Technical Report for FineAction 2022 - Temporal Action Localization

Shimin Chen[1] Yijia Duan[1,2] Wei Li[1] Changlong Li[1,2] Chen Chen[1]
[1]OPPO Research Institute. [2]Shanghai Jiao Tong University.
{chenshimin1, liwei19, chenchen}@oppo.com
{yjduan, hurray-long}@sjtu.edu.cn

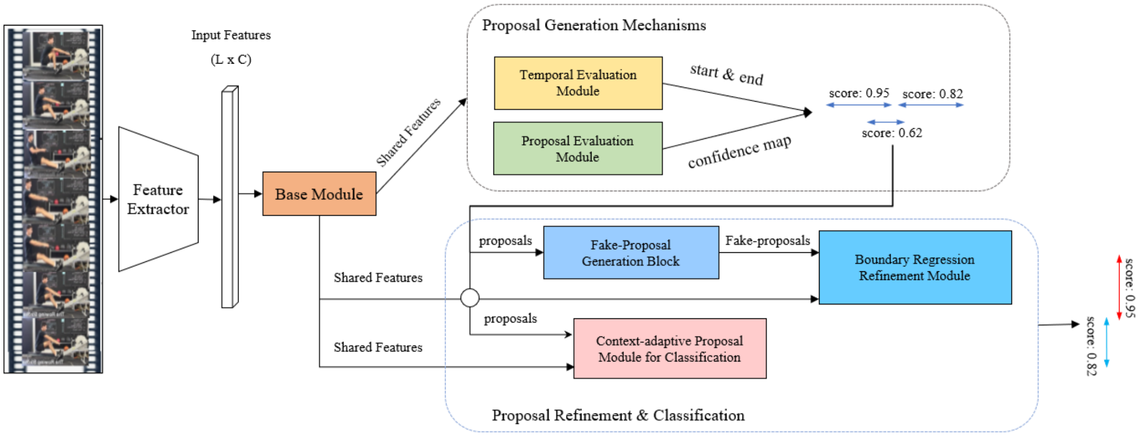
Input Features
(L x C)
Feature Extractor
Base Module
Shared Features
Proposal Generation Mechanisms
Temporal Evaluation Module
Proposal Evaluation Module
start & end
confidence map
score: 0.95
score: 0.82
score: 0.62
proposals
Fake-Proposal Generation Block
Fake-proposals
Boundary Regression Refinement Module
Shared Features
proposals
Context-adaptive Proposal Module for Classification
Shared Features
Proposal Refinement & Classification
score: 0.95
score: 0.82

Figure 1: Overview of our method. Given an untrimmed video, Faster-TAD can generate proposals and simultaneously (1) refine the boundary and (2) classify the proposal in a context-adaptive way. We construct our Faster-TAD with feature sequences extracted from raw video as inputs.

1. Method

In the task of temporal action localization of Fineaction dataset, we propose to locate the temporal boundaries of each action and predict action class in untrimmed videos. We first apply VideoSwinTransformer [1] as feature extractor to extract different features. Then we apply a unified network following Faster-TAD[2] to simultaneously obtain proposals and semantic labels. Last, we ensemble the results of different temporal action detection models which complement each other. Faster-TAD simplifies the pipeline of TAD and gets remarkable performance. Also, to take into account both short-term and long-term temporal instances, we introduce a multi-size sliding window strategy. Besides, We add "negative sample windows" which without action instances to the training set to reduce false positives.

1.1. Feature Engineering

With detailed analysis of Fineaction dataset, we recognize that this dataset includes more fine-grained actions than other public datasets like Kinetics-700, HACS Clips and ActivityNet-1.3, and a lot of action durations are less than 1.2-seconds. Considering this fine-grainedness, we construct two features, auxiliary feature and temporal boundary feature, which respectively focus on the action duration and the beginning and end of the action. For auxiliary feature, we train the model with 2-seconds clips and expand to two seconds from the center point for the actions shorter than 1.2s. Besides, we add three informative public datasets for joint training with fineaction: Kinetics-700, HACS Clips and ActivityNet-1.3 datasets. For temporal boundary features, we just choose the actions longer than 1.2s in order to better distinguish the training data at the start and end. The details are shown in the figure 2. To better make use of the background information of Fineaction, we set the background of each class as a supplementary class in training, and that change the duration number of categories in Fineaction from 106 to 212, and change that of start-end from 106 to 318.

We train two VideoSwinTransformer models with two clips to extract two features with window size=64 and stride=32. The results of each atomic classifier are shown in the Table 1.

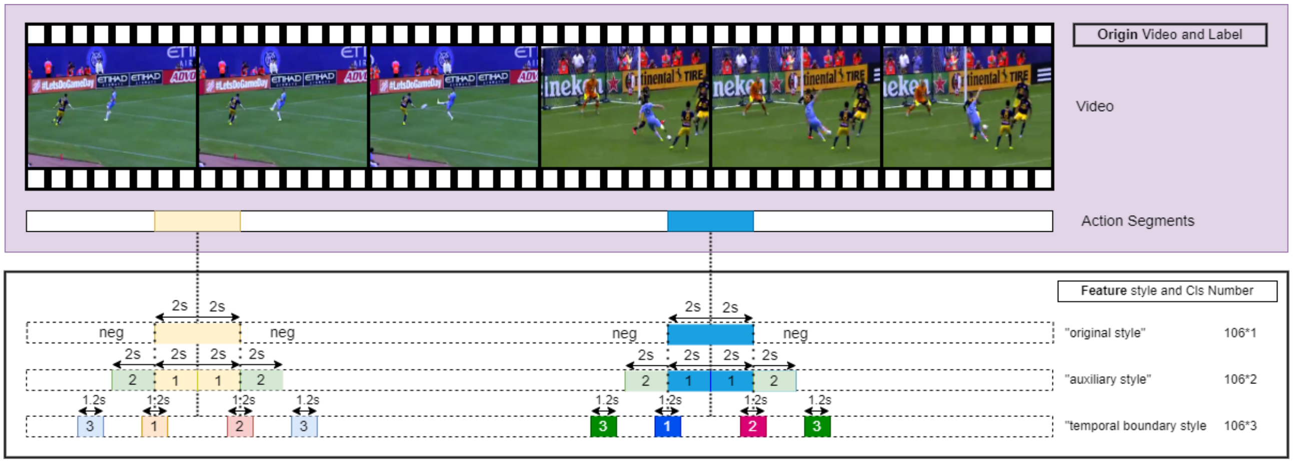

Figure 2: The data construction of the atomic classifier. Based on the original dataset, two different construction methods of atomic actions result in the generation of two different features, which are sensitive to temporal boundary in different degrees. Action segments with different colors present different annotated label classes.

Table 1: Action classification results on validation set of Fineaction dataset, measured by Accuracy(%). *Auxiliary style* stands for the training data with background classes. *Temporal boundary style* stands for the training data supplemented with start-end classes.

Model	Train Data	Top1 ACC	Top5 ACC
Swin	Auxiliary style	60.87	90.23
Swin	Temporal boundary style	49.73	90.45

1.2. Temporal Action Detection

1.2.1 Multi-Size Sliding Windows

FineAction[3] is a large-scale and fine-grained video dataset, containing 103K temporal instances of 106 action categories. We analyze the training and validation set, observing that the temporal duration is widely distributed, ranging from 0.5s to 400s. Although 70% of the instances fall within 3s, instances with longer duration are still a non-negligible component for TAL task. To take into account both short-term and long-term temporal instances, we introduce a multi-size sliding window strategy in the preprocessing of video features and train the model on these features separately. We set 4 different window sizes, spanning from 5s to 160s, with stride to be half of the corresponding window size. The final proposals from several models are integrated via the post-processing.

The compatibility of single window size to temporal duration is restricted, and instances with too long or too short duration will hinder feature learning. Thus, we employ double-sided thresholds to remedy this deficiency. The higher threshold excludes instances in annotations whose duration is longer than the window size. Conversely, the lower threshold filters out instances that are too short relative to the window size, set to be one tenth of the window size smaller one level than the sliding window. This double-sided criteria effectively enhances data centralization and avoids data omission. Ground truth for each window size follows the same filtering criteria at the video level. In addition, fake instances are synthesized manually to expand the training data scale. We randomly selected clip features without any instances, and covered part of them with instance features selected randomly from the same video. The ratio of fake to real in the training set is set 1:1. Finally, 4 sets of clip features are generated, and resized to 100 along the temporal dimension for subsequent prediction.

1.2.2 Temporal Proposal Generation

We apply a Faster-RCNN like network in this temporal action detection task, dubbed Faster-TAD[2]. By jointing temporal proposal generation and action classification with multi-task loss and shared features, Faster-TAD simplifies the pipeline of TAD.

As shown in figure 1, we construct our Faster-TAD with feature sequences extracted from raw video as inputs by VideoSwinTransformer[1] Extractor. We process the feature sequences with a base module to extract shared features, which consists of a CNN Layer, a Relu Layer, and a GCNeXt[4] Block. We then exert a Proposal Generation Mechanism to obtain most credible K coarse proposals, where K is 120. Proposals and shared features are further utilized to get more accurate boundaries by Boundary Regression Refinement Module[5]. At the same time, shared features and proposals are employed to get the semantic labels of action instances with Context-Adaptive Proposal Module.

Table 2: Action detection results on our new validation set of FineAction, measured by AUC and the average mAP(%). We construct new training and validation sets, by adding 4/5 of the original validation data to the training set, and taking the remaining as the validation set. CEL stands for cross entropy loss. $NS\ \ Window$ stands for negative sample windows.$TB\ style$ stands for Temporal boundary style.

Method	Feature	Class-Loss	Window Size	NS Window	AR@20	AUC
Faster-TAD	TB style	CEL+Triplet	5	✓	23.06	29.89
Faster-TAD	TB style	CEL+Triplet	10	✓	27.32	33.85
Faster-TAD	TB style	CEL	40	✓	68.11	71.53
Faster-TAD	TB style	CEL	160	✓	75.22	76.51
Faster-TAD	Auxiliary style+TB style	CEL+Triplet	5	✓	21.71	29.32
Faster-TAD	Auxiliary style+TB style	CEL+Triplet	10	✓	25.43	32.49
Faster-TAD	Auxiliary style+TB style	CEL	40	✓	65.67	69.12
Faster-TAD	Auxiliary style+TB style	CEL	160	✓	73.12	74.72
Faster-TAD	TB style	CEL	5	×	17.54	25.61
Faster-TAD	TB style	CEL	10	×	21.36	28.71
Faster-TAD	TB style	CEL	40	×	55.14	60.13
Faster-TAD	TB style	CEL	160	×	47.48	50.54
Ensemble Video Level Results			all windows	mAP 22.62	29.67	36.95

We make some improvements to tackle the challenges in temporal action detection. Faster-TAD includes Context-Adaptive Proposal Module to adaptively learn the semantic information of proposals by introducing attention mechanism across proposals to whole video and considering context as proximity-category proposals. Then the Fake Proposal based on the ground truth boundary with different offsets improves the Boundary Regression Module. Also, we found diverse features representation can complement each other, like “Auxiliary style” and “Temporal boundary style” in this paper. We employ Auxiliary-Features Block to adapt to the two streams input, and obtains remarkable performance.

The FineAction dataset contains many raw videos with diverse duration, which have lots of action instances within 2 seconds. This results in many sliding windows without any action instances, named as “negative sample windows” in this paper. To reduce false positives, we add “negative sample windows” to the training set, and set their confidence map label all to zero. This strategy greatly improves the AUC (Area under the ROC Curve) of the results.

1.2.3 Proposal Classification

In order to get clear classification boundaries, we propose to involve metric learning loss functions for explicit constraints of embedded feature distributions. In addition to the commonly utilized cross entropy loss, we adopt a metric learning loss function: triplet loss [6]. In order to explicitly constrain the similarity relationships between positive and negative sample pairs, during the training process, a mini-batch is grouped with P unique categories, each with K samples. As a sample may contain more than 1 category, only the first is taken into consideration at the batch sampling stage. Metric learning losses aim to form compact clusters for each category.

For an anchor sample in the mini-batch as x^i, whose similarity to positive and negative samples as s_p^i and s_n^i, the triplet loss [6] can be formulated with:

$$\mathcal{L}_{tr} = \left[s_n^i - s_p^i + m\right]_+, \tag{1}$$

where m represents the margin between clusters, and $[]_+$ stands for $\max(\cdot, 0)$. Triplet loss directly pulls close samples of the same category and pushes away those of different categories.

1.3. Ensemble

In the Chapter 1.1 mentioned before, we can generate discriminate features for temporal action detection. In this section, we synthesize the proposal classification results to form the final classification results, and apply soft-NMS [7] to the proposal localization results with different thresholds for different category. Besides, Boundary-Matching confidence map mentioned in BSN[8] enumerates all possible combination of temporal locations, bringing promotion in both efficiency and effectiveness.

2. Experiment

We train our TAD model in a single network, with batch size of 64 on 8 gpus. The learning rate is 6×10^{-4} for the first 3 epochs, and is reduced by 10 in epoch 3 and 7. We train the model with total 10 epochs. In inference, we apply Soft-NMS[7] for post-processing, and select the top-M prediction for final evaluation. M is 120.

We construct new training and validation sets, by adding 4/5 of the original validation data to the training set, and taking the remaining as the validation set. The results of TAD on the val dataset are shown in Table 2, which measured by AUC and the average mAP(%) as ActivityNet-1.3 [9].

References

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

[2] Shimin Chen, Chen Chen, Wei Li, Xunqiang Tao, and Yandong Guo. Faster-tad: Towards temporal action detection with proposal generation and classification in a unified network. *arXiv preprint arXiv:2204.02674*, 2022.

[3] 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.

[4] 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.

[5] 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.

[6] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[7] 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*, pages 5561–5569, 2017.

[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.

[9] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.