition. In *ICCV*, pages 3192–3199, 2013.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[12] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV 2011*.

[13] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[14] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *neural networks*, 8(1):98–113, 1997.

[15] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *IEEE CVPR 2009*.

[16] K. Rapantzikos, Y. Avrithis, and S. Kollias. Dense saliency-based spatiotemporal feature points for action recognition. In *CVPR, 2009. CVPR 2009. IEEE Conference on*, pages 1454–1461. IEEE, 2009.

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[18] K. Soomro, H. Idrees, and M. Shah. Action Localization in Videos Through Context Walk. *ICCV*, pages 3280–3288, 2015.

[19] W. Sultani and I. Saleemi. Human action recognition across datasets by foreground-weighted histogram decomposition. In *Conference on CVPR*, pages 764–771, 2014.

[20] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *Conference on CVPR*, pages 2642–2649, 2013.

[21] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[22] M. Ullah, S. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC*, volume 10, pages 95–1, 2010.

[23] J. Van De Weijer and C. Schmid. Coloring local feature extraction. In *Computer Vision–ECCV 2006*, pages 334–348. Springer, 2006.

[24] J. van Gemert, M. Jain, E. Gati, and C. Snoek. Apt: Action localization proposals from dense trajectories. In *BMVC*, volume 2, page 4, 2015.

[25] H. Wang, A. Klaser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *IEEE CVPR*, pages 3169–3176, 2011.

[26] H. Wang, C. Yuan, W. Hu, and C. Sun. Supervised class-specific dictionary learning for sparse modeling in action recognition. *Pattern Recognition*, 2012.

[27] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, pages 3164–3172, 2015.