TEMPORAL ACTION LOCALIZATION WITH TWO-STREAM SEGMENT-BASED RNN

Tianwei Lin, Xu Zhao, Zhaoxuan Fan*

Key Laboratory of System Control and Information Processing MOE
Department of Automation, Shanghai Jiao Tong University

ABSTRACT

Temporal Action localization is a more challenging vision task than action recognition because videos to be analyzed are usually untrimmed and contain multiple action instances. In this paper, we investigate the potential of recurrent neural network, toward three critical aspects for solving this problem, namely, high-performance **feature**, high-quality **temporal segments** and effective recurrent neural network **architecture**. First of all, we introduce the two-stream (spatial and temporal) network for feature extraction. Then, we propose a novel temporal selective search method to generate temporal segments with variable lengths. Finally, we design a two-branch LSTM architecture for category prediction and confidence score computation. Our proposed approach to action localization, along with the key components, say, segments generation and classification architecture, are evaluated on the THUMOS'14 dataset and achieve promising performance by comparing with other state-of-the-art methods.

Index Terms— Temporal action localization, Two-stream ConvNet, RNN, LSTM, Temporal segment

1. INTRODUCTION

With the continuous booming number of videos on internet, automatic video content analysis is widely required and draws great research interest from both academic and industry fields. An important direction of video content analysis is action recognition, which aims to label manually trimmed short videos. There are many works focusing on action recognition with various datasets [1, 2, 3]. However, videos from real scenarios are often long, untrimmed and contain multiple action instances with very low time proportion. This problem motivates a more challenging vision task: temporal action localization, which aims to localize action instances in long untrimmed videos and classify their categories. It can be used in many areas such as surveillance and home care.

The action detection task of THUMOS Challenge [4] is developed for temporal action localization. This dataset provides a large number of untrimmed videos with temporal annotations. Most approaches [5, 6, 7] report their results in THUMOS using improved Dense Trajectory (iDT) feature with Fisher Vector coding [8]. There is also method which uses 3D ConvNets to capture motion characteristics in video [9]. Recently, two-stream architecture has shown better performance in action recognition task over iDT and 3D ConvNets [2, 3, 10]. So we adopt two-stream networks for features extraction in our approach.

How to generate temporal segments from video sequence is another difficult problem for action localization. Sliding window is a widely used method [7, 9]. However, it's difficult to choose the scale for sliding window since the duration of action instances are varied vastly. We propose a temporal segments generation method called temporal selective search, which is inspired by selective search [11] used in RCNN [12]. We combine temporal selective search and multi-scale sliding window to generate temporal segments with variable length.

Since the two-stream network only explores the temporal consistency in short period, we combine two-stream features with the Recurrent Neural Network (RNN) to explore the temporal consistency in long sequence. There are also some approaches using RNN for temporal action localization [6, 13, 14, 15]. In our approach, we adopt bi-directional long short-term memory (LSTM) [16, 17] units. We propose a two-branch framework including a classification RNN network and a confidence RNN network. These two networks are trained separately, and their outputs are combined to form the final prediction results.

Combining contents discussed above, we propose the Two-Stream Segment-based Recurrent Neural Network (TSS-RNN) framework for temporal action localization. The main contributions of our work are summarized as follows.

(1) An effective two-branch bi-directional LSTM framework for temporal action localization is proposed. Although LSTM have been widely used before in this area, we are the first to predict category and confidence using separate LSTM networks. It achieves better performance than one-branch classification network in evaluation.

(2) A new method combining multi-scale sliding window and temporal selective search is proposed, which can improve the quality of temporal segments.

(3) Our proposed approach reaches start-of-the-art results on the large-scale THUMOS' 2014 dataset. When the overlap threshold is set as 0.3, the mAP is improved from 36.3 to 36.9.

This research has been supported by the funding from NSFC (61673269, 61375019, 61273285). * Corresponding author

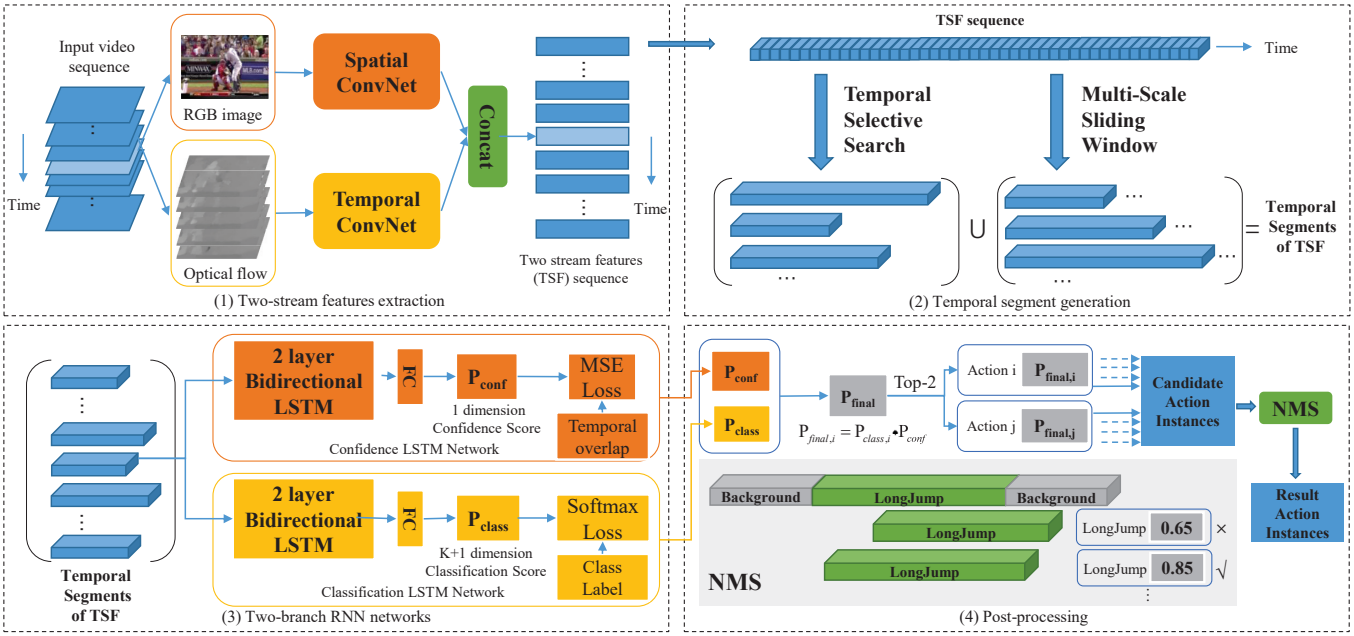

Fig. 1: Framework of our approach. (1) Two stream features extraction: given an untrimmed video, we use two stream network to extract features. (2) Temporal segments generation: we combine temporal selective search and multi-scale sliding window to generate candidate temporal segments. (3) Two-branch RNN network: we adopt two-branch RNN architecture for classification where each branch is a bi-directional LSTM network. (4) Post-processing: during prediction, we choose top-2 result of each segment to form the candidate action instances and remove redundance by NMS to obtain the final action instances.

2. OUR APPROACH

The framework of our approach is shown in Fig. 1. In this section, we give detailed descriptions of our TSS-RNN framework including two-stream features extraction, segments generating, LSTM network architecture, training procedure and post-processing method.

2.1. Two-Stream Network for Features Extraction

Feature extraction is built upon the two-stream networks proposed in [2], including a spatial stream ConvNet and a temporal stream ConvNet. The spatial stream ConvNet operates on still video frame and the temporal stream ConvNet takes optical flow stacking of 10 frames as input. We compute optical flow using GPU implementation of [18] from the OpenCV toolbox. We concat two-stream networks' fc8 layer outputs to create a joint feature with twice the length of total amount of categories. This feature extraction process is shown in Fig. 1.1. We call the two-stream network feature as TSF for short.

2.2. Temporal Segments Generating Method

Inspired by the selective search [11] method used in RCNN [12], we propose the temporal selective search method which can generate segments with variable length and match the ground-truth action instances better. Our temporal selective search method takes TSF sequence as input. First we divide TSF sequence into 5 frames length mini-segments and calculate average TSF within mini-segments. Then we merge two adjacent mini-segments with minimum Manhattan distance of TSF, and their TSF are averaged. Merging is continued until there is only one segment in the sequence. Finally, we output all segments existed in the merging procedure with length between the lower and upper threshold. The lower and upper length threshold is set to 30 and 300 frames respectively in our experiments.

For each untrimmed video, we combine temporal selective search and multi-scale sliding window to generate temporal segments: $S_{full} = \{s_n : (TSF_{t_{start}}, ..., TSF_{t_{end}})\}_{n=1}^{N}$ as shown in Fig. 1.2. We conduct sliding windows with variable lengths of 30, 60, 90, 180, 300 frames with 90% overlaps.

2.3. LSTM Network Architecture

We use a common RNN type, the Long Short-Term Memory (LSTM) network with peephole implementation [19], to generate categories and confidence score of candidate segments. We use TSF as the input of networks.

To exploit both past and future temporal consistency in segments, we use bi-directional architecture in our model. The forward and backward LSTM network have the same architecture with a two layers LSTM network with Peephole

implementation. The number of hidden states in each layer is 256. We use normal distributed random numbers to initialize the initial states of LSTM network. Fully connected layer follows with the LSTM network and projects features from 256 dimensions to target dimension D. D equals to 1 in the confidence network and K+1 in the classification network, where K is number of action categories. The outputs of the final fully connected layer of forward and backward networks are averaged to form the output of networks.

2.4. Training Procedure

Two Stream Network. We train the two stream networks using the same strategy described in [20], which adopts VGG-16 architecture. Each input RGB frame for the spatial network is resized to $224 \cdot 224 \cdot 3$, where 3 is the number of image channels. The temporal network takes the input of stacking optical flow with shape $224 \cdot 224 \cdot 20$, where 20 is ten stacking optical flow images with horizontal and vertical channels. We train the two stream networks on trimmed videos from training set.

The Confidence Network. We train a bi-directional LSTM network for the confidence score regression. The higher confidence score means that this segment has higher intersection over union (IoU) overlap with ground truth action instances.

We use the following strategy to construct training data $S_{conf} = \{(s_n, u_n)\}_{n=1}^{N}$, where $u_n \in [0, 1]$ is the highest IoU between the candidate segment and all ground-truth action instances. To make our model more distinguishable, we only take candidate segments with u_n larger than 0.6 or smaller than 0.2 for training. We assign a segment with $u_n \geq 0.6$ a positive label and $u_n \leq 0.2$ a negative label which indicates the background. We randomly sample the background segments to make sure both positive and background segments have similar proportion in training set.

To train the confidence network, we combine the mean square error (MSE) loss and L2 regularization loss to form the loss function:

$$L_{conf} = \frac{1}{N} \sum_{1}^{N} (y_{pred} - y_u)^2 + \lambda \cdot L_2(\Theta_{conf}) \qquad (1)$$

where y_{pred} and y_u are prediction result and ground truth u_n of a segment repectively. λ balances the MSE loss and l_2 regularization loss, and Θ_{conf} is the confidence network. Through empirical validation, we find $\lambda = 10^{-5}$ works well. As for parameters used in SGD, the learning rate is 10^{-4} and decayed to 10^{-5} after 30 epochs. The batch size is set to 256.

The Classification Network. Then we train a classification model for the K+1 action categories including the background category.

We use similar strategy to prepare the training data $S_{class} = \{(s_n, k_n)\}_{n=1}^{N}$, where $k_n \in \{0, 1, ..., K\}$ and 0 means background here. We reduce the amount of background segments by randomly sampling to keep the background category having roughly similar amount as every other K action categories. And we combine the softmax loss and l_2 regularization loss to form the loss function of classification network:

$$L_{class} = L_{softmax} + \lambda \cdot L_2(\Theta_{class}). \qquad (2)$$

We set $\lambda = 5 \times 10^{-5}$ through empirical validation. For parameters in SGD, we take the same set as confidence network.

2.5. Prediction and Post-Processing

During prediction, we use the same method used in training to extract TSF and generate candidate segments. For each segment, we implement both confidence and classification network on it and get classification score P_{class} and confidence score P_{conf}. Then we multiply P_{class} with P_{conf} and get the final score of segment $P_{final} = P_{class} \cdot P_{conf}$. Furthermore, we choose top 2 categories of P_{final} to represent top two action categories most likely to occur. Every segment's top-2 categories and corresponding P_{final} form the candidate action instances. In this paper, we keep segments with $P_{conf} \geq 0.5$. Finally, we implement non-maximum suppression (NMS) on candidate action instances to remove redundant detections and get the final action instances. We set the overlap threshold as 0.01 in NMS, so almost all overlapping segments are removed.

3. EXPERIMENT

3.1. Dataset and Experimental Setup

For temporal action detection task of THUMOS Challenge 2014 [4], only 20 action categories (as shown in Fig. 2) are involved and temporally annotated. The training set is the UCF-101 [21] dataset. The validation and test sets contain 1010 and 1574 untrimmed videos with temporal annotations of 3007 and 3358 instances respectively. We exclude background videos in validation and test sets.

The evaluation metrics is based on mean average precision (mAP) averaged over all categories. A result instance is masked as correct if it gets the same category label with ground truth instance and its highest IoU is larger than the overlap threshold θ.

Parameters used in each part of approach have been given before. We train the two stream networks using caffe [22]. And we implement two-branch LSTM networks using TensorFlow [23].

3.2. Evaluation on Two-Branch LSTM Networks

To evaluate the performance of each LSTM branch network, we compare the two-branch TSS-RNN with one-branch TSS-RNN. Because the confidence network can't be tested alone

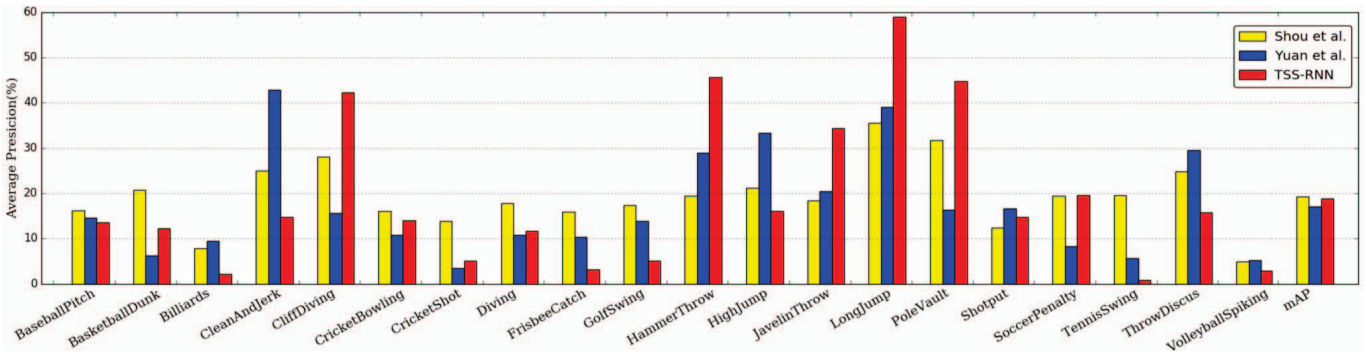

Fig. 2: Detection AP over different action classes with overlap=0.5

Table 1: Comparisons between two-branch TSS-RNN and one-branch classification TSS-RNN on THUMOS'14.

Networks	mAP ($\theta = 0.5$)
TSS-RNN(w/o confidence network)	9.8
TSS-RNN	**18.8**

without the prediction from classification network, we only test one-branch classification TSS-RNN. As shown in Table 1, TSS-RNN with two-branch networks has significantly better performance than TSS-RNN model with one-branch network.

3.3. Evaluation on Temporal Segments Generation Method

To check the effects of different components of our temporal segments generation method, we evaluate multi-scale sliding window, temporal selective search and their combination respectively during testing. We use the strictest threshold 0.5 during evaluation. As shown in Table 2, multi-scale sliding window shows better performance than temporal selective search. The best result can be obtained when combining these two methods. These results indicate that temporal selective search we proposed works well as the supplement of multi-scale sliding window and improves the quality of candidate temporal segments.

Table 2: Comparisons between different components of our temporal segment generation method on THUMOS'14.

Method	mAP ($\theta = 0.5$)
Temporal Selective Search	14.3
Multi-Scale Sliding Window	17.8
Combined two methods	**18.8**

3.4. Comparison with the State-of-the-Art Methods

Our approach is also compared with some state-of-the-art methods [5, 6, 7, 9, 13]. In [5, 6, 7], improved Dense Trajectory (iDT) with Fisher Vector coding [8] is used. Yeung et al. [13] introduce a fully end-to-end approach for action detection in videos, using recurrent neural network-based agent and reinforce learning strategy. Shou et al. [9] introduce a multi-stage segment-based 3D ConvNets, including proposal, classification and localization network, where sliding window method is used to generate temporal segments. Comparison results are shown in Table 3. As can be seen, our approach has similar performance with other state-of-the-art method, and we improve mAP from 36.3 to 36.9 when $\theta = 0.3$. On categories level, we compare our approach with [9] and [13]. As shown in Fig 2, our TSS-RNN approach outperforms other state-of-the-art systems for 6 out of 20 action categories.

Table 3: mAP results on THUMOS'14 with variable IoU threshold θ used in evaluation

θ	0.5	0.4	0.3	0.2	0.1
Wang et al. [7]	8.5	12.1	14.6	17.8	19.2
Oneata et al. [5]	15.0	21.8	28.8	36.2	39.8
Yeung et al. [13]	17.1	26.4	36.0	44.0	48.9
Yuan et al. [6]	18.8	26.1	33.6	42.6	**51.4**
Shou et al. [9]	**19.0**	28.7	36.3	**43.5**	47.7
TSS-RNN(Ours)	18.8	**28.9**	**36.9**	42.9	46.1

4. CONCLUSION

We propose an effective two-stream temporal segment-based RNN approach for temporal action localization. In our approach, we develop the temporal selective search method and combine it with multi-scale sliding window to generate segments, which can improve quality of temporal segments. And the two-branch TSS-RNN architecture we proposed shows great improvement than one-branch architecture during evaluation. Our approach achieves state-of-the-art results on the THUMOS'14 dataset. And an important direction for future work is to further improve our method to generate higher quality temporal segments with lower redundance.

5. REFERENCES

[1] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[2] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[3] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," *arXiv preprint arXiv:1604.06573*, 2016.

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

[5] D. Oneata, J. Verbeek, and C. Schmid, "The lear submission at thumos 2014," *ECCV THUMOS Workshop*, 2014.

[6] J. Yuan, B. Ni, X. Yang, and A. A. Kassim, "Temporal action localization with pyramid of score distribution features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3093–3102.

[7] L. Wang, Y. Qiao, and X. Tang, "Action recognition and detection by combining motion and appearance features," *THUMOS14 Action Recognition Challenge*, vol. 1, pp. 2, 2014.

[8] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[9] Z. Shou, D. Wang, and S. Chang, "Action temporal localization in untrimmed videos via multi-stage cnns," *arXiv preprint arXiv:1601.02129*, 2016.

[10] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: towards good practices for deep action recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 20–36.

[11] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[13] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," *arXiv preprint arXiv:1511.06984*, 2015.

[14] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, "A multi-stream bi-directional recurrent neural network for fine-grained action detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1961–1970.

[15] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1942–1950.

[16] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European conference on computer vision*. Springer, 2004, pp. 25–36.

[19] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling.," in *Interspeech*, 2014, pp. 338–342.

[20] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.

[21] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[23] M. Abadi, A. Agarwal, P. Barham, et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.