Multi-modal fusion network based on relation-aware pyramid network for temporal action localization

Jialin Gao[1,2], Tianwei Lin[1], Xiang Long[1], Dongliang He[1], Fu Li[1], Xin Li[1], Shilei Wen[1], Errui Ding[1]
[1] VIS, Baidu Inc. [2] Shanghai Jiao Tong University

Abstract

This technical report aims to illustrate the overview of our solutions for the two tasks in HACS Temporal Action Localization Challenge: Weakly-supervised and Supervised Learning Track. Anchor-based and boundary-based approaches are two main categories in this field. Different modality, such RGB images and optical flow images, also is prone to classify different action class. Inspired by these, we propose a multi-modal fusion network to not only explore the complementary characteristics between the two types of advanced approaches, but also exploit the rich information in videos, including audio, RGB images and optical flow. First, RapNet and BMN are used to generate anchor-based and boundary-based action proposal separately. Then, we fuse them for retrieving and finally combine the proposals with video-level classification predicted by our classification network. All methods adopted in our solution are implemented using PaddlePaddle.

1. Introduction

Temporal action localization has become one of the most challenging and promising tasks in video analytic and understanding. It is required to predict accurate start and end time stamps of different human actions. Similar to the advanced solutions in object detection, approaches in this task also observes the popular two-stage pipeline, which divide the problem into proposal generation and multi-label video classification. Due to the latter could be performed well with convincing classification accuracy in action recognition, many recent works, including ours, focus on the former in order to generate temporal action proposals with highly precise boundaries.

The methods in action proposal generation could be divided into two categories: anchor based and boundaries based. The former [3, 7] usually defines or clusters the manually defined number of anchors and employs pyramid-like neural networks for proposal generation in cope with the various duration of actions. Boundary-based methods [12, 8] produce candidates with high precision boundaries by evaluating starting and ending probability on each temporal location or directly evaluating confidence score on densely distributed proposals [6].

In this report, we intend to take the advantage of these two kind of methods. Since RapNet [3] and BMN [6] serve as the advanced approaches in their respective categories, we use them to generate accurate temporal action proposal for fusion. In addition, we also fully exploit the complementary characteristics between RGB images and optical flow images. In the following, we will introduce the fused temporal action proposal generation method and action classification method separately.

2. Action Proposal Generation

In this section, we will simply introduce RapNet and BMN for better understanding. Please refer to the papers for more details if interested.

(1) RapNet

In order to explore the ignored global contextual information in previous methods, the RapNet proposes a relation-aware module to exploit the bi-directional long-range relations between local features and then integrates it into feature pyramid network for multi-granularity temporal proposal generation. We modify the original architecture by adding residual connection and removing the boundary adjustment and ranking part. Its details are illustrated in Fig.1.

(2) BMN

In order to capture rich context for confidence score evaluation, BMN introduce the boundary-matching mechanism based on the densely distributed proposals. we use it to augment the proposal generated by RapNet in the fusion processing. To further exploit ActivityNet train + validation subsets (removing video included in HACS testing set) as unlabeled data to train BMN in semi-supervised fashion.

After generating action proposals by RapNet and BMN, we merge them together for post-processing with soft-NMS. In this challenge, we extract three modalities features with TSM [5] on RGB images, optical flow and VGG on audio.

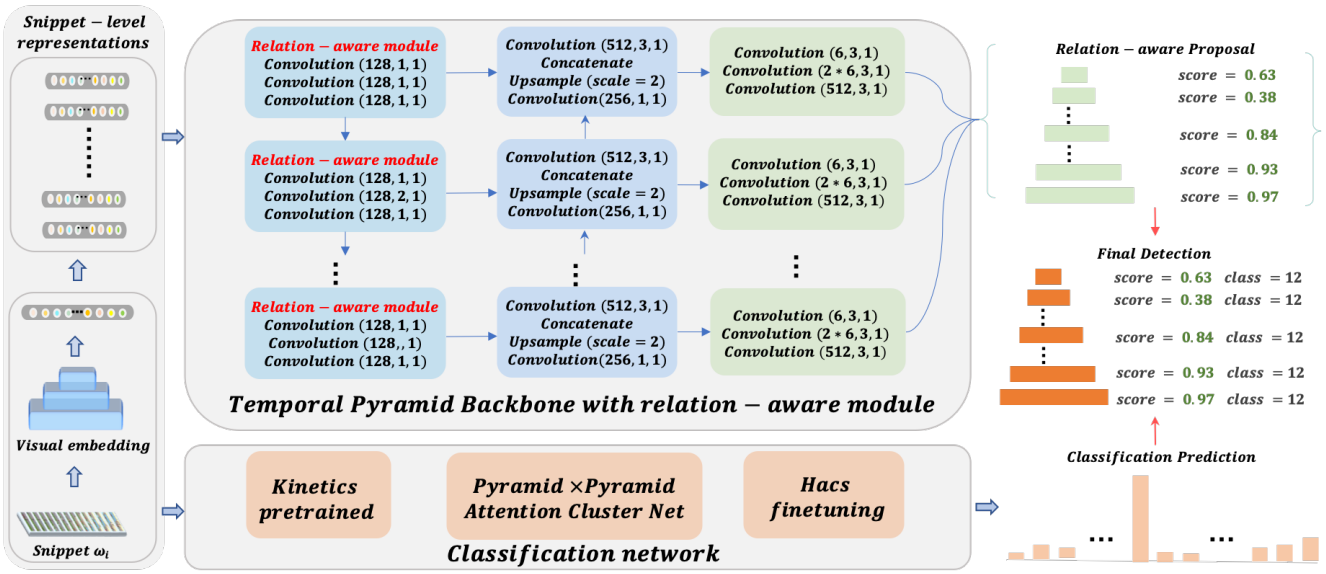

Figure 1. The framework of our approach consists of two components according to the conventional two-stage pipeline: proposal generation and video-level classification. First, a visual embedding network [5] is used for extracting snippet-level video representations. Then the following temporal pyramid backbone integrated with relation-aware modules [3] aims to separately generate candidate instances with different duration via clustered anchor. Finally, our relation-aware proposals are combined with classification predictions to perform temporal action localization.

3. Action Classification

The existing state-of-the-art action classification methods, such as Attention Cluster [10], have leveraged attention mechanism to generate final descriptor from feature sequence of a video. Though effectiveness have been achieved, the attention schema of these models can be further improved from the following two aspects: 1) The attention granularity over different channels could be better designed. 2) The existing attention-based local feature integration solutions have ignored temporal order which is critical for action recognition under some circumstance. To alleviate the aforementioned shortcomings, we propose our Pyramid×Pyramid Attention Network, which combines both channel pyramid and temporal pyramid.

3.1. Pyramid×Pyramid Attention Cluster

For each video, first we extract L segment features, which are arranged in a temporal order, forming a segment feature sequence:

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_L)\,. \tag{1}$$

3.1.1 Channel Pyramid Attention Cluster

We use shift attention (SAtt) as attention unit following Attention Cluster [10]. The shift attention unit essentially calculates a weighted average of local features to obtain the global feature.

$$\mathbf{y} = \sum_{k=1}^{L} f(\mathbf{x}_k, \mathbf{X}) \cdot \mathbf{x}_k, \tag{2}$$

where $f(\cdot,\cdot)$ is the weighting function defined in Attention Cluster [10].

The Channel Pyramid Attention Cluster (CPAC) has a total of N levels. For the n-th level, we split each local feature into 2^{n-1} sub-features, such that:

$$\mathbf{x} = \left[\mathbf{x}^{(n)1}, \mathbf{x}^{(n)2}, ..., \mathbf{x}^{(n)2^{n-1}}\right], \tag{3}$$

where [] is concatenate, $\mathbf{x}^{(n)i}$ is the sub-feature. Then we can construct 2^{n-1} sub-feature sequence for the n-th level:

$$\mathbf{X}^{(n)i} = \left(\mathbf{x}_1^{(n)i}, \mathbf{x}_2^{(n)i}, ..., \mathbf{x}_L^{(n)i}\right), \tag{4}$$

where $i \in 1, 2, 3, ..., 2^{n-1}$.

Next, we apply shift attention to each sub-feature sequence and concatenate their outputs together as the output of n-th level:

$$\mathbf{y}^{(n)} = \left[\mathrm{SAtt}(X^{(n)1}), ..., \mathrm{SAtt}(X^{(n)2^{n-1}})\right]. \tag{5}$$

Finally, outputs of each level is concatenated as the final output of the channel pyramid attention after ℓ_2-normalization.

$$\mathbf{y} = \left[\frac{\mathbf{y}^{(1)}}{\|\mathbf{y}^{(1)}\|_2}, \frac{\mathbf{y}^{(2)}}{\|\mathbf{y}^{(2)}\|_2}, ..., \frac{\mathbf{y}^{(N)}}{\|\mathbf{y}^{(N)}\|_2}\right]. \tag{6}$$

Figure 2 shows an efficient implementation of a 3 level CPAC.

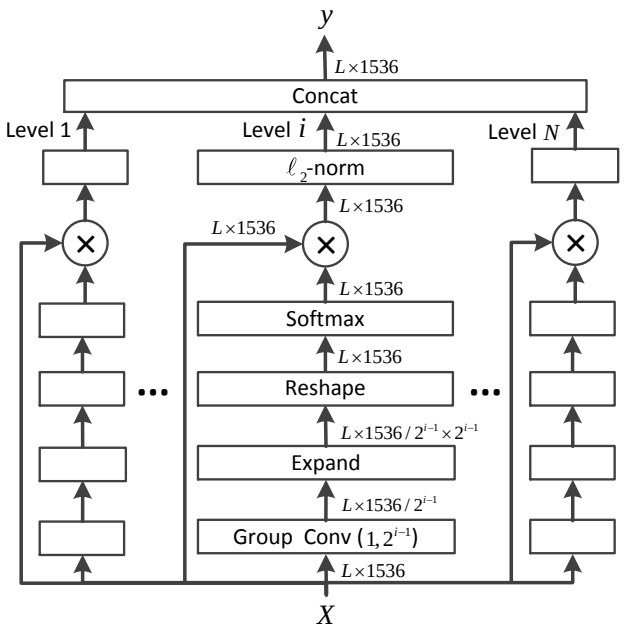

Figure 2. Implementation of Channel Pyramid Attention Clusters: The calculation of attention weights can be efficiently done with 1D group convolution followed by expand, reshape and softmax. In this figure, for illustration purpose, we use 1536 as an example channel size of local features.

3.1.2 Temporal Pyramid Attention Cluster

Temporal Pyramid Attention Cluster has M levels in total. For the m-th level, we split the feature sequence $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_L)$ into 2^{m-1} sub-sequences in temporal order:

$$\mathbf{X}_{(m)j} = \left(\mathbf{x}_{(j-1)L/2^{m-1}+1}, ..., \mathbf{x}_{jL/2^{m-1}}\right), \quad (7)$$

where $j \in 1, 2, 3, ..., 2^{m-1}$.

CPACs are then applied to each sub-sequence separately, and the outputs are concatenated as output of the m-th level:

$$\mathbf{y}_{(m)} = [\mathrm{CPAC}(X_{(m)1}), ..., \mathrm{CPAC}(X_{(m)2^{m-1}})]. \quad (8)$$

Finally, we concatenate the outputs of all levels as the final output of temporal pyramid attention:

$$\mathbf{y} = \left[\mathbf{y}_{(1)}, \mathbf{y}_{(2)}, ..., \mathbf{y}_{(M)}\right]. \quad (9)$$

Since we combine two kinds of pyramid attention on two orthogonal dimension, channel and temporal, we call it Pyramid×Pyramid Attention Cluster (PPAC).

3.1.3 Overall Architecture

For action classification, we extract multiple feature sequences from three modalities, including RGB, flow, audio. We apply different Pyramid×Pyramid Attention to each feature sequences separately. Then the outputs are concatenated together as the multi-modal fused feature of the video. Finally, a fully-connected layer is applied for classification.

Model (%)	Hit@1	Hit@3	mAP
Video-Level LSTM	93.9	99.2	97.1
Video-Level GRU	93.9	99.1	96.9
Segment-Level PPAC	94.2	99.2	97.2
Video-Level PPAC	**94.6**	**99.3**	**97.5**
Ensemble	**95.5**	**99.5**	**97.9**

Table 1. Action Classification Results on HACS validation split.

3.2. Experimental Results

Different kinds of segment-level features are extracted using different backbone methods, *i.e.*, TSN [11], I3D [2] and TSM [5]. Backbone models are first pretrained on Kinetics[2] and then fine tuned on HACS [13]. We set the number of segments L to 32, channel pyramid level N to 4, temporal pyramid level M to 3.

We combine ActivityNet [4] and HACS [13] dataset for training. We merge the training and validation set of Activitynet and the training set of HACS, and then remove the video included in HACS testing set to construct our final training set.

We train two types of action classification models, video level and segment level. At the video level, the whole video sequence is used as input for training. During prediction, we random sample 10 video sequences evenly for each video, and the result is averaged to get video level action classification result. For training segment level model, we use the annotated segments of each action as inputs in training. During prediction, top 10 predicted proposals are used as inputs to predict segment level results, and then segment level results are averaged to get the video level result.

We also train several other video-level sequence models including LSTM and GRU [9]. The final action classification result is the ensemble of all sequence models.The results are summarized in Table 1.

4. Post-Processing

First, we train several RapNet models based on audio, optical flow and RGB images. Then, we use the results from BMN models to decay the confidence score. Finally, we apply the soft-NMS [1] algorithm to suppress the abundant proposals. The Hyper-parameter alpha and threshold are set as 0.3 and 0.5, respectively.

5. Experiments

5.1. Feature extraction

We adopt the TSM [5] in advance to encode the visual content of an input video, where the RGB images are used

Modality	AUC	mAP (val)
Audio	40.65	15.89
Optical flow	61.87	34.01
TSM-K700	62.63	33.88
TSM-FT201	63.00	35.15
All features	64.79	36.86
Ensemble	–	38.65

Table 2. The action detection performance of RapNet on different feature modalities, shown in percentage.

to capture the appearance features and the optical flow images are employed to extract the motion features. In addition, we utilize the VGG to embed the audio information. In order to obtain compact features, we compose snippets sequence $\Omega = \{\omega_i\}_{i=1}^{T'}$ of a given video, where each snippet ω_i with L frames and T' is the number of snippets. we resize the length of each feature sequence from one video to a fixed size (T=256) by linear interpolation before feeding it into the RapNet.

5.2. Action localization

In this section, we will show the temporal action detection results of our RapNet on single modality, such as the first four rows in Tab.2. Then, we concatenate all the modality (denoted as "All features") as input to feed it into the RapNet and achieve the 36.86 map in validation. In our final results, after fusion RapNet and BMN, we achieve 39.20 at validation set, and achieve 39.33 at testing set.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5561–5569, 2017. 3

[2] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *arXiv preprint arXiv:1705.07750*, 2017. 3

[3] Jialin Gao, Zhixiang Shi, Yufeng Yuan, Shiming Ge, and Xi Zhou. Accurate temporal action proposal generation with relation-aware pyramid network. 1, 2

[4] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 3

[5] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 1, 2, 3

[6] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019. 1

[7] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 988–996. ACM, 2017. 1

[8] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1

[9] Xiang Long, Chuang Gan, Gerard de Melo, Xiao Liu, Yandong Li, Fu Li, and Shilei Wen. Multimodal keyless attention fusion for video classification. In *AAAI*, 2018. 3

[10] Xiang Long, Chuang Gan, Gerard de Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*, 2018. 2

[11] 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. In *ECCV*, 2016. 3

[12] 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. 1

[13] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *ICCV*, 2019. 3