BOUNDARY INFORMATION MATTERS MORE: ACCURATE TEMPORAL ACTION DETECTION WITH TEMPORAL BOUNDARY NETWORK

[1,2]*Tao Zhang,* [3]*Shan Liu,* [4]*Thomas Li,* [1,2]*Ge Li*

[1]School of Electronic and Computer Engineering, Peking University, Shenzhen, China
[2]Peng Cheng Laboratory
[3]Media Lab, Tencent, Shenzhen
[4] Advanced Institute of Information Technology, Peking University, Hangzhou, China

ABSTRACT

Temporal action detection in untrimmed videos is an important yet challenging task. How to locate complex actions accurately is still an open question due to the ambiguous boundaries between action instances and the background. Recently a newly proposed work exploits Structured Segment Networks (SSN) for temporal action detection, which models temporal structure of action instances via structured temporal pyramids, and comprises two classifiers, respectively for classifying actions and determining proposal completeness. In this paper we attempt to delve the temporal boundary information when modeling temporal structure of action instance, by introducing to SSN the structured temporal boundary attention pyramid. On top of the pyramid, we add another set of classifiers for unit-wise completeness evaluation, which enables proposal recycling for efficient action detection. Experimental results on two challenging benchmarks, THUMOS'14 and ActivityNet, indicate that our Temporal Boundary Network shows a significant performance improvement compared with SSN, and achieves a competitive performance compared with state-of-the-arts.

***Index Terms*—** deep learning, temporal action detection, feature pyramid, temporal boundary

1. INTRODUCTION

Temporal action detection aims to figure out both the action category and the accurate temporal location of action instances in untrimmed videos. During the past few years, temporal action detection has drawn increasing interest [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], due to its considerable applications in security surveillance, human-machine interaction, *etc*. However, how to locate long complex action instances accurately is still an open question.

Most existing methods follow a two-step fashion. First, proposals probably containing actions are generated, and then the proposals are classified and refined. Recently, it is reported in [6] that temporal structure analysis is vital for distinguishing complete actions from incomplete ones when classifying proposals. In [6], each proposal is expanded and divided into three stages, *e.g. starting*, *course* and *ending*. Upon these stages, structured temporal pyramids (STPP) are built to explicitly model the temporal structure of actions. Then the proposals are classified and refined based on the prediction of action classifier and completeness classifier. SSN [6] achieves a promising detection performance. However, SSN ignores the importance of temporal boundary information by treating these three stages equally. Additionally, SSN simply discards proposals containing incomplete actions, resulting in a waste of resources.

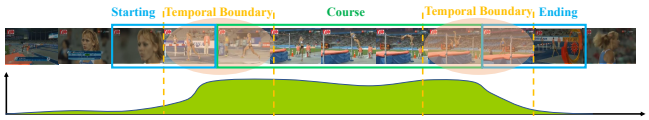


Fig.1. **Temporal boundary**. A proposal (green box) is expanded and then divided into three stages, $i.e.$ *starting*, *course*, *ending*. The response of TBN shows a steep change near temporal boundaries.

In this paper we focus on the classification and refinement of action proposal. We inherit the manner to detect actions through temporal structure modeling, but go beyond these two aforementioned limitations of SSN. To determine whether a video segment contains a complete action instance, generally human beings first inspect the course of the segment briefly, and then focus on its two boundaries to see whether it is complete. Following this observation, we replace the STPP in SSN with structured temporal boundary attention pyramid (STBAP). The difference between STBAP and STPP is that STPP simply pools and concatenates the feature of the three stages, while STBAP executes an additional downsampling operation before concatenation to highlight the temporal boundary information. The boundary attention mechanism enables the network to pay more attention to temporal boundaries between action instances and the background (Fig.1). Furthermore, we evenly divide each expanded proposal into several units, and add another set of unit-wise completeness classifiers upon the pyramid in addition to the two discriminative models in SSN. With the fine-grained unit-wise completeness prediction, we use the grouping scheme proposed in [15] to compose additional detection results from the union of incomplete action instances. The recycling of proposals containing incomplete actions brings higher proposal usage efficiency and better detection performance. The new network is named **Temporal Boundary Network (TBN)**, which achieves a significant performance improvement compared with SSN. The **main contributions** of this paper: (1)To the best of our knowledge, this is the first work to proposal temporal boundary attention mechanism for accurate temporal action detection. (2)We propose a proposal recycling strategy, which improves both the proposal usage efficiency and detection performance. (3)TBN achieves competitive performance compared with state-of-the-arts on THUMOS'14 and ActivityNet dataset.

Thanks to Shenzhen Municipal Science and Technology Program under Grant JCYJ20170818141146428, Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (No. ZDSYS201703031405467), National Engineering Laboratory for Video Technology - Shenzhen Division, Peng Cheng Laboratory, Shenzhen Fundamental Research Program (NO.JCYJ20170818141120312) for funding.

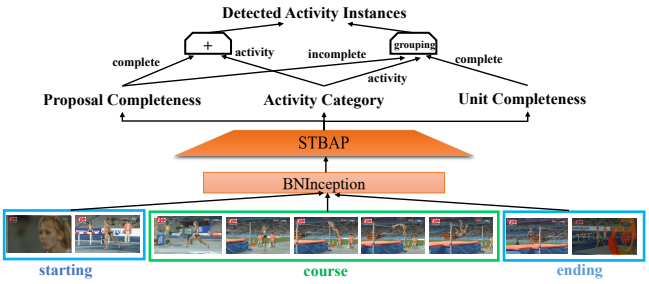


Fig.2. **An overview of TBN**. First a proposal is extended to $2X$ length. Then the augmented proposal is divided into three stage: *starting*, *course*, *ending*. STBAPs are built on the two-stream features of these three stages. Upon the pyramids, three types of classifiers are built, respectively for action category prediction, proposal-wise completeness determination and unit-wise completeness determination.

2. RELATED WORK

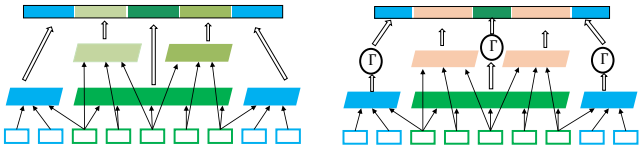
Action Recognition. Lots of methods have been introduced to action recognition task [16, 17, 18, 19, 20]. The two-stream network [16] and 3D CNN [17] have been proposed to incorporate both appearance and motion cues in different manners. Wang *et.al.* [19] proposed Temporal Segment Network (TSN) to cope with long videos. Both SSN [6] and TBN are based on two-stream structure and use the sparse sampling scheme introduced in [19] to select input frames from videos.

Temporal Action Detection. Recently temporal action detection has received much attention ([1, 21, 3, 4, 6, 22, 8, 23, 10, 24, 11, 14]). Shou *et al.* [4] utilized a multi-stage 3D CNN network [17] for action localization. Chao *et al.* [14] proposed to detect actions based on Faster RCNN [25] structure. Zhao *et al.* [6] introduced Structured Segment Network, which distinguish complete and incomplete action instances by modeling the temporal structure of action instances. Different from them, we utilize STBAP to delve temporal boundary information of action instances. Furthermore, we recycle incomplete action proposals by grouping action units.

3. METHOD

An overview of TBN is shown in Fig 2. In this section we delve into the two main contribution of this paper, *i.e.*, the STBAP, and the proposal recycling strategy. We also introduce the multi-task loss function to train the network.

3.1. Structured Temporal Boundary Attention Pyramid



(a). STPP in SSN (b). STBAP.

Fig.3. Each hollow box represents the feature of a snippet. **(a)**. **STPP in SSN** [6]. Feature of three stages (blue boxes for *starting* and *ending*, green box for *course*) is pooled and concatenated. **(b)**. **STBAP**. Before concatenation we downsample the features of all three stages except temporal boundary parts. Γ is 1 * 1 convolution.

We first give a brief description of the STPP in SSN. Given a expanded proposal $p_i = [s_i, e_i]$, where s_i and e_i are the starting and ending frame of the i-th expanded proposal, STPP first divides it into three stages, *i.e.*, *starting*, *course* and *ending*. On the feature of each stage, a K-level STPP is constructed, where each level evenly divides a stage into T_k parts. Taking the *course* stage for instance, the j-th part of the k-th level, whose interval is $[s_{kj}, e_{kj}]$, can be denoted as

$$u_j^{(k)} = \frac{1}{|e_{kj} - s_{kj} + 1|} \sum_{t=s_{kj}}^{e_{kj}} v_t \tag{1}$$

where v_t is the two-stream feature of t-th snippet, and an expanded proposal consists of 9 snippets. The overall representation of *course* stage F_i^c is the concatenation of the pooled feature across all parts at all levels, which can be denoted as

$$F_i^c = (u_j^{(k)} | k = 1, ..., K, j = 1, ..., T_k) \tag{2}$$

Note that each $u_j^{(k)}$ is a vector with the same length L_u. Hence, every part is processed with the same importance. Different from that, STBAP derives a global representation for each proposal while focusing more on the temporal boundaries between action instance and its surrounding background. As shown in Fig.3 (b), STBAP downsamples the feature of each stage except the temporal boundary part. More specifically, for level k, the feature representation contains two parts: the temporal boundary part $u_i^b = (u_j^{(k)} | j = 1, T_k)$ and the non-temporal boundary part $u_i^n = (u_j^{(k)} | j = 2, ..., T_k - 1)$. For each $u_j^{(k)} \in u_i^n$, we downsample it with operation Γ. We concatenate u_i^b and u_i^n as feature representation $\mathcal{H}_i^c$ of stage *course*, which can be denoted as

$$\mathcal{H}_i^c = ((\Gamma(u_j^{(k)}) | j = 2, ..., T_k - 1), (u_j^{(k)} | j = 1, T_k)) \tag{3}$$

where $k = 1, ..., K$. Specifically, we use 1*1 convolution to downsample u_i^n to length L_d ($L_d < L_u$). This construction not only leverages the temporal structure of each action instance, but also explicitly delves the temporal boundary information, which is vital for distinguishing complete action instances from incomplete ones. Following [6], we use a two-level pyramid, *i.e.* $K = 2, T_1 = 1, T_2 = 2$ for the *course* stage, while using a simpler one-level pyramid (standard average pooling) for the *starting* stage and *ending* stage.

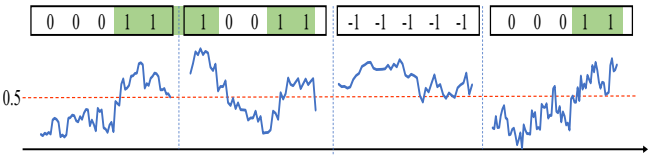


Fig.4. **Unit grouping**. Each box represents a proposal divided into 5 units. 0 and 1 represents background and complete unit respectively. Units in complete action proposals are marked with -1, which would not be taken in when perform unit grouping. Green boxes represents the grouped detection results.

3.2. Proposal Recycling Strategy

First we briefly introduce the two types of classifiers that TBN inherits from SSN. The action classifier A is built on the *course* stage, and makes classification prediction on C action categories and the background category. The completeness classifiers $\{B_k\}_{k=1}^{C}$ determines whether a proposal covers a complete action instance, and it is fed the concatenated feature of all three stages.

We introduce another set of unit-wise completeness classifiers $\{C_k\}_{k=1}^{C+1}$. Here a unit is a single part of an action proposal. Specif-

ically, given an expanded proposal $p_i = [s_i, e_i]$ with a duration $d_i = e_i - s_i$, we evenly divide it into N_s individual units, each with a duration d_i/N_s. The j-th unit of p_i is represented by $\omega_j^{(i)}$: $[s_j^{(i)}, e_j^{(i)}]$ $(j = 1, ..., N_s)$, where $s_j^{(i)} = s_i + (j-1) \cdot d_i/N_s$ and $e_j^{(i)} = s_i + j \cdot d_i/N_s$. For each $\omega_j^{(i)}$, the unit-wise completeness classifiers output a probability indicating whether $\omega_j^{(i)}$ is background or part of an action instance. With unit-wise completeness prediction, we recycle proposals that contain incomplete action instances, by grouping isolated, fine-grained action assessments at each unit into meaningful detection results, as shown in Fig.4. This proposal recycling strategy aims to make full use of action proposals, thus improve the efficiency and detection performance.

The predictions of the three classifiers (category, completeness, unit-wise completeness) for p_i are denoted by $P(c_i|p_i)$, $P(b_i|c_i, p_i)$ and $P(b_j|c_j, \omega_j, p_{\tilde{j}})$ respectively, where the unit ω_j is included in the proposal $p_{\tilde{j}}$. In [6], the joint probability of $P(c_i|p_i)$ and $P(b_i|c_i, p_i)$ is used as confidence score to evaluate the quality of p_i, as defined in Eq. 4:

$$P(c_i, b_i|p_i) = P(c_i|p_i) \cdot P(b_i|c_i, p_i) \ (c_i \geq 1) \tag{4}$$

Defining that $\{\nu_k\}_{k=1}^{N_g}$ represents new action proposals collected by grouping complete units, where ν_k is comprised of units $\{\omega_j\}_{j=1}^{M_k}$, we compute the confidence score for ν_k by:

$$P(c_k, b_k|\nu_k) = \frac{1}{M_k}^2 \sum_{j=1}^{M_k} P(c_j|\omega_j, p_{\tilde{j}}) \cdot \sum_{j=1}^{M_k} P(b_j|c_j, \omega_j, p_{\tilde{j}}) \tag{5}$$

where $c_j \geq 1$ and $b_{\tilde{j}} = 0$. When $c_j \geq 1$, $p_{\tilde{j}}$ and ω_j share the same action category, $p(c_j|\omega_j, p_{\tilde{j}}) = p(c_{\tilde{j}}|p_{\tilde{j}})$. Both $P(c_i, b_i|p_i)$ and $P(c_k, b_k|\nu_k)$ are used evaluate the detection results when performing Non-Maximum Suppression.

3.3. Loss Function Formulation

Let us consider proposal p_i and the units $\left\{\omega_j^{(i)}\right\}_{j=1}^{N_s}$ separated from it. For $\omega_j^{(i)}$ we label its completeness with $b_j^{(i)} = 1$ if its overlap with the groundtruth is higher than 0.5. Otherwise $b_j^{(i)} = 0$. The optimization objective of unit-wise completeness classifiers is shown in Eq. 6:

$$\mathcal{L}_{unit}(b^{(i)}, c^{(i)}, p_i) = \frac{1}{N_s} \sum_{j=1}^{N_s} (-log(P_j^{c_j^{(i)}}[p_i])) \tag{6}$$

We utilize location regression to refine the position and span of proposal. We define a multi-task loss to train the network in an end-to-end manner, illustrated by Eq.7:

$$\begin{aligned} \mathcal{L}oss = \mathcal{L}_{cls}(c_i, p_i) &+ \lambda_1 \cdot 1_{(c_i \geq 1)} \mathcal{L}_{com}(b_i, c_i, p_i) \\ &+ \lambda_2 \cdot 1_{(c_i \geq 1 \& b_i = 0)} \mathcal{L}_{unit}(b^{(i)}, c^{(i)}, p_i) \\ &+ \lambda_3 \cdot 1_{(c_i \geq 1 \& b_i = 1)} \mathcal{L}_{reg}(\mu_i, \phi_i; p_i) \end{aligned} \tag{7}$$

where $\mathcal{L}_{cls}(c_i, b_i)$ is cross-entropy loss function, and $\mathcal{L}_{com}(b_i, c_i, p_i)$ is computed based on online hard negative mining [26]. $\mathcal{L}_{reg}(\mu_i, \phi_i; p_i)$ utilizes the smooth L1 loss function.

4. EXPERIMENTS

In this section, we first introduce the datasets and implementation details. Then we go deeper for ablation studies. Finally we compare the overall temporal action detection performance of our method with state-of-the-arts, and discuss the generalizability of our methods.

4.1. Dataset and Implementation

Dataset. We validate our method on two widely-used public dataset: THUMOS14 [27] and ActivityNet v1.2 [28]. **THUMOS'14** contains 20 action class, with 200 videos for training and 213 videos for testing. **ActivityNet v1.2** has 100 different type of actions, with 4819 training videos, 2383 validation videos and 2480 test videos.
Implementation. Inception with Batch Normalization (BN-Inception) [29] is used as feature extractor. Each classifier consists of one fully connnected layer. For fair comparison, we follow most hyper-parameter settings of SSN. The mini-batch size is 144. The initial learning rate for the RGB branch and the Flow branch are 0.001 and 0.005 respectively. The momentum is 0.9 for both streams. On THUMOS'14, the RGB branch and flow branch are trained for 1K iterations and 6K iterations respectively, and the learning rate is scaled down by 0.1 per 400 and 2500 iterations respectively. On ActivityNet v1.2, the RGB branch and flow branch are respectively trained for 12K and 20K iterations, with learning rate reduced by a factor of 10 after per 4K and 8K iterations. Each augmented proposal is divided into 5 units, $i.e.$ N_s = 5. In the loss function, λ_1=0.1, λ_1=0.1, λ_1=0.3.

4.2. Ablation Study

To validate the effectiveness of our proposed method, and explore the best hyper-parameter setting, we conduct ablation study experiments on THUMOS'14 dataset. We use ImageNet pre-trained weights to initialize TBN when performing ablation study. mean Average Precision (mAP) at tIoU=0.5 is used to measure the performance. For convenience, "Grouping" represents the unit grouping scheme introduced in [15], and TBN is the combination of "SSN", "STBAP", "Grouping".

SSN(RGB+Flow)	27.36		
	$L_d = L_u$	$L_d = 0.75L_u$	$L_d = 0.5L_u$
SSN(RGB) + STBAP	18.02	**18.68**	17.84
SSN(Flow) + STBAP	22.64	22.92	**23.34**
SSN + STBAP	27.66	**28.83**	27.91

Table 1. Detection performance(%mAP at tIoU=0.5) with various L_d value .

	without Grouping	with Grouping
TBN(RGB)	19.68	20.10
TBN(Flow)	24.31	24.44
TBN	30.06	30.30

Table 2. Experimental results (%mAP at tIoU=0.5) on the effect of proposal recycling.

Structured Temporal Boundary Attention Pyramid. Here we explore the effectiveness of STBAP. In STBAP, 1*1 convolution is used to downsample the feature of each stage to length L_d, while the length of temporal boundary feature is kept as L_u. Here L_u is determined by feature extraction backbone network. The experiment results with various L_d values are shown in Table 1. The performance is significantly improved when SSN is equipped with STABP. While an additional 1*1 convolution layer ($L_d = L_u$) helps improve the performance slightly, rescaling feature vectors that contain little boundary information ($L_d < L_u$) enables the network to perform much better. It verifies the idea that the temporal boundary information should be paid more information when doing accurate temporal action detection. As we can see in Table 1, the best value for L_d is

$0.75L_u$, which retains the detailed temporal boundary information and condenses the feature from other parts of proposals simultaneously. Therefore, we fix $L_d = 0.75L_u$ in following experiments.

Proposal Recycling. In [15], Xiong *et al.* introduce a scheme to group action proposals from snippet-wise actionness estimation. Similarly, we adopt the grouping scheme to generate extra detection results from incomplete action proposals (Fig.4). To study the influences of unit-wise completeness classifiers and the grouping scheme separately, we train TBN with unit-wise completeness classifiers, but test it with or without the grouping method. The experimental results are shown in Table 2. As shown in Table 2, even though we don't supplement the extra detection results generated by grouping, the performance is still improved, which indicates that the unit-wise classifiers help boost the performances of the other two types of classifiers. This is reasonable since additional supervision information from unit-wise completeness classifiers promotes the whole network to learn about the temporal structure of action proposal more accurately. Besides, we can observe that the grouping scheme does help to improve the detection performance.

tIoU	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Karaman *et al.* [30]	4.6	3.4	2.1	1.4	0.9	-	-
Oneata *et al.* [3]	36.6	33.6	27.0	20.8	14.4	8.5	3.2
S-CNN [4]	47.7	43.5	36.3	28.7	19.0	10.3	5.3
Yeung *et al.* [7]	48.9	44.0	36.0	26.4	17.1	-	-
Yuan *et al.* [8]	51.4	42.6	33.6	26.1	18.8	-	-
CDC [1]	-	-	40.1	29.4	23.3	13.1	7.9
R-C3D [2]	54.5	51.5	44.8	35.6	28.9	-	-
CBR-TS [31]	60.1	56.7	50.1	41.3	31.0	19.1	9.9
SS-TAD [5]	-	-	45.7	-	29.2	-	9.6
Chao *et al.* [14]	59.8	57.1	53.2	48.5	**42.8**	33.8	20.8
SSN(ImageNet) [6]	-	-	-	-	27.4	-	-
SSN[1](ImageNet) [6]	-	-	-	-	29.8	-	-
SSN(Kinetics) [6]	-	-	-	-	32.5	-	-
TBN(ImageNet)	61.9	55.6	48.8	39.1	**30.3**	20.7	11.3
TBN(Kinetics)	**67.2**	**61.2**	**54.8**	44.6	33.8	22.8	13.0

Table 3. Action detection results using TAG proposals [6] on THUMOS'14 testing set. Performance is measured by mAP at multiple tIoU thresholds, ranging from 0.1 to 0.7.

4.3. Comparison With State-of-the-arts

THUMO'14. First we validate our method on the proposal produced by TAG [6]. The overall experimental results on THUMOS'14 are show in Table 3. mAP at different tIoU thresholds is used to compare the performance of different methods. A good temporal action detection method is expected to achieve high mAP at high tIoU threshold. From Table 3 we can observe that TBN shows about 3% mAP improvement over SSN at tIoU=0.5 when using ImageNet [32] pre-trained weights for initialization, while showing a 1.3% improvement when TBN is initialized with Kinetics [18] pre-trained weights. Compared with most of other state-of-the-art methods, TBN also demonstrates superior performance. The experimental results demonstrate that, our proposed STBAP and proposal recycling strategy significantly help to locate action instances more accurately.

ActivityNet v1.2. ActivityNet has more diversity in terms of the number of action categories and the number of videos. Evaluation in ActivityNet helps validate the effectiveness of our method, and exclude the possibility that it is overfitting that brings performance improvement. Since only action proposals (generated by TAG [6]) on the training and validation sets of ActivityNet v1.2 are publicly available, we measure the performance of TBN on the validation set. As shown in Table 4, compared with other approaches, our proposed TBN achieves superior temporal action detection performance. The performance is improved significantly compared with SSN, though we only slightly modify the architecture of SSN.

tIoU	0.5	0.75	0.95	Average
Xiong *et al*. [15]	41.1	24.1	5.0	24.9
SSN (ImageNet) [6]	-	-	-	24.5
TBN (ImageNet)	42.9	28.1	7.4	**27.6**

Table 4. Action detection results using TAG proposals [6] on ActivityNet v1.2 validation set. The performances are measured by mAP at multiple tIoU thresholds $\{0.5, 0.75, 0.95\}$ and the average mAP of thresholds from 0.5:0.05:0.95.

tIoU	0.3	0.4	0.5	0.6	0.7
BSN [33]	53.5	45.0	36.9	28.4	20.0
SSN(ImageNet) [6]	52.6	46.2	37.9	28.8	20.2
TBN(ImageNet)	**53.8**	**47.1**	**39.1**	**29.7**	**20.8**

Table 5. Action detection results using proposals generated by BSN [33] on THUMOS'14 testing set.

4.4. Discussion

Though TBN outperforms most of other temporal action detection methods, there is still a performance gap between TBN and the Faster RCNN based approach in [14]. One possible reason is that the detection performance is limited by the quality of proposals. To verify this conjecture, we validate our method on another better proposal method [33]. As we can see from Table 5, given proposals of higher quality, TBN obtain a much better detection performance. Besides, the network in [14] uses a more discriminative feature extracter (I3D [18]) than BN-Inception, which promises better performance. The performance gain compared to baseline network demonstrates that the temporal boundary attention mechanism and proposal recycling method can bring significant performance improvement. The temporal boundary attention mechanism can be implanted into any other proposal-based temporal action detection framework to improve detection performance.

5. CONCLUSION

In this paper, we address the challenging problem of temporal action detection. Our proposed **Temporal Boundary Network (TBN)** utilizes a structured temporal boundary attention pyramid to explicitly emphasize temporal boundary information of action instances. On top of the pyramid, three types of classifiers, respectively for action category prediction, proposal-wise completeness evaluation and unit-wise completeness determination, are applied. With the fine-grained unit-wise completeness prediction, we recycle proposals containing incomplete action instance by bottom-to-up units grouping. Extensive experiments demonstrate that TBN achieves competitive performance compared with other state-of-the-arts method on both THUMOS14 (mAP @tIoU=0.5: 39.1) and ActivityNet v1.2 (Average mAP 27.6) datasets. The proposed generalizable temporal boundary attention mechanism and proposal recycling strategy significantly help to locate actions more accurately .

[1]The detection results is filtered with UntrimmedNets [11] to keep only those from the top-2 predicted action classes

6. REFERENCES

[1] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang, "Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," *CVPR*, pp. 1417–1426, 2017.

[2] Huijuan Xu, Abir Das, and Kate Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," *ICCV*, vol. 6, pp. 8, 2017.

[3] Dan Oneata, Jakob Verbeek, and Cordelia Schmid, "The lear submission at thumos 2014," 2014.

[4] Zheng Shou, Dongang Wang, and Shih-Fu Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," *CVPR*, pp. 1049–1058, 2016.

[5] S Buch, V Escorcia, B Ghanem, L Fei-Fei, and JC Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," *BMVC*, 2017.

[6] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin, "Temporal action detection with structured segment networks," *ICCV*, vol. 8, 2017.

[7] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," *CVPR*, pp. 2678–2687, 2016.

[8] Jun Yuan, Bingbing Ni, Xiaokang Yang, and Ashraf A Kassim, "Temporal action localization with pyramid of score distribution features," *CVPR*, pp. 3093–3102, 2016.

[9] F Caba Heilbron, Wayner Barrios, Victor Escorcia, and Bernard Ghanem, "Scc: Semantic context cascade for efficient action detection," *CVPR*, vol. 2, 2017.

[10] Tianwei Lin, Xu Zhao, and Zheng Shou, "Single shot temporal action detection," *ACMMM*, pp. 988–996, 2017.

[11] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool, "Untrimmednets for weakly supervised action recognition and detection," *CVPR*, 2017.

[12] Alexander Richard and Juergen Gall, "Temporal action detection using a statistical language model," *CVPR*, pp. 3131–3140, 2016.

[13] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem, "Daps: Deep action proposals for action understanding," *ECCV*, pp. 768–784, 2016.

[14] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," *CVPR*, pp. 1130–1139, 2018.

[15] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang, "A pursuit of temporal accuracy in general activity detection," *arXiv preprint arXiv:1703.02716*, 2017.

[16] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," *NIPS*, pp. 568–576, 2014.

[17] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," *ICCV*, pp. 4489–4497, 2015.

[18] Joao Carreira and Andrew Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," *CVPR*, pp. 4724–4733, 2017.

[19] 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," *ECCV*, pp. 20–36, 2016.

[20] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid, "Potion: Pose motion representation for action recognition," *CVPR*, 2018.

[21] Tao Zhang, Nannan Li, Jingjia Huang, Jia-Xing Zhong, and Ge Li, "An active action proposal method based on reinforcement learning," in *ICIP*, 2018, pp. 4053–4057.

[22] Jingjia Huang, Nannan Li, Tao Zhang, and Ge Li, "A self-adaptive proposal model for temporal action detection based on reinforcement learning," *arXiv preprint arXiv:1706.07251*, 2017.

[23] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen, "Temporal context network for activity localization in videos," *ICCV*, pp. 5727–5736, 2017.

[24] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao, "A multi-stream bi-directional recurrent neural network for fine-grained action detection," *CVPR*, pp. 1961–1970, 2016.

[25] Ross Girshick, "Fast r-cnn," *ICCV*, pp. 1440–1448, 2015.

[26] Abhinav Shrivastava, Abhinav Gupta, and Ross B Girshick, "Training region-based object detectors with online hard example mining," *CVPR*, pp. 761–769, 2016.

[27] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "THUMOS challenge: Action recognition with a large number of classes," `http://crcv.ucf.edu/THUMOS14/`, 2014.

[28] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," *CVPR*, pp. 961–970, 2015.

[29] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[30] Svebor Karaman, Lorenzo Seidenari, and Alberto Del Bimbo, "Fast saliency based pooling of fisher encoded dense trajectories," *ECCV THUMOS Workshop*, vol. 1, no. 2, pp. 5, 2014.

[31] Jiyang Gao, Zhenheng Yang, and Ram Nevatia, "Cascaded boundary regression for temporal action detection," *arXiv preprint arXiv:1705.01180*, 2017.

[32] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," *CVPR*, pp. 248–255, 2009.

[33] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," *arXiv preprint arXiv:1806.02964*, 2018.