electronics

Article

Temporal Context Modeling Network with Local-Global Complementary Architecture for Temporal Proposal Generation

Yunfeng Yuan [1,2], Wenzhu Yang [1,2,*], Zifei Luo [1,2] and Ruru Gou [1,2]

[1] School of Cyber Security and Computer, Hebei University, Baoding 071002, China
[2] Hebei Machine Vision Engineering Research Center, Hebei University, Baoding 071002, China
* Correspondence: wenzhuyang@hbu.edu.cn; Tel.: +86-15720127565

Abstract: Temporal Action Proposal Generation (TAPG) is a promising but challenging task with a wide range of practical applications. Although state-of-the-art methods have made significant progress in TAPG, most ignore the impact of the temporal scales of action and lack the exploitation of effective boundary contexts. In this paper, we propose a simple but effective unified framework named Temporal Context Modeling Network (TCMNet) that generates temporal action proposals. TCMNet innovatively uses convolutional filters with different dilation rates to address the temporal scale issue. Specifically, TCMNet contains a BaseNet with dilated convolutions (DBNet), an Action Completeness Module (ACM), and a Temporal Boundary Generator (TBG). The DBNet aims to model temporal information. It handles input video features through different dilated convolutional layers and outputs a feature sequence as the input of ACM and TBG. The ACM aims to evaluate the confidence scores of densely distributed proposals. The TBG is designed to enrich the boundary context of an action instance. The TBG can generate action boundaries with high precision and high recall through a local–global complementary structure. We conduct comprehensive evaluations on two challenging video benchmarks: ActivityNet-1.3 and THUMOS14. Extensive experiments demonstrate the effectiveness of the proposed TCMNet on tasks of temporal action proposal generation and temporal action detection.

Keywords: temporal action proposal generation; temporal action detection; boundary context; action completeness module; temporal boundary generator

Citation: Yuan, Y.; Yang, W.; Luo, Z.; Gou, R. Temporal Context Modeling Network with Local-Global Complementary Architecture for Temporal Proposal Generation. *Electronics* **2022**, *11*, 2674. https://doi.org/10.3390/electronics11172674

Academic Editor: Silvia Liberata Ullo

Received: 20 July 2022
Accepted: 23 August 2022
Published: 26 August 2022

1. Introduction

Temporal action detection is one of the most fundamental tasks in video understanding, which is aimed at not only classifying every action instance in each video, but also looking for their accurate temporal locations. In general, the temporal action detection task is composed of two subtasks: temporal action proposal generation and action classification. Although current action recognition methods [1,2] can achieve convincing classification accuracy, the performance of temporal action detection is still unsatisfactory on mainstream benchmarks. Object detection aims to find as many tight bounding box locations and classes of objects as possible. With the continuous in-depth research of many works, a quite number of recent methods [3–5] have achieved remarkable progress and superior performance. Akin to object proposals for object detection in images, temporal action proposal has a crucial impact on the quality of action detection. As a result, more and more works are therefore devoted to improving the quality of temporal action proposals. Temporal Action Proposal Generation (TAPG) gradually became a research focus in video understanding tasks. TAPG is not only used for temporal action detection, but is also the core of several downstream tasks such as video recommendation, video captioning, and summarization.

Proposals generated by a robust TAPG method usually have two essential properties: (1) The generated temporal proposals should cover ground-truth action instances accurately

and exhaustively, and have flexible durations and accurate boundaries. (2) The generated temporal proposals should be precisely evaluated so that redundant proposals can be effectively suppressed. Existing TAPG methods can be roughly divided into two categories. The first category follows a top-down fashion. Such methods generate proposals by predefining sliding windows [6] or designing a set of regularly distributed anchors [7] of different scales for each video segment. The generated proposals are then evaluated by a binary classifier. However, as sliding windows and anchors are defined manually, the generated proposals are doomed to have imprecise boundaries. Under this circumstance, more and more researchers begin to study bottom-up TAPG methods. By using local clues on the video sequence to evaluate each temporal location, these types of approaches can generate more precise boundaries and more flexible durations.

Although recent methods have made significant progress in TAPG, they still have unresolved problems. (1) The duration of ground-truth action instances varies, typically ranging from seconds to minutes. However, existing methods use a fixed temporal receptive field for all action instances and thus ignore the temporal scale issue of action instances. (2) Most of the existing methods only exploit the local details around the boundaries to predict starting and ending time, but do not pay much attention to the global context in the video sequence. Figure 1 shows the diversity of ground-truth action instances' durations on two challenging video benchmarks: ActivityNet-1.3 and THUMOS14.

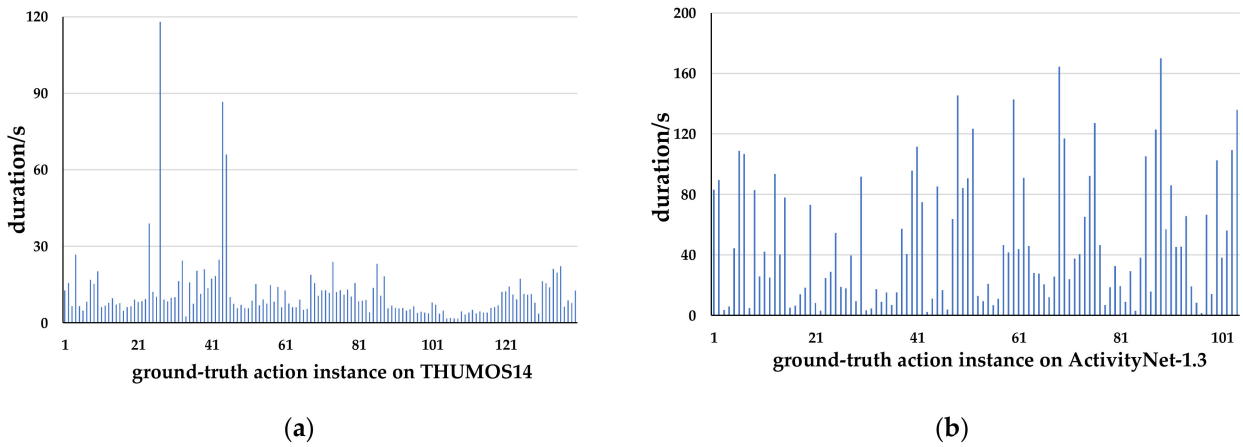

Figure 1. Diversity of a ground-truth action instance's durations on THUMOS14 and ActivityNet-1.3 benchmarks: (**a**) ground-truth action instance's durations on THUMOS14; (**b**) ground-truth action instance's durations on ActivityNet-1.3.

Motivated by the above observations, we propose a Temporal Context Modeling Network (TCMNet) to efficiently model video action instances with different durations and make full use of global context to generate more accurate temporal boundaries. In our framework, a BaseNet named DBNet takes the extracted video features as input and provides a shared feature sequence for subsequent modules. To efficiently model video action instances with different durations, DBNet contains multiple convolution layers with different dilated rates. These convolution filters have different receptive fields that are most effective at a specific temporal scale. An Action Completeness Module (ACM) is designed to take the shared feature sequence as input and generate action completeness maps to evaluate densely distributed proposals. A Temporal Boundary Generator (TBG) is designed to generate temporal boundaries with high precision and high recall. The TBG contains a local branch and a global branch. The local branch consists of only two temporal convolution layers. It focuses on the local abrupt background-to-action (action-to-background) change in the input feature sequence and generates rough boundaries with high recall but low precision. The global branch generates temporal boundaries with high precision but low recall by using our improved contextual U-shaped network structure. It uses multiple temporal convolutional layers followed by down-sampling steps

to extract semantic information from different granularities. To restore the resolution of the temporal feature sequence, the up-sampling operation is repeated multiple times and the features of the same resolution are fused. All in all, the contributions of our work can be summarized fourfold:

(1) We propose a Temporal Context Modeling Network (TCMNet) for temporal action proposal generation. TCMNet adopts multiple dilated temporal convolutions. Each of them is most effective for action instances with a specific duration, to obtain different receptive fields. The responses of all temporal convolutions are fused to generate more reliable temporal action proposals.
(2) To achieve the complete action proposal generation, we embed an improved U-shaped network in the temporal boundary generator. Therefore, TCMNet can improve performance by leveraging the global context for boundary detection through local–global structures.
(3) We propose a pooling operation to obtain more useful deep semantic information and an aggregation function to achieve adaptive fusion of semantic features. The pooling operation and the aggregation function are embedded in the U-shaped network to reduce the disturbance of noise.
(4) We conduct extensive experiments on the THUMOS14 and ActivityNet-1.3 benchmarks. The results show that TCMNet can achieve significant proposal generation performance. Combined with the existing action classifiers, TCMNet can also achieve remarkable temporal action detection performance compared with other approaches.

2. Related Works

2.1. Temporal Anchoring Methods

With the continuous development of deep learning networks [8,9], great progress has been made in video analysis tasks. Temporal action detection is a key part of video analysis tasks, and extracting high-quality temporal proposals is crucial for action detection. Temporal proposals have different time spans to align with action instances. However, fixed-size features must be extracted from each proposal to be fed to fully connected layers for proposal evaluation [10]. Bottom-up methods [11,12] first obtain the boundary candidates and then use 1D RoI pooling to estimate all possible combinations. Multi-scale anchor methods [13] extend image detection to temporal action localization. They generate class-agnostic proposals by jointly classifying and regressing a fixed set of multi-scale anchors at each location. Anchor-free methods [14] directly predict the confidence score, the center offset, and the length of time through the center point feature. Continuous representation [15] proposes modeling action segments by maximizing the confidence scores in a 2D function. It enables a more flexible and efficient data sampling space.

2.2. Action Recognition

Action recognition is a fundamental and important task in the video understanding area, and deep learning models have recently achieved significant performance promotion in the action recognition task. Ref. [16] uses human boxes and key points to represent instance-level features, and the action region features of this framework are used as the input of the temporal action head network, which makes the framework more discriminative. The author of [17] proposed a multi-scale feature extraction method used to extract richer feature information. At the same time, a multi-task learning model is introduced. It can further improve classification accuracy by sharing data among multiple tasks. Due to the continuous development of deep models in the field of action recognition, some works [18,19] begin to solve the difficult problems of deep models in real-life applications, so those deep models can be used in practice.

2.3. Temporal Action Proposal Generation

Temporal action proposal generation (TAPG) aims to generate proposals with precise temporal action boundaries and confidence in untrimmed videos. Existing proposal gener-

ation methods are mainly divided into two branches: top-down and bottom-up methods. Top-down methods mainly generate proposals based on sliding windows or uniformly distributed anchors, and then use a binary classifier to evaluate the generated proposals. SCNN [6] first uses sliding windows of different scales to generate some proposals with a fixed overlap rate. TURN [20] draws on the classic algorithm Faster R-CNN [21] in object detection. It generates proposals through uniformly distributed anchors. GTAN [22] introduces Gaussian kernels to dynamically optimize the temporal scale of each action proposal. Those methods are inspired by the achievements of anchor-based object detectors in still images; they discretize the proposal task into a classification task where multiple predefined anchors with different lengths are used as classes and a class that best fits the ground-truth action length is regarded as a ground-truth class for training.

As for the bottom-up methods [23–26], they generate proposals by locating temporal boundaries and then combining the boundaries in a certain strategy. TAG [27] designs a temporal watershed algorithm to generate proposals but lacks confidence scores for retrieval. On the basis of TAG, BSN [11] utilizes a temporal evaluation module to locate temporal boundaries and adopts a proposal evaluation module to regress the confidence of proposals. However, BSN is inefficient because it conducts proposal feature construction and confidence evaluation procedure for each proposal, respectively. To solve this problem, BMN [28] designs a boundary-matching (BM) mechanism for the confidence evaluation of densely distributed proposals. Bottom-Up-TAL [12] introduces two regularization terms to mutually regularize the learning procedure. Jointly optimizing these two terms, the entire framework is aware of potential constraints during an end-to-end optimization process. Considering that proposals generated by the methods using only local clues are susceptible to noise. TSI [29] leverages temporal context for boundary detection with the local–global-complementary structure to improve performance. TSI also designs a scale-invariant loss function to improve detection performance for short actions. RTD-Net [30] adopts Transformer architecture to directly generate action proposals in untrimmed videos. It models dependencies between proposals from a global perspective and avoids non-maximum suppression post-processing through simple but efficient design.

2.4. Temporal Action Detection

Temporal action detection can be divided into two types of methods. One is the one-stage method, which aims to localize an action and predict its class simultaneously. The other is the two-stage approach, which works by classifying proposals and detecting them. As one-stage methods, PBRNet [31] and AFSD [14] skip the proposal generation by directly detecting action instances in untrimmed videos. P-GCN [32] exploits the proposal–proposal relations for temporal action detection in videos. G-TAD [33] adaptively incorporates multi-level semantic context into video features and casts temporal action detection as a sub-graph localization problem to localize actions in video graphs. As for two-stage temporal action detection methods, TCANet [34] and RCL [15] adopt the progressive boundary refinement method to achieve precise boundaries and reliable confidence of proposals, thus improving the efficiency of action detection.

3. Methodology

3.1. Problem Definition

We are given an untrimmed video sequence $V = \{v_t\}_{t=1}^{l_v}$, where v_t denotes the t-th frame in the video sequence and l_v is the length of the video. The temporal annotation set corresponding to the video V is composed of a set of action instances $\psi_g = \{\varphi_{g,n} = (t_n^s, t_n^e)\}_{n=1}^{N_g}$, where N_g is the number of ground-truth action instances and t_n^s and t_n^e are the starting and ending time of action instance $\varphi_{g,n}$. TAPG aims to predict proposals $\psi_p = \{\varphi_{p,n} = (t_n^s, t_n^e, p_n)\}_{n=1}^{N_p}$ to cover ψ_g with high recall and high temporal overlap, where p_n is action completeness score of predicted proposal $\varphi_{p,n}$, and it will be further used for proposal ranking.

3.2. Feature Encoding

We employ two-stream networks to encode raw video sequence and generate a visual feature sequence. Specifically, given an untrimmed video V containing l_v frames, we can extract a visual feature sequence $F = \{f_i\}_{i=1}^{l_s}$ by concatenating the output of the last FC-layer in the two-stream networks, where l_s denote the length of visual feature F. Like previous works [11,24,28,29], we extract features at regular frame interval δ to reduce computational cost; thus $l_s = l_v/\delta$.

3.3. Temporal Context Modeling Network (TCMNet)

TCMNet is designed to generate densely distributed proposals directly in a unified network. It generates action completeness maps that represent the confidence of densely distributed proposals and local–global boundary probability sequences that represent boundary information simultaneously. The framework of TCMNet is illustrated in Figure 2, which contains three main modules: BaseNet with dilated convolutions (DBNet), Action Completeness Module (ACM) and Temporal Boundary Generator (TBG). DBNet can be seen as the backbone of TCMNet, which aims to handle the input video features through different dilated convolutional layers to better model the temporal information. It receives the video feature sequence as input and outputs a feature sequence as the input to ACM and TBG. ACM generates action completeness maps of dense proposals through Boundary-Matching (BM) layers proposed in BMN [28]. In addition, dilated convolutional layers are embedded in ACM to obtain different receptive fields. TBG contains a local branch and a global branch. The local branch focuses on local sudden changes in the input feature sequence and generates rough boundaries with high recall. The global branch extracts contextual features and generates high-precision boundaries through our improved U-shaped architecture.

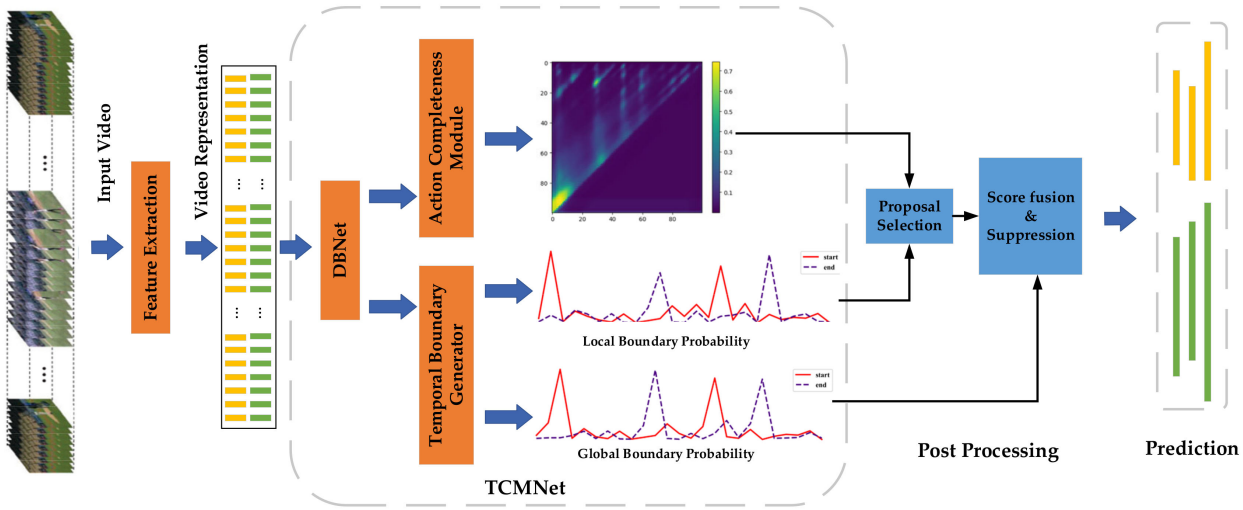

Figure 2. The framework of our method. TCMNet contains three main modules: BaseNet with dilated convolutions (DBNet) handles the extracted features for temporal relationship modeling. Action Completeness Module (ACM) evaluates the confidence of densely distributed proposals. Temporal Boundary Generator (TBG) generates high-recall and high-precision boundaries.

3.3.1. DBNet

In order to faithfully detect boundaries, each action instance in the video sequence needs to have the appropriate temporal receptive fields. However, the duration of different action instances in the video generally varies widely, so it is impossible to find a one-for-all temporal receptive field. As a natural solution, we embed a set of convolutional filters with different dilation rates in BaseNet and name it DBNet. The goal of DBNet is to receive the two-stream video feature sequence F as input and output a feature sequence $F_{BaseNet}$ shared

by ACM and TBG. As shown in Figure 3, we embed a dilated convolutional layer consisting of several different dilated convolutional filters after the traditional temporal convolution. The outputs from all dilated convolutions are simply averaged, returning fused contextual information. Note that a skip connection is inserted after the average operation, such that the dilated convolutions are reinforced to focus on learning the residual. This is written as

$$F_{BaseNet} = dc(conv2(\mathrm{dc}(conv1(F)))), \tag{1}$$

$$dc = \frac{1}{N_{conv}} \sum_{i=1}^{N_{conv}} conv_i(F) + F, \tag{2}$$

where $conv1$ and $conv2$ denote two traditional temporal convolutions and $dc(\cdot)$ denotes the dilated convolutional layer. By combining convolutions with different dilation rates, DBNet can better model the temporal relationship of action instances with different durations.

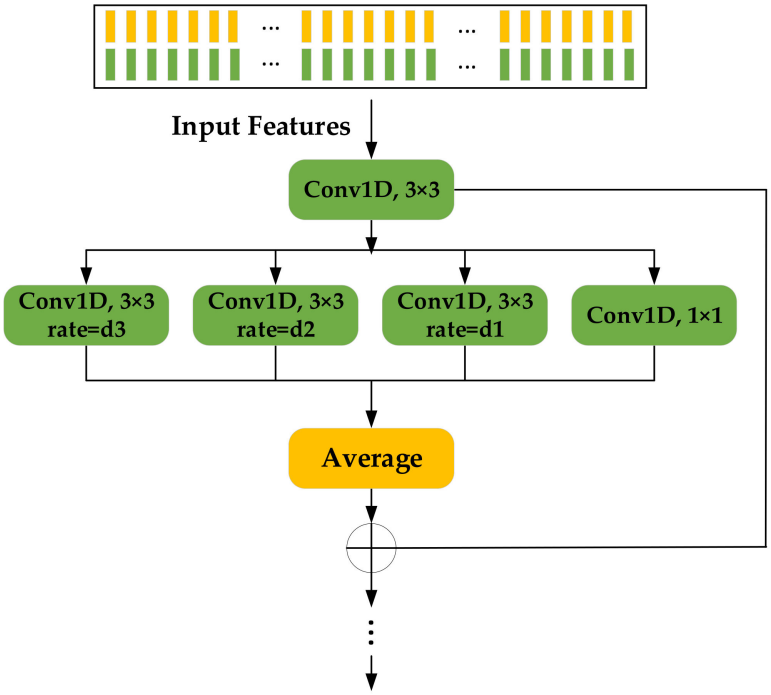

Figure 3. Architecture of DBNet. $d1$, $d2$ and $d3$ are the dilation rates of temporal 1D convolutions.

3.3.2. Action Completeness Module (ACM)

The ACM module receives the shared feature sequence generated by DBNet as input and outputs action completeness maps to evaluate the confidence score of dense proposals. To achieve this goal, we adopt the Boundary-Matching (BM) mechanism proposed in BMN [28]. As shown in Figure 4, the BM layer can transfer temporal feature sequence $F_{BaseNet} \in R^{C \times D}$ to proposal feature maps $M_F \in R^{D \times T \times 128 \times 32}$, where T is the length of the feature sequence and D is the maximum duration of proposals. The proposal feature maps are then fed into several 2D convolutional and dilated convolutional layers to generate new feature maps $M_F^{'} \in R^{D \times T \times 128}$. After going through the ACM module, each proposal is predicted as two confidence scores, which are supervised by the IoU classification loss and the IoU regression loss.

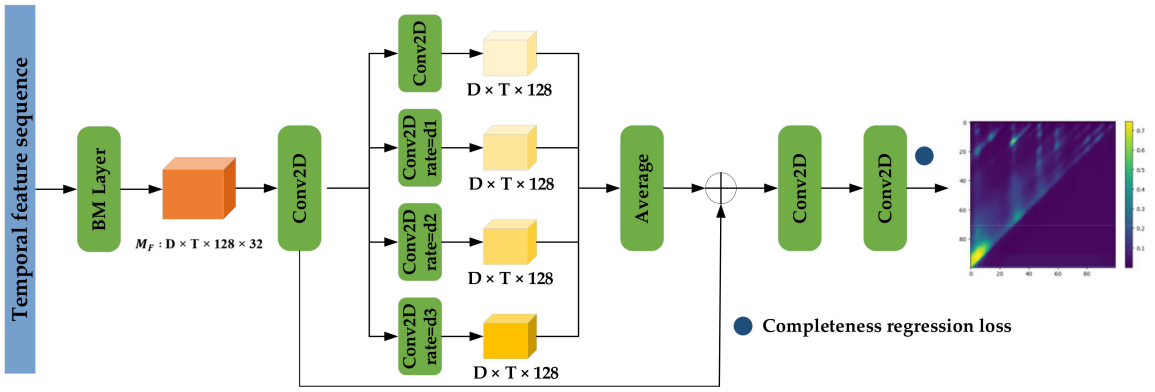

Figure 4. The architecture of Action Completeness Module (ACM). ACM evaluates the completeness of all densely distributed proposals.

3.3.3. Temporal Boundary Generator (TBG)

The goal of TBG is to accurately evaluate the start and end probabilities of all temporal locations in untrimmed videos. These boundary probability sequences are then used to generate proposals in the post-processing stage. Previous methods treat the boundary as a kind of local information but do not pay enough attention to global context or deep semantic features, which makes the detection of the boundary vulnerable to noise [35]. To remedy this defect, we follow the structural details of TSI [29] to accurately detect temporal boundaries through a local–global complementary architecture. The architecture of TBG is shown in Figure 5. The local branch in TBG contains only two temporal convolutional layers. This branch focuses on local abrupt changes and generates a rough boundary with high recall but low precision to cover all actual start/end points. Inspired by the UNet [36] used in image segmentation, the global branch is designed to represent the action boundary through a U-shaped contextual architecture. The global branch uses multiple temporal convolutional layers followed by down-sampling to extract semantic information from different granularities. In order to restore the resolution of the temporal feature sequence, the up-sampling operation is repeated multiple times, and the features of the same resolution are adaptively fused.

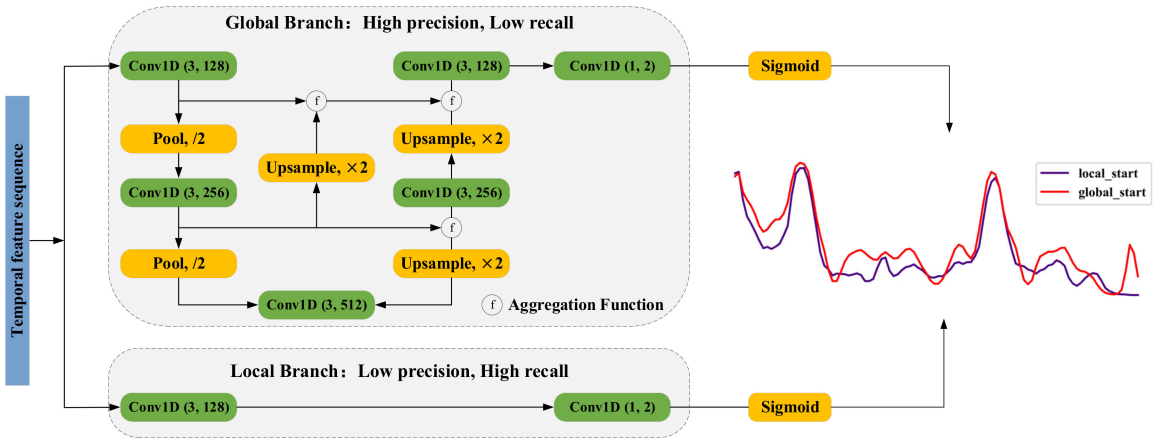

Figure 5. The architecture of Temporal Boundary Generator (TBG). TBG contains local and global branches to generate high-recall and high-precision boundaries. Pool stands for the down-sample operation and Ⓕ stands for aggregation function.

Unlike the TBD proposed in TSI [29], we argue that: (1) Deep semantic features obtained through temporal max pooling during down-sampling are not enough because the fine-grained temporal information critical for localizing boundaries is lost. Therefore, as shown in Figure 6a, we design a new pooling method called Pool, which uses both

average-pooling and max-pooling operations to generate two different temporal context descriptors. The two descriptors are then forwarded to the shared MLP to produce our deep semantic features, written as

$$F_{\max} = MaxPool(X),\ F_{\text{avg}} = AvgPool(X), \tag{3}$$

$$Pool = MLP(F_{\text{avg}}) + MLP(F_{\max}), \tag{4}$$

where + is element-wise addition. (2) Semantic features of different granularities contribute to boundary detection differently, so it is not the most appropriate way to concatenate semantic features directly. Therefore, as shown in Figure 6b, we design an aggregation function to achieve adaptive fusion of the same resolution features. Specifically, we first concatenate each input feature in the channel dimension to obtain the new feature

$$F_{TBG}^{upsample} = \left[f_1^{upsample} \parallel f_2^{upsample} \parallel \ldots \parallel f_n^{upsample}\right], \tag{5}$$

where $\parallel$ denotes concatenation and $f_1^{upsample}, \ldots, f_n^{upsample}$ are semantic features of different granularities. Then, we feed $F_{TBG}^{upsample}$ into squeeze-excitation architecture consisting of several temporal convolutions to explicitly model the channel relationship; the channel scaling factor in the squeeze-excitation architecture is denoted as r. Then, we normalize the output using the Softmax function to get the weight of semantic features with different granularity.

$$\alpha = [\alpha_1 \parallel \alpha_2 \parallel \ldots \parallel \alpha_n], \tag{6}$$

where $\alpha_1, \ldots, \alpha_n$ denote the weight of semantic features of different granularities. Finally, we can get the feature of adaptive fusion, written by $F_{TBG}^{upsample} = \sum_{i=1}^{n} (f_i^{upsample} \cdot \alpha_i)$.

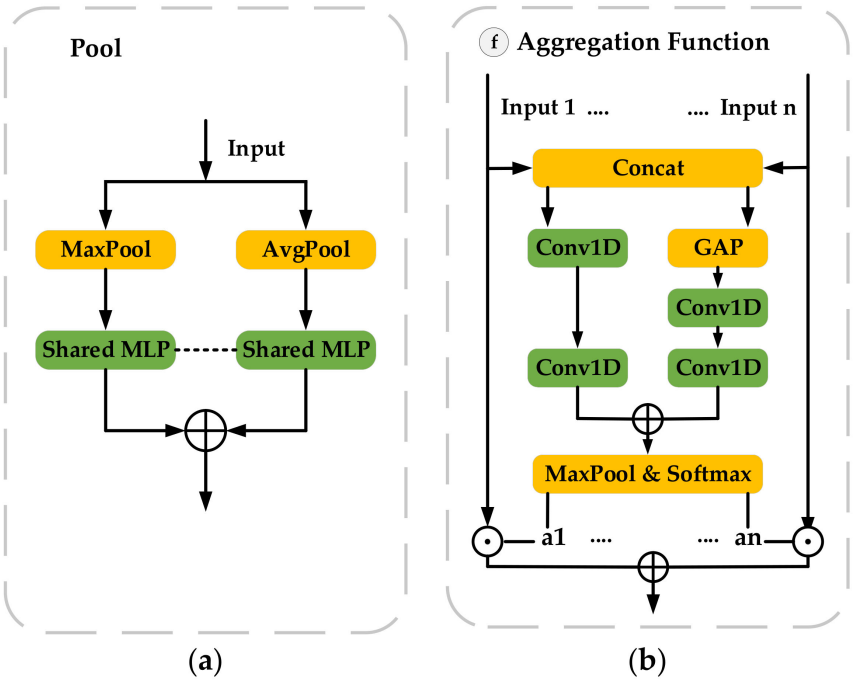

Figure 6. Details of the down-sample operation and aggregation function: (**a**) Architecture of Pool; (**b**) Architecture of Aggregation Function.

4. Training and Inference

4.1. Training of TCMNet

4.1.1. Label Assignment

For each action instance $\varphi_{g,n} = (t_n^s, t_n^e)$ in the annotation ψ_g, its starting region and ending region are defined as $r_s = [t_s - d/10, t_s + d/10]$ and $r_e = [t_e - d/10, t_e + d/10]$, respectively, where $d = t_e - t_s$ is the duration of $\varphi_{g,n}$. Then, by computing the maximum overlap ratio of each temporal interval with r_s, we can obtain $G_s = \left\{g_{t_n}^s\right\}$ as the starting label of TBG. The ending label $G_e = \left\{g_{t_n}^e\right\}$ can be obtained through the same label assignment process. For ACM, we follow the definition in BMN [28] to get the label of the action completeness map $G_c = \left\{g_{i,j}^c\right\}$.

4.1.2. Loss of ACM

ACM outputs action completeness map p_c with two channels. The training loss is defined as regression loss L_{reg} and binary classification loss L_{cls}, respectively:

$$L_{ACM} = L_{cls}(P_c^{cls}, G_c) + \beta \cdot L_{reg}(P_c^{reg}, G_c), \tag{7}$$

where L2 loss is adopted as L_{reg} and SI loss proposed in TSI [29] is adopted as L_{cls}.

4.1.3. Loss of TBG

TBG outputs the starting and ending probability sequence of local and global branches, denoted as P_l^s, P_l^e, P_g^s, P_g^e, respectively. We follow BMN [28] to adopt binary logistic loss L_{bl} as starting and ending losses to supervise the boundary prediction with G_s, G_e, denoted as

$$L_{TBG} = \frac{1}{2}(L_{bl}(P_l^s, G_s) + L_{bl}(P_l^e, G_e) + L_{bl}(P_g^s, G_s) + L_{bl}(P_g^e, G_e)), \tag{8}$$

4.1.4. The Training Objective of TCMNet

The multi-task loss function of TCMNet consists of TBG loss, ACM loss and the L2 regularization term, which is defined as:

$$L = L_{TBG} + \beta \cdot L_{ACM} + \lambda \cdot L_2(\theta), \tag{9}$$

where weight term β and λ are set to 1 and 0.0001 separately to ensure different modules are trained evenly, and $L_2(\theta)$ is the L2 regularization term.

4.2. Inference of TCMNet

4.2.1. Proposal Selection

To ensure the diversity of proposals and guarantee high recall, we only use the local starting and ending probability sequences of TBG for proposal selection. When temporal locations in the probability sequences satisfy (1) local peak of boundary probabilities or (2) probabilities higher than 0.5·max(P), these temporal locations are regarded as the starting and ending locations. Then, we match all starting and ending locations to generate redundant candidate proposals ψ_p.

4.2.2. Score Fusion and Proposal Suppression

To generate a more reliable confidence score, for each proposal φ, we multiply its boundary probability by the confidence score to generate the final confidence score p_f,

$$p_f = p_{start} \cdot p_{end} \cdot p_c, \tag{10}$$

where $p_{start} = \sqrt{p_l^s \cdot p_g^s}$ is the starting probability, $p_{end} = \sqrt{p_l^e \cdot p_g^e}$ is the ending probability and p_c is the action completeness score, which is the fusion of the classification score and the regression score, written by $p_c = p_c^{cls} \cdot p_c^{reg}$. Then, we need Soft-NMS [37] to suppress

redundant proposals to retrieve the final high-quality proposals. After the Soft-NMS step, we employ a confidence threshold to get the final sparse candidate proposals.

5. Experimental Results and Discussion

5.1. Datasets and Settings

5.1.1. ActivityNet-1.3 and THUMOS14

The ActivityNet-1.3 [38] dataset consists of 19,994 untrimmed videos with annotations for the action proposal task. The dataset has 200 action categories and is divided into training, validation and test sets by a ratio of 2:1:1. The THUMOS14 [39] dataset contains 200 annotated untrimmed validation videos with 20 action categories and 213 annotated untrimmed test videos with 20 action categories. We train TCMNet on the validation set and evaluate it on the test set.

5.1.2. Implementation Details

For video representation, we adopt two-stream networks TSN [40] and I3D [41] for feature encoding. During THUMOS14 feature extraction, the frame strides of I3D and TSN are set to 8 and 5, respectively. For ActivityNet-1.3, the sampling frame stride is 16. On ActivityNet-1.3, the feature sequence is rescaled to 100 by linear interpolation. On the THUMOS dataset, the length of the sliding window is set to 128 and the overlap ratio is set to 0.5. When training TCMNet, we use Adam for optimization. The batch size is set to 8. The learning rate is set to 0.001 for the first seven epochs, and we decay it to 0.0001 for the other two epochs.

5.2. Temporal Action Proposal Generation

Following previous works, we compute the average recall (AR) under different average numbers of proposals (AN) and the area under the AR versus the AN curve for each video, denoted by AR@AN and AUC, to evaluate the quality of generated proposals. We use tIoU thresholds set [0.5:0.05:0.95] on ActivityNet-1.3 and [0.5:0.05:1.0] on THUMOS14 [10].

5.2.1. Comparison with State-of-the-Art Methods on ActivityNet-1.3

Table 1 illustrates the performance of our proposal generation method compared with other state-of-the-art methods on the validation set of the ActivityNet-1.3 dataset. It should be pointed out that due to the limitations of experimental equipment, several TAPG methods (DBG, TSI) that we reimplemented on ActivityNet-1.3 did not achieve the results proposed in the original paper. As can be seen from the table, our TCMNet outperforms other state-of-the-art proposal generation methods. Specifically, the TCMNet outperforms BMN [28] with 0.92% and 1.07% in terms of AR@100 and AUC. In addition, TCMNet improves AUC from 67.93% to 68.17% on the validation set compared to TSI [29]. Additionally, when AN is one, our TCMNet significantly improves AR from 32.57% to 33.69% by 1.12%. It should be pointed out that action proposal generation focuses on the diversity of the retrieved proposals and judges the performance by the recall of top-N proposals, while the action detection task focuses on the accuracy of the top-N proposals. Therefore, some methods, such as DBG [23], can retrieve the actions with good diversity, but sacrifice the accuracy of top-N proposals, which leads to lower action detection performance. The results in Table 5 also prove this point, the performance of DBG on action detection is much lower than other methods.

5.2.2. Comparison with State-of-the-Art Methods on THUMOS14

We also compare the performance of our method with other state-of-the-art methods on the THUMOS14 dataset, as shown in Table 2. Due to the excellent performance achieved by I3D and TSN in action recognition tasks, we use them in our TCMNet to extract features. For a fair comparison, we also re-implement BMN [28] and TSI [29] using the same TSN and I3D features through publicly available code. As can be seen from the table, our method using TSN_GTAD or I3D_PGCN video features outperforms BMN [28] and TSI [29] significantly when the proposal number is set within [50,100,200,500,1000]. Specifically,

(1) based on the I3D_PGCN features, when the number of proposals varies from 50 to 1000, our method outperforms TSI by 2.09%, 1.63%, 1.58%, 1.18% and 0.93%. (2) Based on the TSN_GTAD features, when the number of proposals varies from 50 to 1000, our method outperforms TSI by 2.39%, 1.29%, 0.84%, 0.72% and 0.40%.

5.3. Ablation Study

In this section, we comprehensively evaluate our proposed TCMNet on the THUMOS14 dataset. We use I3D_PGCN feature as the visual feature sequence for ablation experiments.

5.3.1. Effectiveness and Efficiency of Modules in TCMNet

We conduct ablation studies using different architectural settings to verify the effectiveness and efficiency of each module proposed in TCMNet. The evaluation results shown in Table 3 indicate that: (1) Integrating convolutional filters with different dilation rates effectively achieves different temporal receptive fields optimized for specific-duration actions. (2) Unlike TSI [29] which employs max pooling for down-sampling, our proposed pooling operation for down-sampling can obtain fine-grained temporal information critical for localizing boundaries. (3) By further utilizing aggregation functions in TBG, deep semantic information of different granularities can be adaptively fused to reduce the impact of noise. (4) Finally, by integrating all the separated modules into an end-to-end framework, we can obtain competitive performance gains.

5.3.2. Study on Channel Scaling Factor *r* in TBG

Drawing on the idea in SENet [42], we explicitly model the weight of each feature channel through the squeeze-excitation architecture. We then use this weight to enhance useful channels and suppress channels that are not useful for boundary detection. The parameter r in the TBG module needs to be adjusted during the experiment, where the range of r is 1, 2, 4 and 8. In Table 4, we notice that without channel dimension reduction, the average recall (AR) under different average number of proposals (AN) drops severely, and AR@AN also drops as r exceeds a certain range. A reasonable explanation is that when the value of r is too large, the intermediate representation vector will lose key information, but when r is too small, the action-independent information contained in the intermediate representation vector will dominate. We finally adopt $r = 2$ by default for all experiments, with which we obtained the best results for AR@AN.

5.3.3. Effectiveness of Locating Actions with Different Durations

To further verify the effectiveness of locating actions with different durations, we follow the details of TSI [29] and conduct several ablation experiments, which are shown in Table 5. We divide the dataset into three groups according to the value of s (s stands for the scale of ground truth): small-scale actions that $0 \leq s < 0.06$, middle-scale actions in which $0.06 \leq s < 0.65$, and large-scale actions in which $0.65 \leq s \leq 1.0$. Each of these subsets has almost the same amount of ground truth to ensure fair comparisons. We then evaluate the methods on each sub-dataset. As can be seen from the table, TCMNet has a better performance on actions of different durations.

5.3.4. Visualization of Qualitative Results

We also visualize qualitative results. The top five proposal predictions of BMN [28] and TCMNet on the ActivityNet-1.3 dataset are shown in Figure 7 The demonstrated canoeing video has three ground-truth action instances. However, due to the excessive learning for long actions, BMN may regard two individual action instances as only one and predict more proposals with a long duration. Additionally, the temporal boundary of BMN is also not accurate enough because it only treats boundaries as local clues and does not pay enough attention to the global context. Compared with BMN, our proposed method can retrieve three action instances independently with higher overlap and more accurate boundaries.

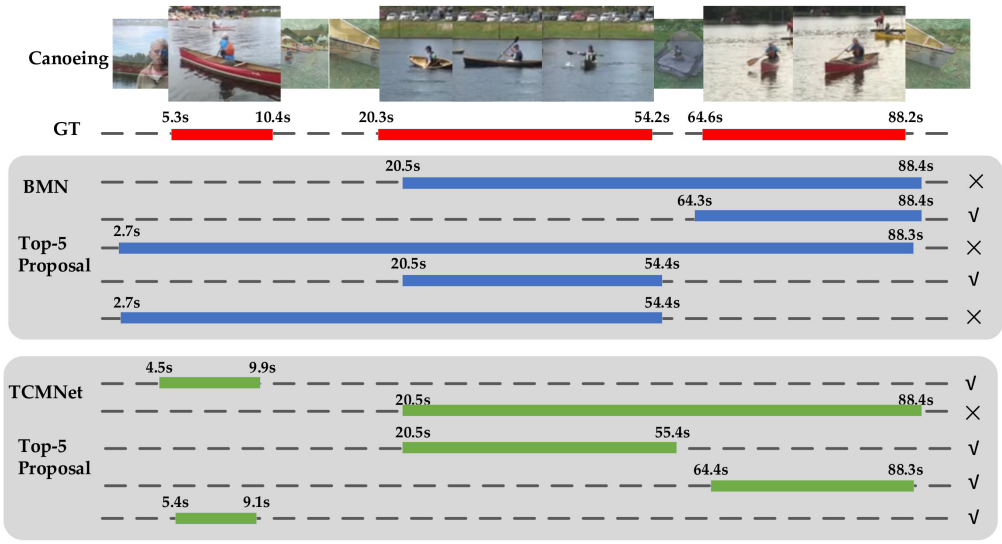

Figure 7. Qualitative results of BMN [28] and TCMNet on an example from the ActivityNet-1.3. The proposals shown are the top five predictions for corresponding ground truths.

5.4. Temporal Action Proposal Detection

In this section, we put the proposals into a temporal action detection framework to evaluate its detection performance. We adopt Mean Average Precision (mAP) as an evaluation metric for the temporal action detection task. On THUMOS14, mAP with tIoU thresholds set [0.3:0.1:0.7] are calculated. On ActivityNet-1.3, mAP with tIoU thresholds set {0.5,0.75,0.95} and average mAP with tIoU thresholds set [0.5:0.05:0.95] are reported [10].

For ActivityNet-1.3, we first use TCMNet to generate a set of action proposals for each video and keep the top 100 proposals for subsequent detection. Then, we adopt the top video-level classification result provided by CUHK [43] as the detection result. The experimental results are shown in Table 6; we can see that our method outperforms TSI by 1.53%, 1.42% and 0.94% when tIoU varies from 0.5 to 0.95 and achieves the mAP of 34.03%. Furthermore, compared to recent methods, TCMNet can achieve state-of-the-art results at tIoU = 0.5 and tIoU = 0.95.

For THUMOS14, we first use TCMNet to generate 200 temporal proposals per video. Then, we use the top two video-level classification results generated by UntrimmedNet [44] classifier to generate classification results for each proposal. As can be seen from Table 7, TCMNet achieves the best results at tIoU 0.6 (44.8%) and 0.7 (32.1%). Specifically, our TCMNet outperforms TSI by 3.8%, 4.9%, 4.9%, 5.2% and 4.4% when tIoU varies from 0.3 to 0.7. These results indicate that proposals generated by TCMNet are more accurate.

Table 1. Comparison between TCMNet and other state-of-the-art temporal action proposal generation methods on the validation set of ActivityNet-1.3 in terms of AR@AN and AUC. "re" denotes re-implementation by ourselves.

Method	BSN [11]	MGG [45]	BMN [28]	DBG (re) [23]	TSI (re) [29]	RTD-Net [30]	TCMNet
AR@1 (val) (%)	32.17	-	-	30.52	32.57	33.05	33.69
AR@100 (val) (%)	74.16	74.56	75.01	76.04	75.99	73.21	75.93
AUC (val) (%)	66.17	66.54	67.10	68.13	67.93	67.10	68.17

Table 2. Comparison between TCMNet and other state-of-the-art temporal action proposal generation methods on THUMOS14 in terms of AR@AN. Results with "*" are reported based on I3D_PGCN features and results with "^" are reported based on TSN_GTAD features.

Method	Feature	@50 (%)	@100 (%)	@200 (%)	@500 (%)	@1000 (%)
MGG [45]	2-Stream	39.93	47.75	54.65	61.36	64.06
BSN [11]	2-Stream	37.46	46.06	53.21	60.64	64.52
BMN [28]	2-Stream	39.36	47.72	54.70	62.07	65.49
BMN* [28]	I3D_PGCN	37.03	44.12	49.49	54.27	-
BMN^ [28]	TSN_GTAD	40.61	49.79	57.40	65.75	70.72
BSN++ [24]	2-Stream	42.44	49.84	57.61	65.17	66.83
TSI [29]	2-Stream	42.30	50.51	57.24	63.43	-
TSI* [29]	I3D_PGCN	39.12	47.79	55.02	63.88	67.81
TSI^ [29]	TSN_GTAD	40.93	50.23	57.88	66.46	71.95
TCANet [34]	2-Stream	42.05	50.48	57.13	63.61	66.88
RapNet [46]	2-Stream	40.35	48.23	54.29	61.41	64.47
RTD-Net* [30]	I3D_PGCN	41.52	49.32	56.41	62.91	-
ABN [47]	2-Stream	40.87	49.09	56.24	63.53	67.29
TCMNet *	I3D_PGCN	41.21	49.42	56.60	65.06	68.74
TCMNet ^	TSN_GTAD	43.32	51.52	58.72	67.18	72.35

Table 3. Ablation study on the performance of the proposed module on the THUMOS14 dataset, measured by AR@AN. "f" is the proposed feature aggregation function.

Module	@50 (%)	@100 (%)	@200 (%)	@500 (%)
TSI [29]	39.12	47.79	55.02	63.88
+DBNet	41.02	49.11	56.24	64.10
+ACM	39.64	48.24	55.28	63.86
+TBG (w/o f)	39.91	48.13	55.63	64.03
+TBG	40.35	48.80	56.11	64.07
+DBNet and ACM and TBG	41.21	49.42	56.60	65.06

Table 4. Analysis of hyperparameter settings on THUMOS14 dataset, measured by AR@AN. r is the channel scaling factor in TBG.

Scaling Factor r	@50 (%)	@100 (%)	@200 (%)	@500 (%)
1	40.66	48.79	55.83	64.22
2	41.21	49.42	56.60	65.06
4	40.92	49.29	56.42	64.47
8	40.89	49.19	56.26	64.12

Table 5. Effectiveness of locating actions with different durations on the ActivityNet-1.3 validation set. s stands for the scale of ground truth.

Method	AUC (%)	$0.0 \leq s < 0.06$ (%)	$0.06 \leq s < 0.65$ (%)	$0.65 \leq s \leq 1.0$ (%)
BMN [28]	67.10	36.53	70.43	94.48
DBG [23]	68.13	39.07	72.18	93.08
TSI [29]	67.93	39.25	71.06	94.59
TCMNet	68.17	40.24	71.55	94.71

Table 6. Temporal action detection results on the validation set of the ActivityNet-1.3 in terms of mAP at different tIoU thresholds. "re" denotes that this method is re-implemented by ourselves.

Method	0.5 (%)	0.75 (%)	0.95 (%)	Average (%)
BSN [11]	46.45	29.96	8.02	30.03
BMN [28]	50.07	34.78	8.29	33.85
DBG [23]	42.59	26.24	6.56	29.72
TSI (re) [29]	49.32	32.84	8.40	32.64
P-GCN [32]	48.26	33.16	3.27	31.11
G-TAD [33]	50.36	35.02	9.02	34.09
RTD-Net [30]	47.21	30.68	8.61	30.83
TCMNet	50.85	34.26	9.34	34.03

Table 7. Temporal action detection results on the test set of THUMOS14 in terms of mAP at different tIoU thresholds. "re" denotes that this method is re-implemented by ourselves.

Method	0.3 (%)	0.4 (%)	0.5 (%)	0.6 (%)	0.7 (%)
BMN [28]	56.0	47.4	38.8	29.7	20.5
P-GCN [32]	60.1	54.3	45.5	33.5	19.8
PBRNet [31]	58.5	54.6	51.3	41.8	29.5
TSI (re) [29]	63.6	57.7	49.9	39.6	27.7
PcmNet [25]	61.5	55.4	47.2	37.5	27.3
TCANet [34]	60.6	53.2	44.6	36.8	26.7
AFSD [14]	67.3	62.4	55.5	43.7	31.1
ABN [47]	59.9	54.0	46.1	37.0	25.6
RCL [15]	70.1	62.3	52.9	42.7	30.7
TCMNet	67.4	62.6	54.8	44.8	32.1

6. Conclusions

In this paper, we proposed a Temporal Context Modeling Network (TCMNet) for generating temporal action proposals. TCMNet effectively achieved different temporal receptive fields optimized for specific-duration actions by embedding convolutional layers containing different dilation rates. To predict precise action boundaries, the Temporal Boundary Generator (TBG) module improved the local–global complementary architecture in TSI. TBG obtained useful deep semantic information by embedding the proposed pooling operation and achieved an adaptive fusion of semantic features through an aggregation function to reduce noise disturbance. Extensive experiments on ActivityNet-1.3 and THUMOS14 datasets demonstrated the effectiveness of our method in terms of temporal action proposal and detection performance. In the beginning, we considered that the contextual information exploited in previous work was often characterized by the similarity between frames (or proposals) at the semantic feature level, without taking into account the temporal location contextual interactions between frames (or proposals). Temporal location contextual interactions are valuable prior knowledge. Therefore, we tried to embed position encoding in the temporal action proposal generation framework, but we did not achieve the desired effect. A possible reason is that fixed sinusoidal position encoding can only provide relative distance information without direction. In future work, we will try to augment feature representations with directed temporal positional encoding for more precise localization of actions.

Author Contributions: Conceptualization, Y.Y. and W.Y.; methodology, Y.Y. and W.Y.; software, Y.Y. and R.G.; validation, Y.Y.; formal analysis, W.Y. and Y.Y.; investigation, Z.L. and R.G.; resources, Y.Y. and R.G.; data curation, Y.Y. and Z.L.; writing—original draft preparation, Y.Y. and W.Y.; writing—review and editing, W.Y., Z.L. and R.G.; visualization, Y.Y.; supervision, W.Y. and Z.L.; project administration, Y.Y. and W.Y.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Hebei Province under Grant F2022201003 and the Post-graduate's Innovation Fund Project of Hebei University under Grant HBU2022ss037.

Acknowledgments: We thank the machine vision lab in Hebei University for the equipment and other help offered.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dos Santos, L.L.; Winkler, I.; Nascimento, E.G.S.J.E. RL-SSI Model: Adapting a Supervised Learning Approach to a Semi-Supervised Approach for Human Action Recognition. *Electronics* **2022**, *11*, 1471. [CrossRef]
2. Tweit, N.; Obaidat, M.A.; Rawashdeh, M.; Bsoul, A.K.; Al Zamil, M.G.J.E. A Novel Feature-Selection Method for Human Activity Recognition in Videos. *Electronics* **2022**, *11*, 732. [CrossRef]
3. Fu, R.; He, J.; Liu, G.; Li, W.; Mao, J.; He, M.; Lin, Y. Fast Seismic Landslide Detection Based on Improved Mask R-CNN. *Remote Sens.* **2022**, *14*, 3928. [CrossRef]
4. Akshatha, K.R.; Karunakar, A.K.; Shenoy, S.B.; Pai, A.K.; Nagaraj, N.H.; Rohatgi, S.S. Human Detection in Aerial Thermal Images Using Faster R-CNN and SSD Algorithms. *Electronics* **2022**, *11*, 1151. [CrossRef]
5. Lee, D.; Kim, J.; Jung, K. Improving object detection quality by incorporating global contexts via self-attention. *Electronics* **2021**, *10*, 90. [CrossRef]
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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1049–1058.
7. Lin, T.; Zhao, X.; Shou, Z. Single shot temporal action detection. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 988–996.
8. Abadía-Heredia, R.; López-Martín, M.; Carro, B.; Arribas, J.I.; Pérez, J.M.; Le Clainche, S. A predictive hybrid reduced order model based on proper orthogonal decomposition combined with deep learning architectures. *Expert Syst. Appl.* **2022**, *187*, 115910. [CrossRef]
9. Lopez-Martin, M.; Le Clainche, S.; Carro, B. Model-free short-term fluid dynamics estimator with a deep 3D-convolutional neural network. *Expert Syst. Appl.* **2021**, *177*, 114924. [CrossRef]
10. Vahdani, E.; Tian, Y. Deep learning-based action detection in untrimmed videos: A survey. *arXiv* **2021**, arXiv:2110.00111. [CrossRef] [PubMed]
11. Lin, T.; Zhao, X.; Su, H.; Wang, C.; Yang, M. BSN: Boundary sensitive network for temporal action proposal generation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
12. Zhao, P.; Xie, L.; Ju, C.; Zhang, Y.; Wang, Y.; Tian, Q. Bottom-up temporal action localization with mutual regularization. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 539–555.
13. 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, Venice, Italy, 22–29 October 2017; pp. 5783–5792.
14. Lin, C.; Xu, C.; Luo, D.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; Fu, Y. Learning salient boundary feature for anchor-free temporal action localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 3320–3329.
15. Wang, Q.; Zhang, Y.; Zheng, Y.; Pan, P. RCL: Recurrent Continuous Localization for Temporal Action Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2022; pp. 13566–13575.
16. Lee, I.; Kim, D.; Wee, D.; Lee, S. An efficient human instance-guided framework for video action recognition. *Sensors* **2021**, *21*, 8309. [CrossRef] [PubMed]
17. Xu, Y.; Zhou, F.; Wang, L.; Peng, W.; Zhang, K. Optimization of Action Recognition Model Based on Multi-Task Learning and Boundary Gradient. *Electronics* **2021**, *10*, 2380. [CrossRef]
18. Silva, V.; Soares, F.; Leão, C.P.; Esteves, J.S.; Vercelli, G. Skeleton driven action recognition using an image-based spatial-temporal representation and convolution neural network. *Sensors* **2021**, *21*, 4342. [CrossRef] [PubMed]
19. Habib, S.; Hussain, A.; Albattah, W.; Islam, M.; Khan, S.; Khan, R.U.; Khan, K. Abnormal Activity Recognition from Surveillance Videos Using Convolutional Neural Network. *Sensors* **2021**, *21*, 8291. [CrossRef] [PubMed]
20. Gao, J.; Yang, Z.; Chen, K.; Sun, C.; Nevatia, R. Turn tap: Temporal unit regression network for temporal action proposals. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 4 August 2017; pp. 3628–3636.
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
22. Long, F.; Yao, T.; Qiu, Z.; Tian, X.; Luo, J.; Mei, T. Gaussian temporal awareness networks for action localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 344–353.
23. Lin, C.; Li, J.; Wang, Y.; Tai, Y.; Luo, D.; Cui, Z.; Wang, C.; Li, J.; Huang, F.; Ji, R. Fast learning of temporal action proposal via dense boundary generator. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11499–11506.

24. Su, H.; Gan, W.; Wu, W.; Qiao, Y.; Yan, J. BSN++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 2–9 February 2021; pp. 2602–2610.
25. Qin, X.; Zhao, H.; Lin, G.; Zeng, H.; Xu, S.; Li, X.J.a.p.a. PcmNet: Position-Sensitive Context Modeling Network for Temporal Action Localization. *arXiv* **2021**, arXiv:2103.05270. [CrossRef]
26. Wang, H.; Damen, D.; Mirmehdi, M.; Perrett, T. TVNet: Temporal Voting Network for Action Localization. *arXiv* **2022**, arXiv:2201.00434.
27. Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; Lin, D. Temporal action detection with structured segment networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2914–2923.
28. Lin, T.; Liu, X.; Li, X.; Ding, E.; Wen, S. Bmn: Boundary-matching network for temporal action proposal generation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 23 July 2019; pp. 3889–3898.
29. Liu, S.; Zhao, X.; Su, H.; Hu, Z. TSI: Temporal scale invariant network for action proposal generation. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November–December 2020.
30. Tan, J.; Tang, J.; Wang, L.; Wu, G. Relaxed transformer decoders for direct action proposal generation. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 13526–13535.
31. Liu, Q.; Wang, Z. Progressive boundary refinement network for temporal action detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11612–11619.
32. Zeng, R.; Huang, W.; Tan, M.; Rong, Y.; Zhao, P.; Huang, J.; Gan, C. Graph convolutional networks for temporal action localization. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 23 July 2019; pp. 7094–7103.
33. Xu, M.; Zhao, C.; Rojas, D.S.; Thabet, A.; Ghanem, B. G-tad: Sub-graph localization for temporal action detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10156–10165.
34. Qing, Z.; Su, H.; Gan, W.; Wang, D.; Wu, W.; Wang, X.; Qiao, Y.; Yan, J.; Gao, C.; Sang, N. Temporal context aggregation network for temporal action proposal refinement. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 485–494.
35. Zhu, Z.; Tang, W.; Wang, L.; Zheng, N.; Hua, G. Enriching local and global contexts for temporal action localization. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 13516–13525.
36. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
37. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS–improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5561–5569.
38. Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; Carlos Niebles, J. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 961–970.
39. Idrees, H.; Zamir, A.R.; Jiang, Y.-G.; Gorban, A.; Laptev, I.; Sukthankar, R.; Shah, M.J.C.V.; Understanding, I. The THUMOS challenge on action recognition for videos "in the wild". *Comput. Vis. Image Underst.* **2017**, *155*, 1–23. [CrossRef]
40. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Gool, L.V. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 20–36.
41. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6299–6308.
42. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
43. Zhao, Y.; Zhang, B.; Wu, Z.; Yang, S.; Zhou, L.; Yan, S.; Wang, L.; Xiong, Y.; Lin, D.; Qiao, Y.; et al. Cuhk & ethz & siat submission to activitynet challenge 2017. *arXiv* **2017**, arXiv:1710.08011.
44. Wang, L.; Xiong, Y.; Lin, D.; Van Gool, L. Untrimmednets for weakly supervised action recognition and detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4325–4334.
45. Liu, Y.; Ma, L.; Zhang, Y.; Liu, W.; Chang, S.-F. Multi-granularity generator for temporal action proposal. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3604–3613.
46. Gao, J.; Shi, Z.; Wang, G.; Li, J.; Yuan, Y.; Ge, S.; Zhou, X. Accurate temporal action proposal generation with relation-aware pyramid network. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 10810–10817.
47. Vo, K.; Yamazaki, K.; Truong, S.; Tran, M.-T.; Sugimoto, A.; Le, N.J.I.A. ABN: Agent-aware boundary networks for temporal action proposal generation. *IEEE Access* **2021**, *9*, 126431–126445. [CrossRef]