Extracting Action Sensitive Features to Facilitate Weakly-Supervised Action Localization

Zijian Kang, Le Wang, Ziyi Liu, Qilin Zhang, Nanning Zheng

Extracting Action Sensitive Features to Facilitate Weakly-supervised Action Localization

Zijian Kang[1], Le Wang[1], Ziyi Liu[1], Qilin Zhang[2], and Nanning Zheng[1]

[1] Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R.China
[2] HERE Technologies, Chicago, IL 60606, USA

Abstract. Weakly-supervised temporal action localization has attracted much attention among researchers in video content analytics, thanks to its relaxed requirements of video-level annotations instead of frame-level labels. However, many current weakly-supervised action localization methods depend heavily on naive feature combination and empirical thresholds to determine temporal action boundaries, which is practically feasible but could still be sub-optimal. Inspired by the momentum term, we propose a general-purpose action recognition criterion that replaces explicit empirical thresholds. Based on such criterion, we analyze different combination of streams and propose the Action Sensitive Extractor (ASE) that produces action sensitive features. Our ASE sets temporal stream as main stream and extends with complementary spatial streams. We build our Action Sensitive Network (ASN) and evaluate on THUMOS14 and ActivityNet1.2 with different selection method. Our network yields state-of-art performance in both datasets.

Keywords: Action localization · Weaky-supervised · Two-stream.

1 Introduction

Temporal action localization (TAL) in untrimmed videos has attracted more and more attention in recent years, many methods [23, 38, 14, 11, 9, 28, 43, 41] that greatly enhanced performance have been developed. Because labeling action boundary in untrimmed video is expensive, some researchers [26, 38, 30, 33, 25] proposed to use video-level action annotation to produce snippet level action localization results, which greatly reduced demand for human laboring and yield comparable performance. These studies combine Multiple Instance Learning (MIL) [7] and attention mechanism [38, 26, 25] with Deep Convolutional Neural Networks (DCNN) to produce clip presentations. Then, action detection criterions maps clip presentations to Class Activation Sequence (CAS) which determines which snippet includes action.

However, these weakly-supervised methods share two convenient assumptions that might be too optimistic in the real world. The first assumption is empirical thresholds to determine temporal action boundaries could be obtained in a trivial manner. This implicit assumption could be far from reality, given the

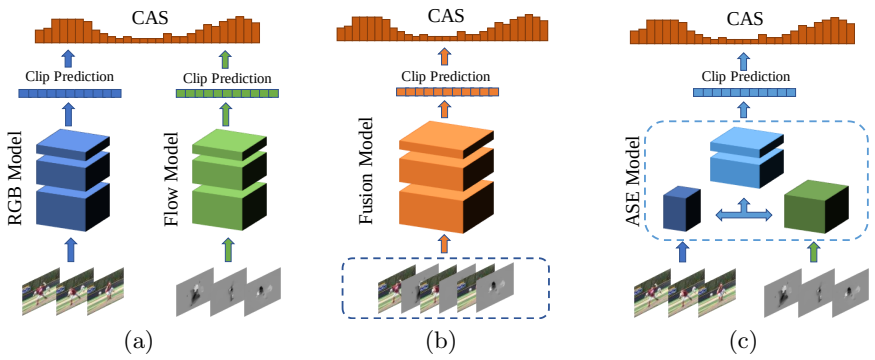


Fig. 1. Illustration of different strategies to combine two-stream features. (a) Lateral fusion of two-stream features. (b) Concatenating two-stream features for processing. (c) Our action sensitive extractor.

diversity of datasets and applications. The second assumption is that straightforward fusion strategies are adequate in weakly-supervised TAL because of the prevailing two-stream networks [3, 32, 39], where CAS is either generated separately and fused by weighted average [38, 30, 25], or generated by concatenated features [26] and [30] regression methods. With the two-stream network, each stream is independently trained via backpropagation and no interactions happen between streams. These two strategies are straightforward to implemented but we argue that there could possibly be a better alternative.

To address these challenges, we design a general-purpose action detection criterion and an alternative stream fusion strategy. Specifically, we design the action detection criterion based on attention mechanism with momentum-inspired threshold generated in training stage. An analysis in stream combination options results in the proposed Action Sensitive Extractor(ASE) as shown in Fig. 1. Inspired by recent literature in spatial and temporal interaction [35, 10], the proposed ASE prudently selects action sensitive features between two streams and produces activations. In the ASE, we handle spatial and temporal stream asymmetrically with respect to different sensitivities in actions. With our action detection criterion and the ASE, we build **Action Sensitive Network (ASN)** for Weakly-supervised TAL.

Main contributions of this paper include (1) a comparative analysis on stream fusion strategies with the proposed Action Sensitive Extractor (ASE), and (2) a new flexible action localization criterion which generates high quality CAS. The performance gains of the proposed ASN algorithm are verified on two challenging public datasets.

2 Related Works

Video Action analyze has been wildly discussed in several years. Most studies focus on action recognition in trimmed videos. Many novel structures have been

proposed for videos [15, 8, 19, 3] based on DCNN neural networks [16, 17, 37]. Two-stream network [32] was one design which employs RGB images and optical Flow with lateral fusion. Based on two-stream network, temporal segment network (TSN) [39] was proposed to analyze long-term temporal data. TSN has been used as backbone in different tasks [43, 38] with good performance. To further leverage optical Flow, [35] proposed a novel structure for optical Flow. Recent proposed SlowFast Network [10] uses two path way to process videos similar to two-stream network. In SlowFast, a fast pathway handles wide temporal motions and a slow pathway handles rich local details.

Action localization has been greatly improved based on video action analyze. Many neural architectures and methods [21, 13, 24, 9] have been developed for supervised learning. However, those studies heavy rely on data annotations of action sequences, which are expensive to acquire. To incorporate more data in training, Sun et al. [34] proposed to use web images and video-level annotation to handle TAL. Moreover, hide and seek [33] discovered how to force network focus on most discriminating part. UntrimmedNet [38] designed a novel structures that trains high-quality network on untrimmed videos and proposed a method which efficiently selects action segments. UntrimmedNet not only provide a good solution for Localization but is also a good baseline model that generates local representation. Based on extracted feature representation, AutoLoc [30] discovered an anchor generation and selection standard on feature sequences. W-TALC [26] and Nguyen et al. [25] discovered feature based networks with different auxiliary loss functions and attention mechanism.

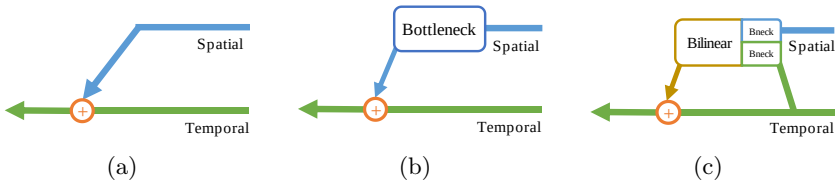
3 Action Sensitive Network

In this section, our proposed Action Sensitive Network will be introduced. Section 3.1 describes the ASE we proposed. Section 3.2 describes our momentum-inspired action detection criterion. In the last section, we introduce details of ASN.

3.1 Action Sensitive Extractor

In this section, we propose models to extract action sensitive features. Our proposal is to train a network that maximally leverage action sensitive features in two streams. Because actions are described in moving image in videos, spatial stream with only one frame perception unlikely to recognize action directly. While temporal stream have wider temporal perception and inherently sensitive at motion boundary [27]. Detial analysis can be found in our experiment section. Inspired by SlowFast network [10], where spatial (slow) and temporal (fast) are fed into different network architectures with different channels and different temporal perception, we propose our models that treat temporal and spatial asymmetrically.

In general, we use learned action sensitive knowledge(inherit from temporal stream) as main stream. We discover different structures to extract beneficial



(a) (b) (c)

Fig. 2. Data flow of different ASE settings. We set temporal stream as main stream and spatial stream as reinforce stream. Reinforced streams are fed into classification and activation branch then. (a) Fusion Model. (b) Bottleneck Model. (c) Bilinear Bottleneck Model.

features to reinforce main stream. We adopt strategy in DenseNet [17] that we concatenate our main stream and reinforce stream together for classification and attention calculation. We call our extraction model **Action Sensitive Extractor(ASE)**, different settings of ASE are shown in Fig. 2. For simplicity, we still use single fully-connected layer for classification or attention branch. ASE with classification and attention branch is referred as ASE model.

Fusion with Temporal Knowledge To leverage temporal features that are related to actions, we propose to build a network that initialized with temporal features and extended with spatial features. To achieve this goal, we adopt methods from [4], our network on fusion(concatenated) features is initiated with pretrained temporal weights and zero spatial weights. For example, equation 1 shows the classification branch for fusioned features. To inherit knowledge in temporal classifier, we set $\mathbf{W}^t$ and $\mathbf{b}^t$ to pretrained temporal weights, while $\mathbf{W}^s$ and $\mathbf{b}^s$ are set to 0. We also apply same method to attention branch.

$$\begin{aligned} \mathbf{c} &= \mathbf{W}^f \cdot \mathbf{x}^f + \mathbf{b}^f \\ &= \left[\mathbf{W}^t, \mathbf{W}^s\right] \cdot \begin{bmatrix} \mathbf{x}^t \\ \mathbf{x}^s \end{bmatrix} + \begin{bmatrix} \mathbf{b}^t \\ \mathbf{b}^s \end{bmatrix} \end{aligned} \tag{1}$$

Bottleneck Model To limit overfitting with spatial features, we further study on limiting and distilling spatial features. Different from former studies that enforce loss [26], we simply use special designed network architecture. As a naive attempt, we use a bottleneck layer to extract knowledge from spatial features. The bottleneck layer includes a dropout, a fully-connected layer and ReLU activation. The features extracted from bottleneck layer are concatenated with temporal features and feed to classification and attention branch. Our bottleneck layer extracts most expressive spatial features that help to identify actions.

Bilinear Bottleneck Model Bottleneck model will remove unnecessary spatial feature but can't introduce interactions. In recent works [40, 5], bilinear layers are proposed to aggregate spatial-temporal features. To make use of connection between streams, we propose to use bilinear block to aggregate features. In

our work, we propose to use two fully-connected bottleneck layer to aggregate features in each stream and use bilinear layer to combine temporal and spatial features. We use 0.5 dropout before bottleneck layer and bilinear layer. We use ReLU activation after fc layers and bilinear layer. The aggregated features are concatenated with temporal features as in bottleneck model. Hidden layer size of bottleneck layer and bilinear layer are set to same for simplicity.

3.2 Action Detection Criterion

Here we represent our action detection criterion. In our study, we propose to trim fixed proportion of clips as background, since proportion of background frames is relative stable in each dataset. We set our threshold as quantile of attention values during training, similar to batch normalization [18], where mean and standard deviation in each batch is recorded and reused, to deal with fluctuation.

Quantile level describes desired proportion of clips. Level of quantile defines how much proportion the quantile divide, e.g. quantile at 30% means around 30% of clips in each batch have lower attention than the quantile. For each batch in training, we sort attentions of each clip in this batch and sample a attention value at desired level. Current quantile is updated by a momentum factor according to equation 2. Quantile is fixed during testing.

$$q^{t+1} = \alpha q^t + (1 - \alpha)q \tag{2}$$

Our method is simple and cross modality, it is easy to apply our action detection criterion to any attention based localization problem across different settings. Note that the quantile maybe different across datasets, since proportion of background frames maybe different.

3.3 Network Details

Having explained key components, now we introduce details of our ASN as shown in Fig. 3. To efficiently look over long video, we break videos into different levels. In bottom level, each frame is represented separately as frame. We use features from our two-stream pretrained DCNN model as representation. Then the middle level, which is clip level. We average our features sampled in short temporal period as clip representation since close frames in videos should be correlated. To distill key knowledge and trim noise, we use Action Sensitive Extractor to extract features and feed to classification and attention branch. The highest level is video-level, which is aggregated by attention mechanism. This level is symmetrical to annotations.

In our study, we discover a setting based on extracted features from UntrimmedNet [38]. Following UntrimmedNet [38], we randomly sample 7 clips for untrimmed videos, 1 clips for video clips. For each clip, 3 frames are sparsely sampled as in TSN [39] and averaged as clip representation. Two fully-connected layers are used to produce classification and attention respectively. Dropout of 0.5 is used only before classification layer. To fuse clip level activations, we apply

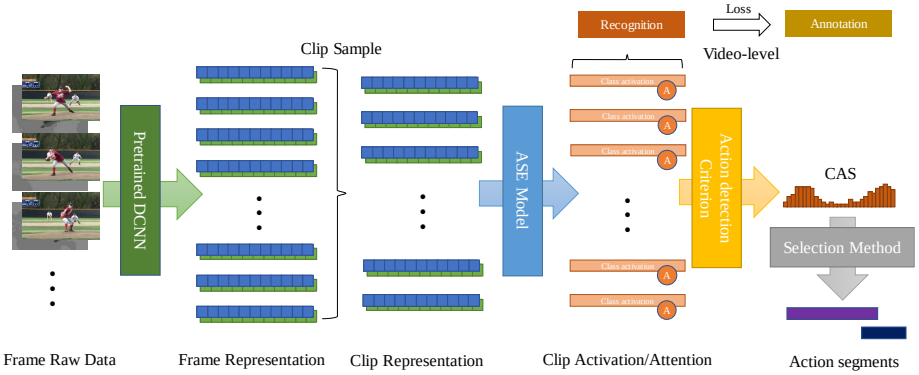


Fig. 3. Our full network for action recognition and detection. We use our ASE model to produce frame activations and attentions.Video-level classification activations are optimized with video-level annotations. CAS is generated by action detection criterion. Action segments are selected based on CAS.

softmax operation on attentions x from clip 1 to t. The normalized attentions $\bar{x_i^a} = \frac{exp(x_i^a)}{\sum_{j=1}^{t} exp(x_j^a)}$ are used to fuse clip level classifications into video level prediction $\mathbf{x}^c$, where $\mathbf{x}^c = \sum_{i=1}^{t} \bar{x_i^a} \mathbf{x}_i^c$. Next, we apply softmax operations among each dimension of prediction and optimize with multi-label cross-entropy loss.

$$l(\mathbf{x}^c, \mathbf{y}) = \sum_{i=1}^{t} y_i log(\frac{exp(x_i^c)}{\sum_{j=1}^{t} exp(x_j^c)}) \tag{3}$$

During testing, we use strategy similarly to [38] and [30]. Each clip is aggregated every 15 frames. ASE model produces classification and attention activations for each clip. For video recognition, we soften our attentions by a factor (sets to 3) at first. Then, clips are fused to video representation according to their attentions as in training. For video detection, we generate CAS of size $clip\_number \times class\_number$ and feed it into selection method. Firstly, we apply softmax operation on clip classification activations. Then, we apply threshold on video-level prediction, clip activations of video unrelated class are set to 0 in CAS. Thirdly, we apply attention level threshold, clips with attentions lower than threshold are set to 0 in CAS. Finally we feed our CAS into selection method to generate action segments.

4 Experiments

4.1 Dataset

THUMOS14 [20] has 101 classes for recognition and 20 classes out of 101 for action detection. THUMOS14 includes training set, validation set and testing set. Training set includes action video clips, validation and testing set includes untrimmed videos. In THUMOS14, 15 instances of actions covers 29% of video

on average [28]. We train our model on training set and validation set, we test our model on testing set.

ActivityNet1.2 [2] has 100 classes for both detection and recognition. It is divided into training set, validation set and test set. In ActivityNet, 1.5 instances of actions covers 64% of video on average [28]. We train our model on training set and test on validation set.

4.2 Implementation Details

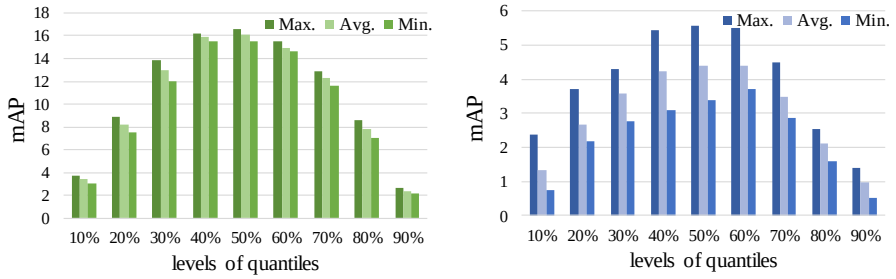
We train our ASN using features extracted by UntrimmedNet pretrained model, which trained on same dataset and subsets as UntrimmedNet. We train our network with Nestrov momentum [36] of 0.9, weights decay of 0.0005. Batch size is set to 512 for THUMOS14 validation set and 8192 for THUMOS14 training set. Batch size is set to 512 for ActivityNet1.2. On THUMOS14 [20], we train 80 epochs jointly on training set and validation set. Our learning rate is set to 0.1 and decay 10 times on 40th and 60th epoch. On ActivityNet1.2 [2], we train 160 epochs jointly on training set. Learning rate is set to 0.1 and decay on 80th and 120th epoch.

4.3 Ablation study

In this section, we explore our action detection criterion at different levels of quantiles and different model settings. For simplicity and efficiency, we use naive approach in UntrimmedNet [38] as selection method in ablation study on THUMOS14. This method only selects consecutive activated frames in CAS. For a selected snippet from clip timestamp n to $k+n$ with label v, confidence scores s are evaluated by video-level activation c_v and average activation as shown in Equation 4. Where we use $\lambda = 0.2$ in our experiment.

$$s = \frac{1}{n+1} \sum_{i=k}^{k+n} c_v^i + \lambda c_v \tag{4}$$

Evaluation of Action Detection Criterion To demonstrate efficiency of our action detection criterion, we train our network 10 times and record performance on testing set under different quantiles. As baseline, spatial (RGB) and temporal (Flow) models are treated separately. Different levels of quantiles are recorded and tested on localization task. We train our network 10 times and quantiles are recorded at level 10%, 20%, 30% to 90%. To compare with former studies, we also apply our methods on pretrained weights provided by [38], the quantiles of pretrained models are recorded by running on THUMOS14 validation set. CAS of two-stream model in localization are generated by two steps. First, clip level classifications after softmax are averaged. Second, attention scores of each stream are normalized by each threshold and averaged. Video-level recognition results are set to average of two streams.



(a) Flow (b) rgb

Fig. 4. Localization mAP of flow and rgb model under different quantiles on THUMOS14. mAP is recorded under 0.5 IoU threshold.

Results of spatial and temporal model under IoU threshold of 0.5 are shown in Fig. 4. The performance of pretrained model on different quantiles are shown in Fig. 5. For different models, performance peaks locate near 50% quantiles. During training, we find attention quantiles are fluctuating but performances are generally stable. Notably, spatial performances are worse and more unstable than temporal. We also compare our methods with original UntrimmedNet [38]. Performances of our best models under different settings are shown in Table 1. Our action detection criterion can achieve high performance with only temporal stream.

Evaluation of Streams We evaluate different combination of streams as shown in Table 1. We evaluate on spatial (RGB), temporal (Flow), two-stream and fusion stream (concatenated features of RGB and Flow). We also discover attention quality of each stream.

Table 1. Comparison with different settings on THUMOS14. We compare localization mAP under common IoU threshold and recognition accuracy. UntrimmedNet use a slightly different recognition strategy.

Models	Localization (IoU threshold)					Recognition
	0.3	0.4	0.5	0.6	0.7	
Flow pretrained	27.68	21.26	14.87	9.79	5.64	73.93%
RGB pretrained	15.32	8.76	5.02	2.80	1.35	72.29%
Two-stream pretrained	28.50	21.06	14.40	8.75	4.78	82.04%
UntrimmedNet [38]	28.2	21.1	13.7	-	-	*82.2%
Flow stream	28.67	22.43	16.60	10.42	5.63	74.15%
RGB stream	15.32	8.76	5.02	2.80	1.35	72.29%
Fusion stream	20.40	14.50	9.50	5.56	3.03	75.61%
Two-stream	27.87	20.76	14.46	8.59	4.68	79.95%
Two-stream (RGB)	21.88	14.73	9.19	4.97	2.29	-
Two-stream (Flow)	28.57	22.23	16.40	10.04	5.62	-

Surprisingly, temporal stream yield the best localization performance. Streams with spatial features perform poorly. The bad behavior of spatial related streams may because of trivial details in spatial features cause overfitting. In addition, we analyze attention in each stream. For two-stream model, we fix CAS and apply only temporal or spatial attention to our criterion. We find two-stream with temporal attentions yield high performance similar to temporal stream and two-stream with spatial attentions yield low performance similar to fusion stream.

Our experiment shows differences in action sensitivity between two streams. Combining with temporal and spatial information usually yield higher performance in action recognition but lower in localization. We also find commonly used two-stream or fusion strategies are inefficient in weakly-supervised localization task, which are worse than single temporal stream.

Evaluation of ASE We evaluate different ASE model settings. For inherit strategy, we our best flow model as initial weight. For fusion model, bottleneck

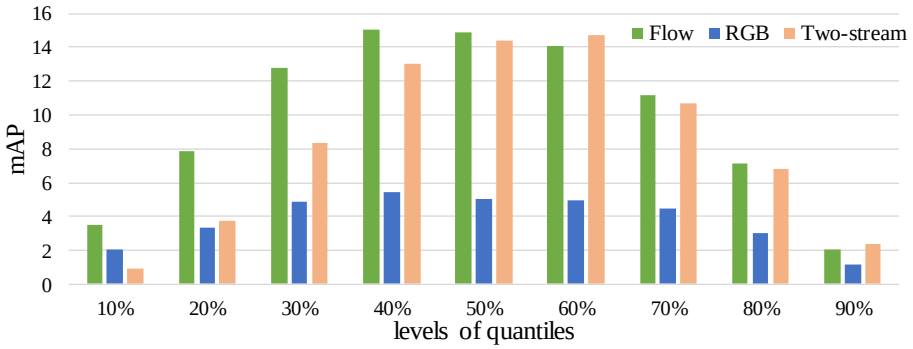


Fig. 5. Localization mAP of pretrained model under different quantiles on THUMOS14. mAP is recorded under 0.5 IoU threshold.

Table 2. Compare with different ASE model settings on THUMOS14.

Models	Localization (IoU threshold)					Recognition
	0.3	0.4	0.5	0.6	0.7	
Flow ours	28.67	22.43	16.60	10.42	5.63	74.15%
Two-stream ours	27.87	20.76	14.46	8.59	4.68	79.95%
Fusion from scratch	20.40	14.50	9.50	5.56	3.03	75.61%
Fusion inherit	26.21	19.38	12.72	7.45	4.02	81.39%
Bottleneck64 inherit	32.73	24.84	17.36	11.12	6.42	78.16%
Bottleneck64 scratch	29.35	22.61	15.91	9.94	5.39	80.92%
Bottleneck128 inherit	32.33	25.13	17.60	10.69	5.64	79.10%
BiBottleneck64 inherit	31.89	25.35	17.74	11.29	6.23	78.20%
BiBottleneck64 scratch	29.12	22.84	16.27	10.30	5.99	74.73%
BiBottleneck128 inherit	32.21	25.34	18.16	11.42	6.23	78.57%

model and bilinear bottleneck model, we compare training from scratch and inherit strategy with feature size of 64. We compare inherit strategy of feature size of 64 and 128 in bottleneck model and bilinear bottleneck model. Our results are shown in Table 2.

Compare with training from scratch, inherit strategy greatly improves recognition and localization except for bottleneck model. For bottleneck model, only localization is slightly improved. This phenomenon may denotes that our bottleneck model has already restrained overfitting. For bottleneck models and bilinear bottleneck models, feature size from 64 to 128 slightly improves performance.

In recognition tasks, fusion model has the highest performance because it can access full information, it also proves that our bottleneck structure does restrain information. For localization, bottleneck model and bilinear bottleneck model performs much higher than fusion model. Bilinear bottleneck models perform slightly higher than bottleneck model, which denotes that bilinear layer does improve interaction. High performance of our proposed ASE models shows its ability to extract action sensitive features.

4.4 Experiments on AutoLoc

To evaluate our final Action Sensitive Network, we use AutoLoc [30] as selection methods and compare with state-of-art results. AutoLoc incorporate Outer-Inter-Contrastive (OIC) loss that evaluate action snippet accurately. To further adjust performance, we increase weights for outer boundary in OIC as follow:

$$L_{OIC} = \lambda A_o(\phi) + A_i(\phi) \tag{5}$$

On THUMOS14, we set λ to 2. We also increase boundary inflation rate to 0.35. These settings help AutoLoc select most distinguishable action snippets. We add more offset anchors to AutoLoc and only use AutoLoc as a selection method over CAS. We show performance of our bilinear bottleneck model with feature size 128 and inherit strategy in Table 4. For ActivityNet1.2 [2], we set λ to 5 and boundary inflation to 0.7. We use quanile at 10% for ActivityNet1.2. Our results are shown in Table 3. Compare with other weakly-supervised TAL methods, our method have advantage especially under higher IoU and reach state-of-art level in both datasets.

Table 3. Comparison with state-of-art methods on ActivityNet1.2 in terms of action localization mAP under different IoU. We only list weakly-supervised methods. All results in this table are based on UntrimmedNet features. We describe selection methods we used in brackets.

Models	Localization (IoU threshold)										Avg.
	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	
UntrimmedNet [38]	7.4	6.1	5.2	4.5	3.9	3.2	2.5	1.8	1.2	0.7	3.6
AutoLoc [30]	27.3	24.9	22.5	19.9	17.5	15.1	13.0	10.0	6.8	3.3	16.0
W-TALC [26]	37.0	33.5	30.4	25.7	14.6	12.7	10.0	7.0	4.2	1.5	18.0
ASN (AutoLoc)	29.8	27.1	25.0	23.1	21.2	18.6	16.1	13.1	9.6	4.4	18.8

Table 4. Comparison with state-of-art methods on THUMOS14 in terms of action localization mAP under different IoU. All weakly-supervised results are based on UntrimmedNet features. We describe selection methods we used in brackets.

		Localization (IoU threshold)				
Supervision	Models	0.3	0.4	0.5	0.6	0.7
Full	S-CNN [31]	36.3	28.7	17.0	10.3	5.3
Full	Yuan et al. [42]	33.6	26.1	18.8	-	-
Full	CDC [29]	40.1	29.4	23.3	13.1	7.9
Full	Dai et al. [6]	-	33.3	25.6	15.9	9.0
Full	SSAD [22]	43.0	35.0	24.6	-	-
Full	Turn Tap [12]	44.1	34.9	25.6	-	-
Full	R-C3D [41]	44.7	35.6	28.9	-	-
Full	SS-TAD [1]	45.7	-	29.2	-	9.6
Full	Gao et al. [11]	50.1	41.3	31.0	19.1	9.9
Full	SSN [43]	51.9	41.0	29.8	19.6	10.7
Full	BSN [23]	53.5	45.0	36.9	28.4	20.0
Weak	Sun et al. [34]	8.5	5.2	4.4	-	-
Weak	Hide and Seek [33]	19.5	12.7	6.8	-	-
Weak	UntrimmedNet [38]	28.2	21.1	13.7	-	-
Weak	AutoLoc [30]	35.8	29.0	21.2	13.4	5.8
Weak	W-TALC [26]	32.0	26.0	18.8	-	6.2
Weak	STPN [25]	31.1	23.5	16.2	9.8	5.1
Weak	ASN (Naive)	32.2	25.3	18.2	11.4	6.2
Weak	ASN (AutoLoc)	35.9	29.4	22.8	15.2	7.3

5 Conclusion

We propose a general action detection criterion which can generate high quality CAS and can apply to different modalities. Based on this threholding method, we analyze performance of different combinations of streams. According to our experiments, spatial and temporal stream contains different information and have different sensitivity in actions. To combine two streams properly, we propose our novel Action Sensitive Network. Two-stream features are treated asymmetry to produce accurate representation without losing sensitivity in actions. We use ASE model to produce clip features and CAS that can be applied to different selection methods. Our network yields state-of-art performance with AutoLoc as selection method. In the future, we can investigate higher level relationship between different streams and apply our method to more modalities.

Acknowledgement

This work was supported partly by National Key R&D Program of China Grant 2017YFA0700800, National Natural Science Foundation of China Grants 61629301 and 61773312, Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001.

References

1. Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.: End-to-end, single-stream temporal action detection in untrimmed videos. In: BMVC (2017)
2. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR (2015)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
4. Chen, T., Goodfellow, I., Shlens, J.: Net2net: Accelerating learning via knowledge transfer. arXiv preprint arXiv:1511.05641 (2015)
5. Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J.: Aˆ 2-nets: Double attention networks. In: NIPS (2018)
6. Dai, X., Singh, B., Zhang, G., Davis, L.S., Qiu Chen, Y.: Temporal context network for activity localization in videos. In: ICCV (2017)
7. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence (1997)
8. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
9. Escorcia, V., Heilbron, F.C., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: ECCV (2016)
10. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. arXiv preprint arXiv:1812.03982 (2018)
11. Gao, J., Yang, Z., Nevatia, R.: Cascaded boundary regression for temporal action detection. arXiv preprint arXiv:1705.01180 (2017)
12. Gao, J., Yang, Z., Sun, C., Chen, K., Nevatia, R.: Turn tap: Temporal unit regression network for temporal action proposals. In: ICCV (2017)
13. Gkioxari, G., Malik, J.: Finding action tubes. In: CVPR (2015)
14. Gudi, A., van Rosmalen, N., Loog, M., van Gemert, J.: Object-extent pooling for weakly supervised single-shot localization. arXiv preprint arXiv:1707.06180 (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV (2014)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
17. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
18. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
19. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence (2013)
20. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/ (2014)
21. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
22. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: Proceedings of the 2017 ACM on Multimedia Conference (2017)
23. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. arXiv preprint arXiv:1806.02964 (2018)

24. Ma, S., Sigal, L., Sclaroff, S.: Learning activity progression in lstms for activity detection and early detection. In: CVPR (2016)
25. Nguyen, P., Liu, T., Prasad, G., Han, B.: Weakly supervised action localization by sparse temporal pooling network. In: CVPR (2018)
26. Paul, S., Roy, S., Roy-Chowdhury, A.K.: W-talc: Weakly-supervised temporal activity localization and classification. arXiv preprint arXiv:1807.10418 (2018)
27. Sevilla-Lara, L., Liao, Y., Guney, F., Jampani, V., Geiger, A., Black, M.J.: On the integration of optical flow and action recognition. arXiv preprint arXiv:1712.08416 (2017)
28. Seybold, B., Ross, D., Deng, J., Sukthankar, R., Vijayanarasimhan, S., Chao, Y.W.: Rethinking the faster r-cnn architecture for temporal action localization (2018)
29. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: CVPR (2017)
30. Shou, Z., Gao, H., Zhang, L., Miyazawa, K., Chang, S.F.: Autoloc: Weaklysupervised temporal action localization in untrimmed videos. In: ECCV (2018)
31. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: CVPR (2016)
32. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems (2014)
33. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: ICCV (2017)
34. Sun, C., Shetty, S., Sukthankar, R., Nevatia, R.: Temporal localization of fine-grained actions in videos by domain transfer from web images. In: Proceedings of the 23rd ACM international conference on Multimedia (2015)
35. Sun, S., Kuang, Z., Sheng, L., Ouyang, W., Zhang, W.: Optical flow guided feature: a fast and robust motion representation for video action recognition. In: CVPR (2018)
36. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: ICML (2013)
37. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. vol. 4, p. 12 (2017)
38. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: CVPR (2017)
39. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV (2016)
40. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
41. Xu, H., Das, A., Saenko, K.: R-c3d: region convolutional 3d network for temporal activity detection. In: ICCV (2017)
42. Yuan, J., Ni, B., Yang, X., Kassim, A.A.: Temporal action localization with pyramid of score distribution features. In: CVPR (2016)
43. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. ICCV, Oct (2017)