Temporal Fusion Network for Temporal Action Localization: Submission to ActivityNet Challenge 2020 (Task E)

Zhiwu Qing[1] Xiang Wang[1] Yongpeng Sang[2] Changxin Gao[1]
Shiwei Zhang[3*] Nong Sang[1*]
[1]School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]School of Cyber Science and Engineering, Huazhong University of Science and Technology
[3]DAMO Academy, Alibaba Group
{qzw, u201613707, ypsang, cgao, nsang}@hust.edu.cn
zhangjin.zsw@alibaba-inc.com

Abstract

This technical report analyzes a temporal action localization method we used in the HACS competition which is hosted in Activitynet Challenge 2020. The goal of our task is to locate the start time and end time of the action in the untrimmed video, and predict action category. Firstly, we utilize the video-level feature information to train multiple video-level action classification models. In this way, we can get the category of action in the video. Secondly, we focus on generating high quality temporal proposals. For this purpose, we apply BMN to generate a large number of proposals to obtain high recall rates. We then refine these proposals by employing a cascade structure network called Refine Network, which can predict position offset and new IOU under the supervision of ground truth. To make the proposals more accurate, we use bidirectional LSTM, Non-local and Transformer to capture temporal relationships between local features of each proposal and global features of the video data. Finally, by fusing the results of multiple models, our method obtains 40.55% *on the validation set and* 40.53% *on the test set in terms of mAP, and achieves* ***Rank*** 1 *in this challenge.*

1. Our Approach

Inspired by current state-of-the-art method [9], we decouple the task of temporal action localization into two subtasks, *i.e.*, video classification and proposal generation. First of all, we use Slowfast-101 [6] as a backbone to train a video-level action classification model. We then use the trained backbone to extract features for each video, which allows us to generate a large number of high-quality proposals using BMN [9]. Finally, we adopt the cascade scheme [2] to further fine-tune the proposals.

1.1. Video Classification

To improve the performance of video classification, we try to encode more temporal and spatial information. The Slowfast [6] achieves excellent performance on video classification by decoupling spatial and temporal information in the temporal-spatial space. Specifically, we choose Slowfast101 as our backbone, and each input clip has 32 frames by 15fps. We first pre-train our backbone on Kinetics-600 [3] and then fine-tune it on the HACS dataset [13]. The output of the network is a prediction of 200 categories. Because there is no background class, we only sample frames in segments that involve with action in training. Finally, we add Batch Nuclear-norm [5] to the loss function to improve the generalization capabilities.

1.2. Generation of proposals

Features. The fine-tuned model has a better representation of the HACS dataset. So we use the fine-tuned classification model's backbone to extract features for all videos. To encode more temporal information, we apply a dense sample strategy. Specifically, we sample 32 frames with a 0.5s stride, and resize the short side of the clip to 256 pixels, and input the backbone to extract discriminative features. Then we take the features of stage "Res5" in Slowfast and use global average pooling to get a 2034-dimensional vector as the feature expression for each clip. Considering that every 32 frames represents a video length of about 2.13s, the overlaps between different clips will cause some frames to be repeatedly sampled. However, same frames in different clips will have a different temporal context, so dense sampling can provide a more fine grain feature representation, which can improve the quality of the generated proposals

*Corresponding authors

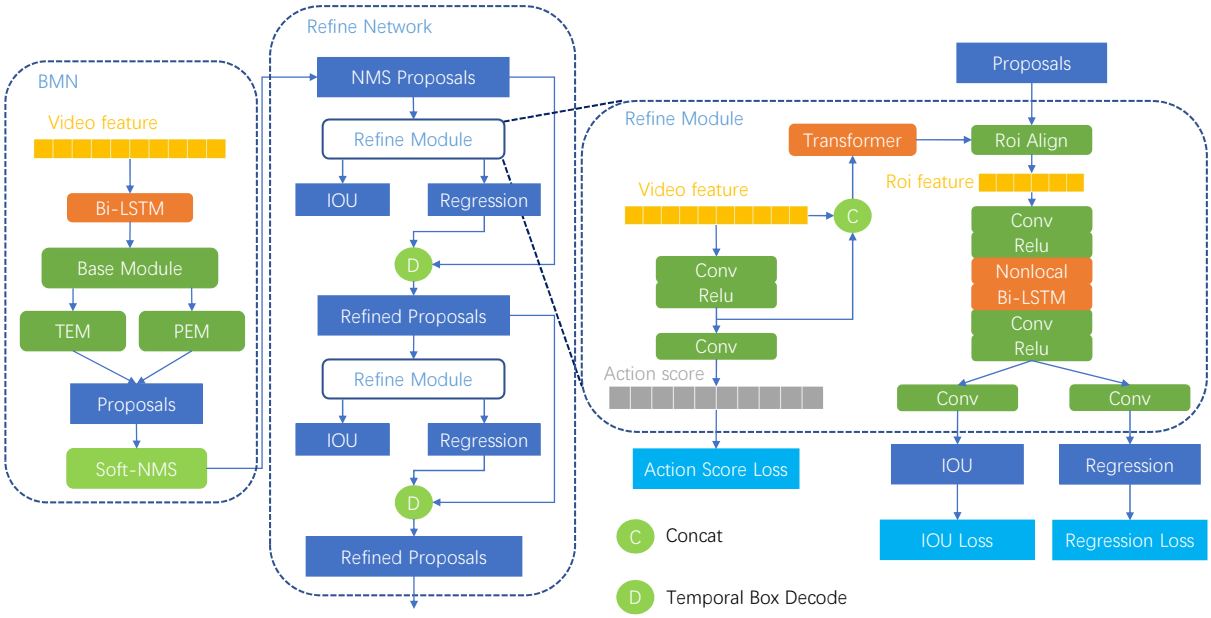

Figure 1. Overview of our proposed framework. We first apply BMN to generate proposals and perform soft-nms on these proposals. Then we input these proposals into the Refine Network to refine these temporal positions and re-predict IoU. Actually, this is a standard cascade architecture. Then we embed a bidirectional LSTM layer in both BMN and the Refine Module to capture the forward and backward temporal information. In the Refine Module, we apply both Nonlocal and Transformer to obtain richer context information. At the same time, by adding the intermediate supervision of Action Score, the obtained features are more discriminative for local proposals.

effectively.

Boundary-Matching Network. In order to generate high-quality temporal proposals, Lin *et.al* [9] proposed the Boundary-Matching Network(BMN). BMN utilizes Temporal Evaluation Module (TEM) to generate temporal proposals. Each proposal generated by TEM can map to BM confidence map which is generated by Proposal Evaluation Module (PEM), and provide confidence for proposals. BMN generates proposals with precise temporal boundaries as well as reliable confidence scores simultaneously. For a video with a duration of t seconds, we can use Slow-fast101 to extract features with a temporal length of $T = round(2t)$. In training and inference process of BMN, we resize the feature temporal length of each video to D. In our experiments we set D to 200. This is because the quality of the proposals is relatively high at a temporal length of 200 in the comparison of different temporal lengths. LSTM can encode sequences very well, and in fact video also belongs to sequence information. We use a bidirectional LSTM to capture forward temporal information and backward temporal information of the video. Our experimental results show that adding bi-directional LSTM to the base module of BMN can also improve performance.

1.3. Fine-tuning of proposals

Refine Network. BMN can generate a good deal of proposals. In the HACS dataset, most videos have long duration, while the temporal length of the above-mentioned features can be only set as a small number, *e.g.*, 200 in our method, limited by the amount of calculation and memory. Obviously, downsampling procedure will lose some temporal information. To solve this problem, we propose to further refine the proposals generated by BMN, which we call Refine Network. We first perform soft-nms [1] on the proposals, and sort them according to the corresponding scores. By this mean, a large number of redundant proposals can be removed, but only some good quality proposals would be fed for further retaining. Then we input these proposals into the Refine Network. The Refine Module employs RoI Align Pooling [7] technology to extract features for each proposal from the enhanced original video features. By applying the features, we can further predict the corresponding IoU and temporal offsets of the proposals. Then we repeat the Refine Modules several times to refine the proposals iteratively. In this way, a cascade [2] architecture is formed. In our method, we embed three Refine Modules in the refine network. As in [2], we set IoU thresholds for assigning positive labels to 0.5, 0.6 and 0.7, respectively.

Refine Module. The refine module aims to further refine the candidate proposals and predict the corresponding IoUs. Firstly, we predict score for each point on video feature whether there is an action, which we call Action Score, Similar to Actionness in [10]. Note that it is a supervised procedure to learn Action Score. By adopting the supervi-

Learning Rate	Remarks	Top1(%)	Top5(%)
0.001		89.71	98.11
0.001	+BNM	90.12	97.89
0.0005		91.19	98.98
0.0001		91.61	99.13
0.0001	+BNM	91.75	99.13
0.0001	+Transformer	91.86	99.06
0.0001	+BNM+Transformer	91.84	98.90
N/A	Ensemble 14 models	**94.32**	**99.68**

Table 1. Part of the results of the classification model. After ensemble, the model Top1 can obtain an absolute improvement to 2.36%, which shows that there is a strong complementarity between different models. At the same time we found that BNM does not always improve performance. But it does not matter, what we need is the complementarity between different models.

Bi-LSTM	Nonlocal	Action Score	Transformer	AUC(%)	mAP(%)
				65.89	38.75
✓				65.75	38.90
✓	✓			65.88	39.24
✓	✓	✓		65.88	39.48
✓	✓	✓	✓	65.90	39.65
Ensemble				**66.08**	**40.55**

Table 2. Part of the results of Refine Network in the validation set. The addition of each module can improve the final mAP. The result of the ensemble of multiple models also shows that using different modules, the preferences of the models are different, and there is complementarity between the models. Our final ensemble result achieves **40.53** in terms of mAP(%) on the test set.

sion information, the learned features can possess discriminative power. Secondly, in order to predict IoU as accurately as possible, we should encode long-term temporal information. It is obvious that local features can not know the actual duration of the action (especially when proposals intercept part of the action). Therefore, we need to extract features which have a global perception of videos. In order to achieve the purpose, before performing RoI Align Pooling [7], we treat the video features as sequences and use Transformer [11] in temporal dimension to obtain a global receptive field. Finally, we also use bidirectional LSTM network [8] and Nonlocal [12] to enhance local features in the RoI features, which allows the network to fully integrate local temporal information.

2. Experimental Results

2.1. Video Classification

We choose Slowfast101 as the backbone of our classification model. In order to make the classification model more feature-rich, we train a variety of classification models based on this backbone. For example, different learning rates, adding Batch Nuclear-norm [5] in training, and adding Transformer [11] to video features. While improving the results of a single model as much as possible, we also need the complementarity of features between different models. So even if the result of the single model is not the best, it can still improve the final classification results when performing ensemble procedure. Finally, we fuse these classification results as our final classification results. When there are N classification models for ensemble, we set an adaptive weight for each model. These N parameters are multiplied by the prediction results of each model before Softmax, and they will automatically converge to the value that makes the classification performance the best. Some of our experimental results are shown in the Table 1.

2.2. BMN

The purpose of BMN [9] is to initially obtain a large number of proposals for the fine network. We also train many different models with different parameters or embedded modules to increase the complementarity between different results for ensemble scheme. For the ensemble of multiple BMN models, we first take the two D-dimensional vectors output by the TEM in each model, which are the action starting and ending probabilities, respectively. Then we take the two $D * D$ Boundary-Matching confidence maps output by PEM. We perform weighted summation on the four maps between all models to obtain the results of model ensemble. When multiple BMN models have different D, we resize the four maps to the same scale by linear interpolation. The experimental results are shown in Table 3.

2.3. Refine Network

All Refine Networks in our experiments apply 3 Refine Module cascades. During the inference, because each Refine Module will change the position of input proposals, so we only take the prediction result of the last layer of Refine Module as our final proposals. We use the batch of proposals obtained after the BMN model ensemble as the input of N different Refine Networks, and we can get N batches of different adjusted proposals. Finally, we use weighted summation to fuse these proposals as our final result. Our final detection result achieves 40.55% on the validation set and 40.53% on the test set in terms of mAP. The experimental results are shown in Table 2.

3. Conclusion

In this technical report, we introduce the method designed for the HACS2020 competition. The experiment results show that the proposals generated by BMN can also be further improved. At the same time, for the fusion of temporal information, simply applying a larger convolution kernel to expand the receptive field does not effectively improve the quality of proposals. The effectiveness of LSTM,

Feature	D	Bi-LSTM	AR@1(%)	AR@5(%)	AR@10(%)	AR@100(%)	AUC(%)
I3D [4]	200	0	18.26	37.94	47.47	70.89	60.99
Slowfast [6]	256	0	19.91	41.04	50.80	73.47	64.10
	200	0	19.87	41.15	50.92	73.10	63.83
	180	0	19.89	41.16	50.99	73.22	63.91
	160	0	19.80	40.98	50.65	72.60	63.49
	200	1	20.31	41.26	50.72	72.31	63.44
	200	2	20.25	41.33	51.08	72.61	63.73
Ensemble 10 models			**20.83**	**42.77**	**52.92**	**74.32**	**65.51**

Table 3. Part of the results of BMN. We found that the results obtained using the Slowfast101 feature are significantly higher than obtained using the I3D feature. The test results of different temporal scales are not particularly different. But with the decrease of D, the performance is obviously reduced. We noticed that the result of adding LSTM has decreased AUC. Nevertheless, AR@1 and AR@5 have significantly improved, which is very helpful for detection. In the ensemble process, we also found that the BMN result with LSTM is very complementary to other results, which can greatly improve the detection performance.

Nonlocal and Transformer shows that high-order information in temporal space is still an important research direction. In future works, we will further to explore for how to better encode temporal information.

References

[1] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[3] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[5] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and Q. Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. *arXiv preprint arXiv:2003.12237*, 2020.

[6] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019.

[10] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[12] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[13] H. Zhao, A. Torralba, L. Torresani, and Z. Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8668–8678, 2019.