GRAPH REGULARIZATION NETWORK WITH SEMANTIC AFFINITY FOR WEAKLY-SUPERVISED TEMPORAL ACTION LOCALIZATION

Jungin Park[†] *Jiyoung Lee*[†] *Sangryul Jeon*[†] *Seungryong Kim*[‡] *Kwanghoon Sohn*[†]

[†]School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
[‡]School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
E-mail: khsohn@yonsei.ac.kr

ABSTRACT

This paper presents a novel deep architecture for weakly-supervised temporal action localization that predicts temporal boundaries with graph regularization. Our model not only generates segment-level action responses but also propagates segment-level responses to neighborhood in a form of graph Laplacian regularization. Specifically, our approach consists of two sub-modules; a class activation module to estimate the action score map over time through the action classifiers, and a graph regularization module to refine the estimated action score map by solving a quadratic programming problem with the predicted segment-level semantic affinities. Since these two modules are integrated with fully differentiable layers, the proposed network can be jointly trained in an end-to-end manner. Experimental results on Thumos14 and ActivityNet1.2 demonstrate that the proposed method provides outstanding performances in weakly-supervised temporal action localization.

Index Terms— weakly-supervised temporal action localization, graph Laplacian regularization, semantic affinity

1. INTRODUCTION

Temporal action localization in untrimmed videos is essential for comprehensive video understanding tasks including event detection [1], video summarization [2], and visual question answering [3]. Over the past few years, impressive improvement has been made in a fully supervised setting [4, 5, 6, 7, 8], requiring the full annotation of the temporal boundaries as ground-truth labels. However, manually annotating temporal boundaries for each action instance is very expensive and time-consuming. Furthermore, the ambiguous temporal extent of actions induces subjective and imprecise annotations, making supervised learning less feasible.

To alleviate these issues, several recent methods [9, 10, 11, 12] have been proposed to use only video-level action labels that are much easier to collect compared to the temporal boundary annotations. Specifically, given action class labels of videos, they pass several randomly sampled segments through an action classifier and then average the action scores to yield a video-level class prediction. As a pioneering work, UntrimmedNet [9] offered a way to deal with insufficient training data by detecting automatically discriminative parts in the temporal domain to minimize the video action classification error. Similarly, Hide-and-seek [13] carried out temporal action localization by randomly hiding several frames from video and attending on salient regions in remaining frames. More recently, some approaches extend this classification framework by additionally employing attention module [10], supervisions from contrastive boundaries [11], or multiple instances [12]. Nguyen *et al.* [10] proposed to identify a sparse subset of key segments using an attention module and fuse the key segments through adaptive temporal pooling. Shou *et al.* [11] introduced the Outer-Inner-Contrastive (OIC) loss to automatically adjust boundaries of actions and background by maximizing the discrepancy between inner boundary and outer one. Paul *et al.* [12] proposed the co-activity similarity loss which is based on the motivation that a pair of videos having at least one activity category in common should have similar features in the temporal regions which correspond to that activity. Although aforementioned techniques provide promising results without temporal annotations, they have two major weaknesses in their formulation; 1) while features are sufficiently trained to represent the entire video, they have difficulties on localizing fine-grained activities since only video-level supervisions are provided. 2) Moreover, action score map is computed segment-by-segment without considering temporal consistencies, making it difficult to perform the precise temporal localization.

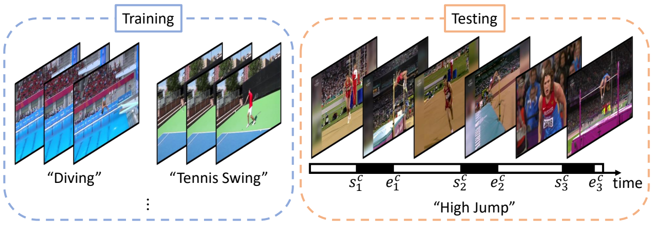

Fig. 1. Illustration of our weakly-supervised temporal action localization: We use only video-level action annotations to learn our framework during training. In testing phase, we classify action classes and localize temporal boundaries of action instances.

To overcome these limitations, we propose a novel deep architecture that imposes temporal smoothness by leveraging graph Laplacian regularization with learned semantic temporal affinities. The key intuition is that a pair of video segments corresponding same action classes should have strong semantic affinity in a graph with segments as a nodes. Specifically, we formulate two subnetworks: *class activation module* to extract a class activation score map and *graph regularization module* to construct a graph to generate class-agnostic affinity matrix by linking semantic neighbor segments that are close in the affinity space. Based on class-agnostic affinity matrix, the class-specific activation scores are propagated to semantically similar segments by using the graph Laplacian regularization techniques. Furthermore, we propose an affinity loss that leverages a pseudo semantic affinity matrix label from the class activation score map. The networks are jointly learned with the classification loss using the video-level annotation and the affinity loss. We employ the two-stream architecture, *i.e.*, RGB and flow streams, to boost the localization performance by considering the spatial and motion information both. In the testing phase, we combine the final

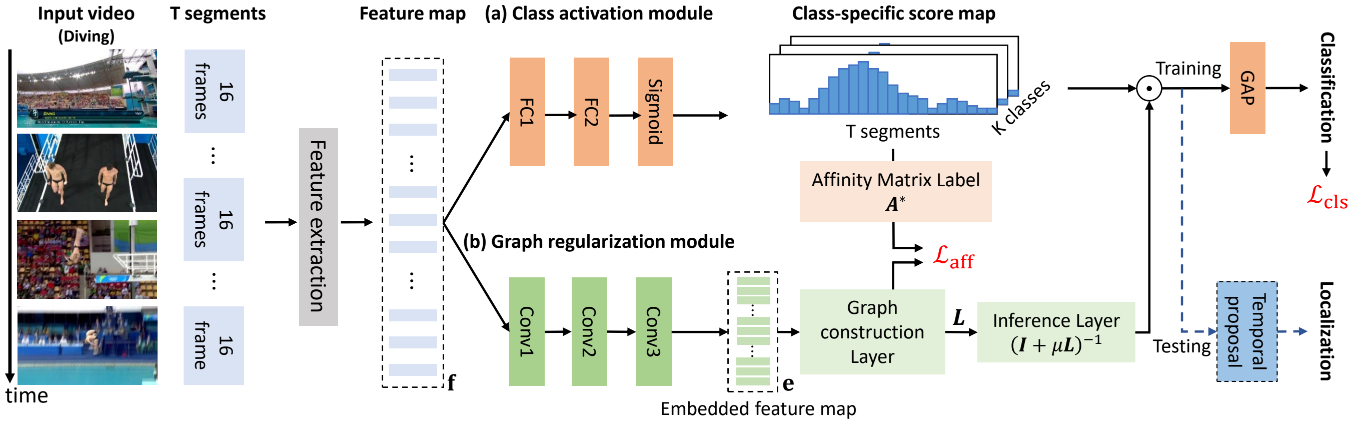

Fig. 2. Illustration of the proposed network for weakly-supervised temporal action localization. It consists of two sub-modules including the class activation module and the graph regularization module. The class activation module estimates a class-specific score map which represents probability for each class with respect to each video segment. The graph regularization module generates a class-agnostic affinity matrix and propagates the score by the graph Laplacian regularization with embedded semantic features.

class score maps from each identical network. Experimental results demonstrate our model outperforms previous approaches on several benchmarks.

2. PROPOSED METHOD

2.1. Problem Formulation

Let us define an untrimmed video V composed of a sequence of N segments as $v_{1:N} = \{v_1, v_2, ..., v_N\}$, where one segment v_n contains several consecutive frames. The objective of temporal action localization is to discover the start time t_s^c and end time t_e^c with one confidence score per action label $c \in \{1, ..., C\}$, where C is the total number of action labels. The overall architecture of our weakly-supervised action localization approach is illustrated in Fig. 2.

2.2. Network Architecture

2.2.1. Class Activation Module

The class activation map indicates the discriminative segment features to identify its action class. Based on the idea in [14], the class activation module learns the weights using video-level labels with global average pooling which outputs the temporal average of the feature map from the last fully-connected (FC) layer. Then, the class activation map is computed as a weighted sum of the feature maps of the previous FC layer.

Specifically, the class activation module consists of two FC layers which have 512 and C channels, respectively. We add a ReLU function between the two FC layers. Finally, we obtain the class activation score s_t^c for a given class c from the output of the first FC layer $f_{\text{fc1},t}$ of t-th segment, which is given by

$$s_t^c = \sigma(\sum_k w^c(k) f_{\text{fc1},t}(k)), \tag{1}$$

where $w^c(k)$ denotes k-th element of the parameter w in the last fully-connected layer for the class c and $\sigma(\cdot)$ is the sigmoid function to obtain class scores between 0 and 1. Although the scores for each segment are sub-optimal to represent the precise temporal boundaries, semantic affinities between segments are not considered in generating class activation scores.

2.2.2. Graph Regularization Module

Embedding feature affinity. Since the class activation score map is not optimal to identify the time intervals, we employ graph Laplacian regularization techniques [15] to refine the class activation score map by considering semantic affinities between segments. To consider semantic affinities between segments, we embed the features to a latent space to identify the correlations of feature representations from each segment instead of using the input feature itself. The affinity feature space represents the semantic affinities between segments. While preserving the temporal correspondences between the features, a set of the embedded feature $\mathbf{e}$ is obtained through a feed-forward such that,

$$\mathbf{e} = \mathcal{F}(\mathbf{f}; \mathbf{w}), \tag{2}$$

where $\mathbf{w}$ represent model parameters to project input features into feature affinity space and $\mathcal{F}$ is feature embedding network. The feature embedding network is composed of three convolution layers and ReLU layers after every convolution layers with batch normalization. The input feature is projected to the affinity feature space with lower dimensions.

Generating affinity matrix. To compute the Laplacian matrix $\mathbf{L}$, the embedded feature $\mathbf{e}$ is fed into the graph construction layer. Given an appropriate neighborhood graph $\mathcal{G}$ with N vertices, graph Laplacian regularization techniques assume that the optimal class activation scores connected to high affinity are smooth with respect to $\mathcal{G}$. We first compute the adjacency matrix $\mathbf{A} \in \mathbb{R}^{T \times T}$ of graph $\mathcal{G}$ which is used as the affinity matrix in our approach. As defined in [15], the edge weight w_{ij} of $\mathbf{A}$ between two embedded features $\mathbf{e}_i$ and $\mathbf{e}_j$ is computed as,

$$w_{ij} = \exp(-||\mathbf{e}_i - \mathbf{e}_j||^2 / 2\epsilon^2), \tag{3}$$

where ϵ is a constant to regulate sensitivity for the distance between features. The degree matrix $\mathbf{D}$ is a diagonal matrix whose i-th diagonal entry is $\sum_{j=1}^{N} w_{ij}$, and then the graph Laplacian matrix $\mathbf{L}$ is given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, which includes the graph Laplacian regularizer.

Solving graph regularization. To discover the optimal class activation score map $\{\hat{\mathbf{s}}^{*c}\}_{c=1}^{K} \in \mathbb{R}^N$ with respect to the Laplacian matrix L, a maximum a posteriori problem can be formulated as follows:

$$\hat{\mathbf{s}}^{*c} = \arg\min_{\hat{\mathbf{s}}^c} (||\mathbf{s}^c - \hat{\mathbf{s}}^c||_2^2 + \mu \cdot \hat{\mathbf{s}}^{cT} \mathbf{L} \hat{\mathbf{s}}^c), \tag{4}$$

where the prior term is a ℓ_2-norm computing the difference between the score vector $\mathbf{s}^c$ and the refined score vector $\hat{\mathbf{s}}^c$ for the class c,

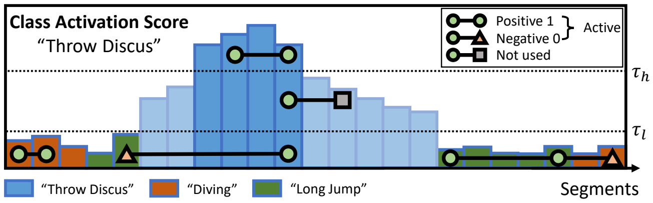

Fig. 3. A toy example of generating affinity matrix labels. Given a class activation score map for the particular class ("Throw Discus"), pairs of segments sampled according to the predefined thresholds. For the pairs which have same class are assigned 1, and 0 otherwise. If there is any segment of the pair that has *neutral* score, the pair is not assigned any labels.

and the posterior term is a graph Laplacian regularizer with hyper-parameter μ. We reformulate the above optimization as an inverse system problem of the linear equation such that

$$\mathbf{s}^{*c} = (\mathbf{I} + \mu\mathbf{L})^{-1}\mathbf{s}^c, \tag{5}$$

where $\mathbf{I}$ is an identical matrix. The solution of (5) is obtained by inference layer which is fully differentiable in a similar way to [16], thus the graph regularization module can be learned in an end-to-end manner. The output of the inference layer is leveraged to refine the class activation score map. Since the Laplacian matrix $\mathbf{L}$ is class-agnostic, class activation scores are refined solely with their semantic affinity.

2.3. Loss Functions

To optimize the proposed network, we define the loss function as the sum of two loss functions:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{aff}} + \lambda \cdot \mathcal{L}_{\text{cls}}, \tag{6}$$

where $\mathcal{L}_{\text{aff}}$ is the affinity loss on the affinity matrix $\mathbf{A}$ and $\mathcal{L}_{\text{cls}}$ denotes the classification loss using only video-level labels. And λ is a hyper-parameter to scale two loss functions.

To propagate local responses via affinity matrix, we carefully consider the class activation score map as another weak supervision in affinity loss. Specifically, we build the *active* set of segments used in training whose class activation scores are higher or lower than threshold τ_h and τ_l corresponding to the ground truth class labels as illustrated in Fig. 3. For each segment in the active set, we temporarily assign class labels z_t based on the class activation score map such that,

$$z_t = \arg\max_c \mathbf{s}_t^c. \tag{7}$$

Then, we define a pseudo affinity label $\mathbf{A}^*_{ij}$ for the active pair of segments v_i and v_j as follows:

$$\mathbf{A}^*_{ij} = \begin{cases} 1 & if \quad z_i = z_j \\ 0 & otherwise \end{cases}. \tag{8}$$

If any segments of pair not belong to *active*, we do not assign labels for the affinity loss. The affinity loss $\mathcal{L}_{\text{aff}}$ is based on the mean squared error between the affinity matrix $\mathbf{A}$ and pseudo affinity label $\mathbf{A}^*$ denoted by:

$$\mathcal{L}_{\text{aff}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}||\mathbf{A}_{ij} - \mathbf{A}^*{}_{ij}||^2, \tag{9}$$

where $\mathcal{N}(i)$ is the semantic neighborhood for v_i.

In addition, the refined class activation score map $\mathbf{s}^{*c}$ of sampled segments is aggregated to generate the video-level score vector $\hat{y}^c$. To learn the weights of two sub-modules, the classification loss function $\mathcal{L}_{\text{cls}}$ employs the standard multi-label cross-entropy loss between the ground truth video labels y^c and the predicted score vector $\hat{y}^c$ such that,

$$\mathcal{L}_{\text{cls}} = -\sum_{c=1}^{K}[y^c\log(\hat{y}^c) + (1-y^c)\log(1-\hat{y}^c)]. \tag{10}$$

Note that the derivative of the classification loss is back-propagated into the both class activation and graph regularization module, but the derivative of the affinity loss can be back-propagated only into the graph regularization module.

2.4. Testing Phase

We employ the two-stream model which is widely used in action recognition and localization areas to get boosted performance [17, 18]. Our networks illustrated in Fig. 2 are trained for the RGB and the optical flow stream with the equivalent settings, and the outputs of each stream are combined to perform the temporal action localization task.

Concretely, to generate temporal proposals including a set of time intervals with respect to each testing video, we compute refined class activation score maps for the RGB and flow stream, represented by $\hat{\mathbf{s}}^c_{t,\text{RGB}}$ and $\hat{\mathbf{s}}^c_{t,\text{FLOW}}$. Similar to [10], we extract temporal segments by applying thresholding strategy to each stream. After that, we integrate the proposals whose overlapping duration is above pre-defined threshold. We then assign a score for each integrated proposal corresponding to each class c as follows:

$$\sum_{t=t_s}^{t_e}\frac{\alpha\cdot\hat{\mathbf{s}}^c_{t,\text{RGB}} + (1-\alpha)\cdot\hat{\mathbf{s}}^c_{t,\text{FLOW}}}{t_e - t_s + 1}, \tag{11}$$

where α is a parameter to balance scores from two modalities. Finally, we apply non-maximum suppression to all proposals with intersection over union (IoU) higher than 0.5.

3. EXPERIMENTAL RESULTS

3.1. Implementation and Training Details

We use two-stream I3D models [17] trained on the Kinetics dataset [20] to extract feature for each video segment. We randomly sample 400 segments in the training set for data augmentation and uniformly sample at the same intervals in the test set. For the RGB stream, the videos are rescaled preserving aspect ratio so that the smallest dimension of a frame is 256 pixels. Then, we perform the random crop of size 224×224 for the training data and the center crop with same size for the test data. For the flow stream, we apply the TV-$L1$ optical flow algorithm [21]. The inputs of the I3D models are the sets of 16 (RGB or flow) frames sampled at 10 frames per second.

We train our model using the Adam optimizer [22] with PyTorch [23]. The learning rate for the class activation module is set to 10^{-4} until the end of training. For the graph regularization module, the learning rate is set to 0 for first $1,000$ iterations and to 10^{-4} after that. At testing time, we collect class activation scores whose video-level score is over 0.1, and extract temporal proposals from each score. We set the balancing parameter α to 0.5.

3.2. Experimental Settings

For the quantitative evaluation, we used mean average precision (mAP) metric according to different IoU threshold values. We evaluated the proposed method on two commonly used action localization benchmark datasets, THUMOS14 [19] and ActivityNet1.2 [24].

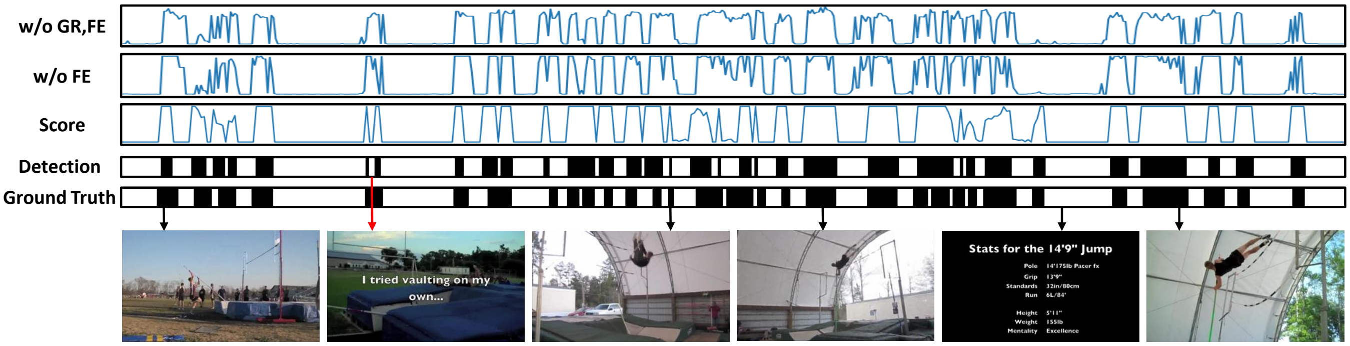

Fig. 4. Localization result for qualitative analysis in the class “Pole Vault” on THUMOS14 dataset [19]. The black arrow shows the successive cases and the red arrow shows failure case due to the occlusion.

Table 1. Localization performance comparisons over the THUMOS14 dataset [19]. w/o GR, FE denotes a result using class activation module only and w/o FE denotes a result without embedded feature.

Supervision	Methods	AP@IoU			
		0.3	0.4	0.5	0.7
Full	Yuan *et al.* [6]	36.5	27.8	17.8	-
	Gao *et al.* [25]	50.1	**41.3**	**31.0**	9.9
	Zhao *et al.* [8]	**51.9**	41.0	29.8	**10.7**
Weak	Wang *et al.* [9]	28.3	21.1	13.7	-
	Nguyen *et al.* [10]	35.5	25.8	16.9	4.3
	Shou *et al.* [11]	35.8	29.0	21.2	5.8
	Paul *et al.* [12]	40.1	31.1	**22.8**	7.6
	Ours w/o GR, FE	25.2	17.8	9.6	2.7
	Ours w/o FE	35.4	26.1	16.7	4.2
	Ours	**40.2**	**32.2**	21.7	**9.2**

Table 2. Localization performance comparisons over the ActivityNet1.2 dataset [24].

Supervision	Methods	AP@IoU			
		0.3	0.4	0.5	0.7
Full	Zhao *et al.* [8]	-	-	41.3	30.4
Weak	Wang *et al.* [9]	-	-	7.4	3.9
	Shou *et al.* [11]	-	-	27.3	17.5
	Paul *et al.* [12]	**45.5**	41.6	**37.0**	14.6
	Ours	45.2	**41.8**	33.7	**18.4**

Specifically, the THUMOS14 dataset has a subset of 200 and 213 untrimmed videos for each the validation and the test set with 20 action classes. We trained our network using the validation set without any temporal annotations and evaluate on the test set with temporal annotations. For the ActivityNet1.2 dataset, we used the training set to learn our model and evaluated on the validation set with 100 classes.

We also conducted ablation studies to investigate the contribution of the feature embedding network and the graph Laplacian regularization on the THUMOS14 dataset. In our experimental results, ‘GR’ means graph Laplacian regularization and ‘FE’ means feature embedding network.

3.3. Results on THUMOS14

Table 1 shows the comparison with state-of-the-art methods for temporal action localization on THUMOS14. We included only a few of the fully-supervised and weakly-supervised approaches in the table. Our algorithm outperforms or shows competitive performance on weakly-supervised learning. Despite the difference in levels of supervision, our algorithm shows competitive performance to recent fully-supervised methods.

We also conducted ablation studies for the graph regularization module and the semantic affinity module to investigate the contribution of components in our model. We observed that the graph regularization module with the feature embedding network sufficiently contributes to the performance improvement. Fig. 4 shows qualitative results for the class “Pole Vault” on the THUMOS14 dataset. Each row represents score signals of two baselines, detection results of our whole networks, and ground truth for example video, respectively. In detection results, we can observe that our model effectively capture the temporal boundaries despite of many action instances. The results of the two baselines presents considering affinity can refine the score signal, and it also confirmed that embedding the feature to affinity space thoroughly improves effect of regularization.

3.4. Results on ActivityNet1.2

Moreover, we presented the evaluations of our model on ActivityNet 1.2 in Table 2. We compared our methods with other state-of-the-art fully-supervised and weakly-supervised temporal action localization methods. The results show our methods obtain state-of-the-art performance at IoU thresholds 0.4 and 0.7 with 41.8 and 18.4 scores, respectively. Also, it even shows competitive performance for other thresholds.

4. CONCLUSION

In this paper, we presented the novel learning framework through CNNs for weakly-supervised temporal action localization in untrimmed videos. Two sub-modules including class activation and graph regularization have been proposed to estimate discontinuity preserved localization results with the learned embedding features. Moreover, the proposed affinity loss and classification loss are used to jointly optimize the networks to effectively embed the features into affinity space. We validated the effectiveness of the proposed method on the THUMOS14 and ActivityNet1.2 benchmark datasets.

5. ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No.2017K1A3A1A16066838). (Corresponding author:Kwanghoon Sohn.)

6. REFERENCES

[1] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," *In: ICCV*, 2007.

[2] W-S. Chu, Y. Song, and A. Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," *In: CVPR*, 2015.

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," *In: ICCV*, 2015.

[4] F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," *In: CVPR*, 2016.

[5] J. Yuan, B. Ni, X. Yang, and A. A. Kassim, "Temporal action localization with pyramid of score distribution features," *In: CVPR*, 2016.

[6] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng, "Temporal action localization by structured maximal sums," *In: CVPR*, 2017.

[7] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," *In: ICCV*, 2017.

[8] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," *In: ICCV*, 2017.

[9] L. Wang, Y.Xiong, D. Lin, and L. van Gool, "Untrimmednets for weakly supervised action recognition and detection," *In: CVPR*, 2017.

[10] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," *In: CVPR*, 2018.

[11] Z. Shou, H. Gao, K. Miyazawa, and S.-F. Chang, "Autoloc: Weakly-supervised temporal action localization in untrimmed videos," *In: ECCV*, 2018.

[12] S. Paul, S. Roy, and A. K. Roy-Chowdhury, "W-talc: Weakly-supervised temporal activity localization and classification," *In: ECCV*, 2018.

[13] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," *In: CVPR*, 2017.

[14] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *In: CVPR*, 2016.

[15] J. Pang and G. Cheung, "Graph laplacian regularization for image denosing: Analysis in the continuous domain," *IEEE Transaction on Image Processing*, vol. 26, 2017.

[16] B. Amos and J.Z. Kolter, "Optnet: Diffentiable optimization as a layer in neural networks," *In: ICML*, 2017.

[17] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," *In: CVPR*, 2017.

[18] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal residual networks for video action regcognition," *In: CVPR*, 2016.

[19] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," 2014.

[20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *arXiv: 1705.06950*, 2017.

[21] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for tv-l^1 optical flow," 2009.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv*, 2014.

[23] Paszke, Adam, Gross, Sam, Chintala, Soumith, Chanan, Gregory, Yang, Edward, DeVito, Zachary, Lin, Zeming, Desmaison, Alban, Antiga, Luca, Lerer, and Adam, "Automatic differentiation in pytorch," *NIPS 2017 Workshop*, 2017.

[24] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: a large-scale video benchmark for human activity understanding," *In: CVPR*, 2015.

[25] J. Gao, Z. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," *In: BMVC*, 2017.