Self-Attention Refinement for Actionness Temporal Action Localization

Jiale Li[1,*], Yanzhu Hu[1], Yingjian Wang[1], Yan Qin[2]

[1] *Beijing University of Posts and Telecommunications, Beijing, China*
[2] *Institute Of Urban Safety And Environmental Science,Beijing Academy Of Science And Technology,Beijing, China*

Abstract

In this paper ,In this paper, for the problem of temporal action localisation, we propose a way to obtain the start and end times and types of actions based on actionness by aggregating action instance segmentation on a sequence of temporal features. In addition we believe that the context of actions is not only reflected in the results of convolution, but also the characteristics of inter class similarity and intra class consistency are essential. For this reason, we designed temporal self -attention mechanism (TSA) and temporal pyramid pooling module (TPP). Our results show that the single-stage model can achieve considerable accuracy after proper feature fusion.

Keywords

actionness; TSA; TPP

1. Introduction

Action recognition is a key technology in the field of computer vision. In recent years, great progress has been made in motion recognition technology, and relevant technologies have also been applied in video understanding, intelligent security and other directions. With the continuous progress of deep learning technology and image algorithm, some large-scale motion recognition network models and complex scene data sets are also produced, which promotes the progress of this field. With the continuous deepening of research, action recognition technology has also changed from simple primitive video classification to action instance identification of complex scenes, and then to action positioning. As network models improve, more information is learned and the model's output becomes more complex.

At present, motion recognition algorithms are mainly divided into video classification, sequential motion recognition and spatio-temporal motion recognition. The video classification technology mainly uses the trimmed video with fixed length as input, and determines the video category after extracting features through the backbone network. On the basis of action recognition, the uncut video is used to predict the action category and starting time in the video through the feature information. Spatiotemporal action recognition not only locates the time of action, but also locates the spatial position of action. Compared with the simple classification of video and the complex location of spatio-temporal motion, temporal motion recognition is the most widely used method in the field of abnormal behavior recognition.

The main method of Temporal action detection is similar to that of target detection. After data processing, different proposal methods are combined with features to complete decoding and obtain output. Many detectors are developed based on target detection network, SSAD[1] developed from SSD method and DaoTAD[2] developed from the RetinaNet, etc. According to the proposed method, timing action recognition technology can be divided into base anchor, anchor free and actionness. The Base anchor method mainly presents starting frames of different sizes and scales, calculates the intersection

ICCEIC2022@3rd International Conference on Computer Engineering and Intelligent Control
EMAIL: *ljl19971013@163.com (Jiale Li)

ratio between the current output position of the model and anchor, and determines the allocation of current positive and negative samples through the intersection ratio. R-C3D[3] uses 3D full convolution to extract features from videos. TAL-Net[4] enhances the feature effectiveness of R-C3D by obtaining global attention to the timing sequence of its sensitive field. STPN[5] conducted constraints by enhancing feature sparsity, MAAN[6] enhanced generalization by reducing model dominant factors, and Zhong[7] et al. introduced fine-processing operations to achieve proposal accuracy of the model. In this paper, we design a method based on feature extension mechanism and receptive field fusion mechanism, and achieve excellent performance.

2. Method

The self-attentive temporal action recognition model uses a backbone feature network to extract three-dimensional features, a fully connected layer to obtain one-dimensional features.

Suppose we give an unclipped video $\{V_n\}_{n=1}^{N}$ and their action instance clips, including category $\{y_n\}_{n=1}^{N}$, where y_n is a one pot encoding vector and C is the action category. Our goal is to output the action category, start time and end time.

We use the mixed mode of optical flow and RGB as the input, and use I3D network pre trained on Kinect as the backbone to extract the network. For the input of a video $X_n \in R^{C\times T\times H\times W}$, the output is $\hat{X}_n \in R^{C'\times T\times H'\times W'}$, where $H'$and W' disappear after dimensional compression to obtain new 1D data $y \in R^{T\times d'}$, where T represents the timing length, d' represents the data dimension, and the new 1D data contains the spatial information of each frame of the video image and the overall timing information. Then the feature map is sent to the TAS module to obtain the global attention, and weighted and fused into the original sequence to obtain a new sequence order$\hat{y} \in R^{T\times d}$. Through different convolutions, action suggestion sequences are obtained through shared channels, including action instance distribution, classifier and time boundary regression. Then, when the obtained action instance distribution is merged with $\hat{y}$, it is sent to TPP, and the refined action instance sequence is output through the shared convolution channel.

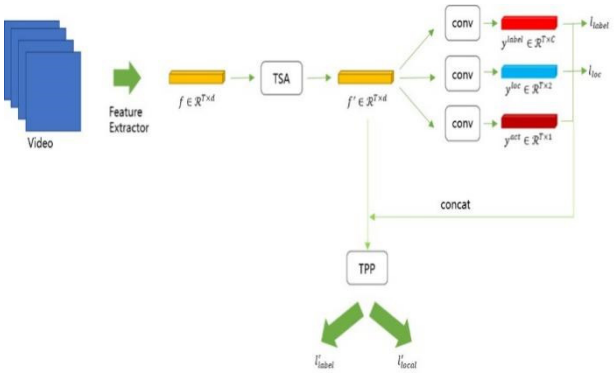

Figure 1 Network overall structure

2.1. Feature Extractor

For the feature extraction part, we used the high performance ResNet I3D as the backbone extraction network. To scale, we used resneti3d with the last layer of the averaging pool removed to extract the video feature information, and used a global averaging pool approach to compress the features. Given the input video clip, 1D features of shape (512,100) were obtained.

2.2. Temporal Self Attention Module

In order to obtain the global attention of action instances throughout the time series and to reduce the impact of the limitations of local feature extraction, we designed the TSA module. We consider that different action instances have some similarity in the sample distribution. In time series information,

due to the continuous nature of actions, we believe that it is valuable to calculate the feature similarity between successive categories and successive action instances. We find that different actions and actions should have different weight coefficients to the background before similar actions, and by doing so, we can avoid learning too much irrelevant information. Similarly, if two sets of patches represent the same action instance, the information they learn is redundant, so we should avoid this situation. To solve this problem, we avoid this situation by calculating the T_{iou}. in such a way that the time span between the two sets of patches is guaranteed to exceed a certain threshold. the working model of TSA is shown in Figure 2.

Formally, two different action instances p_i and p_j, and the context B between them, form a local time series (p_i, B, p_j). A_1carries the temporal information (c_1,w_1,C_1) and A_2 carries the information (c_2,w_2,C_2). By decoding, we obtain the category information $f_1 \in R^{T\times C}$and the location information $f_2 \in R^{T\times 2}$in the input information. we then build the condition matrix A, where $A(i,j)$ represents whether p_i and p_j, belong to the same type of action.

$$A(i,j) = \begin{cases} 0.5\ class_i = class_j \\ 0.4\ class_i \neq class_j\ and\ class_{i,j} \in action \\ 0.1\ one\ of\ class_{i,j} \in action \\ 0\ both\ of\ class_{i,j} \in bg \end{cases} \tag{1}$$

by i, and the maximum value obtained after calculating f_1 for softmax is used as the current prediction category, bg represents the background sample, and action represents that the current category belongs to the action category.

Similarly, in order to ensure that the correlation weight of two groups of different instances is calculated, we construct the distance similarity matrix B, where $B(i,j)$ represents the position similarity between p_i and p_j, and we measure this feature through T_{iou}.

$$T_{iou} = \frac{Interval_i \cap Interval_j}{Interval_i \cup Interval_j} \tag{2}$$

where interval represents the span between the start and end time of the patch prediction action instance.

The core mapping of each patch, so we use the self attention mechanism in Transformer to map the time series y to different feature spaces through space mapping.

$$\begin{aligned} Q &= sigmoid(f_1(y)) \\ K &= sigmoid(f_2(y)) \\ V &= sigmoid(f_3(y)) \end{aligned} \tag{3}$$

$$S(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{4}$$

The feature maps of time series in different spaces are obtained. Finally, the final weight coefficient matrix M is obtained by fusion, where $M(i,j)$ represents the weight map of feature p_i on feature p_j.

The final mapping coefficient matrix $G(i,j)$is formed through matrices A, B and M, representing the correlation system between actions and background, and a new time series vector is obtained through weighted fusion coefficient.

The matrix G is obtained by fusion to effectively ensure that the currently acquired weight matrix does not incorporate redundant information. Through the matrix V, a filtering operation is performed on the feature vector after we have acquired the global attention, and finally a new vector is obtained for the output, which at this point effectively acquires the similarity between action classes and expands the action information.

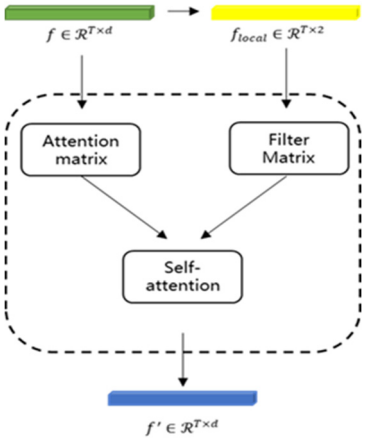

Figure2 TSA(Temporal Self Attention) overall process.

2.3. Temporal Pyramid Pooling Module

In order to better judge the enhanced time series features, we obtain the class and interval information through the convolution channel, and or the distribution of the most important action instances, and pre-judge the current features by calculating $Loss_{class}$ and $Loss_{loc}$.

$$Loss_{class} = Focal(y_{true}, y_{pred}) \tag{5}$$

$$Loss_{act} = Diouloss(Interval_{pred}, Interval_{true}) \tag{6}$$

Next, in order to fully extract the deep features of the time series information and limit the start and end times to the precision boundaries, we use the TPP module to refine on the temporal features to obtain the saliency boundary features, as shown in Figure 3. The features after fusing the action instance distribution feature $y_{act} \in R^{T\times 1}$ and the global attentional temporal feature $\hat{y} \in R^{T\times d}$ are used as input, while to aggregate deeper temporal features, we perform feature downsampling by using a feature pyramid, e.g. for feature f belonging to $y_{fuse} \in R^{T\times h}$, using a convolution operation to process.

$$F_{fuse} = Relu\left(\mathrm{BN}\left(\mathrm{Conv}(y_{fuse})\right)\right) \tag{7}$$

When processing temporal features using the convolution operation, due to the short time span of some temporal features, too much feature information is lost in the process of downsampling, but at the same time, in order to maximize the perceptual field, we have to stack convolution blocks, so we introduced the hole convolution method, through different proportions of hole convolution blocks, to obtain different scales of temporal feature information, for example, for feature f belongs to $y_{fuse} \in R^{T\times h}$, using the convolution operation to process:

$$F_{fuse} = Relu\left(\mathrm{BN}\left(\mathrm{Dilated\_Conv}_{rate}(y_{fuse})\right)\right) \tag{8}$$

Based on the actionness approach, we use a mask to obtain the current action instance distribution as well as the action interval distribution by aggregating the instance distributions. Specifically, we use linear interpolation to upsample features to the original feature sequence length T. By obtaining information about the action instance temporal distribution, we restore the action distribution in the original output temporal sequence by means of feature mapping, and finally obtain the category information by performing softmax operations on the features of each aggregated dimension.

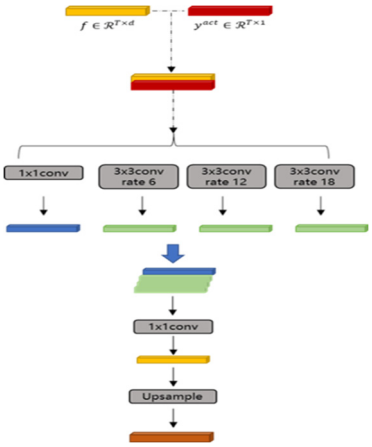

Figure 3 TPP(Temporal Pyramid Pooling) overall process.

3. Experiment

3.1. Datasets

Thumos'14 includes 200 training sets and 212 test sets in the time sequence motion detection direction. Each sample contains 20 types of actions, which are basically daily actions. Frame level annotation includes the start time, end time and kind of each action.

ActivityNet1.2 is a large action recognition data set. The entire data set contains 4819 training sets and 2383 data sets for testing. It also uses frame level annotation for training and testing.

3.2. Training

We sampled the RGB and optical streams at 10 frames per second on the dataset, with each segment limited to 128 frames in length and not having overlapping frames. For the training process, we used temporal random sampling for sampling. Specifically, the model input was limited to contain at least one action instance with a Tiou of 0.75 or higher, and the frame space size was limited to 112x112.To speed up training, we used I3D pre-trained on kinect as the backbone feature to extract network weights.

3.3. Result

It can be seen from Table 1 and Table2 that our model performs at a leading level among all I3D-dominated feature extraction networks. On the Thumos14 dataset, our model handles the optimal level on Map@0.5 and Map@0.6, which benefits from our TAS and TPP modules, and on the ActivityNet dataset we are also at the leading level on Map@0.75. This shows that the modules we have designed are effective.

Table 1 Different algorithms performance on Thumos 14

Model	Backbone	Thumos14			
		0.3	0.4	0.5	0.6
TURN	I3D	44.1	35.9	25.6	——
R-C3D	C3D	44.8	35.6	28.9	——
TAL	I3D	53.2	48.5	42.8	33.8
GTAN	P3D	57.8	47.2	38.8	——
SSN	TS	51.0	41.0	29.8	——
BSN	TS	53.5	45.0	36.9	28.4
BMN	TS	56.0	47.4	38.8	29.7
BU-TAL	I3D	53.9	50.7	45.4	38.0
TSA-TAL	I3D	54.7	51.2	49.8	40.4

Table 2 Different algorithms performance on ActivityNet

Model	Backbone	ActivityNet1.2		
		0.5	0.75	0.95
TURN	I3D	——	——	——
R-C3D	C3D	26.8	——	——
TAL	I3D	38.2	18.3	1.3
GTAN	P3D	52.6	34.1	8.9
SSN	TS	43.2	28.7	5.6
BSN	TS	46.5	30.0	8.0
BMN	TS	50.1	34.8	8.3
BU-TAL	I3D	43.5	33.9	9.2
TSA-TAL	I3D	51.1	35.0	8.9

4. Conclusions

For the purpose of inaccurate localisation accuracy of temporal action recognition, we propose a TSA module for extracting the global attention of the model, while using suitable filters to remove redundant temporal space information, and we use a TPP module to fuse the temporal information of different levels of sensory fields to enhance the performance of the model on different scales of action instances, our model is implemented end-to-end and achieves in effect almost the same level of effectiveness as the two-stage model, while greatly reducing inference time, and we used a single-stream columnar network structure, which is less computational than a large two-stream network like slowfast.

However, our model also has certain limitations when it comes to inference. Firstly, our annotation requirements for real samples are at the frame level, raising the difficulty of acquiring the dataset, and secondly, attentional information can easily be incorrectly fused when faced with action behaviours that are relatively similar. Therefore, much improvement is needed to address this point in the coming time.

5. References

[1] [Tianwei Lin.: Single shot temporal action detection. In Proceedings of the 25th ACM international conference on Multimedia, pages 988–996, 2017.

[2] Wang C.:RGB stream is enough for temporal action detection[J]. arXiv preprint arXiv:2107.04362, 2021.

[3] Huijuan Xu.: R-c3d: Region onvolutional 3d network for temporal activity detection. InProceedings of the IEEE international conference on computer vision, pages 5783–5792, 2017.

[4] Yu-Wei Chao.:Rethinking the faster r-cnn architecture for temporal action localization. In Proceedings ofthe IEEE Conference on Computer Vision and Pattern Recognition, pages 1130–1139,2018.

[5] Phuc Nguyen.: Weakly supervised action localization by sparse temporal pooling network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6752-761, 2018.

[6] Yuan Yuan.: Marginalized average attentional network for weakly-supervised learning. arXiv preprint arXiv: 1905.08586, 2019.

[7] Jia-Xing Zhong.: Step-by-step erasion, one-by-one collection: a weakly supervised temporal action detector. In Proceedings of the 26th ACM international conference on Multimedia, pages 35–44, 2018.