Live Video Action Recognition from Unsupervised Action Proposals

Roberto J. López-Sastre[1], Marcos Baptista-Ríos[2], Francisco J. Acevedo-Rodríguez[1],
Pilar Martín-Martín[1], Saturnino Maldonado-Bascón[1]
[1]GRAM, Department of Signal Theory, University of Alcalá, Alcalá de Henares, Spain.
[2]Multimodal Information Group, Gradiant, Vigo, Spain.

Abstract

The problem of action detection in untrimmed videos consists in localizing those parts of a certain video that can contain an action. Typically, state-of-the-art approaches to this problem use a temporal action proposals (TAPs) generator followed by an action classifier module. Moreover, TAPs solutions are learned from a supervised setting, and need the entire video to be processed to produce effective proposals. These properties become a limitation for certain real applications in which a system requires to know the content of the video in an online fashion. To do so, in this work we introduce a live video action detection application which integrates the action classifier step with an unsupervised and online TAPs generator. We evaluate, for the first time, the precision of this novel pipeline for the problem of action detection in untrimmed videos. We offer a thorough experimental evaluation in ActivityNet dataset, where our unsupervised model can compete with the state-of-the-art supervised solutions.

1 Introduction

The problem of temporal action localization (TAL) in untrimmed videos has drawn a significant amount of attention from the research community [2, 3, 7, 10, 14, 28], owing to its applications in many areas like: intelligent video surveillance, robotics, human-computer interaction and human behavior analysis.

Technically, the goal of any TAL solution consists in identifying in a video what actions occur, and when they appear, by concretely determining the corresponding temporal windows. Interestingly, the best state-of-the-art TAL approaches split the problem in two steps, as it can be seen in Figure 1 Left. First, the methods integrate a Temporal Action Proposals (TAPs) generator. This module is responsible for identifying as many regions in the videos as possible, in order to maximize recall, which have a high probability of containing an action. Then, these action proposals are passed through an action recognition classifier.

All the approaches in this paradigm for TAL need to work in an *offline* fashion. This implies, for example, that in order to properly generate action proposals, these systems need access to the entire video, from beginning to end. From the point of view of applicability, the fact that they are offline is a major limitation. Let's think of a TAL solution for video surveillance, which has to detect when a dangerous situation occurs. Under these offline approaches, the alert would be generated *a posteriori*, once the action has been completed, not being able to anticipate it.

In this paper we want to change the current paradigm so that the TAL problem can be addressed from an online perspective. Figure 1 Right shows the novel approach we introduce. We continue separating the tasks of action classification and generation of action proposals. But we propose to approach the latter task with an *online* model that does not require supervision. This is the main contribution of our work, *i.e.* the integration in a TAL approach of an online and unsupervised TAPs generator. Our action proposals are generated as the video stream evolves, *i.e.* in live mode. As soon as an action proposal is identified, we can pass it to the action recognition module. All the details of our implementation can be found in Section 3. We also offer a thorough experimental evaluation of our model in the challenging ActivityNet [12] dataset (see Section 4). We provide a detailed comparison with the best state-of-the-art supervised TAL solutions, and results reveal that our approach can compete with them.

2 Related work

Currently, the best and most common way to solve the TAL problem involves two stages: temporal action proposals (TAPs) generation and action classification. But how did the research community get to this point?

At the beginning, the TAL task was interpreted in two different ways. On the one hand, some works, *e.g.* [16, 25, 30], considered the problem as an extension of object detection to videos and thus, they proposed to exhaustively process them with a 3D version of the sliding window used to detect objects in images, in order to spatio-temporally locate the action.

On the other hand, several authors have addressed the problem as a retrieval task where given an action query, the method had to search throughout all the video for the segments in which the action was being performed. In this case, the videos involved were untrimmed, *i.e.* irrelevant information could appear at some parts of the video. A classical work representing this interpretation of TAL is the one proposed by [9].

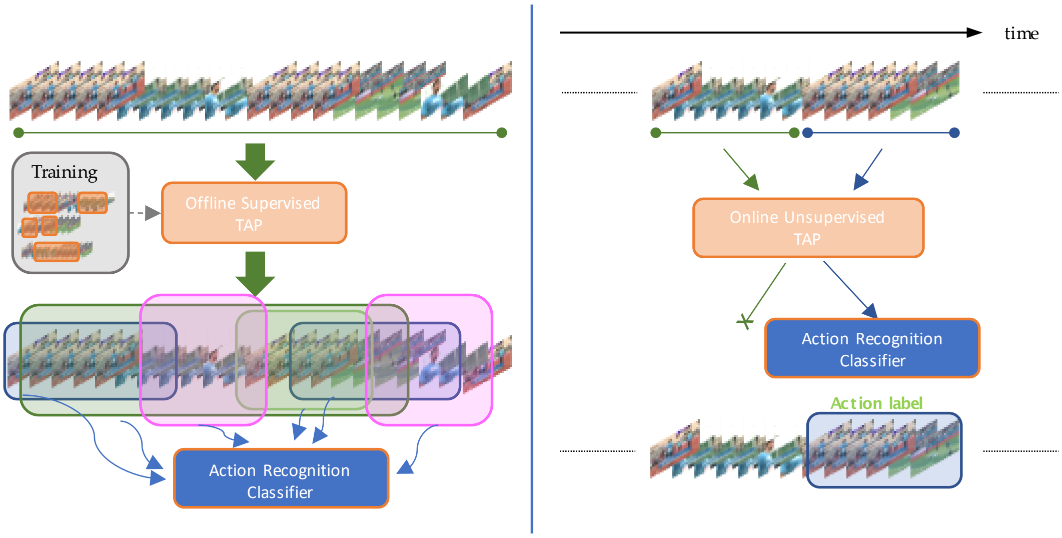

Figure 1. **Left**: We show the current state-of-the-art paradigm for TAL. First, an offline TAP generator is trained with supervised learning. The whole video is processed by the TAP approach to generate overlapped proposals. Then, an action classifier is run over them to predict the action categories. **Right**: This is our live video action recognition approach. The proposed pipeline works with live video streams that enter into the online and unsupervised TAP generator. This module produces action proposals online, or discards video segments considering them as background. Finally, action proposals pass through the action classifier.

Later on, this interpretation evolved to just temporally locate all the instances of a set of actions for a certain dataset without using any query, which is the way the task is understood today (*e.g.* [26, 28, 36]).

Driven by the success of the object proposals concept for detecting objects in still images [15], some authors proposed to extend this idea to the TAL problem, *e.g.* [11, 14, 22, 35]. In [14], Caba *et al.* proposed the topic of Temporal Action Proposals (TAPs), in which methods were only designed to find, in long untrimmed videos, those segments that can contain a human action. Then these segments are used to feed an action classification network (*e.g.* SCNN [28] or Untrimmed-Nets [31]) to cast the final action label. Nowadays, the majority of TAL approaches utilize this pipeline. As an example, TURN-TAP ([10]), CTAP ([11]), BSN ([22]), BMN ([20]), MGG ([24]) and DBG ([19]) have used the SCNN classifier over their proposals.

As a common denominator, all of the above works share that: i) they use TAPs models that have been trained following a supervised learning paradigm; and ii) they have not been designed to tackle the TAL problem from an online perspective. On the contrary, we introduce in this work an approach that integrates an *unsupervised* TAPs generator solution into the TAL pipeline. Moreover, we build with it a live video action classification application, that is able to predict the action in an online fashion. Other works address this online problem (*e.g.* [5, 6]), but they perform a frame-wise video analysis, without action proposals.

3 Live video action recognition application

We describe here a novel approach for live video action recognition from unsupervised temporal action proposals. Our pipeline performs an online video analysis, extracting the action proposal directly from the live video. Then, we integrate an action classifier to perform the action prediction. Figure 1 Right shows the whole system proposed. Unlike all state-of-the-art approaches, our model operates completely online: the video is processed as it arrives to the system, without accessing any information from the future.

3.1 Unsupervised online TAP

We build our application for live video by leveraging previous work on unsupervised TAPs [18, 1].

Technically, we follow [1] where a rank-pooling [8] based filtering is used to learn to discriminate, in an unsupervised fashion, between action and background temporal video segments that dynamically grow, using a Support Vector Classifier (SVC) based clustering model. While the SVC discriminates between contiguous sets of frames to generate candidate action proposal segments, the rank-pooling filtering computes the dynamics of these segments and applies a distance criterion between each segment dynamics and a randomised version of them to confirm or discard them as actual action proposals.

Given an online video stream $\mathcal{V}^i = \{v^i_n\}^{l_i}_{n=1}$ our application starts extracting any state-of-the-art deep learning feature representation for every frame or set of frames. Formally, the online sequence $\mathcal{V}^i$ is converted to a set of visual features $\mathcal{F}^i = \{f_n^i\}_{n=1}^{l_i}$, where $f_n^i \in \mathbb{R}^d$. Given the obtained features, the model begins processing the video $\mathcal{V}^i$ by accessing the first $2 \times N$ features in $\mathcal{F}^i$ to split them into two sets of N consecutive features, $\mathcal{S}^+_{t=1}$ and $\mathcal{S}^-_{t=1}$, *i.e.* $\mathcal{S}^+_{t=1} = \{f_1^i, f_2^i, \dots, f_n^i\}$ and $\mathcal{S}^-_{t=1} = \{f_{n+1}^i, f_{n+2}^i, \dots, f_{2n}^i\}$. Note that $t = 1$ because it is the first iteration of the process and that for every new iteration N new incoming features are evaluated. In the experiments, we fix $N = 32$, this choice gives the best results.

Next step consists in finding whether these two sets belong to the same segment. To do so, the two sets are artificially identified with two different labels $\mathcal{Y}^+_{t=1} = \{+1\}_{n=1}^N$ and $\mathcal{Y}^-_{t=1} = \{-1\}_{n=1}^N$, and the SVC proceeds to learn to separate them according to the labels. For our online application, we have opted for using a simple but effective linear kernel to separate the features.

At iteration t, once the online learning phase of the SVC is finished, the algorithm classifies the provided features and measures its performance by computing the classification error rate (Cer_t). Lastly, it evaluates Cer_t to decide whether to join the initial groups of features. A high Cer_t means that the SVC is not able to correctly separate the two sets. Hence, the two sets of features $\mathcal{S}^+_t$ and $\mathcal{S}^-_t$ should be joined in the same candidate proposal for the next iteration of the algorithm. On the other hand, a low Cer_t implies that the set $\mathcal{S}^+_t$ is different from $\mathcal{S}^-_t$ and can be considered a different proposal. A threshold α is defined to evaluate these conditions: if $Cer_t \geq \alpha$ then $\mathcal{S}^+_{t+1} = \mathcal{S}^+_t \bigcup \mathcal{S}^-_t$, the proposal size is increased for the next iteration; if $Cer_t < \alpha$, then $\mathcal{S}^+_{t+1} = \mathcal{S}^-_t$ and a new action proposal ap_k is generated from the set $\mathcal{S}^+_t$. For the experiments we fix $\alpha = 0.1$.

Following this approach, in each iteration the SVC module decides on which groups to make after learning and classifying based on the initial artificial labels. Each of the candidate segments that are generated has to also be evaluated by the next step: the Rank-pooling based filter.

Let ap_k be a candidate action proposal generated by the SVC module. First, a set $\mathcal{F}^{ap_k} = \{f_n\}_{n=1}^{l_{ap_k}}$ is built, where $f_n \in \mathbb{R}^d$ and l_{ap_k} encodes the size of the proposal. $\mathcal{F}^{ap_k}$ contains the ordered set of features for the video frames included in ap_k. Then, the set of features $\tilde{\mathcal{F}}^{ap_k}$ is generated, which is a randomly disordered version of $\mathcal{F}^{ap_k}$. Finally, a rank-pooling model similar to the one proposed by [8] is used to compute the dynamics of $\mathcal{F}^{ap_k}$ and $\tilde{\mathcal{F}}^{ap_k}$.

As in the rank-pooling model [8], the dynamic of a set of features is summarised as the parameters of a curve in the input space that captures the frame temporal order via linear projections. This is done by optimizing a pairwise-learning-to-rank problem based on an Support Vector Regression (SVR). In particular, we implement a rank-SVR with a linear SVR based formulation, which is known to be a robust point-wise ranking method [23].

Given any set of features $\mathcal{F} = \{f_t\}_{t=1}^{l} = \{f_1, f_2, \dots, f_l\}$, a direct projection of the input vectors f_t to a time variable t is obtained by employing a linear model with parameters $\omega^{\mathcal{F}}$, as follows, $\omega^{\mathcal{F}} = \arg\min_{\omega^{\mathcal{F}}} \sum_t |t - \omega^{\mathcal{F}} \cdot f_t|$, where $\omega^{\mathcal{F}}$ summarises the sequence of dynamics, becoming the pooled dynamics descriptor for $\mathcal{F}$.

The rank-pooling filtering mechanism computes the dynamics for $\mathcal{F}^{ap_k}$ and $\tilde{\mathcal{F}}^{ap_k}$, being them $\omega^{\mathcal{F}^{ap_k}}$ and $\omega^{\tilde{\mathcal{F}}^{ap_k}}$, respectively. As described above, the distance between these two dynamics vectors allows the model to identify action proposals, discarding candidates that might include background information. For a candidate that does not represent to any action, the distance between its dynamics and the dynamics of its randomised version should not be high. The Euclidean distance is used to implement this filtering mechanism in our application. We define a threshold r to discard background segments: if $d(\omega^{\mathcal{F}^{ap_k}}, \omega^{\tilde{\mathcal{F}}^{ap_k}}) < r$, the candidate proposal is rejected. In the experiments, $r = 1$ offers the best results.

3.2 Action Recognition Classifier

The last step of our model (see Figure 1 Right) classifies each of the resulting action proposals. For the classifier, we propose to integrate the Temporal Segment Network (TSN) [32] framework. In our implementation, we feed the TSN with the unsupervised action proposals. Every action proposal is divided into K segments (in the experiments, we fix $K = 7$), and for each segment a short snippet is randomly selected. Every snippet is processed by a spatial convolutional neural network. Note that, for greater efficiency, we do not integrate into our pipeline any temporal optical flow based input modality, contrary to [32]. So, our TSN only works with the RGB frames of the snippets. The class scores of different snippets are fused by a segmental consensus function to yield the action proposal-level prediction.

Formally, given a action proposal video segment ap_i, we divide it into $K = 7$ segments $\{S_1, S_2, \dots, S_7\}$ of equal duration. We then use the TSN to model the sequence of snippets $(T_1, T_2, \dots, T_7)$ as

$$\mathrm{TSN}(T_1, T_2, \dots, T_7) = \mathcal{H}(\mathcal{G}(\mathcal{F}(T_1; \mathbf{W}), \dots, \mathcal{F}(T_7; \mathbf{W}))). \tag{1}$$

$\mathcal{F}(T_i; \mathbf{W})$ is a function that represents a W-parameter convolutional neural network that operates on the short snippet T_i and generates class scores for all classes. $\mathcal{G}$ is the segmental consensus function. It basically incorporates all the snippet outputs to reach at a consensus on a class hypothesis. For our implementation, we use the average of the scores of the same category on all the snippets. Based on this consensus, our prediction function $\mathcal{H}$ is the Softmax function, which predicts the likelihood of each action class for the entire action proposal.

4 Experiments

4.1 Experimental setup

We proceed to evaluate our live video action recognition module from unsupervised TAPs using ActivityNet [12] dataset. As of now, it is the first time a TAL solution based on an unsupervised TAPs generator is evaluated on this dataset.

We report the performance of our model for both the TAPs and the TAL problems. The former is evaluated following the standard Average-Recall versus Average Number of Proposals per Video (AR-AN) metric [14]. As in the official ActivityNet challenge [13], for the TAP task the Average-Recall (AR) is defined as the mean of the recall values computed for the set of $tIoU$ thresholds $[0.5, 0.95]$, using a step of 0.05. To compare different methods, the Area Under the Curve (AUC) for the AR-AN plot is used.

For the experimental evaluation of the TAL problem we again follow the official metric proposed in the ActivityNet challenge. The interpolated Average Precision (AP) is used as the metric for evaluating the results on each activity category. Then, the AP is averaged over all the activity categories (mAP). Finally, we report the average mAP, which is defined as the mean of all mAP values computed with $tIoU$ thresholds 0.5, 0.75 and 0.95.

The implementation of the system is divided into the TAP generator and the action classifier. In the TAP module, videos are first processed with the C3D network [29], which is initialized with weights from the Sport1M [17] dataset. Then, the online action proposal generation is performed over the fc7 features of the network, for which we follow the implementation in [1][1]. On the other hand, for the action classifier we made use of the TSN implementation of [33][2], which relies on the ResNet200 network for the RGB stream.

4.2 Results

For the TAP problem, in Table 1 we start showing a comparison between our fully unsupervised and online approach to generate action proposals, and the current state-of-the-art *supervised* models for TAP. The best supervised model achieves 67.10 of AUC@100 proposals. Our unsupervised model recovers 41% of this model's performance, indicating a promising direction for unsupervised approaches.

Regarding the TAL problem, we report in Table 2 the results of our system, compared with the state of the art in ActivityNet. First, for the lowest $tIoU$ of 0.5, we report 24.55 of mAP. Note that the fully supervised model R-C3D [34] gets a close 26.8 mAP. This supports the effectiveness of our online and unsupervised model for generating proposals. Moreover, if we carefully analyze the results for the most restrictive $tIoU$ of 0.9, suddenly, our pipeline outperforms up to 3 different state-of-the-art models. This suggests that the type of action proposals we are able to generate are more precise, in terms of temporal localizations, than those employed by other fully supervised models. Overall, these results confirm that the live video action recognition approach from unsupervised action proposals can compete with TAL state-of-the-art models, and therefore be integrated into a real application.

Table 1. Comparison with the state-of-the-art for the problem of TAP on ActivityNet. The superscript s indicates the method is supervised. The best supervised model achieves 67.10 of AUC@100. Our unsupervised model achieves 27.61, so it is able to recover 41% of the best performing supervised method.

	AUC
[4]^s	59.58
[21]^s	64.40
CTAP [11]^s	65.72
BSN [22]^s	66.17
BMN [20]^s	67.10
Ours	27.61

Table 2. Comparison with the state-of-the-art for the problem of TAL on ActivityNet.

		$tIoU$		
	0.5	0.75	0.95	**Average mAP**
Ours	24.55	14.08	3.33	14.51
CDC [27]	45.3	26	0.2	23.8
R-C3D [34]	26.8	–	–	–
SSN [37]	39.12	23.48	5.49	23.98
Chao *et al.* [3]	38.23	18.3	1.3	20.22
BSN [22]	46.45	29.96	8.02	30.03
P-GCN [36]	48.26	33.16	3.27	31.11

5 Conclusion

This paper introduces a novel approach for live video action recognition. Technically, we propose to integrate an unsupervised and online TAP generator module, with a Temporal Segment Network action classifier, to tackle the TAL problem in live video streams. We provide a thorough experimental evaluation in the challenging ActivityNet dataset, where our solution offers promising results than can compete with those of state-of-the-art TAL models.

Acknowledgements This research was funded by: (a) project AIRPLANE, with reference PID2019-104323RB-C31, of Spain's Ministry of Science and Innovation; and (b) project CM/JIN/2019-022 of the University of Alcalá.

[1] https://github.com/gramuah/svc-uap

[2] https://github.com/yjxiong/anet2016-cuhk

References

[1] M. Baptista-Ríos, R. J. López-Sastre, F. J. Acevedo-Rodríguez, P. Martín-Martín, and S. Maldonado-Bascón. Unsupervised action proposals using support vector classifiers for online video processing. *Sensors*, 20(10), 2020.

[2] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles. Sst: Single-stream temporal action proposals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6373–6382, July 2017.

[3] Y. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1130–1139, June 2018.

[4] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen. Temporal context network for activity localization in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5727–5736, October 2017.

[5] R. De Geest and T. Tuytelaars. Modeling temporal structure with lstm for online action detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1549–1557, March 2018.

[6] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *European Conference on Computer Vision (ECCV)*, pages 269–284, October 2016.

[7] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision (ECCV)*, pages 768–784, October 2016.

[8] B. Fernando, E. Gavves, J. Oramas M., A. Ghodrati, and T. Tuytelaars. Rank pooling for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(4):773–787, April 2017.

[9] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(11):2782–2795, Nov 2013.

[10] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3648–3656, October 2017.

[11] Jiyang Gao, Kan Chen, and Ram Nevatia. Ctap: Complementary temporal action proposal generation. In *European Conference on Computer Vision (ECCV)*, pages 70–85, September 2018.

[12] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, June 2015.

[13] F. C. Heilbron, B. Ghanem, J. C. Niebles, and C. Snoek. Activitynet challenge 2020, 2020.

[14] F. C. Heilbron, J. C. Niebles, and B. Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1914–1923, June 2016.

[15] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(4):814–830, April 2016.

[16] Junsong Yuan, Zicheng Liu, and Ying Wu. Discriminative subvolume search for efficient action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2442–2449, June 2009.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, June 2014.

[18] Nadjia Khatir, Roberto J. López-Sastre, Marcos Baptista-Ríos, Safia Nait-Bahloul, and Francisco Javier Acevedo-Rodríguez. Combining online clustering and rank pooling dynamics for action proposals. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, pages 77–88, July 2019.

[19] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *AAAI Conference on Artificial Intelligence (AAAI)*, February 2020.

[20] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3888–3897, October 2019.

[21] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *ACM International Conference on Multimedia (ACMM)*, page 988â€“996, 2017.

[22] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *European Conference on Computer Vision (ECCV)*, pages 3–21, Cham, September 2018.

[23] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225â€“331, March 2009.

[24] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S. Chang. Multi-granularity generator for temporal action proposal. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3599–3608, June 2019.

[25] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal localization and categorization of human actions in unsegmented image sequences. *IEEE Transactions on Image Processing (TIP)*, 20(4):1126–1140, April 2011.

[26] D. Oneata, J. Verbeek, and C. Schmid. The lear submission at thumos 2014, 2014.

[27] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1417–1426, July 2017.

[28] Z. Shou, D. Wang, and S. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058, June 2016.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, December 2015.

[30] D. Tran and J. Yuan. Optimal spatio-temporal path discovery for video event detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3321–3328, June 2011.

[31] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6402–6411, July 2017.

[32] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pages 20–36, Cham, October 2016.

[33] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. CUHK & ETHZ & SIAT submission to ActivityNet challenge 2016, 2016.

[34] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5794–5803, October 2017.

[35] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[36] R. Zeng, W. Huang, C. Gan, M. Tan, Y. Rong, P. Zhao, and J. Huang. Graph convolutional networks for temporal action localization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7093–7102, October 2019.

[37] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2933–2942, October 2017.