Timeception Single Shot Action Detector: A Single-Stage Method for Temporal Action Detection

Xiaoqiu Chen, Miao Ma(✉), Zhuoyu Tian, and Jie Ren

School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
mmthp@snnu.edu.cn

Abstract. Temporal action detection is used to detect the start and end times and classify the potentially specific actions in a video. Prior studies in temporal action detection perform weak because they can not fully understand the whole input video's temporal structure and context information, and fail to adapt to the diversity of action time span. We propose a novel Timeception Single Shot Action Detector (TC-SSAD) to solve the problems mentioned above. In detail, we leverage the multiple Timeception layers to generate multi-scale feature sequences, where each Timeception layer uses depthwise-separable temporal convolution with multi-scale convolution kernels to capture the diversity of time spans. Besides, we use the super-event modules to learn the entire input video's temporal structure and contextual information. The experimental results on THUMOS14 dataset show that when IoU threshold is 0.5, our method achieves 38.2% and 44.3% mAP on Two-stream features and Two-stream i3D features respectively, which is better than Decouple-SSAD network based method by 2.4% and 0.6%. Our method on Activitynet-1.3 dataset achieves 20.4% mAP, which is better than Decouple-SSAD network based method by 0.61% as far as Two-stream features on concerned.

Keywords: Temporal action detection · Multi-scale convolution kernel · Super-event module · Temporal structure and context

1 Introduction

With the development of the Internet and the proliferation of personal smart mobile devices, people are generating, storing and using large amounts of video [1, 2]. Most videos are long untrimmed videos, which often contain multiple action instances and have more interference from background and irrelevant actions. Action detection is detecting action instances in long videos, including the start and end times corresponding to action instances and action categories. Action detection is more practical, and its progress can promote a large number of related tasks from real-time applications, such as extracting highlights from sports videos, to automatic video subtitles and other higher-level tasks [3].

Like the object detection task, the common temporal action detection methods can be divided into two-stage and single-stage. The Two-stage method uses sliding windows or

Y. Peng et al. (Eds.): ICIG 2021, LNCS 12888, pp. 340–354, 2021.
https://doi.org/10.1007/978-3-030-87355-4_29

some specific methods (sliding window or action probability curve) to generate proposals and then classifies these action proposals. First of all, the common sliding window method can only produce short proposals no larger than the predefined window size [4]. Second, because the two-stage detection method train action proposal generation and classification separately, the time boundary of action proposals before classification has been fixed. The indirect optimization strategy can not get the optimal solution[5]. At the same time, the single-stage method ignores the action proposal generation and directly predict the time boundary and categories confidence of actions. This type of method encapsulates two subtasks of localization and classification into a single network but ignores the characteristics of each subtask. Since the single-stage methods share the same feature map when predicting the action category and coordinate offset values, this coupling characteristic may affect the accuracy of each task. Therefore, Decouple-SSAD [5] network introduces parallel classification and localization units through deconvolution operation to decouple two subtasks.

The diversity of time spans of action segments in videos is one of the main reasons for the poor performance of current action detection methods. The traditional single-stage action detection methods shorten the length of the feature sequence and increase the receptive field of each temporal position in feature sequences by stacking multiple 1D temporal convolutional layers, thereby predicting the coordinate offsets and categories confidence of the action proposals. These 1D temporal convolutional layers usually use convolution kernels with a fixed scale. Therefore, the receptive fields corresponding to each temporal position in the generated feature map sequence are fixed. Therefore, these methods cannot adapt well to the diversity of the time spans of action segments.

Compared with the spatial contextual information of pictures in object detection, the temporal structure and contextual information of the video may be more important for obtaining accurate time boundaries and classification results [3]. The single-stage methods cannot effectively use the temporal contextual information of action proposals due to the characteristics of generation and classification action proposals at the same time. To solve these problems, we propose TC-SSAD, a Decouple-SSAD based network. The main contributions of this paper are:

(1) In this paper, we use Timeception layer with multi-scale convolution kernel to construct the backbone network, which is used to obtain multi-scale feature sequences. Instead of using 1D temporal convolution layer with a fixed kernel size, the multi-scale convolution kernel can better capture the diversity of action segments during a period.
(2) We also introduce a super-event module to model the whole input video's temporal structure and context information. This module obtains the super-event representation, which can effectively enhance the performance of action detection.

2 Related Work

2.1 Temporal Action Proposal Generation

Temporal action detection can be decomposed into two sub-tasks: action proposal generation and classification. High-quality action proposals are essential to enhance the effectiveness of action detection task.

The methods for generating temporal action proposals can be divided into two major categories: the first type of methods formulates it as a binary classification problem on sliding windows. Among them, SCNN-prop [6] trains a C3D network [7] for action proposal generation. TURN [8] builds video units in a pyramid manner and improves the recall rate of action proposal generation through temporal boundary regression. The second type of methods uses the Temporal Action Grouping (TAG) [9, 10] algorithm to aggregate consecutive high-scoring intervals as action proposals based on Snippet-level action scores. For example, Boundary Sensitive Network (BSN) [11] generates action proposals based on three sets of actions curves, but this kind method based on action scores may be omitted dense and short actions due to the difficulty in distinguishing very close start and end peaks in the action score curves.

2.2 Temporal Action Detection

S-CNN [6] solved this problem by constructing a proposal generation network, a classification network, and a localization network based on 3D convolution. CDC [12] uses convolution-deconvolution operations on the basis of C3D network to predict the actions of frame-level granularity. Inspired by the Faster R-CNN [13] algorithm, R-C3D [14] extended the Faster R-CNN framework to the field of action detection, showed its versatility on different datasets. On the basis of R-C3D, TAL-Net [3] has researched and improved how to deal with the diversity of action time spans and how to use temporal contextual information, obtained state-of-the-art performance on THUMOS14 dataset [15].

The single-stage SSAD [16] network skips the process of generating action proposals and uses traditional 1D temporal convolutional layers to directly perform boundary regression and classification on multiple generated action proposals. Similarly, SS-TAD [17] uses reinforcement learning to train the RNN structure, which is end-to-end and directly performs action detection.

3 Methodology

3.1 Overview of TC-SSAD

Figure 1 presents TC-SSAD, a single stage action detection network based on Decouple-SSAD [5].

First, Two-stream i3D network [18] is used to encode the frame sequence corresponding to a long video into 1D feature sequence. The generated feature sequence goes through two traditional 1D temporal convolutional layers and a maximum pooling layer to reduce the time dimension. After that, the feature sequence is sent to a multi-unit network to generate multi-scale feature sequences. Multi-unit network consists of three parts: backbone unit, classification and localization unit. The backbone unit is composed of multiple Timeception layers [19] and super-event modules [20] in a cascade manner. Furthermore, parallel classification and localization units are constructed from the deep feature sequences of the backbone unit through deconvolution operations and the fusion of the prevIoUs layer feature sequences. The two parallel units focus on the

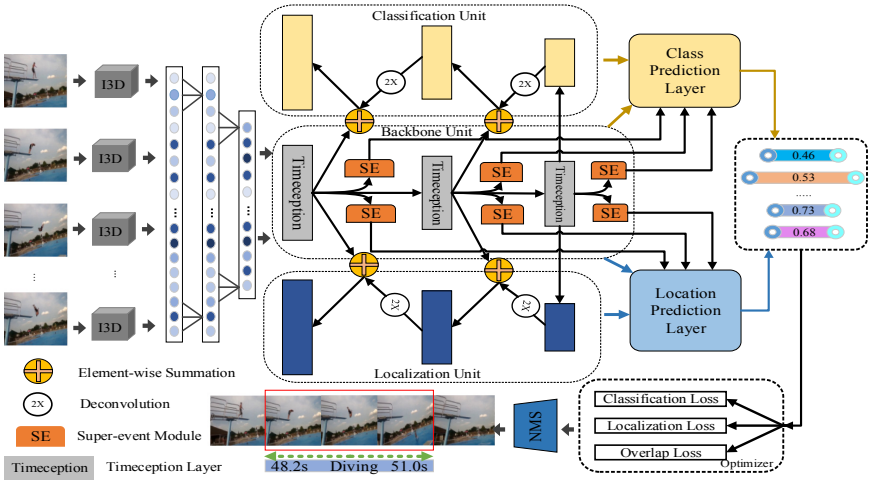


Fig. 1. Overview of TC-SSAD

task of action category confidence generation and coordinate regression, respectively. In addition, in order to learn the temporal structure and contextual information of the entire input video, multi-scale features generated by each Timeception layer of the backbone unit passes the super-event module to obtain the super-event representation, the learned super-event representation is fused with the multi-scale feature sequences generated by two parallel units to obtain the final feature expressions. A series of multi-scale feature sequences generated by three units pass the classification prediction layer and the localization prediction layer to predict categories confidence and coordinate offsets values corresponding to the action proposals. Classification loss, regression loss, and overlap loss are used to optimize different units during the training phase. Post-processing and non-maximum suppress (NMS) are performed on the generated action instances during testing phase to obtain the final results.

3.2 Backbone Unit

The backbone unit of TC-SSAD consists of Timeception layers and the corresponding super-event modules in a cascade manner.

Timeception Layer. For input feature F, hypothesis the feature dimension of F is $d_m = \mathbb{R}^{\{T \times L \times L \times C\}}$, where T is the temporal dimension, L is spatial dimension, C is the number of channels. Figure 2 presents the multi-scale feature sequences of the backbone unit are obtained by the Timeception layers through the following steps:

Firstly, the input features are divided into some groups according to channels to reduce the dependency and complexity between channels. The feature dimension of each group is $g_m = \mathbb{R}^{\{T \times L \times L \times C/N\}}$. Then, each group uses a temporal convolution module to convolve the obtained feature sequences. Specially, each group is further divided into 5 units, and the middle 3 units use depthwise-separable temporal convolution with multi-scale convolution kernels to reduce the amount of network parameters while ensuring

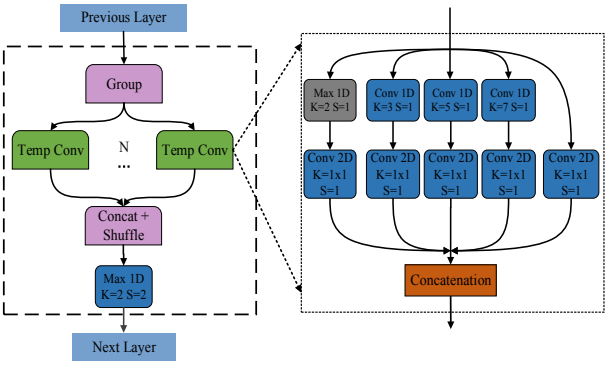


Fig. 2. Architecture of Timeception layer

that the network can well adapt to the diversity of the time spans of action segments. The leftmost unit only uses maximum pooling with kernel size $K = 2$ and stride $S = 1$. A 2D convolution with kernel size after each unit is used to enhance the non-linear expression ability of this Timeception layer. Last, the output of the 5 units through concatenation operation to obtain the final output feature sequences of this group. Finally, for the output features of each group, first perform shuffle operations on these features to exchange information between different channels to ensure the randomness of the channels when the next Timeception layer performs grouping operations. We concat the output features of each group and then go through the maximum pooling layer with kernel size $K = 2$ and stride $S = 2$ to get the final output features of this layer. Specifically, after each Timeception layer, the time dimension of feature sequence is reduced to 1/2, and the number of channels is increased to 1.25 times. For the features input the backbone unit, after three Timeception layers, feature sequences of 3 different scales are finally generated for subsequent processing.

Super-Event Module. As shown in Fig. 3 super-event representation is obtained by the temporal structure filters in the super-event module by learning the soft attention weights of each type of action. Temporal structure filters capture temporal context in videos by paying attention to the frame information of some temporal positions and representing the variable length input video features as fixed length feature vectors.

For the feature sequence with time dimension T, each temporal structure filter can be determined by the following formula:

$$\begin{aligned} \hat{x_n} &= \frac{(T-1)\cdot(\tanh(x_n)+1)}{2} \\ \hat{\gamma_n} &= \exp(1-2\cdot|\tanh(\gamma_n)|) \\ F[t,n] &= \frac{1}{Z_n \pi \hat{\gamma_n}\left(\frac{\left(t-\hat{x_n}\right)}{\hat{\gamma_n}}\right)^2} \end{aligned} \tag{1}$$

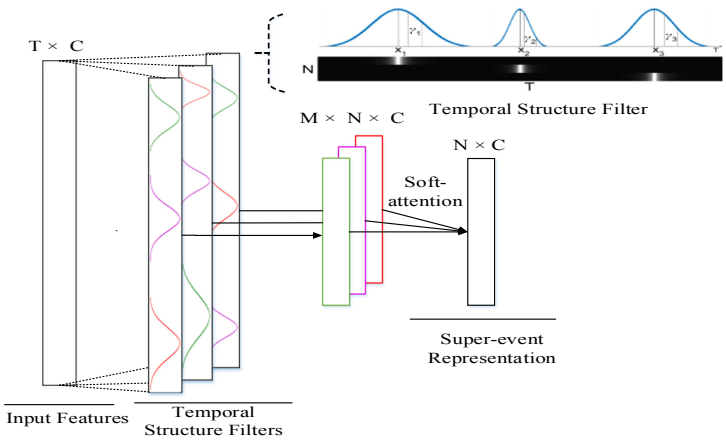


Fig. 3. Architecture of super-events module

where $t \in \{1, \ldots, T\}$, $n \in \{1, \ldots, N\}$, two parameters x_n and γ_n are used to control the center and width of the Cauchy distribution, Z_n is a normalization constant, In particular, for each super-event module, only two parameters x_n and γ_n need to be learned.

Because the number of action categories is much larger than the number of temporal structure filters. In order to use a fixed number of temporal structure filters to represent multiple types of actions, it is necessary to combine the temporal structure filter and soft attention mechanism to obtain the final super-event representation, as shown in the following formula:

$$\begin{aligned} S_C &= \sum_{m}^{M} A_{c,m} \cdot \sum_{t}^{T} F_m[t] \cdot v_t \\ A_{c,m} &= \frac{\exp\left(W_{c,m}\right)}{\sum_{k}^{M} \exp\left(W_{c,k}\right)} \end{aligned} \tag{2}$$

where S_c is the finally obtained super-event representation, M is the number of temporal structure filters, V_t represents the video features, that is, the output of the Timeception layer, $A_{c,m}$ represents the soft attention weight corresponding to each temporal structure filter. The subscript C represents the number of video action categories.

3.3 Classification and Localization Units

Like the backbone unit, each parallel unit contains three kinds of multi-scale feature sequences. Each type feature sequences are obtained by averaging the corresponding deeper feature sequences in the backbone unit through deconvolution operation with shallow feature sequences. The specific decoupling process can be expressed by the

following formula:

$$f_k^L = \begin{cases} C\left(f_n^f\right), & if\ L = N_f \\ C\left(S\left(C\left(f_n^L\right), D\left(f_k^{L+1}\right)\right)\right), & if\ 1 \le L < N_f \end{cases} \tag{3}$$

where C represents the traditional temporal convolution operation, D represents the deconvolution operation, S represents the corresponding element addition and fusion operation, N_f is the number of layers of the Timeception layer in the backbone unit, and L is used to indicate which layer is currently operated on.

3.4 Classification and Localization Prediction Layers

For the multi-scale feature sequences generated by the three units. First, a series of anchors with different basic scale $B_S$and aspect ratio R_S are predefined for each temporal position of the feature sequence.

For a series of predefined anchors obtained above, we send the corresponding feature sequence to the classification and localization prediction layer to generate the prediction result vector $V_{pred} = (S_{cls}, S_{over}, \Delta_c, \Delta_W)$, where S_{cls} and S_{over} are categories confidence and overlapping confidence, Δ_c and Δ_W are predicted centre and width coordinate offset values. It is worth noting that for the classification unit, we generate the result vectors through a multi-class prediction layer because it focuses on classification tasks. The localization unit focuses on localization tasks, and we generate the result vectors through a binary classification prediction layer. For each predefined anchor, the final prediction results can be obtained by the following formula:

$$\begin{aligned} \varphi_C &= \mu_C + \alpha_1 \mu_W \cdot \Delta_C \\ \varphi_W &= \mu_W \cdot \exp(\alpha_2 \cdot \Delta_W) \end{aligned} \tag{4}$$

where μ_c and μ_w are the predefined center point and width, respectively. The parameters α_1 and α_2 are used to control the degree of influence of the predicted value on the result. φ_c and φ_w are final predicted results.

3.5 Loss Function

Training TC-SSAD networks is a multi-task optimization problem. The final loss function is:

$$L = L_{cls} + \lambda L_{reg} + \beta L_{over} \tag{5}$$

where L_{cls}, L_{reg}, L_{over} are classification loss function, localization loss function and overlap loss function. For the multi-classification tasks in this paper, we use the common softmax loss function. We use the Smooth L1 loss function to measure the degree of error between the predicted coordinate values and the true values. Finally, we use the mean square error loss function to measure the overlap between the predicted action proposals and the real annotations.

4 Experiments

4.1 Datasets and Evaluation Metrics

THUMOS14 Dataset [15]. The dataset contains videos from 20 sports action classes, with a total duration of more than 24 h. We use the validation set to train the model and evaluate it on the test set. The validation and test sets contain 200 and 213 untrimmed videos with temporal annotations. The average duration of each video is more than 3 min and contains more than 15 action instances, which makes the dataset particularly challenging.

ActicityNet-1.3 Dataset [21]. The dataset contains 19994 videos, each video contains about 1.5 action instances, about 36% of which are background clips, and there are 200 kinds of actions in total. The whole data set is divided into training set, verification set and test set according to the ratio of 2:1:1, and the test set is not open for competition, so most researches test the model performance on the validation set.

Evaluation Metrics. In this paper, the public evaluation code is used to evaluate the experimental results. We use mean Average Precision (mAP) as the main evaluation metrics. For the predicted action proposals, we only mark the result as correct if the prediction category is correct and the intersection ratio with ground truth is greater than the specified IoU threshold.

4.2 Implementation Details

For THUMOS14 [15] and ActivityNet-1.3 [21] datasets we set batch size is 24; the learning rate for the first 38 epochs to 0.0001 and the learning rate for the last 3 epochs to 0.00001. Adaptive moment estimation (Adam) algorithm is used to optimize the network, and Xavier algorithm randomly initializes network parameters before training. In the test phase, we use non-maximum suppression (NMS) to remove redundant prediction results (NMS threshold is 0.2). For the Timeception layers, we set groups equal to 4. The weight of classification loss is 1, and the weight of location loss and overlap loss is 10. For THUNOS14 dataset, we set $Bs = \{1/16, 1/8, 1/4\}$, $Rs = \{0.5, 0.75, 1, 1.5, 2\}$; For ActivityNet-1.3 dataset, we set $Bs = \{1/16, 1/12, 1/8, 1/6, 1/4\}$, $Rs = \{0.15, 0.25, 0.5, 0.75, 1, 1.5, 2, 3\}$. For two datasets, the convolution kernel size $K = \{3, 5, 7\}$ and the number of temporal structure filters $N = 3$ are used in the following experiments. Due to the large scale of ActicityNet-1.3 dataset, we did not specifically extract its features but directly used the Two-stream features provided by BSN [11].

4.3 Experimental Results

Results of THUMOS14 Dataset. When using a single GTX TITANX GPU, the training time of the verification set composed of 200 videos on the THUMOS14 dataset is about 102 min, and the test time is about 4 min. The experimental results of THUMOS14 dataset [15] are shown in Table 1.

Table 1. Experimental results of THUMOS14 dataset.

Method	Feature	Model	mAP(%)@IoU = 0.5		
			Spatial	Temporal	Fuse
Decouple-SSAD [5]	Bn-Inception	Decouple-SSAD(512)	22.1	33.1	35.8
	Inception_V3	Decouple-SSAD(512)	**30.7**	**44.2**	**43.7**
TC-SSAD	Bn-Inception	DS + TC(512)	23.9	35.1	36.6
	Bn-Inception	DS + TC + SE(512)	23.3	36.9	38.2
	i3D	DS + TC(512)	32.5	35.6	38.2
	i3D	DS + TC + SE(512)	33.1	**41.1**	42.3
	i3D	DS + TC + SE(1024)	**36.4**	41.0	**44.3**

In the feature column, "Bn-Inception" and "Inception_V3" are the original Two-stream feature extraction backbone networks, while Two-Stream i3D network [25] is the feature extraction network used in this paper. In the model column, "Decouple-SSAD" indicates the method adopted in [5], "DS" indicates the use of parallel decoupling units, "TC" indicates the use of Timeception layers, "SE" indicates whether to use the super-event modules, and the bracket is the window size when extract feature. As can be seen from the table, the highest mAP value obtained by our method is 44.3%.

Effectiveness of Timeception Layer and Super-Event Module. In order to verify the validity of the Timeception layers and the super-event modules in our model. First, we use the same features for experiments. The results of the first four rows of Table 1 show that when using only the Timeception layers, there is some improvement over Decouple-SSAD. Spatial and Temporal features increased by 1.8% and 2.0%, respectively, and the result after fusion increased by 0.8%. Besides, when the super-event module is added, the performance is improved significantly, and the fused mAP is increased from 35.8% to 38.2%.

Impact of Different Features on Results. Because a traditional Two-stream network uses 2D convolution, it cannot capture the temporal dependency between frames. As with Decouple-SSAD, we first extract the Two-stream i3D features with 512-frame window size. Based on these features, only the Timeception layers can be used to achieve the same effect after the original super-event representation was added. The results illustrate the effectiveness of i3D network for spatiotemporal modelling.

In addition, we further expanded the window to 1024 frames (about 34 s in duration), and we found that the network's performance was further improved, from 42.3% to 44.3%. We suspect that this is mainly because the long video contains the richer temporal structure and contextual information, and the super-event modules can effectively learn the temporal structure and contextual information in these input videos to further enhance network performance.

Per-class AP. We compared the AP values of each action category after fusion between our method and Decouple-SSAD at IoU = 0.5. The results are shown in Fig. 4 Our

method shows good detection performance on two different video features. Through the analysis of "basketball dunk", "billiards" and other categories of video, we find that the duration of these categories of action is very different, and our method performs well in these categories of video.

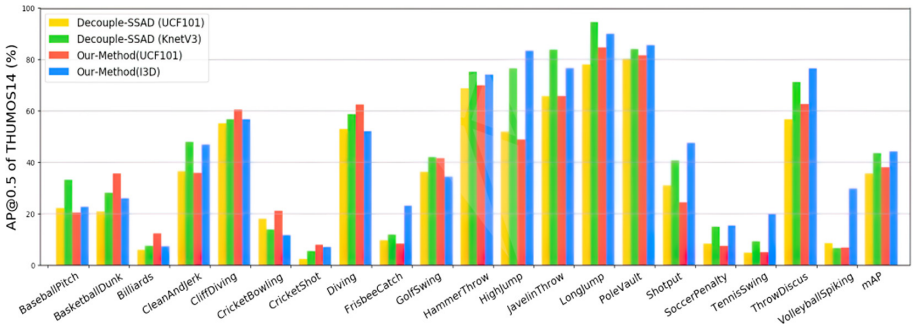


Fig. 4. Comparison of per-class Average Precision and mAP with overlap threshold 0.5 in THUMOS14 test set.

Results of ActivityNet-1.3 Dataset. We also make a comparative experiment on Activitynet-1.3 dataset [21]. During the experiment, we all used the Two-stream features provided by BSN [11]. The results of mAP and average mAP (0.5:0.05:0.95) when IoU = {0.5,0.75,0.95} after fusion are shown in Table 2.

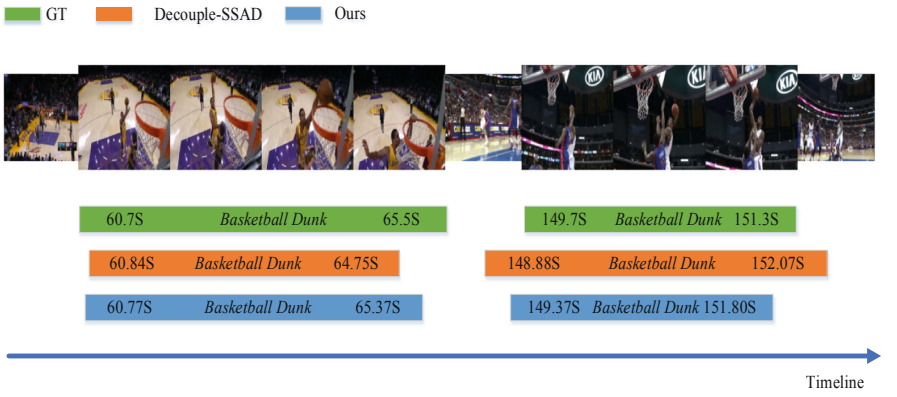
Table 2. Experimental results of activityNet-1.3 dataset.

Method	Feature	Model	mAP(%)@IoU			
			0.5	0.75	0.95	AVG
Decouple-SSAD [5]	Two-stream	Decouple-SSAD	33.15	19.99	1.78	19.81
TC-SSAD	Two-stream	DS + TC	**34.11**	20.17	**2.47**	20.41
	Two-stream	DS + TC + SE	33.61	**20.71**	2.32	**20.42**

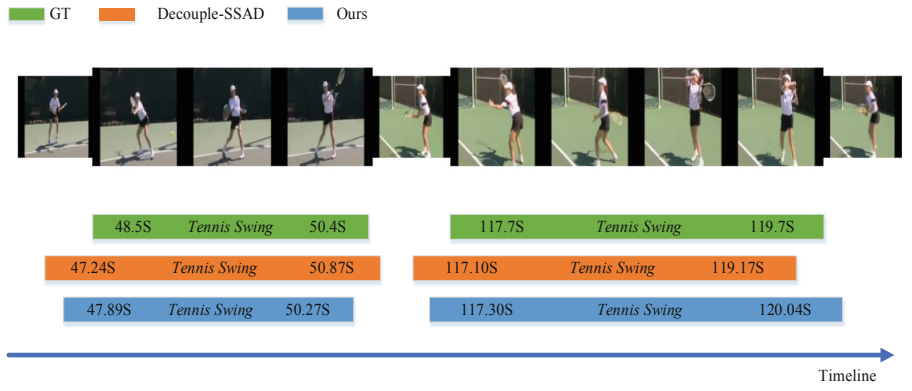
Table 2 shows that the average mAP is still 0.61% higher than that of Decouple-SSAD in ActivityNet-1.3 dataset. In particular, we find that the results are not effectively improved after using super-event modules, and the results under some thresholds are still reduced. This is because ActivityNet-1.3 dataset contains 200 types of actions, and each video contains only about 1.5 action instances. Most of the action instances occupy more than half of the whole video. In this case, there is no rich temporal structure and context information to learn.

4.4 Visualization of Temporal Action Detection Results

In Fig. 5 The temporal action detection results of THUMOS14 dataset are visualized. Each row contains the predicted start and ends time and category of the action, and the sampled frame image is used at the top to display the content of the video. As shown in Fig. 5, Our method is more accurate in predicting the start and end time of actions. The results show that the proposed method is more effective.



(a)



(b)

Fig. 5. Qualitative visualization of TC-SSAD model predictive action instances. (a) Detection results after fusion under UCF101 pre-trained Two-stream features; (b) Detection results after fusion under Kinetics pre-trained Two-stream features, where we use Two-Stream i3D network [18].

4.5 Comparison with State-of-the-Art Methods

The performance comparison between TC-SSAD and current mainstream methods on ActivityNet-1.3 dataset [21] is shown in Table 3. It is not difficult to find that TC-SSAD does not perform well on ActivityNet-1.3 dataset.

Table 3. Comparison of mAP under different IoU thresholds with state-of-the-art methods in ActivityNet-1.3 dataset.

Methods	mAP(%)@IoU			
	0.5	0.75	0.95	Average
Singh *et al.* [22]	26.01	15.22	2.61	14.62
CDC [12]	45.30	26.00	0.20	23.80
TAG-D [9]	39.12	23.48	5.49	23.98
BSN [11]	52.50	33.53	8.85	33.72
GTAN [23]	**52.61**	**34.14**	**8.91**	**34.31**
TC-SSAD	33.61	20.71	2.32	20.42

The first reason is that the action time span of ActivityNet-1.3 dataset changes too much. Some action instances almost occupy the whole video time, while some action instances only take less than 1 s. Using the anchor of preset scale can not capture the actions with too many time spans; Second, due to large scale of ActicityNet-1.3 dataset, we directly use the two-stream feature provided by [11] instead of the sliding window method in Decouple-SSAD for feature extraction. The length of all video feature sequences provided is 100. After multi-layer backbone unit, the length of feature sequence is gradually shortened. Therefore, the receptive field of each time sequence position of the feature sequence output by the deep network layer corresponding to the original video will be too large, resulting in a significant decrease in the sensitivity to some short-term actions, especially for the case of original video with a long time.

For THUMOS14 dataset, Table 4 lists the mAP values of two-stage and single-stage methods under different IoU thresholds. In terms of mAP value, our method is superior to most mainstream temporal action detection methods on THUMOS14 dataset.

Table 4. Comparison of mAP under different IoU thresholds with state-of-the-art methods in THUMOS14 dataset.

Two-stage action detection					
Methods	mAP(%)@IoU				
	0.1	0.2	0.3	0.4	0.5
SCNN [6]	47.7	43.5	36.3	28.7	19.0
SST [24]	–	–	37.8	–	23.0

(*continued*)

Table 4. (*continued*)

Two-stage action detection					
Methods	mAP(%)@IoU				
	0.1	0.2	0.3	0.4	0.5
CDC [12]	–	–	40.1	29.4	23.3
TURN [8]	54.0	50.9	44.1	34.9	25.6
R-C3D [14]	54.5	51.5	44.8	35.6	28.9
SSN [10]	66.0	59.4	51.9	41.0	29.8
BSN [11]	–	–	53.5	45.0	36.9
TAL-Net [3]	59.8	57.1	53.2	48.5	42.8
P-GCN [25]	**69.5**	**67.8**	**63.6**	**57.8**	**49.1**
Single-stage action detection					
SMS [26]	51.0	45.2	36.5	27.8	17.8
SSAD [16]	50.1	47.8	43.0	35.0	24.6
SS-TAD [17]	-–	–	45.7	–	29.2
GTAN [23]	69.1	63.7	57.8	47.2	38.8
Decouple-SSAD [5]	66.4	65.1	60.9	53.4	43.7
TC-SSAD	**69.1**	**67.0**	**63.0**	**55.0**	**44.3**

5 Conclusion

In this paper, we propose a single-stage temporal action detection network TC-SSAD. By cascading the Timeception layer and super-event module, and the network can better adapt to the diversity of action time span in the video and effectively use the temporal structure and context information of the whole input video. The experimental results show that the mAP of TC-SSAD in THUMOS14 datasets is 44.3%, which is 2.4% higher than Decouple-SSAD network. In ActivityNet-1.3 dataset, the average mAP is 20.4%, better than the Decouple-SSAD network 0.61%.

However, the current research is still far from practical applications. In future work, we should consider how to improve the detection accuracy of these difficult action categories, how to build a lightweight end-to-end network and combine it with other video tasks such as video description.

Acknowledgements. This work was supported by National Natural Science Foundation of China (61877038, 61501287, 61902229) and Fundamental Research Funds for the Central Universities (No. TD2020044Y, No. GK201703058, No. GK202103084).

References

1. Ren, J., Yuan, L., Nurmi, P., et al.: Camel: Smart, adaptive energy optimization for mobile web interactions. In: Proceedings of the IEEE INFOCOM Conference on Computer Communications, pp. 119–128 (2020)

2. Qin, Q., Ren, J., Yu, J., et al.: To compress, or not to compress: characterizing deep learning model compression for embedded inference. In: 2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), pp. 729–736 (2018)
3. Y.-W., Chao, S., Vijayanarasimhan, B., et al.: Rethinking the faster R-CNN architecture for temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1130–1139 (2018)
4. Wang, J., Jiang, W., Ma, L., et al.: Bidirectional attentive fusion with context gating for dense video captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7190–7198 (2018)
5. Huang, Y., Dai, Q., Lu, Y.: Decoupling localization and classification in single shot temporal action detection. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 1288–1293 (2019)
6. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1049–1058 (2016)
7. Tran, D., Bourdev, L., Fergus, R., et al.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp. 4489–4497 (2015)
8. Gao, J., Yang, Z., Chen, K., et al.: Turn tap: temporal unit regression network for temporal action proposals. In: Proceedings of the IEEE international conference on computer vision, pp. 3628–3636 (2017)
9. Xiong, Y., Zhao, Y., Wang, L., et al.: A pursuit of temporal accuracy in general activity detection. arXiv preprint arXiv:1703.02716 (2017)
10. Zhao, Y., Xiong, Y., Wang, L., et al.: Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2914–2923 (2017)
11. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: boundary sensitive network for temporal action proposal generation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11208, pp. 3–21. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_1
12. Shou, Z., Chan, J., Zareian, A., et al.: CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5734–5743 (2017)
13. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
14. Xu, H., Das, A., Saenko, K.: R-c3d: region convolutional 3D network for temporal activity detection. In: Proceedings of the IEEE international conference on computer vision, pp. 5783–5792 (2017)
15. Idrees, H., Zamir, A.R., Jiang, Y.G., et al.: The THUMOS challenge on action recognition for videos "in the wild". Comput. Vis. Image Understand. **155**, 1–23 (2017)
16. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: Proceedings of the 25th ACM international conference on Multimedia, pp. 988–996 (2017)
17. Buch, S., Escorcia, V., Ghanem, B., et al.: End-to-end, single-stream temporal action detection in untrimmed videos. In: Procedings of the British Machine Vision Conference 2017. British Machine Vision Association, pp. 1–12 (2017)
18. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017)

19. Hussein, N., Gavves, E., Smeulders, A.W.M.: Timeception for complex action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 254–263 (2019)
20. Piergiovanni, A.J., Ryoo, M.S.: Learning latent super-events to detect multiple activities in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5304–5313 (2018)
21. Heilbron, F.C., Escorcia, V., Ghanem, B., et al.: Activitynet: a large-scale video benchmark for human activity understanding. In: Proceedings of the ieee conference on computer vision and pattern recognition, pp. 961–970 (2015)
22. Singh, B., Marks, T.K., Jones, M., et al.: 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, pp. 1961–1970 (2016)
23. Long, F., Yao, T., Qiu, Z., et al.: Gaussian temporal awareness networks for action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 344–353 (2019)
24. Buch, S., Escorcia, V., Shen, C., et al.: Sst: single-stream temporal action proposals. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 2911–2920 (2017)
25. Zeng, R., Huang, W., Tan, M., et al.: Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7094–7103 (2019)
26. Yuan, Z., Stroud, J.C., Lu, T., et al.: Temporal action localization by structured maximal sums. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3684–3692 (2017)