SCALE MATTERS: TEMPORAL SCALE AGGREGATION NETWORK FOR PRECISE ACTION LOCALIZATION IN UNTRIMMED VIDEOS

Guoqiang Gong[1], *Liangfeng Zheng*[2] *and Yadong Mu*[1,2*]

[1]Wangxuan Institute of Computer Technology, Peking University
[2]Center for Data Science, Peking University
{gonggq,zhengliangfeng,myd}@pku.edu.cn

ABSTRACT

Temporal action localization is a recently-emerging task, aiming to localize video segments from untrimmed videos which contain specific actions. This work proposes a novel integrated temporal scale aggregation network (TSA-Net). Our main insight is that ensembling convolution filters with different dilation rates can effectively enlarge the receptive field with low computational cost, which inspires us to devise multi-dilation temporal convolution (MDC) block. Furthermore, to tackle video action instances with different durations, TSA-Net consists of multiple branches of sub-networks. Each of them adopts stacked MDC blocks with different dilation parameters, accomplishing a temporal receptive field specially optimized for specific-duration actions. We follow the formulation of boundary point detection, novelly detecting three kinds of critical points (i.e., starting / mid-point / ending) and pairing them for proposal generation. Comprehensive evaluations are conducted on THUMOS14. Our proposed TSA-Net demonstrates clear and consistent better performances and recalibrates new state-of-the-art on THUMOS14 benchmark.

***Index Terms*—** Video Analysis,Temporal Action Localization, Action Proposal

1. INTRODUCTION

This paper addresses the task of accurately finding the temporal boundary of a video segment from an untrimmed video that instantiates specific semantic action, referred to as video action localization [1, 2, 3] in the literature. This technique can serve a variety of applications in video analysis, including detecting highlights in a long video, semantic video summarization, etc.

The motivating observation for our work is illustrated in Figure 1(b). As seen, durations of true action instances can diversely distribute, usually varying from few seconds to several minutes. Most of the previous methods utilize a procrustean way to tackle the temporal scale issue. For temporal sliding window or actionness grouping based methods (e.g. [1, 2]), a popular treatment is to build a video pyramid via temporal sub-sampling and extract same-sized video segments from all pyramid levels. An action instance will be detected at specific temporal scales where its re-sized duration is amenable for detection, ignored at other scales. However, the video pyramid significantly complicates learning an effective model and often brings inferior performance than other methods. On the other hand, boundary point-based methods, such as BSN [4], typically ignore the scale issue and use fixed convolutional receptive field for all action instances.

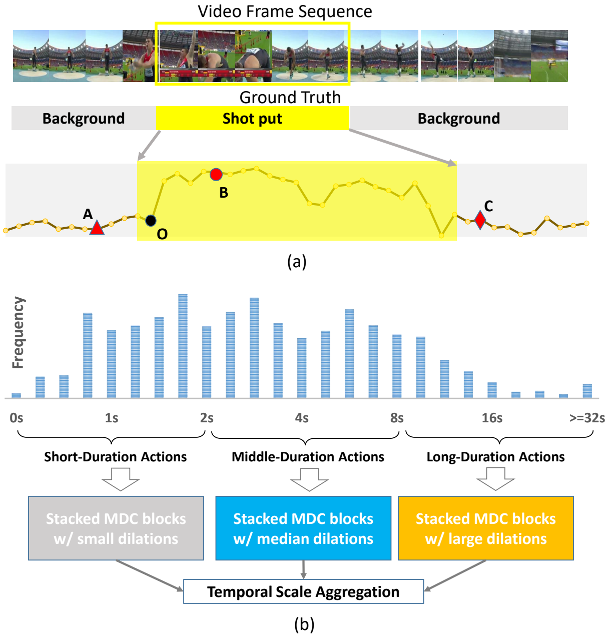

Fig. 1. Motivation of temporal scale aggregation. Sub-figure (a) draws an untrimmed video that contains an action instance, and the corresponding actionness sequence. To faithfully detect a boundary point (such as point O), it is necessary to have a sufficiently large temporal receptive field that reaches points A, B, and C. Sub-figure (b) shows the diversity of a true action instance's time durations on THUMOS14. It is important to ensure the model can combat temporal scale variations. We propose to ensemble three-branched network with varying dilation rates, with each branch specially optimized for the specific duration.

[*] Corresponding Author.

978-1-7281-1331-9/20/$31.00 ©2020 IEEE

This work proposes *temporal scale aggregation* network (TSA-Net) for video action localization. The key differentiator from other competing methods is novelly using multi-dilation temporal convolutions for seriously addressing the temporal scale issue. Figure 1 illustrates our key insight. Investigating temporal context is critical when confidently localizing some boundary points. However, as the duration of an action instance varies, finding a one-for-all temporal receptive field is arguably not possible. As a natural solution, we propose to ensemble a set of parameterized temporal convolutional building blocks. Each of them has different receptive field that is most effective at specific temporal scale. The responses of all temporal convolutions are fused to more reliably estimate a boundary point. The main technical contributions of this work are summarized as below:

1. We propose an action localization model TSA-Net, which falls into a two-step framework (proposal generation + classification). Unlike previous works, TSA-Net concurrently considers many temporal scales when estimating the boundary points. Specifically, we devise a *multi-dilation temporal convolution* (MDC) block as the core component of TSA-Net. To our best knowledge, it has been seldom explored in video action localization to tackling plenary temporal scales by efficiently manipulating convolutional dilation rates.

2. We design TSA-Net as a marriage of boundary detection and actionness grouping. It simultaneously detects three kinds of points: starting, ending and the mid-point of an instance. The mid-point implicitly encodes the actionness information of a proposal. A starting / ending pair can only be useful together with a confident mid-point. This way it effectively avoids the low-accuracy problem in previous boundary point-based methods (e.g. BSN [4]).

3. Comprehensive evaluations are conducted on THUMOS14. TSA-Net outperforms all competitors by large margins, re-calibrating state-of-the-art performance on both benchmarks. For example, our new record on THUMOS14 is 53.0% while the previous best is 49.1% for mAP@0.5.

2. RELATED WORK

The goal of temporal action localization is precisely detecting video segments with specific actions from untrimmed videos. A majority of action localization methods [1, 3, 5] adopt a two-stage computational pipeline, strongly inspired by the success of two-stage image object detection [6]. Besides the two-stage methods, there also exist other methods which adopt a framework of reinforcement learning [7] or single-shot detection [8].

As briefly summarized in Section 1, popular proposal-generating schemes can be roughly categorized as sliding windows [9, 1, 10, 11], temporal action grouping [2], or boundary point detection [4] etc. Two caveats for obtaining good temporal boundaries are doing the job at finer temporal scales, as shown by CDC [12], and fully utilizing temporal contextual information, as demonstrated by SSN [2] that decomposes an action proposal into starting-course-ending phases or BSN [4] that conducts stacked temporal convolutions.

3. PROPOSED METHOD

This section details our proposed TSA-Net, whose computational pipeline is depicted in Figure 2.

3.1. Video Feature Extraction

Let V be an untrimmed video of T frames. In the case of over-long videos, we proceed to extract video features with a regular frame interval σ in order to reduce the computation cost, resulting in $T \leftarrow T/\sigma$ video snippets. Without causing much confusion, we stick to misuse the variable T for representing frame or snippet count. Let $F = [f_1, f_2, \ldots, f_T]$ be the feature sequence of V. More details regarding F are deferred to Section 4.1.

3.2. Critical Point Detection

The goal of this step is to estimate each video snippet's probability of corresponding to a starting / intermediate / ending moment of an action instance.

Multi-Dilation Temporal Convolution (MDC) Block: Figure 1 emphasizes the significance of choosing proper temporal receptive field when trying to detect a boundary point or mid-point (hereafter we call it *critical point*). Consecutive video frames tend to be visually correlated, and investigating video frames sufficiently far away (via temporal 1-D convolution with video features F) is important for finding critical points. To enlarge the receptive field under budgeted computation, we devise a convolutional unit that ensembles multiple dilation rates. The architecture is illustrated in Figure 3. All dilated temporal 1-D convolutions have the same kernel size, yet with a typical choice $d_1 < d_2 < d_3$ that defines increasingly larger receptive fields. Denote it with the notation MDC-(d_1, d_2, d_3). A dilation rate of 1 boils down to a normal convolution. The outputs from all dilated convolutions are simply averaged, returning fused contextual information. Note that a skip connection is inserted after the average operation, such that the dilated convolutions are reinforced to focus on learning the residual.

Multi-branch Stacked MDCs: The duration of video action instances widely varies, typically ranging from 1/10 seconds to a few minutes. For actions of different durations, the required scales of temporal context information are different. To localize short-duration action instances, over-large receptive field of temporal convolutions can harmfully bring irrelevant information that distracts the parameter optimization. While for long action instances, a small temporal receptive field may miss some key discriminative information, such as

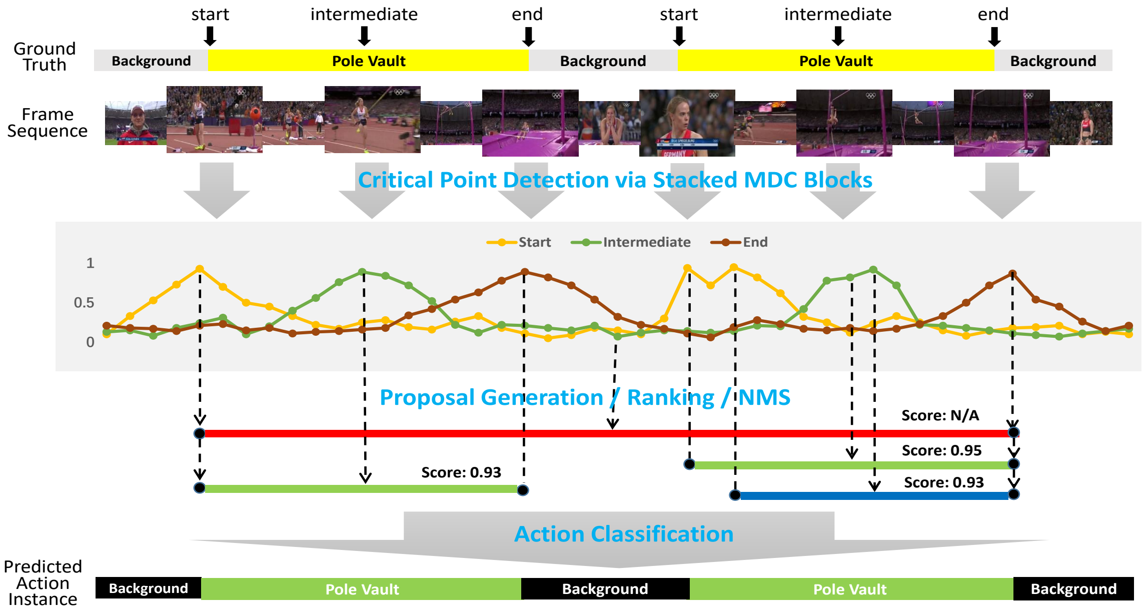

Fig. 2. Pipeline of our proposed temporal scale aggregation (TSA) network. See Section 3 for more details.

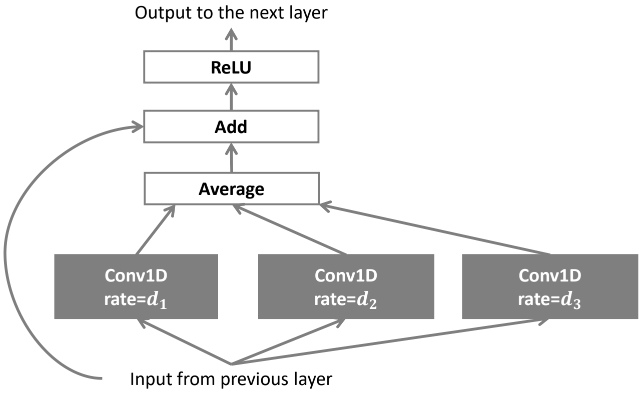

Fig. 3. Multi-dilation temporal convolution (MDC) block. $d1$, $d2$ and $d3$ are the dilation rates of temporal 1D convolutions.

being unable to reach all points A, B and C when judging O as in Figure 1(a).

Unfortunately, it is arguably impossible for finding a one-for-all temporal receptive field. We propose to novelly ensemble multiple network branches. Each of these branches contains stacked MDC blocks parameterized with different dilation rates. State differently, we customize each branch such that it captures videos of specific durations. The network architecture is shown in Figure 4. As seen, the input video snippet features first go through several shared 1-D temporal convolutions for lightly exchanging among adjacent frames. Three branches of stacked MDC blocks are followed. The dilation rates directly control the extent of receptive field. For example, a 2-stack MDC-$(1, 2, 3)$ (i.e., the leftmost branch) or MDC-$(1, 5, 7)$ (i.e., the rightmost) implies a receptive fields of 13 or 29 respectively. One can flexibly tailor the dilation parameters or branch count to fit the videos under inspection. Outputs from all branches are pooled for further processing (we use average pooling).

Objective of Critical Point Detection: Given an untrimmed video V of T frames (or snippets) and the corresponding features $F = [f_1, f_2, \ldots, f_T]$, each starting / ending point pair $(t^{(s)}, t^{(e)})$ temporally de-limits an action instance in this video and also implies an intermediate point $t^{(i)} = (t^{(s)} + t^{(e)})/2$. Let $\Gamma = \{\gamma_k = (t_k^{(s)}, t_k^{(e)}, t_k^{(i)}), k = 1 \ldots K\}$ be a collection of K annotations in the video V.

Let us introduce three notations $Y^{(s)}, Y^{(i)}, Y^{(e)} \in \{0, 1\}^T$ to denote the ground truth of starting / intermediate / ending points, respectively. A direct treatment is to set all $t_k^{(s)}, k = 1 \ldots K$ in $Y^{(s)}$ to 1, otherwise 0. However, since the critical point is highly sparse, such treatment leads to heavy imbalance between positive and negative labels. To mitigate this issue, we inflate each annotated critical point, such as expanding point $t_k^{(s)}$ to a region $[t_k^{(s)} - \delta \cdot L_k, t_k^{(s)} + \delta \cdot L_k]$, where $L_k = t_k^{(e)} - t_k^{(s)}$ and δ is a hyper-parameter (we empirically set to 0.1 in all experiments). All time locations in the expanded region is set to be positive. Likewise, $Y^{(i)}, Y^{(e)}$ are calculated in a similar protocol.

In Figure 4, the outputted feature maps from three branches are aggregated via average pooling. After going through some Conv1D and sigmoid layers, the network eventually renders three probability sequences $P^{(s)}, P^{(i)}, P^{(e)} \in [0, 1]^T$, describing our estimation for a point belonging to starting / intermediate / ending respectively. The pooling-based aggregation ensures a salient probabilistic response, once any branch of stacked MDCs captures the mid-point or boundary-like pattern around this critical point.

The overall objective to be minimized is defined as the sum over three kinds of critical points, i.e., $\mathcal{J} = \mathcal{J}^{(s)} + \mathcal{J}^{(i)} + \mathcal{J}^{(e)}$, where each term on the right hand side adopts a cross-

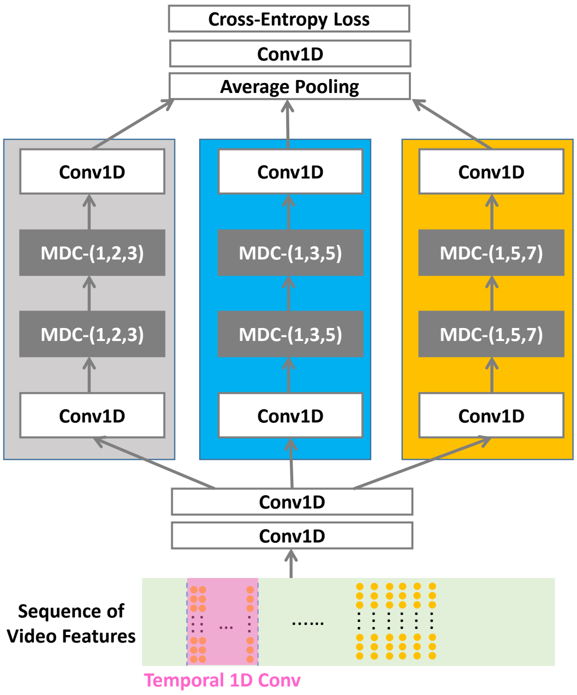

Fig. 4. The architecture of multi-branch stacked MDC blocks which calculates critical point's probability sequences $P^{(s)}, P^{(i)}, P^{(e)} \in [0,1]^T$. Each branch is designed for tackling videos of specific durations. We omit all ReLU and sigmoid activation layers for saving space.

entropy loss. For example,

$$\begin{aligned} \mathcal{J}^{(s)} &= \sum_{i=1\ldots T} -[Y^{(s)}(i)\cdot\log(P^{(s)}(i)) \\ &\quad +(1-Y^{(s)}(i))\cdot\log((1-P^{(s)}(i)))]. \quad (1) \end{aligned}$$

3.3. Proposal Generation and Ranking

The set of action proposals for a video are obtained via sequentially conducting the operations below:

Critical Point Selection and Pairing: The probability sequences $P^{(s)}, P^{(i)}, P^{(e)} \in [0,1]^T$ are designed to indicate how likely a point is a critical point. Therefore one can find critical points by simply thresholding the probability values, say 0.9 as we adopt in all experiments. However, this strategy often misses many true critical points. In order to elevate the recall rate, we also find local maxima by comparing a point with its temporal neighbors and mark them as critical points. We perform above operations on both $P^{(s)}$ and $P^{(e)}$, obtaining candidate starting / ending point sets $C^{(s)} = \{c^{(s)}\}$, $C^{(e)} = \{c^{(e)}\}$ respectively.

Next, any two points $c^{(s)} \in C^{(s)}$ and $c^{(e)} \in C^{(e)}$ will be paired to generate a proposal when they satisfy the conditions below:

1. The distance between $c^{(s)}$ and $c^{(e)}$ is within $[d_{min}, d_{max}]$, where $[d_{min}, d_{max}]$ are the smallest and largest durations estimated from annotated action instances in the training set.

2. Let $c^{(i)} = (c^{(s)} + c^{(e)})/2$ be the mid-point and $P^{(i)}(c^{(i)})$ be the corresponding probability as an intermediate critical point. A low value of $P^{(i)}(c^{(i)})$, as exemplified by the proposal highlighted in red in Figure 2, implies that $c^{(s)}, c^{(e)}$ may indeed come from different action instances and the pair shall be abandoned.

Bayesian Proposal Ranking: The proposal set generated by the previous step tends to be noisy. A follow-up step of proposal ranking can effectively remove many false negatives. In order to score an arbitrary proposal defined by boundary points $c^{(s)}, c^{(e)}$, we adopt a Bayesian formulation as below:

$$P\left(\overline{c_i^{(s)} c_i^{(e)}}\right) = P^{(s)}(c_i^{(s)})\cdot P^{(e)}(c_i^{(e)})\cdot \phi(c_i^{(s)}, c_i^{(e)}), \quad (2)$$

where $P^{(s)}, P^{(e)}$ are afore-mentioned point-wise probabilities of being a critical point. $\phi(c_i^{(s)}, c_i^{(e)}) : \mathbb{R}^+ \times \mathbb{R}^+ \mapsto [0,1]$ represents some compatibility function to be learned.

To learn $\phi(c_i^{(s)}, c_i^{(e)})$, let us first define a feature representation for the segment $\overline{c_i^{(s)} c_i^{(e)}}$. Inspired by recently-proposed temporally-structured segment networks [2], we extend $\overline{c_i^{(s)} c_i^{(e)}}$ outwards (in practice this segment is resized to $2 \times (c^{(e)} - c^{(s)})$) to include more non-action background information. 8 feature vectors are uniformly sampled from the resized segment by reading the feature sequence F. After concatenation, this forms a feature representation for the segment $\overline{c_i^{(s)} c_i^{(e)}}$. Firstly, the feature representation of each segment is fed into a temporal convolution layer to integrate the context information and reduce the feature dimension. Then the feature maps generated by the temporal convolution layer are flattened and fed into a 2-layer fully-connected network to compute a probabilistic value, which is exactly $\phi(c_i^{(s)}, c_i^{(e)})$.

To enforce the learned $\phi(c_i^{(s)}, c_i^{(e)})$ to be informative, we borrow the idea in [4] and solve:

$$\min \sum_i \ell\left(\phi(c_i^{(s)}, c_i^{(e)}), \sup_{\gamma_k \in \Gamma} IoU(\overline{c_i^{(s)} c_i^{(e)}}, \gamma_k)\right), \quad (3)$$

where the sum is calculated over all potential proposals. We choose $\ell(\cdot)$ to the smooth L_1 loss [6]. $IoU(\cdot,\cdot)$ is an intersection-over-union operator between two 1-D segments.

Redundancy Removal: Learning to calculate $P\left(\overline{c_i^{(s)} c_i^{(e)}}\right)$ enables us to conduct non-maximum suppression (NMS) to remove redundant proposals. We experiment with either naive greedy NMS to soft-NMS [13], depending on which the competing method under comparison uses. An example of removed proposal is highlighted in blue color in Figure 2.

3.4. Action Classification

The last step of two-stage video action localization is feeding proposals into an action classifier. It categories the proposal

Table 1. Comparisons in terms of AR@AN on THUMOS14.

Feature	Method	AR@AN		
		@50	@100	@200
C3D	DAPs [9]	13.56	23.83	33.96
C3D	SCNN-prop [1]	17.22	26.17	37.01
C3D	SST [10]	19.90	28.36	37.90
C3D	TURN [11]	19.63	27.96	38.34
C3D	MGG [15]	29.11	36.31	44.32
C3D	BSN + Greedy-NMS [4]	27.19	35.38	43.61
C3D	BSN + Soft-NMS [4]	29.58	37.38	45.55
C3D	BMN + Greedy-NMS [16]	29.04	37.72	46.79
C3D	BMN + Soft-NMS [4]	32.73	40.68	47.86
C3D	Ours + Greedy-NMS	34.42	42.98	48.64
C3D	Ours + Soft-NMS	**36.11**	**43.25**	**49.18**
Flow	TURN [11]	21.86	31.89	43.02
2-Stream	TAG [17]	18.55	29.00	39.41
2-Stream	CTAP [18]	32.49	42.61	51.97
2-Stream	MGG [15]	39.93	47.75	54.65
2-Stream	BSN + Greedy-NMS [4]	35.41	43.55	52.23
2-Stream	BSN + Soft-NMS [4]	37.46	46.06	53.21
2-Stream	BMN + Greedy-NMS [16]	37.15	46.75	54.84
2-Stream	BMN + Soft-NMS [4]	39.36	47.72	54.70
2-Stream	Ours + Greedy-NMS	42.09	49.35	54.24
2-Stream	Ours + Soft-NMS	**43.40**	**50.30**	**55.50**
P3D	Ours + Greedy-NMS	46.37	52.04	55.17
P3D	Ours + Soft-NMS	**48.39**	**54.11**	**58.37**

to one of many pre-defined action classes, or the null class. Since the major scope of this work is about a novel scheme for proposal generation, we directly adopt action classifiers widely used in previous works. This eases more focused comparisons with other action localization methods. Specifically, we use UntrimmedNet (UNet) [14] or SCNN-classifier (SCNN-cls) [1] on THUMOS14.

4. EXPERIMENTS

4.1. Dataset Preparation and Implementation

We conduct experiments on the THUMOS14 [19]. THUMOS14 contains videos from 20 sports action classes. There are 200 and 212 temporally-annotated videos in validation and testing sets respectively. Following the settings in previous works, we use 200 untrimmed videos in the validation set to train our model and evaluate on the test set.

We extract two sets of heterogeneous video features in order to investigate how different features affect the final performance, including: 1) **two-stream features**: The specific implementation of two-stream model in [20] is used. We adopt the version pre-trained on ActivityNet-1.3 in [20]. 2) **P3D [21] or C3D [22] features**:Two separate P3D models are trained from consecutive frames and flows on Kinetics [23] respectively. The C3D model are pre-trained on the UCF-101 dataset [24].

4.2. Evaluation Protocol

Action Proposal Generation: *Average recall* (AR) at different IoU is typically used as the evaluation metric. Following common practice, we calculate AR with different *average number* (AN) of proposals per video to evaluate the relationship between recall and proposal number.

Table 2. Ablation study results on THUMOS14 in terms of AR@AN.

Method	AR@AN		
	@50	@100	@200
multi-Conv	31.48	40.83	47.78
single-MDC	37.18	45.04	51.07
multi-MDC	40.83	47.92	53.22
multi-MDC+ranking	**43.40**	**50.30**	**55.50**

Action Classification: We adopt standard *mean average precision* (mAP) metric. We report mAP at different IoU thresholds. On THUMOS14, the IoU thresholds are 0.3, 0.4, 0.5, 0.6, 0.7.

4.3. Comparisons for Proposal Generation

Table 1 summarizes all comparisons conducted on the test set of THUMOS14. Regarding proposal generation, our proposed TSA-Net consistently outperforms other methods when AN ranges from 50 to 200. Specifically, for AR@50, our method significantly improves the performance from the previous record 39.93% in the literature to 43.40% using identical two-stream features. Using stronger P3D features can further elevate the AR@AN scores.

4.4. Ablative Study for Proposal Generation

We investigate the contributions from several key components in the proposed TSA-Net. All experiments in this ablation study are performed on the THUMOS14 dataset with two-stream features. We compare four settings: 1) replace the MDC blocks of each branch with standard 1-D convolutions (multi-Conv); 2) a single-branch network with MDC blocks (single-MDC). The average performance of 3 single branches are used for comparison; 3) a multi-branch network with MDC blocks (multi-MDC); 4) a multi-branch network with MDC blocks and scoring the proposals by $P\left(\overline{c_i^{(s)} c_i^{(e)}}\right)$ in Eqn. 2 (multi-MDC + ranking). The first three settings scoring the proposals by $P^{(s)}(c_i^{(s)}) \cdot P^{(e)}(c_i^{(e)})$ in Eqn. 2. Table 2 shows that both our design choices and Bayesian proposal ranking contribute in the overall performance.

4.5. Experimental Analysis of Action Localization

The evaluations performed on the testing set of THUMOS14 is shown in Table 3. Most baselines adopt two-stream or C3D features while few use spatio-temporal features I3D [23] (akin to the P3D features that we extract yet more computationally heavy), such as TAL-Net [5].

Table 3 exhibits that our proposed TSA-Net outperforms state-of-the-art action localization methods by significant margins when the IoU threshold varies from 0.3 to 0.7. Specifically, for mAP@0.5, our method significantly improves the performance from 38.8% to 44.1%, when both use the two-stream features. The performance of our method is further improved by using P3D features, and we achieved an mAP of 53.0% when the IoU threshold is 0.5.

Table 3. Action classification comparisons on THUMOS14 in terms of mAP@IoU. "–" denotes that a method uses its own action classifier. "n/a" denotes that the corresponding performance is not reported in the original literature.

Proposal Method	Classifier	0.7	0.6	0.5	0.4	0.3
SCNN [1]	–	5.3	10.3	19	28.7	36.3
CDC [12]	–	8.8	14.3	24.7	30.7	41.3
TCN [25]	–	9.0	15.9	25.6	33.3	n/a
R-C3D [3]	–	9.3	19.1	28.9	35.6	44.8
TAL-Net [5] (I3D)	–	20.8	33.8	42.8	48.5	53.2
P-GCN[26] (I3D)	–	n/a	n/a	49.1	57.8	63.6
SST [10]	SCNN-cls	n/a	n/a	23.0	n/a	n/a
TURN [11]	SCNN-cls	7.7	14.6	25.6	33.2	44.1
BSN [4]	SCNN-cls	15.0	22.4	29.4	36.6	43.1
CTAP [18]	SCNN-cls	n/a	n/a	29.9	n/a	n/a
MGG [15]	SCNN-cls	15.8	23.6	29.9	37.8	44.9
Ours(2-Stream)	SCNN-cls	14.6	25.3	35.3	41.4	46.0
Ours(P3D)	SCNN-cls	**21.8**	**32.0**	**40.2**	**47.3**	**51.4**
SST [10]	UNet	4.7	10.9	20.0	31.5	41.2
TURN [11]	UNet	6.3	14.1	24.5	35.3	46.3
BSN [4]	UNet	20.0	28.4	36.9	45.0	53.5
MGG [15]	UNet	21.3	29.5	37.4	46.8	53.9
BMN [16]	UNet	20.5	29.7	38.8	47.4	56.0
Ours(2-Stream)	UNet	21.8	33.0	44.1	52.0	55.8
Ours(P3D)	UNet	**28.8**	**42.4**	**53.0**	**61.4**	**65.6**

5. CONCLUSIONS

We propose TSA-Net for video action localization which effectively compiles temporal context from multi-scales. The core designs include a multi-branch stacked MDC blocks for temporal aggregation and a light-weight sub-network for regressing the proposal's Bayesian confidence. The proposed TSA-Net outperforms competing methods and re-calibrate the state-of-the-art performances on THUMOS14 dataset.

Acknowledgement: This work is supported by Beijing Municipal Commission of Science and Technology (Z181100008918005), Beijing Natural Science Foundation (Z190001) and Tencent AI Lab Rhino-Bird Focused Research Program (JR202021).

6. REFERENCES

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

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

[3] Huijuan Xu, Abir Das, and Kate Saenko, "R-C3D: region convolutional 3d network for temporal activity detection," in *ICCV*, 2017.

[4] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang, "BSN: boundary sensitive network for temporal action proposal generation," in *ECCV*, 2018.

[5] 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," in *CVPR*, 2018.

[6] Ross B. Girshick, "Fast R-CNN," in *ICCV*, 2015.

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

[8] Tianwei Lin, Xu Zhao, and Zheng Shou, "Single shot temporal action detection," in *ACM Multimedia*, 2017.

[9] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem, "Daps: Deep action proposals for action understanding," in *ECCV*, 2016.

[10] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles, "SST: single-stream temporal action proposals," in *CVPR*, 2017.

[11] Jiyang Gao, Zhenheng Yang, Chen Sun, Kan Chen, and Ram Nevatia, "TURN TAP: temporal unit regression network for temporal action proposals," in *ICCV*, 2017.

[12] 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," in *CVPR*, 2017.

[13] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis, "Soft-nms - improving object detection with one line of code," in *ICCV*, 2017.

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

[15] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang, "Multi-granularity generator for temporal action proposal," in *CVPR*, 2019.

[16] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen, "BMN: boundary-matching network for temporal action proposal generation," in *ICCV*, 2019.

[17] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang, "A pursuit of temporal accuracy in general activity detection," *CoRR*, vol. abs/1703.02716, 2017.

[18] Jiyang Gao, Kan Chen, and Ram Nevatia, "CTAP: complementary temporal action proposal generation," in *ECCV*, 2018.

[19] H. Idrees, A. R. Zamir, Y. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, "The thumos challenge on action recognition for videos "in the wild"," *Computer Vision and Image Understanding*, vol. 155, pp. 1–23, 2017.

[20] 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," *CoRR*, vol. abs/1608.00797, 2016.

[21] Zhaofan Qiu, Ting Yao, and Tao Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *ICCV*, 2017.

[22] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[23] João Carreira and Andrew Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *CVPR*, 2017.

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

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

[26] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan, "Graph convolutional networks for temporal action localization," in *ICCV*, 2019.