IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 12, 2019, accepted September 2, 2019, date of publication September 10, 2019,
date of current version September 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940407

A Novel Method for Temporal Action Localization and Recognition in Untrimmed Video Based on Time Series Segmentation

JICHAO LIU, CHUANXU WANG, AND YUN LIU
School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266100, China
Corresponding author: Chuanxu Wang (wangchuanxu_qd@qust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61472196 and Grant 61672305.

ABSTRACT Positioning of each action in a long complicated video is a challenging task in computer vision. To address this issue we propose a method with temporal boundary regression based on time series segmentation, which can generate proposals with flexible temporal duration. Firstly, we use a clustering algorithm to generate proposals, which is more efficient than sliding window method. It generates proposals by aggregating areas of high-probability behavior in time domain, and uses non-maximum suppression to remove redundancy. Then a multi-layer perceptron is used to refine boundary regression of behavior proposals, the process makes boundary coordinates closer to the real boundaries. Secondly, each behavioral proposal is represented by concatenating a three-subsegment feature description, which includes the proposal segment, its starting subsegment and its ending subsegment. Finally, the behavior proposal including a target action is identified by multi-layer perceptron. Our method is evaluated in two large data sets THUMOS14 and ActivityNet, which are commonly used in time series behavior detection task. The recognition rates can reach 30.1% and 33.19% respectively, which proves that the method can effectively improve the classification accuracy.

INDEX TERMS Temporal action localization, action proposals, two-stream convolutional networks, temporal segmentation.

I. INTRODUCTION

With the rapid growth of surveillance and mobile cameras, the amount and size of video databases have been increasing. Video analysis has become an active research field. Behavior/action recognition is one of the most popular research topics in video analysis. The goal of human behavior recognition is to automatically analyze the ongoing behavior in an untrimmed video or image sequence. At present, the behavior recognition research mainly focuses on individual behaviors in short videos, but in our daily life, most videos are complex long videos containing multiple different behaviors, which needs another recognition algorithm: Temporal Action Localization. This task is to detect human action instances in untrimmed long videos, along with the start and end times of each of the actions. This algorithm can be applied to many aspects, such as automatic retrieval and intelligent monitoring.

The associate editor coordinating the review of this manuscript and approving it for publication was Hiram Ponce.

Temporal Action Localization can usually be divided into two parts, temporal action proposal and action recognition. The main goal of temporal action proposal is to generate some temporal boundaries of action instances without classifying their categories. The task of the classification is to decide the categories of the temporal action proposals and specify the start time and the end time. Although the traditional behavior recognition has reached high accuracy, temporal action detection is still one of the most challenging problems to be resolved [1], [2]. Therefore, how to produce high-quality behavior proposal is the key problem in the study [3]–[6]. In order to obtain a high-quality proposal, the proposal generated in the proposal generation phase needs to be flexible in duration to cope with the problems caused by the different duration and large gaps of the video clips. And the generated proposals should have precise time boundaries. Some recently-raised proposal generation methods [3]–[5], [7] make use of different lengths of sliding windows to generate proposals and evaluate the proposed confidence with the trained models. But the method which predefined

duration and interval to generate the proposal has some obvious shortcomings: (1) the start and end time is not accurate; (2) the length of the fixed behavior segment cannot handle the behaviors of different duration. When the duration of different behavioral actions is large, the requirements for different duration cannot be met, which will thus increase the number of sliding windows and cause redundant calculations.

The recent studies [7]–[9] applied deep neural networks to the detection framework and achieved better performance. S-CNN [7] proposed a multi-stage convolutional neural network, which improves the accuracy recognition by using a positioning network. However, since S-CNN generates behavioral proposals by using a sliding window, and C3D [10], as a feature extractor initially used as unit classifiers, can only hold 16 frames as input, it will take a large amount of time to calculate when dealing with behavior detection tasks of time series. Another study [8] uses a recurrent neural network (RNN) to learn a strategy for predicting the start and end points of an action. This sequential prediction is often time-consuming in long video processing, and it does not support joint training of frame-by-frame CNN for feature extraction.

In the above context, in order to overcome the shortcomings of the sliding window and generate high-quality behavior proposals, this paper proposes a time-series behavior detection model based on the spatial and temporal network [11]. We extract the features from spatial and temporal network. And then using a clustering algorithm to generate proposals which is more efficient than sliding window method. It generates proposals by aggregating areas of high-probability behavior in time domain and uses non-maximum suppression to remove redundancy. After that we use multi-layer perceptron to refine boundary regression of behavior proposals. Each behavioral proposal is represented by concatenating a three-subsegment feature description. Finally, we use multi-layer perceptron to do classification. These components are integrated into a unified novel network for the Temporal Action Localization task. Our experimental results show that the method can effectively improve the classification accuracy.

II. PROPOSED APPROACH

A. OVERVIEW OF THE WHOLE ALGORITHM

This paper proposes a model based on time series segmentation, as is shown in Figure 1. Firstly, the dual-stream convolutional neural network is used to extract the feature sequence of the long video. The Temporal Actionness Grouping (TAG) [12] method is used to flexibly generate the behavior proposal on the feature sequence, which works as the input of the model. The multi-layer perceptron is used to iterate the start and end boundaries of each behavior proposal. This process can process the boundary of the behavior proposal more precisely and make it closer to the real boundary information. Each behavioral proposal will be redesigned using a three-segment feature description. The behavioral proposal

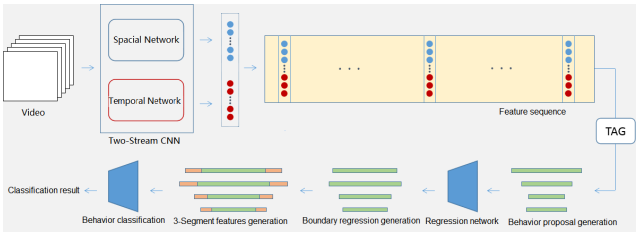

FIGURE 1. Human behavior recognition model based on time series segmentation.

will be split into 3 parts (starting, course and ending), and concatenated to form the global representations. Finally, the behavior proposal including the target action is identified and the classification result is obtained.

B. PROBLEM DESCRIPTION

We denote an untrimmed long video as $X = \{x_n\}_{n=1}^{N}$, where x_n is the n-th frame in X. The action annotation of each video X consists of a set of action instances $\psi_g = \{\phi = (t_{s,n}, t_{e,n})\}_{n=1}^{A_n}$, in which A_n is the total number of real action instances in X, $t_{s,n}$ $t_{e,n}$ are the start time and end time of the action instance ϕ_n.

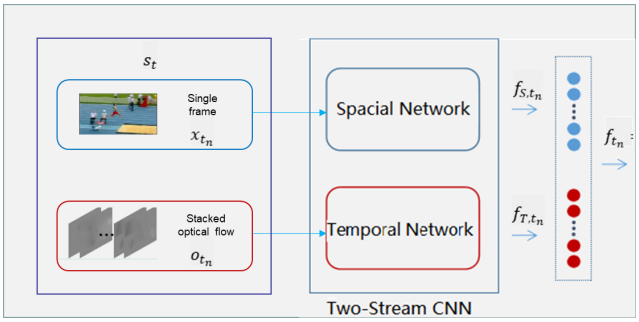

FIGURE 2. Feature extraction.

C. FEATURE EXTRATION

In order to extract the characteristics of the dual-stream convolutional neural network as shown in Figure.2, the video is divided into T consecutive equal-length and non-overlapping units. Thus, the video can be represented as $X = \{s_t\}_{t=1}^{T}$, in which T represents the number of units in the video, and unit $s_t = (x_{t_n}, o_{t_n})$ represents two parts, x_{t_n} is the t_n-th RGB frame in video X, and o_{t_n} is a stacked optical flow field centered on the x_{t_n}-frame. After that the two networks separately capture spacial and temporal information. The spacial part, in the shape of individual frame appearance, conveys information about scenes and objects. The temporal part, in the shape of motion across the frames, carries the movement of the observer and the objects. The two parts are concatenated to feature vector $f_{t_n} = (f_{S,t_n}, f_{T,t_n})$, Where f_{S,t_n}, f_{T,t_n} represent the output vectors of the spacial network and the temporal network, respectively. Therefore, if a sequence of elements S of length l_s is given, the feature sequence $F = \{f_{t_n}\}_{n=1}^{l_s}$ can be

extracted. The dual stream convolution feature sequence will be sent to the TAG network to generate behavioral proposals.

D. BEHAVIOR PROPOSAL

Compared with the sliding window, the TAG method can flexibly generate motion proposals of different lengths without requiring a lot of calculations.

The TAG method uses a behavioral classifier to evaluate the probability of an action occurring in each cell. This behavior classifier is a binary classifier. The basic idea of the method is to find a continuous region of high probability. To achieve this goal, the method redesigns a classical watershed algorithm and applies it to the onedimensional action probability value. The method can obtain a series of ''basins'' by setting different ''water levels'', and each basin corresponds to a high probability region in the time domain.

Given a series of basins G, a clustering method similar to [13] was chosen to try to connect small basins into nominated areas. The work-flow of the program is as follows: starting with a seed basin and continuously absorbing subsequent basins until the portion of the basin falls to a certain threshold below Y over the entire duration (i.e., from the first basin to the end of the last basin). In this way, a set of regions, denoted by, can be generated starting from different seed basins. Note that and are not specific combinations selected, but are evenly sampled between (0, 1) with a step size of 0.05. The combination of these two thresholds will result in multiple sets of regions. Then, combine the sets of regions and use non-maximum suppression to filter areas with high overlap and set the IoU threshold to 0.95. The resulted/generated behavioral proposal will be sent to multi-layer perceptron for boundary regression.

E. BOUNDARY REGRESSION

The basic idea of performing boundary regression on the time domain is to use neural networks to infer the boundaries of behavioral proposals. In this paper, the multi-layer perceptron is used as the regression network, the behavioral proposal is taken as the input, and the coordinate regression offset is the output. The specific calculation is shown in Equation 1.

$$o_s = s_{clip} - s_{gt}, \quad o_e = e_{clip} - e_{gt} \tag{1}$$

Here s_{clip}, e_{clip} are the start and end coordinates of the input behavior proposal, respectively, and s_{gt}, e_{gt} are the start and end coordinates of the corresponding real data, respectively. The coordinate regression model used in this paper has two advantages: First, the use of unit-level coordinate regression matches the way in which the dual-stream convolutional neural network extracts features based on the unit, and the computational cost is also relatively small; Second, the offset of the starting coordinates is directly used as the regression result instead of the coordinate parameterization. This is because the coordinate regression of the behavior proposal is performed in the time domain, and the spatial coordinate regression is performed in the spatial domain. Since the target can be rescaled in the image due to camera projection, it is necessary to first normalize the frame coordinates to a certain standard scale. Time domain coordinates can rely on the time domain itself as a standard scale and do not need to be parameterized.

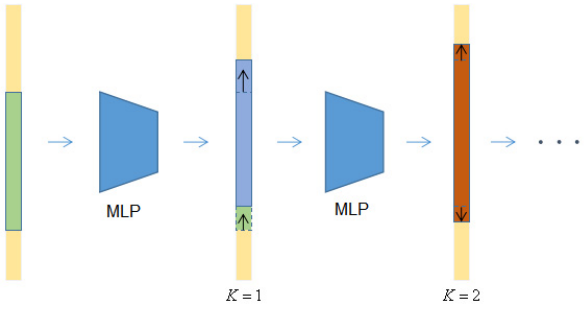

FIGURE 3. Boundary regression network processing behavior proposal boundary.

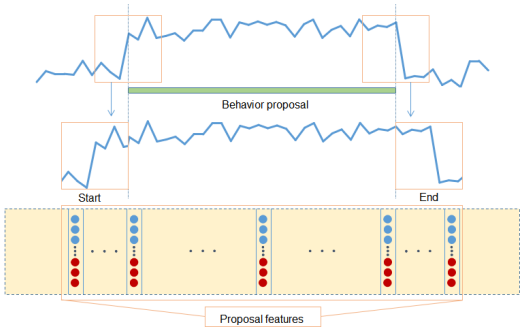

FIGURE 4. Feature construction of behavioral proposal.

As is shown in Figure 2, the boundary regression task of this paper is completed by the multi-layer perceptron using iterative method. The output of the boundary regression is sent to the multi-layer perceptron as an input and calculated repeatedly. With a few rounds of repeated calculations, more accurate results are obtained. By taking the behavior proposal as an input, the regression model outputs the coordinate regression offset in the time domain and obtains the boundary coordinate value after the regression. For this layer network, the input value $p_c = [t_s, t_e]$ is given to a candidate proposed boundary data, the output data $p_c^1 = [t_s^1, t_e^1]$ will be used as an input for the second round of boundary regression calculation. The output of the second round is $c^2 =< s^2, e^2 >$. The iterative process is performed a total of K times, and the final boundary result is:

$$p_c^K =< p_s^K, p_e^K >, \quad p = \prod_{i=1}^{K} p_i \tag{2}$$

F. PROPOSED FEATURE

To establish the proposed feature shown in FIG. 3, the scope of the behavior proposal ϕ, is defined as the interval $p_c = [t_s, t_e]$, and the duration of the proposal ϕ is $d = t_e - t_s$. The start and end intervals associated with it are $p_s = [t_s - d/4, t_e + d/4]$ and $p_e = [t_e - d/4, t_e + d/4]$ respectively. The proposed feature $f_\phi = (f_{ps}, f_{pc}, f_{pe})$ of the

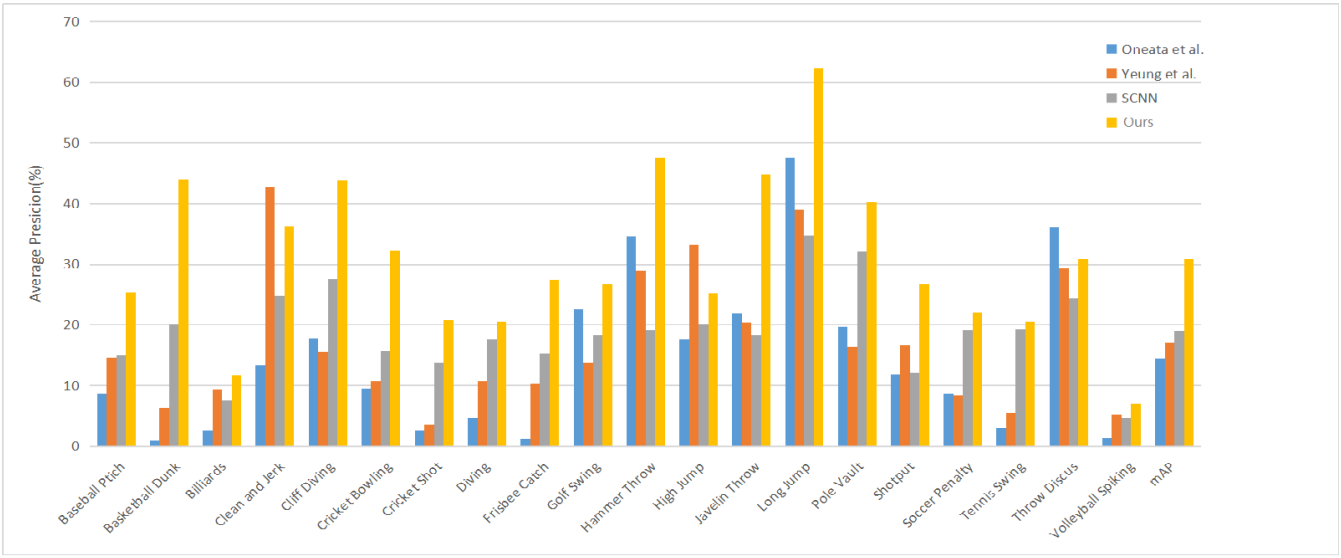

FIGURE 5. Per-class AP at IoU threshold 0.5 on THUMOS14.

candidate offer ϕ can be obtained by splicing the vectors corresponding to the three parts of the start, end and perform of the interval. The proposed feature has good robustness. After the introduction of the start interval and the end interval, the behavior proposal feature is provided with context information.

G. BEHAVIOR CLASSIFICATION

A classifier is commonly used in deep learning networks. This paper chooses a multi-layer perceptron network as a multi-classifier after feature construction. For time-series behavior detection tasks, the multi-layer perceptron network outputs n+1 probability values, where n represents the number of behaviors in the data set and 1 represents the background class. Each probability value represents the probability of belonging to a certain type of behavior, and the behavior corresponding to the maximum probability value is taken as the result of the behavior classification.

In order to obtain better experimental results, this paper adopts a multi-task loss function to jointly train boundary regression and behavior classification networks.

$$L = L_{cls} + \lambda L_{reg}$$

Among the equation, L_{cls} is the classification loss function. For the multi-classification task in this paper, the multi-class cross entropy function is used as the loss function. L_{reg} is the boundary regression loss function and λ is the hyper parameter. The regression loss function is:

$$L_{reg} = \frac{1}{N}\sum_{i=1}^{N}\sum_{z=1}^{n} l_i^z \left[R\left(o_{s,i}^{'z} - o_{s,i}^{z}\right) + R\left(o_{e,i}^{'z} - o_{e,i}^{z}\right)\right]$$

Here R is the Manhattan distance, N is the batch size, n is the total number of behavior categories, l_i^z is the label. When the i-th sample belongs to the z-class, l_i^z, otherwise, $l_i^z = 0$. $o^{'}$ is the regression offset and o is the true value. The learning rate is set to 0.005 and the batch size is set to 128.

III. EXPERIMENTAL CLASSIFICATION RESULTS AND ANALYSIS

This paper conducts experiments on ActivityNet v1.3 [1] and THUMOS 2014 [22] data sets. This section will first introduce these two data sets and other experimental settings, and will then compare the performances with those of other methods, and finally analyze the experimental results.

A. DATASETS INFORMATION AND ITS SPLIT METHOD

As a large data set for timing behavior detection, ActivityNet v1.3 [1] contains 19994 long videos with 200 types of action annotations, and were used in the 2017 and 2018 ActivityNet challenges. ActivityNet is divided into training set, verification set and test set according to the ratio of 2: 1: 1.

In THUMOS 2014 [22], there are 1010 videos for verification and 1574 videos for testing. These videos contain 20 types of target actions with behavioral annotations. The data set has no training set and uses the UCF101 data set as a training set. Since the training set does not provide time annotations, this paper trains the model on the validation set and performs experimental tests on the test set. Therefore, 220 videos with 20 types of behavioral annotations were used for training. In this paper we present the analysis of the experimental results and compare them to other models on THUMOS 2014 and ActivityNet v1.3.

B. EXPERIMENTAL SETTINGS

We implement our model using Caffe. To train with sparse sampling, we use a sparse snippet sampling scheme. The parameters in the model were learned using the SGD method, with a batch size of 128 and a momentum of 0.9. The dual-stream convolutional neural network takes the ResNet [23]

network as the spatial network and the BN-Inception [24] network as the time network. The initial learning rates of the spatial network and the time network are set to 0.001 and 0.005, respectively. In ActivityNet v1.3, the number of iterations of spatial network and time network iterations was 9,500 and 20,000, respectively, and the learning rate was reduced by 0.1 after every 4000 and 1000 iterations. In THUMOS 2014, the spatial network and the time network performed 1000 and 6,000 iterative training respectively, and the learning rate was reduced by 0.1 per 400 and 2,500 times. In the feature extraction process, the cell spacing σ is set to 16. The binary behavior classifier used in the TAG method carries out trains by using the training set of each data set. In the process of boundary regression, $K = 3$.

C. RESULTS AND ANALYSIS

Evaluation Criteria: AvtivityNet v1.3 [1] and THUMOS 2014 [22] have uniform evaluation criteria, so the mean Average Precision (mAP) of different IoU thresholds is tested according to their evaluation criteria. In the ActivityNet v1.3 data set, the IoU threshold for the required test is {0.5, 0.75, 0.95}, and the average of the mAP for the IoU threshold range [0.5: 0.05: 0.95] is used to compare performances between different methods. In the THUMOS 2014 data set, the IoU threshold for the required test is {0.1,0.2, 0.3, 0.4, 0.5}. The mean Average Precision which is obtained when the threshold is 0.5 is used to compare the experimental results of different methods.

The algorithm and other time series behavior detection methods are compared on the THUMOS 2014 data set and the ActivityNet v1.3 data set, as is shown in Table 1 and Table 2. It can be found from Table 1 and Table 2 that on these two data sets, the algorithm identification accuracy proposed in this paper is higher than those of other algorithms, and the recognition effect is better. In this paper, with the combination of the motion surface features and timing information, the features acquired by the dual-stream convolutional neural network better explore the information contained in the video. The behavioral proposal is more accurate after iterative processing by the multi-layer perceptron. The subsequent three-stage feature design is integrated with the context information. On the one hand, it establishes a more comprehensive behavior description. On the other hand, it improves the accuracy of behavior recognition.

We also analyse the average classification accuracy (Average Precision, AP) for each behavior category of the algorithm proposed in this paper and the other three algorithms in the THUMOS14 data set, the time domain overlap rate threshold is 0.5. Figure 4 shows that our method is better than the other three methods in most of the behavior categories, especially in the dunk. It is because the high probability behavior clustering method in the time domain can effectively aggregate the interval of behavior occurring. The behavior boundary in the dunk video is rather vague, so the traditional methods does not work well. The behavior proposal generation method in this paper can produce more accurate behavior proposals and also the feature construction method that introduces context information improved the accuracy of behavior recognition.

TABLE 1. Action detection results on THUMOS14, measured by mAP at different IoU thresholds.

THUMOS14, mAP					
Method	0.1	0.2	0.3	0.4	0.5
Wang et.al. [14]	18.2	17.0	14.0	11.7	8.3
Oneata et. al. [15]	36.6	33.6	27.0	20.8	14.4
Richard et.al. [16]	39.7	35.7	30.0	23.2	15.2
S-CNN [7]	47.7	43.5	36.3	28.7	19.0
Yeung et.al. [8]	48.9	44.0	36.0	26.4	17.1
TCN [18]	-	-	-	33.3	25.6
STPN [17]	52.0	44.7	35.5	25.8	16.9
Ours	68.9	59.0	53.4	40.6	30.1

TABLE 2. Action detection results on ActivityNet v1.3, measured by mean average precision (mAP) for different IoU thresholds and the average mAP of IoU thresholds from 0.5 to 0.95.

ActivityNet v1.3(testing), mAP				
Method	0.5	0.75	0.95	Average
Wang et.al.[19]	42.48	2.88	0.06	14.62
Singh et.al.[20]	28.67	17.78	2.88	17.68
TCN [18]	37.49	23.47	4.47	23.58
R-C3D + Boundary [21]	27.82	15.00	2.82	15.68
Ours	45.73	30.56	7.46	33.19

D. COMPUTATIONAL COST

To train with sparse sampling, we use a sparse snippet sampling scheme. Additionally, a simple yet effective temporal action proposal scheme is used to generate high quality action proposals. And the sparse proposals are conducive to the detection performance. Though we used above methods to improve performance, we found that the computational bottleneck was on the optical flow extraction. It takes about 60 milliseconds to extract optical flow per frame. So further speedup is possible to improve the optical flow extraction.

IV. CONCLUSION

In order to fully acquire the spatio-temporal information in the video, a dual-stream convolutional neural network is used to construct the feature descriptor. Then the candidate behavior proposal is generated by the TAG method.

After multiple iterations, more accurate boundary information is obtained, and the behavior proposal is extended to three segment for behavior recognition. Based on the combination of time series information, the method generates high-quality action nominations with more accurate timing boundary and improved recognition rate. The experimental results show that the method can get good results on the THUMOS14 data collection and the ActivityNet dataset. However, it is still insufficient for the accuracy rate of timing positioning. The future study is to streamline the experimental steps and try to obtain more accurate timing boundaries.

REFERENCES

[1] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 961–970.

[2] Y. G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. S. M. Laptev, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," in *Proc. ECCV Workshop*, Sep. 2014.

[3] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "SST: Single-stream temporal action proposals," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2017, pp. 6373–6382.

[4] F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1914–1923.

[5] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "DAPs: Deep action proposals for action understanding," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 768–784.

[6] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "TURN TAP: Temporal unit regression network for temporal action proposals," 2017, *arXiv:1703.06189*. [Online]. Available: https://arxiv.org/abs/1703.06189

[7] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1049–1058.

[8] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proc. CVPR*, Jun. 2016, pp. 2678–2687.

[9] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, "Online action detection," in *Proc. ECCV*. New York, NY, USA: Springer, 2016, pp. 269–284.

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. ICCV*, Dec. 2015, pp. 4489–4497.

[11] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[12] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, D. Lin, and X. Tang, "Temporal action detection with structured segment networks," 2017, *arXiv:1704.06228*. [Online]. Available: https://arxiv.org/abs/1704.06228

[13] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," Mar. 2015, *arXiv:1503.00848*. [Online]. Available: https://arxiv.org/abs/1503.00848

[14] L. Wang, Y. Qiao, and X. Tang, "Action recognition and detection by combining motion and appearance features," *THUMOS Action Recognit. Challenge*, vol. 1, no. 2, p. 2, Sep. 2014.

[15] D. Oneata, J. Verbeek, and C. Schmid, "The LEAR submission at Thumos 2014," in *THUMOS Action Recognition Challenge*. Aug. 2014. [Online]. Available: https://hal.inria.fr/hal-01074442

[16] A. Richard and J. Gall, "Temporal action detection using a statistical language model," in *Proc. CVPR*, Jun. 2016, pp. 3131–3140.

[17] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6752–6761.

[18] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen, "Temporal context network for activity localization in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5793–5802.

[19] R. Wang and D. Tao, "UTS at activityNet," *AcitivityNet Large Scale Activity Recognit. Challenge*, vol. 8, p. 2016, Jul. 2016.

[20] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, "A multi-stream bi-directional recurrent neural network for fine-grained action detection," in *Proc. CVPR*, Jun. 2016, pp. 1961–1970.

[21] W. Kong, N. Li, T. Li, G. Li, and S. Liu, "BLP—Boundary likelihood pinpointing networks for accurate temporal action localization," 2018, *arXiv:1811.02189*. [Online]. Available: https://arxiv.org/abs/1811.02189

[22] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. (2014). *THUMOS Challenge: Action Recognition With a Large Number of Classes*. [Online]. Available: http://crcv.ucf.edu/THUMOS14/

[23] K. He, X. Zhang, J. Sun, and S. Ren, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.