Learning Efficient Feature Representation for Temporal Action Localization

Chenglu Wu*, Xuefeng Yang*, Fuzhi Duan, Yanxun Yu, Yayun Wang, Jun Yin
Zhejiang Dahua Technology Co., Ltd.
Hangzhou City, Zhejiang Province, China
duanfz@whu.edu.cn
{wu_chenglu,yang_xuefeng,yu_yanxun}@dahuatech.com

Abstract

*This paper presents an overview of our solution used in the submission to **ICCV DeeperAction Challenge 2021 Track 1 (temporal action localization)**. Temporal action localization requires to precisely locate the temporal boundaries of action instances and accurately classify the action instances into specific categories. The performance of temporal action localization depends on feature extraction, proposal generation and video classification. For the feature extraction, we analyze the impact of different video features on the quality of generated proposals. In order to improve the quality of proposals, we use the attention mechanism, graph convolution, and dilate convolution to deform the receptive field. At the same time, NMS and refining module cascades were applied to TCAnet to further refine the proposal. Finally, we ensemble different classifiers to improve the accuracy of video classification effectively. With these methods, we achieve **Rank 1** in this competition.*

1. Introduction

With the flourish of Streaming Media, the number of videos increases rapidly, which leads to the increasing demand for video understanding. The temporal action localization (TAL) is one of the main branches of video understanding, which aims to precisely locate the temporal boundaries of action instances and accurately classify the action instances into specific categories. Similar to the object detection, TAL can be divided into two categories: one-stage and two-stage[9]. The generation and classification of candidate temporal boundaries are performed simultaneously in the one-stage method, which can be trained end-to-end. This method is simple in the training process, but slightly inferior in accuracy[1, 10]. The two-stage method first generates candidate boundaries and then performs action classification on each proposal, effectively improving the accuracy of TAL, but it will inevitably bring a speed reduction[5, 16, 18]. In this competition, we use a two-stage method to explore the best performance of the FineAction.

Video type	Number	Proportion
one-label	11259	89.26%
multi-label	1355	10.74%

Table 1. **One-label, multi-label frequency statistics.** Frequency statistics of different labels on training and validation set.

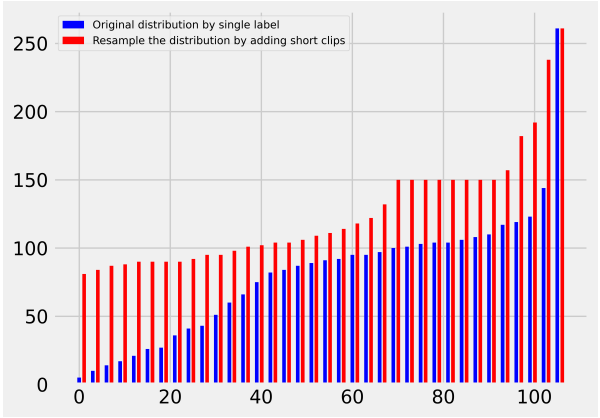

Figure 1. **Original distribution VS. resample distribution.** The blue bar indicates the distribution of original number of different categories. The red bar indicates the distribution of the number of unbalanced video clips after resampling.

2. Dataset

FineAction[11] is a fine-grained dataset, which is composed of 8,440 train videos, 4,174 validation videos and 4,118 test videos. It contains 106 categories, covering Household Activities, Personal Care, Socializing Relaxing and Sports Exercise. We analyzed the number of labels in

* Equal contribution

This work is supported by Jinn Platform of Dahua Technology Co., Ltd.

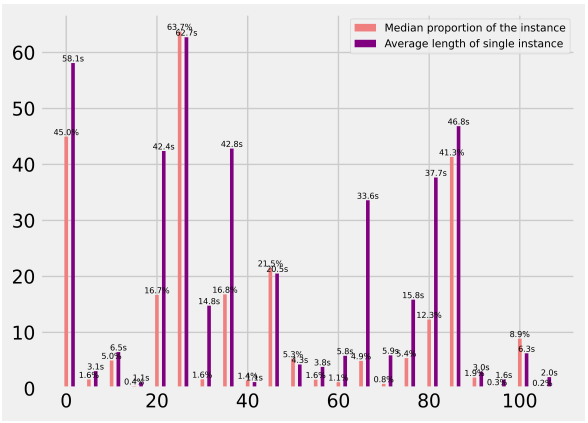

Figure 2. **Median proportion and the average length of instances.** Light coral bar indicates the median percentage of the total video duration for a single instance. The purple bar indicates the average length of a single instance of each category. Each category label is defined as a value from 0 to 105.

the training and validation set and observed that the labels of action instances on the same video are mostly the same, and the label distribution is shown in Table 1. To reduce the difficulty of training the video classification network, we redefined the label of each video as the category of the most frequent occurrence.

After redefining the video labels, we visualized the distribution of 35 categories uniformly selected on the training set, as shown in Figure 1. We can observe a particular imbalance in the number of different categories from the figure. For this reason, we used a resampling strategy to cut out some sparsely labeled instances from the original videos to supplement the training dataset. The adjusted distribution of video labels is shown in the red histogram in Figure 1. In addition, after excluding the dirty annotations in the ground truth which end time earlier than the start time or greater than the total video duration, we analyzed the average length and the median proportion of different instances to total duration ratio under 106 categories of individual videos in the training set. Figure 2 shows the distribution of the 21 categories of actions sampled at random. Light coral bars are the median proportion of instances in the video, and purple bars are the average length of individual instances. Finally, we eliminated the extremely short instances based on the resolution of the instances supported by the grid. It has been shown that the elimination of this gap can improve the AR@1 of the proposal.

3. Method

The structure of our proposed method is shown in Figure 3 and consists of feature extraction of video clips, proposal generation, proposal refinement, and video classification.

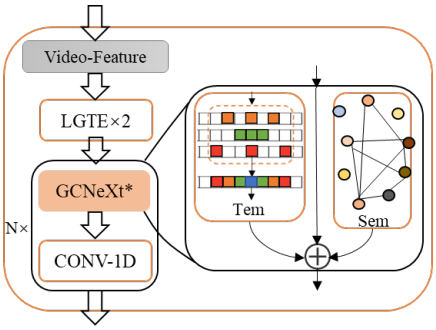

Figure 3. **The improved base-feature layer and the modified GCNeXt layer.** We used the innovative LGTE module in TCAnet[13] to encode local and global temporal relationships simultaneously. We also followed the semantic graph of the GCNeXt module in GTAD[17] and added the dilated convolution to reconstruct the temporal graph to deform the receptive field and enhance the aggregation of context information.

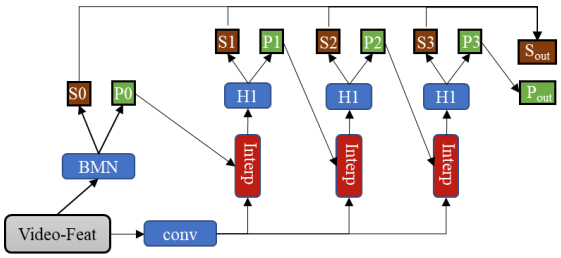

Figure 4. **The architecture of cascade proposals refinement module.** We followed the Cascade RCNN[2], a productive two-stage object detection method, to refine the proposal. "S" is the proposal score, "P" refers to the generated proposal, "Interp" denotes feature sampling based on "P" and "H" denotes the feature processing layer.

3.1. Feature Extraction

The quality of video features can seriously affect the performance of TAL, as our proposed TAL network takes video features as input. First, we used TSN[15] and SlowOnly[6] to extract features from the untrimmed videos and compare them to the official I3D feature. Specifically, we extracted TSN features on three models, including pre-trained on Kinetics-700 dataset[3], pre-trained on Kinetics-600 dataset, and pre-trained on Kinetics-400 dataset fine-tuned on FineAction. The TSN input we used contains only RGB information, as extracting optical flow images is time-consuming. We used the first two models to extract features before the softmax layer. For the fine-tuned model, we used only the features before the fully connected layer with a dimension of 1,024. For the SlowOnly pre-trained on Kinetics-700, the feature dimensions were chosen to be consistent with TSN-K700. In addition, we chose the video swin transformer[12], which currently performs SOTA on

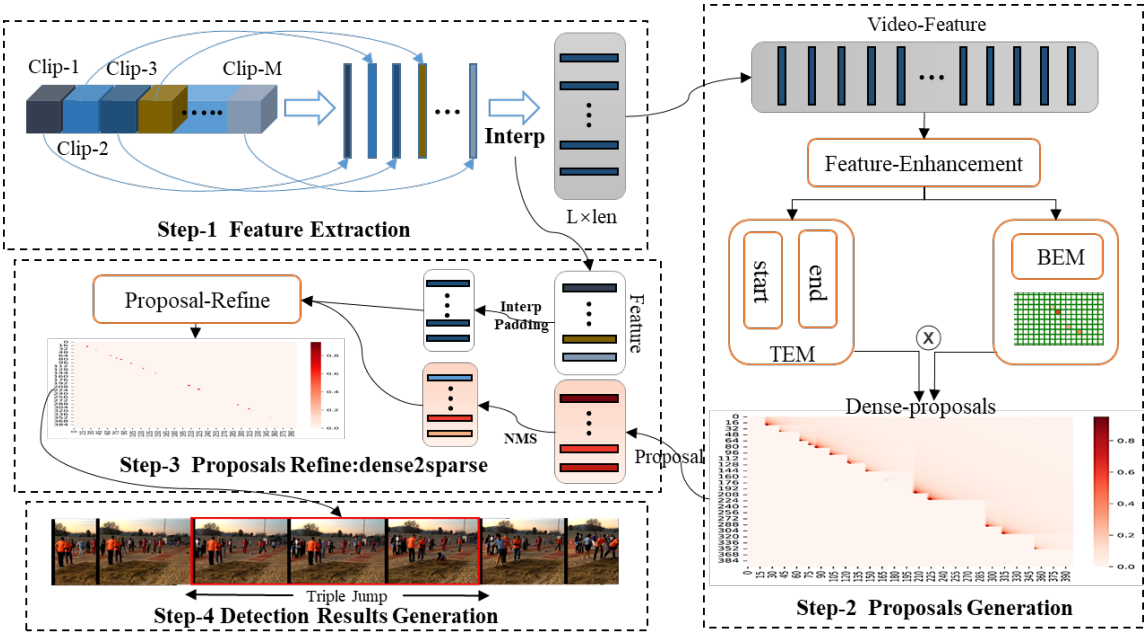

Figure 5. **The architecture diagram of our proposed method.** The whole architecture contains feature extraction, proposal generation, proposal refinement and proposal classification. The L in feature extraction denotes the dense proposal grid size, and the len is the individual temporal clips feature dimension.

the kinetics-600 dataset, as the feature extractor. The features selected were those before the fully connected layer and softmax layer, resulting in a higher mAP of our final submission. Due to the varying length of the videos in the FineAction dataset, the above-extracted clip features are normalized by linear-Interpolation.

3.2. Temporal Proposal Generation

We normalized the length of video features to a fixed size as the input to the temporal proposal network, and this network outputs a set of proposals containing the action scores, as shown in Figure 3. In this competition, we followed the dense proposal generation framework BMN[8]. We set the grid size equal to L to generate $L\times L/2$ potential action clips and supervised training of the BMN to learn the action scores that fall into the corresponding grid. The lengths of the detectable actions depend on the size of the grid, with the shortest detectable length being the duration of per video divided by L. Given the high proportion of short actions in the FineAction dataset, the grid length was fixed at 400×400.

Compared to the original BMN, we have improved the base-feature layer, which directly processes video features. Specifically, we first used the innovative LGTE module in TCAnet[13] as a pre-processing module for video features reconstruction to encode video features' local and global temporal relationships simultaneously. Secondly, we followed the semantic graph of the GCNeXt module in GTAD[17] and added the dilated convolution with dilate=2 and dilate=3 to reconstruct the temporal graph to enhance the aggregation of context information of adjacent frames while minimizing the growth of parameters, as shown in Figure 3.

3.3. Proposal Refinement

The proposal generation network will generate a large number of action proposals to improve recall. However, the grid-scale of BMN is fixed, which limits the shortest resolution at detectable actions, and proposals are inflexible in predicting start and end boundaries. To alleviate it, we were inspired by the ideas of Cascade RCNN[2] and TCAnet, and designed a three-level cascade dense2sparse network to increase the accuracy of proposals. Specifically, the modified BMN was used to generate the training and validation set proposals as the anchors of detection region, which were combined with the extracted features and fed into the dense2sparse network for training. The proposal input to the dense2sparse network sorted by topK in descending order of scores, was reduced the dense-proposal to sparse-proposal. To reduce the difficulty of the network training, the length of the original features is fixed to 1,000. To ensure that the features of each clip is not affected by the interpolated, we followed the TCAnet by interpolating the video features with length greater than 1000. To increase the generalization ability of the dense2sparse network, we added the NMS on the proposals to remove pro-

Method	SlowFast	CSN	TSN	NeXtVLAD	Video-swin-transformer
backbone	Res101+50	Res152	Swin-Base	Swin-Base	Swin3D-Base
head	SlowFastHead	I3DHead	TSNHead	NextVLADHead	I3DHead
clip_len	32	32	1	2	32
frame_interval	2	2	1	1	3
num_clips	1	1	32	32	1

Table 2. **Training details for video classification networks.** The "num_clips" is the number of clips the whole video will be divided into, the "clip_len" denotes the number of frames selected for clip, and the "frame_interval" is defined as the number of frames between each clip.

num_clips	Accuracy
3	86.08
4	86.77
5	87.61
6	87.42
7	87.03
8	**87.90**
9	87.58
10	87.63

Table 3. **The Top-1 accuracy of Video-SwinB at different num_clips.** The Video-SwinB achieved the highest Top-1 accuracy 87.90% when num_clips equals 8 in FineAction.

Model	Top-1	Top-2	Top-3	Top-5
TSN	81.54	91.46	94.19	96.87
SlowFast (nc=8)	84.92	94.11	96.47	98.63
CSN (nc=8)	85.70	94.39	96.96	98.78
Video-SwinB (nc=5)	87.61	94.90	97.41	98.82
Video-SwinB (nc=8)	87.90	95.23	97.84	99.04
NeXtVLAD	87.21	94.87	97.24	98.70
Ensemble	**90.03**	**96.72**	**98.43**	**99.41**

Table 4. **Comparison between different backbones for video classification.** The nc denotes num_clips.We used NeXtVLAD, CSN and Video-SwinB to ensemble the model.

posals with similar scores and boundaries. Inspired by the Cascade RCNN, which suggests that the best performance is achieved by picking samples corresponding to a specific iou for training, we set the training iou threshold to 0.5 in $h1$. We sent the fine-tuned proposals from $h1$ to $h2$, and finally, sent the fine-tuned results from $h2$ to $h3$. By setting different thresholds in different stages and cascading each other, the accuracy of the proposal prediction is gradually boosted, as shown in Figure 3.

3.4. Video Classification

After refining the proposals to predict the start and end boundaries of the action, we need to further cascade the video classifier to output the action detection results. We refer to the Section 2 for the analysis of the ratio of single-video multi-label in the FineAction dataset. We followed the experience from the ActivityNet competition to generate classification results based on the single-label and train the video classification network separately. Specifically, we used TSN, CSN[14], SlowFast, and video-swin-transformer as classifiers, respectively. In TSN, we used only RGB images for training and swin-transformer as the backbone. We added the NeXtVLAD[7] to the classifier network after the fully connected layer as well. The configuration and validation set accuracy of different classifiers can be found in the experiments in Section 4.

4. Experiments

4.1. Action Recognition

The video classification was trained by the mmaction framework[4], and four models were explored as classifiers in the development phase. The specific training details are shown in Table 2, and the NeXtVLAD is a modified scheme of the TSN. All backbone were fine-tuned on the FineAction dataset using the pre-trained weights on the Kinetics.

In Table 2, num_clips, $clip\_len$, and $frame\_interval$ are the configuration parameters for reading untrimmed videos. The num_clips is defined as the number of clips the whole video will be divided into, $clip\_len$ is defined as the number of frames actually selected for each clip, and $frame\_interval$ is defined as the number of frames between each clip. Different configurations have an impact on the accuracy of the model. In video-swin-transformer (SwinB), for example, num_clips is set to 1 during training, and TTA (Test Time Augmentation) is performed by increasing num_clips during testing to improve the test accuracy, which is used to cover the variable duration of untrimmed videos. Table 3 shows the Top-1 accuracy of SwinB under different num_clips in detail.

The video classification accuracy under different models is given in Table 4. The NeXtVLAD achieves an accuracy second only to SwinB and is superior to the original TSN. Finally, we ensembled the CSN, SwinB, NeXtVLAD, and SlowFast models with different num_clips parameters to determine the validation dataset accuracy.

Video feat	L	AR@1	AR@5	AR@10	AR@100	AUC
I3D	100	4.92	10.35	13.38	24.64	19.57
	200	4.87	10.40	13.75	27.78	21.28
TSN-K700	200	5.15	11.44	15.10	29.61	23.07
	250	5.05	10.90	14.41	28.64	22.04
Slowonly-k700	200	5.09	11.19	14.86	29.25	22.67
TSN-full-2048	200	5.42	12.15	16.08	31.29	24.53
SwinB-k600	200	5.80	12.98	16.94	31.73	25.05
	256	5.82	12.90	17.09	32.26	25.32
	325	6.21	14.23	18.75	35.00	27.69
SwinB-full-1024	256	6.07	13.97	18.60	33.96	27.28
	325	6.21	14.44	19.32	36.84	29.25
	400	**6.49**	15.13	20.30	**38.80**	**30.81**
	450	6.34	**15.20**	**20.32**	38.61	30.67

Table 5. **Proposals AUC comparison between different backbones used to extract features.** The L denotes the length of video feature.

BMN	LGTE	GCNeXt	Dilate	TCAnet	NMS	Cascade	Average-mAP(%)	Promotion
✓							17.32	-
✓	✓						17.61	+1.67%
✓	✓	✓					18.21	+3.41%
✓	✓	✓	✓				18.63	+2.31%
✓	✓	✓	✓	✓			19.14	+2.74%
✓	✓	✓	✓	✓	✓		21.19	**+10.71**%
✓	✓	✓	✓	✓	✓	✓	**22.05**	+4.06%

Table 6. **Influence of different modules on the performance of FineAction.** We used the BMN network trained with the standardized video feature length of 400 as the baseline.

4.2. Extracting Features

I3D, TSN, SlowOnly, SwinB were used as feature extractors. 2048-D of I3D features are directly selected from the official features. In Table 5, TSN-K700, SlowOnly-K700 are defined as features extracted before the Softmax using Kinetics-700 pre-training weights. TSN-full-2048 is defined as the model using Kinetics-400 pre-training weights, fine-tuned by the classifier adding the NeXtVLAD. We used the features before the fully connected layer as a representative of video clips. SwinB-K600 is defined as the feature extracted before the Softmax using the Kinetics-600 pre-training weights, and SwinB-1024 is represented as the feature before the fully connected layer after fine-tuning using the Kinetics-600 weights. It should be noted that the frame interval δ of each video is 16 for I3D, TSN-K700 and SlowOnly-K700, 5 for TSN-full-2048, and 8 for SwinB-K600 and SwinB-full-1024.

In the Table 5, we find that the results obtained with the TSN-full features are higher than TSN-K700 and the results of SwinB-full are higher than SwinB-K600 using the same grid scale. The AUC of the proposal at L400 is higher than L325 and L450 from the validation results of different temporal scales. Moreover, the feature extracted by the SwinB model showed a significant improvement compared to the previous features. We believe that the features within the temporal clips significantly impact on the quality of proposal generation. We speculate that the operation mechanism of the SwinB is more accurate in describing short clip features, which is a privilege for the detection of a large number of short instances in the FineAction.

4.3. Proposal Generation

The proposal generation network was trained with the standardized video feature length of 400. At the same time, we employed an AdamW optimizer for 32 epochs using a step decay learning rate scheduler. A batch size of 8, an initial learning rate of 0.001, and a weight decay of 0.0001 were used. In the training of the TCAnet, the input video feature length L was 1000, and the top 128 highest scoring proposals by the NMS of each video were selected as an input. We employed an AdamW optimizer for 8 epochs. An initial learning rate of 0.0004, weight decay of 0.00001, batch size of 16 were used. All experiments were trained and tested on the NVIDIA-A40 GPU.

Firstly, We added the LGTE, GCNeXt, and dilate convolution to enhance the features in BMN and then used the

dense proposals generated by BMN to feed the TCAnet. When training TCAnet, the NMS on dense proposals and the cascade refinement on sparse proposals were used, respectively. The experimental results are shown in Table 6. From Table 6, we can find that the use of additional modules in the baseline both improve performance, with the NMS on the proposals bringing the largest improvement of 10.71%, indicating that a large number of generated proposals is duplicated and that after the NMS, and the TCAnet can learn a broader range of features. Cascading different models also achieved a significant improvement of 4.06% in proposal refinement, which indicates that cascading different refinement can improve the quality of proposals by training on samples of specific iou threshold quality. Finally, our detection result achieves 22.05% on the validation set and 23.35% on the test set in terms of mAP.

5. Conclusion

In this paper, we introduce the method designed for the FineAction competition, including feature extraction, proposal generation, proposal refinement, and video classification. We find that the model using the video clips for action recognition has a greater performance on proposals than the model using single frame. Increasing the grid size of BMN can further improve the recognition accuracy of short actions in the FineAction. However, with the increase of grid size, the total number of parameters of the model will increase exponentially, and the model convergence epoch will move backward as well. Considering the cumbersome nature of this method, we will improve the one-stage TAL framework to locate and identify the extremely short instances in the future.

References

[1] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *Procedings of the British Machine Vision Conference 2017*. British Machine Vision Association, 2019.

[2] Zhaowei Cai and Nuno 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] Joao Carreira and Andrew 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.

[4] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. `https://github.com/open-mmlab/mmaction2`, 2020.

[5] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5793–5802, 2017.

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[7] Rongcheng Lin, Jing Xiao, and Jianping Fan. Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[8] 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/CVF International Conference on Computer Vision*, pages 3889–3898, 2019.

[9] Tianwei Lin, Xu Zhao, and Zheng Shou. Temporal convolution based action proposal: Submission to activitynet 2017. *arXiv preprint arXiv:1707.06750*, 2017.

[10] Chenchen Liu, Yang Jin, Kehan Xu, Guoqiang Gong, and Yadong Mu. Beyond short-term snippet: Video relation detection with spatio-temporal global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10840–10849, 2020.

[11] Yi Liu, Limin Wang, Xiao Ma, Yali Wang, and Yu Qiao. Fineaction: A fine-grained video dataset for temporal action localization. *arXiv preprint arXiv:2105.11107*, 2021.

[12] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[13] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 485–494, 2021.

[14] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019.

[15] 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 *European conference on computer vision*, pages 20–36. Springer, 2016.

[16] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.

[17] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020.

[18] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019.