Multi-granularity Generator for Temporal Action Proposal

Yuan Liu[♯*] Lin Ma[♮†] Yifeng Zhang[♯†] Wei Liu[♮] Shih-Fu Chang[§]
[♮]Tencent AI Lab [♯]Southeast University [§]Columbia University
{lhy19930911,forest.linma}@gmail.com yfz@seu.edu.cn {wl2223,sc250}@columbia.edu

Abstract

Temporal action proposal generation is an important task, aiming to localize the video segments containing human actions in an untrimmed video. In this paper, we propose a multi-granularity generator (MGG) to perform the temporal action proposal from different granularity perspectives, relying on the video visual features equipped with the position embedding information. First, we propose to use a bilinear matching model to exploit the rich local information within the video sequence. Afterwards, two components, namely segment proposal producer (SPP) and frame actionness producer (FAP), are combined to perform the task of temporal action proposal at two distinct granularities. SPP considers the whole video in the form of feature pyramid and generates segment proposals from one coarse perspective, while FAP carries out a finer actionness evaluation for each video frame. Our proposed MGG can be trained in an end-to-end fashion. By temporally adjusting the segment proposals with fine-grained frame actionness information, MGG achieves the superior performance over state-of-the-art methods on the public THUMOS-14 and ActivityNet-1.3 datasets. Moreover, we employ existing action classifiers to perform the classification of the proposals generated by MGG, leading to significant improvements compared against the competing methods for the video detection task.

1. Introduction

Temporal action proposal [10, 14] aims at capturing video temporal intervals that are likely to contain an action in an untrimmed video. This task plays an important role in video analysis and can thus be applied in many areas, such as action recognition [3, 19–21], summarization [45,47], grounding [6,7] and captioning [39,40]. Many methods [13,43] have been proposed to handle this task, and have shown that, akin to object proposals for object detection [30], temporal action proposal has a crucial impact on the quality of action detection.

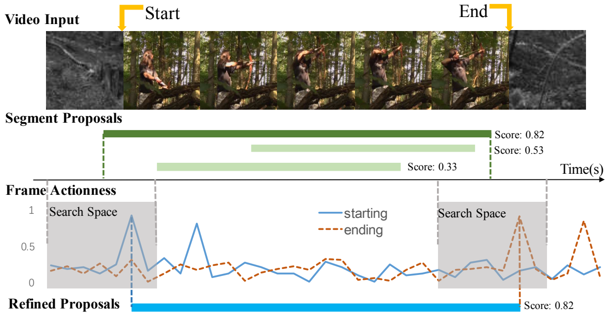


Figure 1: Our proposed MGG can generate segment proposals and frame actionness simultaneously, which helps discover information about possible actions at both the coarse and fine levels. By temporally adjusting the boundaries of the segment within the search space determined by the computed frame actionness, MGG can yield refined action proposals with both high recall and precision.

High-quality action proposal methods should capture temporal action instances with both high recall and high temporal overlapping with ground-truths, meanwhile producing proposals without many false alarms. One type of existing methods focuses on generating segment proposals [14, 35], where the initial segments are regularly distributed or manually defined over the video sequence. A binary classier is thereafter trained to evaluate the confidence scores of the segments. Such methods are able to generate proposals of various temporal spans. However, since the segments are regularly distributed or manually defined, the generated proposals naturally have imprecise boundary information, even though boundary regressors are further applied. Another thread of work, like [33, 43, 50], tackles the action proposal task in the form of evaluating frame actionness. These methods densely evaluate the confidence score for each frame and group consecutive frames together as candidate proposals. The whole video sequence is analyzed at a finer level, in contrast with the segment proposal based methods. As a result, the boundaries of the generated

[*] This work was done while Yuan Liu was a Research Intern with Tencent AI Lab.
[†] Corresponding authors.

proposals are of high precision. However, such methods often produce low confidence scores for long video segments, resulting in misses of true action segments and thus low recalls.

Obviously, these two types of methods are complementary to each other. Boundary sensitive network (BSN) [26] adopts a "local to global" scheme for action proposal, which locally detects the boudary information and globally ranks the candidate proposals. Complementary temporal action proposal (CTAP) [13] consists of three stages, which are initial proposal generation, complementary proposal collection, and boundary adjustment and proposal ranking, respectively. However, both of these two methods are multi-stage models with the modules in different stages trained independently, without overall optimization of the models. Another drawback is the neglect of the temporal position information, which conveys the temporal ordering information of the video sequence and is thereby expected to be helpful for precisely localizing the proposal boundary.

In order to address the aforementioned drawbacks, we propose a multi-granularity generator (MGG) by taking full advantage of both segment proposal and frame actionness based methods. At the beginning, the frame position embedding, realized with cosine and sine functions of different wavelengths, is combined with the video frame features. The combined features are then fed to MGG to perform the temporal action proposal. Specifically, a bilinear matching model is first proposed to exploit the rich local information of the video sequence. Afterwards, two components, namely segment proposal producer (SPP) and frame actionness producer (FAP), are coupling together and responsible for generating coarse segment proposals and evaluating fine frame actionness, respectively. SPP uses a U-shape architecture with lateral connections to generate candidate proposals of different temporal spans with high recall. For FAP, we densely evaluate the probabilities of each frame being the starting point, ending point, and inside a correct proposal (middle point). During the inference, MGG can further temporally adjust the segment boundaries with respect to the frame actionness information as shown in Fig. 1, and consequently produce refined action proposals.

In summary, the main contributions of our work are fourfold:

- We propose an end-to-end multi-granularity generator (MGG) for temporal action proposal, using a novel representation integrating video features and the position embedding information. MGG simultaneously generates coarse segment proposals by perceiving the whole video sequence, and predicts the frame actionness by densely evaluating each video frame.

- A bilinear matching model is proposed to exploit the rich local information within the video sequence, which is thereafter harnessed by the following SPP and FAP.

- SPP is realized in a U-shape architecture with lateral connections, capturing temporal proposals of various spans with high recall, while FAP evaluates the probabilities of each frame being the stating point, ending point, and middle point.

- Through temporally adjusting the segment proposal boundaries using the complementary information in the frame actionness, our proposed MGG achieves the state-of-the-art performances on the THUMOS-14 and ActivityNet-1.3 datasets for the temporal action proposal task.

2. Related Work

A large number of existing approaches have been proposed to tackle the problem of temporal action detection [24, 31, 34, 48, 49, 51]. Inspired by the success of two-stage detectors like RCNN [17], many recent methods adopt a proposal-plus-classification framework [5, 9, 33, 44], where classifiers are applied on a smaller number of class agnostic segment proposals for detection. The proposal stage and classification stage can be trained separately [33, 35, 51] or jointly [5, 44], and demonstrate very competitive results. Regarding temporal action proposal, DAP [10] and SST [1] introduce RNNs to process video sequences in a single pass. However, LSTM [18] and GRU [8] fail to handle video segments with long time spans. Alternatively, [9,35,41] directly generate proposals from sliding windows. R-C3D [44] and TAL-Net [5] follow the Faster R-CNN [30] paradigm to predict locations of temporal proposals and the corresponding categories. These methods perceive the whole videos in a coarser level, while the pre-defined temporal intervals may limit the accuracy of generated proposals. Methods like temporal action grouping (TAG) [43] and CDC [33] produce final proposals by densely giving evaluation to each frame. Analyzing videos in a finer level, the generated proposals are quite accurate in boundaries. In our work, MGG tackles the problem of temporal action proposal in both coarse and fine perspectives, being better at both recall and overlapping.

3. Our Approach

Given an untrimmed video sequence $\mathbf{s} = \{s_n\}_{n=1}^{l_s}$ with its length as l_s, temporal action proposal aims at detecting action instances $\varphi_p = \{\xi_n = [t_{s,n}, t_{e,n}]\}_{n=1}^{M_s}$, where M_s is the total number of action instances, and $[t_{s,n}, t_{e,n}]$ denote the starting and ending points of an action instance ξ_n, respectively.

We propose one novel neural network, namely MGG shown in Fig. 2, which analyzes the video and performs

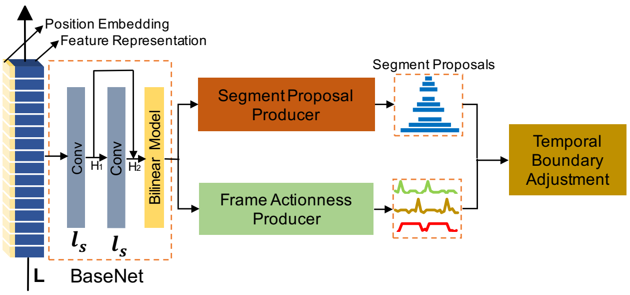


Figure 2: The architecture of our proposed MGG. The video visual features are first combined with the position embedding information to form the video representations. The proposed BaseNet relies on a blinear model to exploit the rich local information within the sequential video representations. Segment proposal producer (SPP) is realized by using a U-shape architecture with lateral connections to generate proposals of different temporal lengths, while frame actionness producer (FAP) evaluates each frame whether it is the starting point, ending point, or middle point. With the temporal boundary adjustment (TBA) module, boundaries of the segment proposals are temporally adjusted based on computed frame actionness, and the refined accurate action proposals are therefore generated.

temporal action proposal at different granularities. Specifically, our proposed MGG consists of four components. The video visual features are first combined with the position embedding information to yield the video representations. The subsequent BaseNet relies on a blinear model to exploit the rich local information within the sequential video representations. Afterwards, SPP and FAP are used to produce the action proposals from the coarse (segment) and fine (frame) perspectives, respectively. Finally, the temporal boundary adjustment (TBA) module adjusts the segment proposal boundaries regarding the frame actionness and therefore generates action proposals of both high recall and precision.

3.1. Video Representation

First, we need to encode the video sequence and generate the corresponding representations. Same as the previous work [13, 26], one convolutional neural network (CNN) is used to convert one video sequence $\mathbf{s} = \{s_n\}_{n=1}^{l_s}$ into one visual feature sequence $\mathbf{f} = \{f_n\}_{n=1}^{l_s}$ with $f_n \in R^{d_f}$. d_f is the dimension of each feature representation. However, the temporal ordering information of the video sequence is not considered. Inspired by [15, 38], we embed the position information to explicitly characterize the ordering information of each visual feature, which is believed to benefit the action proposal generation. The position information of the n-th ($n \in [1, l_s]$) visual feature f_n is embedded into a feature p_n with a dimension d_p by computing cosine and sine functions of different wavelengths:

$$\begin{aligned} p_n(2i) &= \sin(n/10000^{2i/d_p}), \\ p_n(2i+1) &= \cos(n/10000^{2i/d_p}), \end{aligned} \tag{1}$$

where i is the index of the dimension. The generated position embedding p_n will be equipped with the visual feature representation f_n via concatenation, denoted by $l_n = [f_n, p_n]$. As such, the final video representations $L = \{l_n\}_{n=1}^{l_s} \in R^{l_s \times d_l}$ are obtained, where $d_l = d_f + d_p$ denotes the dimension of the fused representations.

3.2. BaseNet

Based on the video representations, we propose a novel BaseNet to exploit the rich local behaviors within the video sequence. As shown in Fig. 2, two temporal convolutional layers are first stacked to exploit video temporal relationships. A typical temporal convolutional layer is denoted as $\mathrm{Conv}(n_f, n_k, \Omega)$, where n_f, n_k, and Ω are filter numbers, kernel size, and activation function, respectively. In our proposed BaseNet, the two convolutional layers are of the same architecture, specifically $\mathrm{Conv}(d_h, k, \mathrm{ReLU})$, where d_h is set to 512, k is set to 5, and ReLU refers to the activation of rectified linear units [29]. The outputs of these two temporal convolutional layers are denoted as H_1 and H_2, respectively.

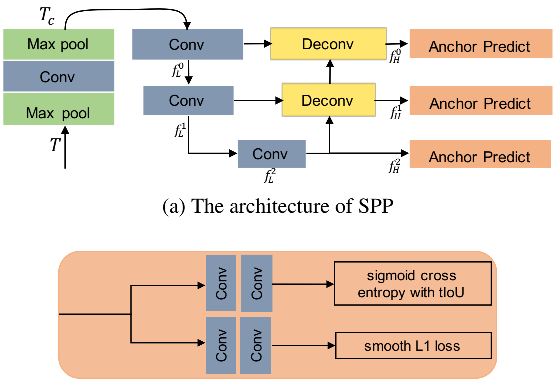
The intermediate representations H_1 and H_2 express the semantic information of the video sequence at different levels, which are rich in characterizing the local information. We propose a bilinear matching model [28] to capture the interaction behaviors between H_1 and H_2. Due to a large number of parameters contained in a traditional bilinear matching model, which result in an increased computational complexity and a higher convergence difficulty, we turn to pursue a factorized bilinear matching model [11, 23]:

$$\begin{aligned} \hat{H}_1^n &= H_1^n W_i + b_i, \\ \hat{H}_2^n &= H_2^n W_i + b_i, \\ T_i^n &= \hat{H}_1^n \hat{H}_2^{n\top}, \end{aligned} \tag{2}$$

where $H_1^n \in R^{1 \times d_h}$ and $H_2^n \in R^{1 \times d_h}$ denote the corresponding representations at the n-th location of H_1 and H_2, respectively. $W_i \in R^{d_h \times g}$ and $b_i \in R^{1 \times g}$ are the parameters to be learned, with g denoting a hyperparameter and being much smaller than d_h. Due to the smaller value of g, fewer parameters are introduced, which are easier for training. As such, the matching video representations $T = [T^1, .., T^{l_s}]$, with $T^n = [T_1^n, T_2^n, .., T_{d_h}^n]$ denoting the n-th feature, is obtained and used as the input to the following SPP and FAP for proposal generation.

3.3. Segment Proposal Producer

Due to large variations of action duration, capturing proposals of different temporal lengths with high recall is a big



(a) The architecture of SPP

(b) The anchor predict module

Figure 3: (a) Overview of SPP with pyramid levels $M = 3$. With a U-shape architecture and lateral connections, the generated feature pyramid F_H is helpful for capturing proposals with different temporal durations. (b) The anchor predict module has two branches which are used for classification and boundary regression, respectively.

challenge. Xu *et al.* [44] used one feature map to locate proposals of various temporal spans, yielding low average recall. SSAD [24] and TAL-Net [5] use a feature pyramid network, with each layer being responsible for proposal localization with specific time spans. However, each pyramid layer, especially the lower ones being unaware of high-level semantic information, is unable to localize temporal proposals accurately. To deal with this issue, we adopt a U-shape architecture with lateral connections between the convolutional and deconvolutional layers, as shown in Fig. 3.

With yielded matching video representations T as input, SPP first stacks three layers, specifically one temporal convolutional layer and two max-pooling layers, to reduce the temporal dimension and hence increase the size of the receptive field accordingly. As a result, the temporal feature T_c with temporal dimension $l_s/8$ is taken as the input of the U-shape architecture.

Same as the previous work, such as Unet [32], FPN [27], and DSSD [12], our U-shape architecture also consists of a contracting path and an expansive path as well as the lateral connections. Regarding the contracting path, with repeated temporal convolutions with stride 2 for downsampling, the feature pyramid (FP) $F_L = \{f_L^{(0)}, f_L^{(1)}, ...f_L^{(M-1)}\}$ is obtained, where $f_L^{(n)}$ is the n-th level feature map of F_L with temporal dimension $\frac{l_s}{8*2^n}$. M denotes the total number of pyramid levels. For the expansive path, temporal deconvolutions are adopted on multiple layers with an upscaling factor of 2. Via lateral connections, high-level features from the expansive path are combined with the corresponding low-level features, with the fused features denoted as $f_H^{(n)}$. Repeating this operation, the fused feature pyramid is defined as $F_H = \{f_H^{(0)}, f_H^{(1)}, ...f_H^{(M-1)}\}$. Different levels of feature pyramids are of different receptive fields, which are responsible for locating proposals of different temporal spans.

A set of anchors are regularly distributed over each level of feature pyramid F_H, based on which segment proposals are produced. As shown in Fig. 3, each f_H is followed by two branches, with each branch realized by stacking two layers of temporal convolutions. Specifically, one branch is the classification module to predict the probability of a ground-truth proposal being present at each temporal location for each of the ρ anchors, where ρ is the number of anchors per location of the feature pyramid. The other branch is the boundary regression module to yield the relative offset between the anchor and the ground-truth proposal.

3.4. Frame Actionness Producer

Based on the yielded matching video representations T, the frame actionness producer (FAP) is proposed to evaluate the actionness of each frame. Specifically, three two-layer temporal convolutional networks are used to generate the starting point, ending point, and middle point probabilities for each frame, respectively. Please note that two-layer temporal convolutional networks share the same configuration, where the first one is defined as $\text{Conv}(d_f, k, \text{ReLU})$ and the second one is $\text{Conv}(1, k, \text{Sigmoid})$. d_f is set to 64, while k, as the kernel size, is set to 3. And their weights are not shared. As a result, we obtain three probability sequences, namely the starting probability sequence $P_s = \{p_n^s\}_{n=1}^{l_s}$, the ending probability sequence $P_e = \{p_n^e\}_{n=1}^{l_s}$, and the middle probability sequence $P_m = \{p_n^m\}_{n=1}^{l_s}$, with p_n^s, p_n^e, and p_n^m denoting the starting, ending, and middle probabilities of the n-th feature, respectively. Compared with the generated segment proposals by SPP, the frame actionness yielded by FAP densely evaluates each frame in a finer manner.

4. Training and Inference

In this section, we will first introduce how to train our proposed MGG network, which can subsequently generate segment proposals and frame actionness. During the inference, we propose one novel fusion strategy by temporally adjusting the segment boundary information with respect to the frame actionness.

4.1. Training

As introduced in Sec. 3, our proposed MGG considers both the SPP and FAP together with a shared BaseNet. During the training process, these three components cooperate with each other and are jointly trained in an end-to-end fashion. Specifically, the objective function of our proposed

MGG is defined as:

$$L_{MGG} = L_{SPP} + \beta L_{FAP}, \tag{3}$$

where L_{SPP} and L_{FAP} are the objective functions defined for SPP and FAP, respectively. β is a parameter to adjust their relative contributions, which is empirically set to 0.1. Detailed information about L_{SPP} and L_{FAP} will be introduced in what follows.

4.1.1 SPP Training

Our proposed SPP produces a set of anchor segments for each level of the fused feature pyramids F_H. We first introduce how to assign labels to the corresponding anchor segments. Subsequently, the objective function by referring to the assigned labels is introduced.

Label Assignment. Same as Faster RCNN [30], we assign a binary class label to each anchor segment. A positive label is assigned if it overlaps with some ground-truth proposals with temporal Intersection-over-Union (tIoU) higher than 0.7, or has the highest tIoU with a ground-truth proposal. Anchors are regarded as negative if the maximum tIoU with all ground-truth proposals is lower than 0.3. Anchors that are neither positive nor negative are filtered out. To ease the issue of class imbalance, we sample the positive and negative examples with a ratio of 1:1 for training.

Objective Function. As shown in Fig. 3 (b), we perform a multi-task training for SPP, which not only predicts the actionness of each anchor segment but also regresses its boundary information. For actionness prediction, the cross-entropy function is used, while the smooth L_1 loss function, as introduced in [16], is used for boundary regression. Specifically, the objective function is defined as:

$$\begin{aligned} L_{SPP} =& \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \\ & \gamma \frac{1}{N_{reg}} \sum_i [p_i^* \geqslant 1] L_{reg}(W_i, W_i^*), \end{aligned} \tag{4}$$

where γ is a trade-off parameter, which is set to 0.001 empirically. N_{cls} is the total number of training examples. p_i stands for the yielded score. p_i^* is the label, 1 for positive samples and 0 for negative samples. L_{cls} is the cross-entropy loss function between p_i and p_i^*. The smooth L_1 loss function L_{reg} is activated only when the ground-truth label p_i^* is positive, and disabled otherwise. N_{reg} is the number of training examples whose p_i^* is positive. $W_i = \{t_c, t_l\}$ represents the predicted relative offsets of anchor segments. $W_i^* = \{t_c^*, t_l^*\}$ indicates the relative offsets between ground-truth proposals and the anchors, which can be computed:

$$\begin{cases} t_c^* = & (c_i^* - c_i)/l_i, \\ t_l^* = & log(l_i^*/l^i), \end{cases} \tag{5}$$

where c_i and l_i indicate the center and length of anchor segments, respectively. c_i^* and l_i^* represent the center and length of the ground-truth action instances.

4.1.2 FAP Training

FAP takes the matching video representations with their length as l_s as input and outputs three probability sequences, namely the starting probability sequence $P_s = \{p_n^s\}_{n=1}^{l_s}$, the ending probability sequence $P_e = \{p_n^e\}_{n=1}^{l_s}$, and the middle probability sequence $P_m = \{p_n^m\}_{n=1}^{l_s}$.

Label Assignment. The ground-truth annotations of temporal action proposals are denoted as $\pi = \{\psi_n = [t_{s,n}, t_{e,n}]\}_{n=1}^{M_a}$, where M_a is the total number of annotations. For each action instance $\psi_n \in \pi$, we define the starting, ending, and middle regions as $[t_{s,n} - d_{d,n}/\eta, t_{s,n} + d_{d,n}/\eta]$, $[t_{e,n} - d_{d,n}/\eta, t_{e,n} + d_{d,n}/\eta]$, and $[t_{s,n}, t_{e,n}]$, respectively, where $d_{d,n} = t_{e,n} - t_{s,n}$ is the duration of the annotated action instance and η is set to 10 empirically. For each visual feature, if it lies in the starting, ending, or middle regions of any action instances, its corresponding starting, ending, or middle label will be set to 1, otherwise 0. In this way, we obtain the ground-truth label for the three sequences, which are denoted as $G_s = \{g_n^s\}_{n=1}^{l_s}$, $G_e = \{g_n^e\}_{n=1}^{l_s}$, and $G_m = \{g_n^m\}_{n=1}^{l_s}$, respectively.

Objective Function. Given the predicted probability sequences and ground-truth labels, the objective function for FAP is defined as:

$$L_{FAP}^{all} = \lambda_s L_{FAP}^s + \lambda_e L_{FAP}^e + \lambda_m L_{FAP}^m. \tag{6}$$

The cross-entropy loss function is used for calculating all the three losses L_{FAP}^s, L_{FAP}^e, and L_{FAP}^m, where a weighting factor set by an inverse class frequency is introduced to address class imbalance. L_{FAP}^{all} is the sum of the starting loss L_{FAP}^s, ending loss L_{FAP}^e, and middle loss L_{FAP}^m, where λ_s, λ_e, and λ_m are the weights specifying the relative importance of each part. In our experiments, we set $\lambda_s = \lambda_e = \lambda_m = 1$.

4.2. Inference

As aforementioned, SPP aims to locate segment proposals of various temporal spans, thus yielding segment proposals with inaccurate boundary information. On the contrary, FAP gives an evaluation of each video frame in a finer level, which makes it sensitive to boundaries of action proposals. Obviously, SPP and FAP are complementary to each other. Therefore, during the inference phase, we propose the temporal boundary adjustment (TBA) module realized in a two-stage fusion strategy to improve the boundary accuracy of segment proposals with respect to the frame actionness.

Stage I. We first use non-maximum suppression (NMS) to post-process the segment-level action instances detected by SPP. The generated results are denoted as $\varphi_p = \{\xi_n = [t_{s,n}, t_{e,n}]\}_{n=1}^{M_s}$, where M_s is the total number of the detected action instances, and $t_{s,n}$ and $t_{e,n}$ denote the corresponding starting and ending times of an action instance ξ_n, respectively. We will adjust $t_{s,n}$ and $t_{e,n}$ by referring to the starting and ending scores detected in FAP. Firstly, we set two context regions ξ_n^s and ξ_n^e, which are named as the searching space:

$$\begin{aligned} \xi_n^s &= [t_{s,n} - d_{d,n}/\varepsilon, t_{s,n} + d_{d,n}/\varepsilon], \\ \xi_n^e &= [t_{e,n} - d_{d,n}/\varepsilon, t_{e,n} + d_{d,n}/\varepsilon], \end{aligned} \tag{7}$$

where $d_{d,n} = t_{e,n} - t_{s,n}$ is the duration of ξ_n. ε which controls the size of the searching space is set to 5 . The max starting score and the corresponding time in the region of ξ_n^s are defined as c_n^s and $t_{s,n}^{max}$, respectively , and the max ending score and the corresponding time in the region of ξ_n^e are defined as c_n^e and $t_{e,n}^{max}$, respectively. If c_n^s or c_n^e is higher than a threshold $\sigma \in [0, 1]$, which is set manually for each specific dataset, we adjust the starting or ending point of ξ_n with a weighting factor δ to control the contribution of $t_{s,n}^{max}$ and $t_{e,n}^{max}$ and yield the refined action instance $\xi_n^\star$. As such, the new segment-level action instance set is refined to be $\varphi_p^\star = \{\xi_n^\star\}_{n=1}^{M_s}$.

Stage II. The middle probability sequence illustrates the probability of each frame whether it is inside one action proposal or not. We use the grouping scheme similar to TAG [43] to group the consecutive frames with high middle probability into regions as the candidate action instances. Such generated action instances are denoted by $\varphi_{tag} = \{\phi_n\}_{n=1}^{M_t}$ with M_t indicating the total number of grouped action instances. We propose to make a further position adjustment by considering both φ_{tag} and $\varphi_p^\star$. Specifically, for each action instance $\xi_n^\star$ in $\varphi_p^\star$, its tIoU with all the action instances in φ_{tag} are computed. If the maximum tIoU is higher than 0.8, the boundaries of $\xi_n^\star$ will be replaced by the corresponding action instance ϕ_n in φ_{tag}. Via such an operation, the substituted proposals are sensitive to boundaries and the overall boundary accuracy is improved accordingly.

5. Experiments

5.1. Datasets

THUMOS-14 [22]. It includes 1,010 videos and 1,574 videos with 20 action classes in the validation and test sets, respectively. There are 200 and 212 videos with temporal annotations of actions labeled in the validation and testing sets, respectively. We conduct the experiments on the same public split as [13, 43].

ActivityNet-1.3 [2]. The whole dataset consists of 19,994 videos with 200 classes annotated, with 50% for training, 25% for validation, and the rest 25% for testing. We train our model on the training set and perform evaluations on the validation and testing sets, respectively.

Table 1: Performance comparisons with DAPs [10], SCNN-prop [35], SST [1], TURN [14], BSN [26], TAG [43], and CTAP [13] on THUMOS-14 in terms of AR@AN.

Feature	Method	@50	@100	@200	@500	@1000
Flow	TURN	21.86	31.89	43.02	57.63	64.17
2-Stream	TAG	18.55	29.00	39.61	-	-
2-stream	CTAP	32.49	42.61	51.97	-	-
2-Stream	BSN+NMS	35.41	43.55	52.23	61.35	**65.10**
2-Stream	**MGG**	**39.93**	**47.75**	**54.65**	**61.36**	64.06
C3D	DAPs	13.56	23.83	33.96	49.29	57.64
C3D	SCNN-prop	17.22	26.17	37.01	51.57	58.20
C3D	SST	19.90	28.36	37.90	51.58	60.27
C3D	TURN	19.63	27.96	38.34	53.52	60.75
C3D	BSN+NMS	27.19	35.38	43.61	53.77	59.50
C3D	**MGG**	**29.11**	**36.31**	**44.32**	**54.95**	**60.98**

5.2. Temporal Proposal Generation

In this section, we compare our proposed MGG against the existing state-of-the-art methods on both THUMOS-14 and ActivityNet-1.3 datasets.

For temporal action proposal, Average Recall (AR) computed with different tIoUs is usually adopted for performance evaluation. Following traditional practice, tIoU thresholds set from 0.5 to 0.95 with a step size of 0.05 are used on ActivityNet-1.3, while tIoU thresholds set from 0.5 to 1.0 with a step size of 0.05 are used on THUMOS-14. We also measure AR with different Average Numbers (ANs) of proposals, denoted as AR@AN. Moreover, the area under the AR-AN curve (AUC) is also used as one metric on ActivityNet-1.3, where AN ranges from 0 to 100.

Table 1 illustrates the performance comparisons on the testing set of THUMOS-14. Different feature representations will significantly affect the performances. As such, we adopt the two-stream [36] and C3D [37] features for fair comparisons. Taking the two-stream features as input, the AR@AN performances are consistently improved for AN ranging from 50 to 500, while BSN+NMS achieves a better performance with AN equal to 1000. While the C3D features are adopted, the AR@AN of MGG is higher than those of the other methods, with AN ranging from 50 to 1000. Such experiments clearly indicate the effectiveness of MGG in temporal proposal generation.

Furthermore, Fig. 4 illustrates the AR-AN and recall@100-tIoU curves of different models on the testing split of THUMOS-14. It can be observed that our proposed MGG outperforms the other methods in terms of AR-AN curves. Specifically, when AN equals 40, MGG significantly improves the performance from 33.02% to 37.01%. For recall@100-tIoU, MGG gains a significantly higher recall when tIoU ranges from 0.5 to 1, indicating high accuracy of our proposal results.

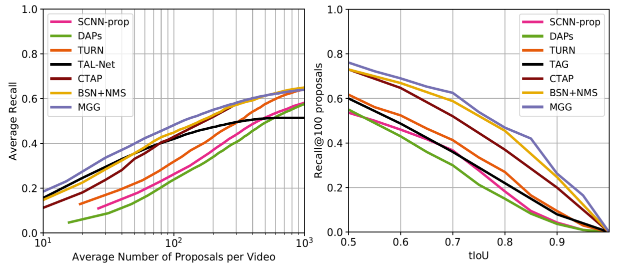


Figure 4: AR-AN and recall@AN=100 curves of different temporal action proposal methods on the testing set of THUMOS-14.

Table 2: Performance comparisons with TCN [9], MSRA [46], Prop-SSAD [25], CTAP [13], and BSN [26] on the validation and testing splits of ActivityNet-1.3.

Method	TCN	MSRA	Prop-SSAD	CTAP	BSN	**MGG**
AUC (val)	59.58	63.12	64.40	65.72	66.17	**66.43**
AUC (test)	61.56	64.18	64.80	-	66.26	**66.47**
AR@100	-	-	73.01	73.17	74.16	**74.54**

Table 2 illustrates the performance comparisons on the ActivityNet-1.3 dataset, where a two-stream Inflated 3D ConvNet (I3D) model [4] is used to extract features. Specifically, we compare our proposed MGG with the state-of-the-art methods, namely TCN [9], MSRA [46], Prop-SSAD [25], CTAP [13], and BSN [26], in terms of AUC and AR@100. It can be observed that the proposed MGG outperforms the other methods on both the validation and testing sets. Specifically, MGG improves AR@100 on the validation set from 74.16 of the state-of-the-art method BSN to 74.54.

Fig. 5 illustrates some qualitative results of the generated proposals by MGG on ActivityNet-1.3 and THUMOS-14. Each is composed of a sequence of frames sampled from a full video. By analyzing videos from both coarse and fine perspectives, MGG generates the refined proposals, with high overlapping with ground-truth proposals.

5.3. Ablation Study

In this subsection, the effect of each component in MGG is studied in detail. We ablate the studies on the validation set of ActivityNet-1.3. Specifically, in order to verify the component effectiveness of MGG: position embedding, bilinear matching, U-shape architecture in SPP, FAP, and SPP, we perform the ablation studies as follows:

MGG-P: We discard the position information of the input video sequence and directly feed the visual feature representations into MGG.

MGG-B: We discard the bilinear matching model which exploits the interactions between the two temporal convolutions within BaseNet, and instead feed the output of the second convolutional layer to the following SPP and FAP.

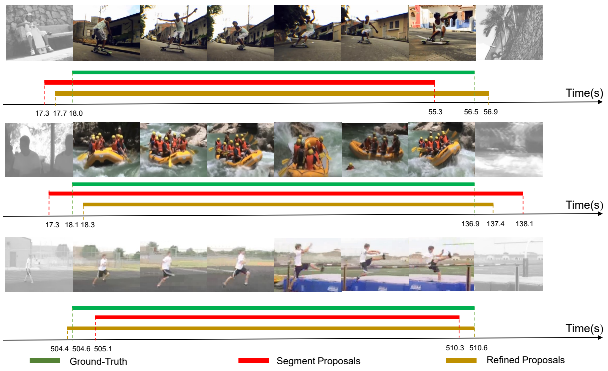


Figure 5: Qualitative results of the proposals generated by MGG on ActivityNet-1.3 (top and middle) and THUMOS-14 (bottom). It can be observed that the boundary information of the segment proposals generated by SPP is further adjusted using FAP, resulting in more precise proposals.

Table 3: Ablation studies on the validation set of ActivityNet-1.3 in terms of AUC and AR@AN.

Method	AUC (val)	@30	@50	@80	@100
MGG-P	65.59	65.21	69.93	72.88	73.92
MGG-B	65.88	65.56	70.41	73.19	73.89
MGG-U	65.02	64.85	69.41	72.95	73.71
MGG-F	64.31	63.76	67.91	71.04	72.24
MGG-S	59.91	59.53	63.05	67.18	68.96
MGG	**66.43**	**66.21**	**70.97**	**73.87**	**74.54**

MGG-U: We discard the U-shape architecture which is proposed in SPP to increase semantic information of the lower layers. Correspondingly, only the expansive path of the feature pyramid is used.

MGG-F: We only consider SPP to generate the final proposals, without considering FAP and the following TBA module.

MGG-S: We only consider FAP to generate the final proposals, without considering SPP and the following TBA module.

As shown in Table 3, our full model MGG outperforms all its variants, namely MGG-P, MGG-B, MGG-U, MGG-F, and MGG-S, which verifies the effectiveness of the components. In order to examine the detailed effectiveness of the U-shape architecture, we compare the recall rate of generated proposals in different lengths. As shown in Table 4, the recall rate of short proposals drops dramatically, when the U-shape architecture is removed. The reason is that the U-shape architecture transfers higher semantic information to the lower layers, which can perceive global information of the video sequence, and is thus helpful for capturing proposals with short temporal extents.

Table 4: Recall rates of MGG-U and MGG on generated proposals of different temporal extents on the validation set of ActivityNet-1.3, where AN and tIoU thresholds are set to 100 and 0.75, respectively.

Method	0-5s	5-10s	10-15s	15-20s	25-30s	35-40s	40-45s
MGG-U	0.15	0.63	0.73	0.80	0.91	0.93	0.94
MGG	**0.21**	**0.73**	**0.82**	**0.90**	0.93	0.93	0.92

Table 5: Performance comparisons of the two-stage TBA on the validation set of ActivityNet-1.3 in both end-to-end training and stagewise training manners.

	Stagewise			End-to-end		
MGG-F	✓	✓	✓	✓	✓	✓
Stage I		✓	✓		✓	✓
Stage II			✓			✓
AUC(val)	64.12	65.40	66.28	64.31	65.54	66.43
AR@100	72.05	73.41	74.19	72.24	73.48	74.54

Moreover, it can be observed that MGG-F and MGG-S both perform inferiorly to our full MGG. The main reason is that SPP and FAP generate proposals at different granularities. Our proposed TBA can exploit their complementary behaviors and fuse them together to produce proposals with more precise boundary information. As introduced in Sec. 4.2, TBA performs in two stages:

Stage I: The starting and ending probability sequences generated by FAP are used to adjust boundaries of segment proposals from SPP.

Stage II: The middle probability sequence is grouped into proposals with the method similar to [43] and gives a final adjustment to boundaries of proposals from Stage I.

Table 5 illustrates the effectiveness of each stage in TBA. It can be observed that the two stages of TBA can both refine boundaries of segment proposals, thus consistently improving the performances, with AUC increasing from 64.31% to 66.43%.

Training: Stagewise v.s. End-to-end. MGG is designed to jointly optimize SPP and FAP in an end-to-end fashion. It is also possible to train SPP and FAP separately, in which they do not work together. Such a training scheme is referred to as the stagewise training. Table 5 illustrates the performance comparisons between end-to-end training and stagewise training. It can be observed that models trained in an end-to-end fashion can outperform those learned with stagewise training under the same settings. It clearly demonstrates the importance of jointly optimizing SPP and FAP with BaseNet as a shared block to provide intermediate video representations.

Table 6: Performance comparisons between MGG and the other proposal generation methods in terms of video detection on the testing set of THUMOS-14, where mAP is reported with tIoU set from 0.3 to 0.7.

Proposal Method	Classifier	0.7	0.6	0.5	0.4	0.3
SST [1]	SCNN-cls	-	-	23.0	-	-
TURN [14]	SCNN-cls	7.7	14.6	25.6	34.9	44.1
CTAP [13]	SCNN-cls	-	-	26.9	-	-
BSN [26]	SCNN-cls	15.0	22.4	29.4	36.6	43.1
MGG	SCNN-cls	**15.8**	**23.6**	**29.9**	**37.8**	**44.9**
SST [1]	UNet	4.7	10.9	20.0	31.5	41.2
TURN [14]	UNet	6.3	14.1	24.5	35.3	46.3
BSN [26]	UNet	20.0	28.4	36.9	45.0	53.5
MGG	UNet	**21.3**	**29.5**	**37.4**	**46.8**	**53.9**

5.4. Action Detection

In order to further examine the quality of generated proposals by MGG, we feed the detected proposals into the state-of-the-art action classifiers, including SCNN [35] and UntrimmedNet [42]. For fair comparisons, the same classifiers are also used for other proposal generation methods, including SST [1], TURN [14], CTAP, and BSN. We adopt the conventional mean Average Precision (mAP) metric, where Average Precision (AP) reports the performance of each activity category. Specifically, mAP with tIoU thresholds $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ is used on THUMOS-14.

Table 6 illustrates the performance comparisons, which are evaluated on the testing set of THUMOS-14. With the same classifier, MGG achieves better performance than the other proposal generators, and outperforms the state-of-the-art proposal methods, namely CTAP [13] and BSN [26], thus demonstrating the effectiveness of our proposed MGG.

6. Conclusion

In this paper, we proposed a novel architecture, namely MGG, for the temporal action proposal generation. MGG holds two branches: one is SPP perceiving the whole video in a coarse level and the other is FAP working in a finer level. SPP and FAP couple together and integrate into MGG, which can be trained in an end-to-end fashion. By analyzing whole videos from both coarse and fine perspectives, MGG generates proposals with high recall and more precise boundary information. As such, MGG achieves better performance than the other state-of-the-art methods on the THUMOS-14 and ActivityNet-1.3 datasets. The superior performance of video detection relying on the generated proposals further demonstrates the effectiveness of the proposed MGG.

Acknowledgements. This work was supported in part by the Natural Science Foundation of Jiangsu under Grant BK20151102, in part by the State Key Laboratory for Novel Software Technology, Nanjing University under Grant KFKT2017B17, and in part by the Natural Science Foundation of China under Grant 61673108.

References

[1] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, pages 6373–6382, 2017.

[2] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[3] L. Cao, R. Ji, Y. Gao, W. Liu, and Q. Tian. Mining spatiotemporal video patterns towards robust action retrieval. *Neurocomputing*, 105:61–69, 2013.

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 4724–4733. IEEE, 2017.

[5] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, pages 1130–1139, 2018.

[6] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua. Temporally grounding natural sentence in video. In *EMNLP*, 2018.

[7] J. Chen, L. Ma, X. Chen, Z. Jie, and J. Luo. Localizing natural language in videos. In *AAAI*, 2019.

[8] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen. Temporal context network for activity localization in videos. In *ICCV*, pages 5727–5736, 2017.

[10] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, pages 768–784, 2016.

[11] Y. Feng, L. Ma, W. Liu, T. Zhang, and J. Luo. Video relocalization. In *ECCV*, pages 51–66, 2018.

[12] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[13] J. Gao, K. Chen, and R. Nevatia. Ctap: Complementary temporal action proposal generation. *arXiv preprint arXiv:1807.04821*, 2018.

[14] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, pages 3648–3656, 2017.

[15] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[16] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[19] W. Huang, L. Fan, M. Harandi, L. Ma, H. Liu, W. Liu, and C. Gan. Towards efficient action recognition: Principal backpropagation for training two-stream networks. *IEEE Transactions on Image Processing*, 28(4):1773–1782, 2019.

[20] Y. Jiang, Q. Dai, W. Liu, X. Xue, and C. Ngo. Human action recognition in unconstrained videos by explicit motion modeling. *IEEE Transactions on Image Processing*, 24(11):3781–3795, 2015.

[21] Y. Jiang, Q. Dai, X. Xue, W. Liu, and C. Ngo. Trajectory-based modeling of human actions with motion reference points. In *ECCV*, pages 425–438, 2012.

[22] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. In *ECCV Workshop*, 2014.

[23] Y. Li, N. Wang, J. Liu, and X. Hou. Factorized bilinear models for image recognition. *arXiv preprint*, 2017.

[24] T. Lin, X. Zhao, and Z. Shou. Single shot temporal action detection. In *ACM MM*, pages 988–996, 2017.

[25] T. Lin, X. Zhao, and Z. Shou. Temporal convolution based action proposal: Submission to activitynet 2017. *arXiv preprint arXiv:1707.06750*, 2017.

[26] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. Bsn: Boundary sensitive network for temporal action proposal generation. *arXiv preprint arXiv:1806.02964*, 2018.

[27] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[28] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, pages 1449–1457, 2015.

[29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[31] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *CVPR*, pages 3131–3140, 2016.

[32] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[33] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, pages 1417–1426, 2017.

[34] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang. Autoloc: Weaklysupervised temporal action localization in untrimmed videos. In *ECCV*, pages 162–179, 2018.

[35] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016.

[36] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[37] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[39] B. Wang, L. Ma, W. Zhang, and W. Liu. Reconstruction network for video captioning. In *CVPR*, 2018.

[40] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018.

[41] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 1(2):2, 2014.

[42] L. Wang, Y. Xiong, D. Lin, and L. V. Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, pages 6402–6411, 2017.

[43] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017.

[44] H. Xu, A. Das, and K. Saenko. R-c3d: region convolutional 3d network for temporal activity detection. In *ICCV*, pages 5794–5803, 2017.

[45] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 199–211, 2015.

[46] T. Yao, Y. Li, Z. Qiu, F. Long, Y. Pan, D. Li, and T. Mei. Msr asia msm at activitynet challenge 2017: Trimmed action recognition, temporal action proposals and densecaptioning events in videos. In *CVPR Workshop*, 2017.

[47] T. Yao, T. Mei, and Y. Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *CVPR*, pages 982–990, 2016.

[48] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, pages 2678–2687, 2016.

[49] J. Yuan, B. Ni, X. Yang, and A. A. Kassim. Temporal action localization with pyramid of score distribution features. In *CVPR*, pages 3093–3102, 2016.

[50] Z.-H. Yuan, J. C. Stroud, T. Lu, and J. Deng. Temporal action localization by structured maximal sums. In *CVPR*, page 7, 2017.

[51] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2933–2942, 2017.