Temporal Action Localization With Coarse-to-Fine Network

MIN ZHANG¹, HAIYANG HU², AND ZHONGJIN LI²

¹Department of Design and Art, Zhejiang Industry Polytechnic College, Shaoxing 312000, China
²School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Min Zhang (minzhang15@126.com)

This work was supported by the Scientific Research Fund of Zhejiang Provincial Education Department under Grant Y202146915 (A Project Supported by Scientific Research Fund of Zhejiang Provincial Education Department (Y202146915)).

ABSTRACT Precisely localizing temporal intervals for each action segment in long raw videos is essential challenge in practical video content analysis (e.g., activity detection or video caption generation). Most of previous works often neglect the hierarchical action granularity and eventually fail to identify precise action boundaries. (e.g., embracing approaching or turning a screw in mechanical maintenance). In this paper, we introduce a simple yet efficient coarse-to-fine network (CFNet) to solve the challenging issue of temporal action localization by progressively refining action boundary at multiple action granularities. The proposed CFNet is mainly composed of three components: a coarse proposal module (CPM) to generate coarse action candidates, a fusion block (FB) to enhance feature representation by fusing the coarse candidate features and corresponding features of raw input frames, and a boundary transformer module (BTM) to further refine action boundaries. Specifically, CPM exploits framewise, matching and gated actionness curves to complement each other for coarse candidate generation at different levels, while FB is devised to enrich feature representation by fusing the last feature map of CPM and corresponding raw frame input. Finally, BTM learns long-term temporal dependency with a transformer structure to further refine action boundaries at a finer granularity. Thus, the fine-grained action intervals can be incrementally obtained. Compared with previous state-of-the-art techniques, the proposed coarse-to-fine network can asymptotically approach fine-grained action boundary. Comprehensive experiments are conducted on both publicly available THUMOS14 and ActivityNet-v1.3 datasets, and show the outstanding improvements of our method when compared with the prior methods on various video action parsing tasks.

INDEX TERMS Temporal action localization, action detection, action granularity, progressive learning.

I. INTRODUCTION

With the rapidly increasing usage of various mobile devices and online media archives, large collections of videos are producing, storing and consuming each hour. How to efficiently and effectively parse video data and decode semantic content becomes especially important [1], [2], [3]. In general, parsing human actions in videos includes action detection [4], [5] and temporal action localization [6], [7]. Currently, these active topics have been widely studied due to many practical applications, such as human computer intersection, smart surveillance and online video retrieval, etc.

In recent years, impressive progress for action recognition has been reported by using deep learning networks [8]. However, compared to action detection, temporal action localization remains large development space in technique and performance due to fundamental challenge, which aims not only to recognize the actions with a set of category labels but also to locate the exact time stamps of the starting and ending boundaries of different action instances in long raw videos.

Inspired by object detection, which is required to generate spatial bounding box to precisely locate object instance from images, temporal action localization attempts to determine temporal intervals of action instances in videos. Considering the similarity between the two tasks, many works [9], [10] on action detection inherit the merit of pipeline for image
object detection [11], [12]. Previous detectors can be sketchily classified into two types: one-stage methods [13], [14] and two-stage methods [15], [16]. The latter is generally considered to have better accuracy due to the default anchor boxes, while the former is generally more efficient by eliminating the proposal step.

Compared with the object in image, which generally possesses clear boundary, the boundary points of an action are often blurred, since the subtle changes between successive frames are difficult to detect. As discussed in the work [17], bad localization results are the most influential factor leading to performance degradation. To address this problem, one line of works [10], [18], [19], [20] employs boundary regression to improve the boundary accurate. Among them, the pipeline of cascaded boundary regression is a significant yet effective paradigm to enhance the boundary precision. For example, Cascaded Boundary Regression (CBR) model [21] is designed to perform boundary regression in two cascaded steps, the generated boundaries are feedback as a new input to the model for a next round of refinement. But the CBR model just reuses temporal coordinate regression and the features in cascaded steps are inconsistent with the prerequisite for progressive learning. There are some other works based on regression mechanism [10], [16], [19] that directly employ two-stage pipeline Faster R-CNN to address the issue of activity detection, and thus have a self-contained two-stage cascaded steps. However, all these approaches just achieve rough boundaries rather than accurate boundaries in a fine-grained level. Besides, extra proposal step costs a lot of computing time and reduces detection speed severely. Another direction of works [22], [23], [24] explores frame-level scores to evaluate actionness in a dense manner, they adopt pre-determined thresholds on the scores to identify the boundaries of action intervals. In this way, the boundaries of these candidates are considered to be precise. Nevertheless, the preset threshold or selection metric is heavily relied by these frame-level approaches, and directly determines the key point of the action boundaries.

In this paper, we devise a novel coarse-to-fine network (CFNet) to enhance the precision and speed of temporal action localization by exploring temporal boundary transformer. The proposed network is composed of three components: a coarse proposal module (CPM) to produce coarse action candidates with fusion of multiple actionness grouping, a fusion block (FB) to generate frame-level features by fusing the last feature map of CPM and corresponding raw frame input, and a boundary transformer module (BTM) to further refine precise action intervals by modeling long-term temporal dependency. Specifically, the CPM generates the actionness curves by using multiple actionness evaluation, i.e., frame-wise, matching and gated scores, which can provide complementary information to each other. Then, an efficient watershed actionness grouping algorithm is utilized to generate coarse action candidates. Afterward, FB is devised to enhance feature representation by fusing the features of coarse action candidates and corresponding raw frame input. Finally, BTM is introduced to refine action boundary by modeling long-term temporal dependency at a finer granularity. Thus, fine-grained action intervals can be identified. Comprehensive experiments are carried out on THUMOS14 and ActivityNet-v1.3, and display outstanding improvements of the proposed detector over prior approaches on action detection.

Overall, the major contributions are summed up as follows:

- We devise an effective framework to enhance the performance of temporal action localization in a coarse-to-fine fashion, which is composed of three components: coarse proposal module, fusion block and boundary transformer module.
- To improve the coarse proposal generation, multiple actionness curves are combined together to provide complementary information to each other. Then, an efficient watershed actionness grouping algorithm is applied to produce coarse action candidates.
- A boundary transformer module is employed to model long-term temporal dependency at frame granularity, thus, the accuracy of action boundary can be significantly improved.
- Comprehensive experiments are carried out on THUMOS14 and ActivityNet-v1.3, and exhibit outstanding performance over prior methods on the challenging issue of temporal action localization.

II. RELATED WORK

In this section, we briefly review the most relevant works: two-stage and one-stage pipelines for temporal action localization, and transformers used in computer vision community.

Two-stage pipeline usually adopts proposal and classification step, which is widely studies in works [22], [23], [25], [26], [27], [28]. Among them, a few works ( [22], [23], [25]) try to improve the quality of proposals, while other works [24], [29], [30] aim to design exquisite classifier to accurately classify the proposals. However, the cascaded two steps in most of them are often treated separately and trained in different stages. Recently, inspired by the widely used Faster R-CNN for object detection on images [12], some efforts [16], [19] attempt to train the two-stage pipeline in an end-to-end fashion.

Motivated by the advance of anchor-free object detectors such as SSD [11], one-stage pipelines [6], [13], [31] discard proposal step in favour of one-stage architecture. To address the inflexibility of action boundary during proposal stage, Lin et al. [13] introduced single shot action detector network to directly predict action instances without proposal generation step. Buch et al. [31] exploited a single-stream temporal action detection framework, which effectively combines activity detection and its semantic sub-tasks into a unified end-to-end architecture. Recently, Long et al. [6] introduced a novel one-stage framework GTAN to exploit temporal structure of each action instance based on gaussian kernels.

Transformer [32] has attracted extensive attention in nature language processing, which is based on self-attention
mechanism to learn the relations between elements of a sequence. Dosovitskiy et al. [33] applied a pure transformer to directly classify image patches. Recently, many works employ transformer structure for video understanding task. Girdhar et al. [34] proposed an action transformer model to detect human actions through learning spatio-temporal context information. Bertasius et al. [35] introduced “TimeSformer” to video classification exclusively using self-attention over temporal and spatial dimension. In contrast to these methods, Lui et al. [37] combined a destination classifier and destination-specific trajectory models into a unified deep network architecture to predict pedestrian trajectory. The effective destination-driven model achieves significant performance improvement and outperforms other state-of-the-art methods on both the NYGC and ATC datasets. Cao et al. [38] built a deep clustering network to learn discriminative features. They exploited siamese network to find an embedding space and adopted virtual adversarial training to synthesize different samples. The experimental results verify the superiority and effectiveness of the enhanced network. Gan et al. [39] proposed a pyramid structure to learn long-range dependencies of action proposals. They utilized attention mechanism to focus on salient part of proposals and devised a multi-scale function to generate final fixed-length features while retaining the important temporal information.

Different from above works, we propose a novel coarse-to-fine asymptotic approximation model with self-attention mechanism in pursuit of incorporating the merit of two-stage pipeline into one-stage architecture. The network mainly consists of coarse proposal stage cascaded by boundary transformer module to achieve fine-grained action boundary.

III. COARSE-TO-FINE NETWORK

In this section, we describe the proposed coarse-to-fine network for temporal action localization in detail. The overview of the architecture is shown in Figure 1, which is composed of three stages: a coarse proposal module (CPM) with integration of three actionness scores on different granularity to segment coarse action candidates, a fusion block (FB) to fuse the coarse features of action candidates and corresponding features of raw input frames, and a boundary transformer module (BTM) to further refine action boundaries. Specifically, the CPM is exploited with fusing three actionness grouping at framewise, matching and gated to discriminate actions at different levels. The three actionness measurements can provide complementary information to each other and work together to achieve final actionness.

1) FRAMEWISE ACTIONNESS

The framewise actionness prediction (Figure 2(a)) is essentially learning a binary classifier, which deals with every frame separately. Considering $n$ consecutive frames, from which we can extract a feature set $\mathcal{F} = \{f_i \mid f_i \in \mathbb{R}^d\}$, then the feature set is divided into positive subset $\mathcal{F}^+$ and negative subset $\mathcal{F}^-$ according to the ground truth. Finally, the binary classifier $\phi_b$ can be obtained through optimizing the following loss function:

$$
Loss_b = - \sum_{f \in \mathcal{F}^+} \log(\phi_b(f)) - \sum_{f \in \mathcal{F}^-} \log(1 - \phi_b(f))
$$

(1)

2) MATCHING ACTIONNESS

In the coarse proposal process, we perform a relative evaluation by exploiting actionness degree within a video, which can be considered as a supervised sequence problem. Assuming a couple of frames, which are fetched from the positive subset and negative subset respectively, we seek to train a matching classifier (Figure 2(b)) generating higher score of the positive feature than negative feature. Technically, given feature pairs $\mathcal{P} = \{(f^+, f^-) \mid f^+ \in \mathcal{F}^+, f^- \in \mathcal{F}^-\}$, where each pair $(f^+, f^-)$ contains a positive and negative feature $f^+, f^-$ extracted from the same video respectively. To optimize matching actionness classifier $\phi_p$, the loss function is formulated as follows:

$$
Loss_p = \sum_{(f^+, f^-) \in \mathcal{P}} \max(0, 1 - \phi_p(f^+) + \phi_p(f^-))
$$

(2)

3) GATED ACTIONNESS

It is noted that there is high correlation between action labels of consecutive frames, recurrent information is exploited to predict actionness. To capture the recurrent information from a sequence of frames, we devise a single-layer GRU model and formulate it as a recurrent classification task. The gated classifier $\phi_r$ (Figure 2(c)) is trained to output action categories.

4) SCORE FUSION

After obtaining the three kinds of actionness classifiers: $\phi_b$, $\phi_p$ and $\phi_r$, the corresponding actionness score $S_b$, $S_p$ and $S_r$ can be generated by each classifier. We employ a sigmoid function to normalize the scores to $[0, 1]$, and the final confidence score can be computed by linearly combing the three
FIGURE 1. Overview of the proposed coarse-to-fine network (CFNet), which mainly consists of three stages: coarse proposal module (CPM), fusion block (FB) and boundary transformer module (BTM). First, the CPM generates coarse action candidates by exploiting three complementary action curves (i.e., framewise, matching and gated actionness). Then, the FB is devised to generate frame-level features by fusing the coarse proposal features and corresponding features of raw frame input. Afterward, the action boundaries are predicted with a transformer structure by modeling long-term temporal dependency. Finally, we use matching strategy to update confidence score and further refine the boundary of each action candidate. Thus, each action interval can be achieved in a finer granularity. The three stages are sequentially cascaded together to progressively refine action boundaries by incrementally improving feature representation.

FIGURE 2. Illustration of three kinds of actionness prediction: (a) Framewise actionness prediction. (b) Matching actionness prediction. (c) Gated actionness prediction.

actionness scores as follows:

\[ S = \lambda_1 S_b + \lambda_2 S_p + \lambda_3 S_r \]  

where \( \lambda_1, \lambda_2 \) and \( \lambda_3 \) are weight parameters to be learned by cross validation. The fused confidence score aggregates the scores of three levels and shows an obvious wave peak in actionness curve indicating each action interval.

5) TEMPORAL ACTIONNESS MERGING

After obtaining the final actionness scores, we employ temporal actionness merging algorithm to produce action candidates, the detailed procedure is depicted in Algorithm 1. The step sizes of watershed level \( h \) and merging length \( d \) are set to 0.085 and 0.025 respectively. The non-maximum suppression (NMS) algorithm with threshold IoU=0.95 is performed to remove highly overlapped candidates. Since the action curve can not always accurately reflect shot boundaries, temporal adjacent action intervals are easy to merge together. Thus, the proposals generated in this stage are largely rough and needed to be further refined.

B. FUSION BLOCK

To facilitate subsequent boundary refinement, we design the fusion block (FB) to generate frame-level features. The detailed structure of FB is illustrated in Figure 3. As we can see, the FB is composed of two branches. One branch contains three convolutional layers cascaded by a pooling layer to extract spatiotemporal features. The other branch is based on the last feature map of CPM, and is stacked with three deconvolutional layers to reshape the features. The two branch features are concatenated together to generate enhanced feature representation through a convolutional layer. Intuitively, the features from raw input frames can provide more spatial details.

C. BOUNDARY TRANSFORMER MODULE

It is observed that long-term temporal dependency is essential important for boundary refinement. To capture temporal relationship, previous approaches often adopt temporal convolution [23] or global temporal pooling [40]. However, 1D temporal convolution can learn local relation...
We introduce boundary transformer module (BTM) to learn both local temporal relationship and long-term temporal dependency between frames but fails to capture long-term context information due to the limit of kernel size. In contrast, the global temporal pooling can effectively learn global features over frames but lacks the capability of modeling fine-grained temporal structure and easily causes unnecessary noise. To address this issue, we introduce boundary transformer module (BTM) to learn both local temporal relationship and long-term temporal dependency with a transformer structure [32]. Specifically, we stack M-layer transformers to predict boundary probabilities at frame-level and the encoder-decoder structure in each layer is configured as follows:

The transformer encoder is used to estimate the relevance between frames, which consists of a multi-head self-attention layer and a feed forward network (FFN). First, the frame-level feature map \( F \in \mathbb{R}^{T \times C} \) generated by FB is fed into encoder and output enhanced feature sequence \( F' \in \mathbb{R}^{T \times C} \) incorporating global temporal context.

Different from the encoder, in addition to self-attention mechanism and feed forward network, the decoder also adopts encoder-decoder attention strategy. The input to multi-head self-attention is identical with the encoder, namely feature sequence \( F' \in \mathbb{R}^{T \times C} \). The output of decoder is feature sequence \( \hat{F} \). \( F' \) is transformed into \( K, V \), while \( \hat{F} \) is transformed into \( Q \). Thus, the relevant features can be significantly enhanced, meanwhile the over-smoothing effect is substantially suppressed.

D. TRAINING AND INFERENCE

1) PROPOSAL MATCHING STRATEGY

Given the coarse proposals generated by CPM and the boundary probability sequences generated by BTM, we aim to match them by computing temporal intersection over union (tIoU). Assuming a coarse proposal \( \psi_c = [t_c^l, t_c^r] \), we select the matched boundary sequence \( \omega_c = [t_c^l, t_c^r] \) with the max tIoU value. Then, the coarse proposal can be refined as follows:

\[
[\hat{t}_c, \hat{t}_c] = \begin{cases} 
\left[\frac{t_c^l + t_c^r}{2}, \frac{t_c^l + t_c^r}{2}\right], & \text{if } p_1 > \epsilon_1 \text{ and } p_2 > \epsilon_2, \\
[t_c^l, t_c^r], & \text{otherwise}.
\end{cases}
\]

(4)

where \( p_1 \) and \( p_2 \) are confidence scores for boundary classification and regression respectively. \( \epsilon_1 \) and \( \epsilon_2 \) are pruning thresholds. Finally, we can obtain the fine-grained proposal \( [\hat{t}_c, \hat{t}_c] \) with more precise boundaries.
2) LOSS FUNCTIONS

An elaborate multi-task loss function is utilized to optimize the network. The whole loss \( L_{\text{all}} \) can be defined as follows:

\[
L_{\text{all}} = \alpha L_{\text{coarse}} + \beta L_{\text{fine}}
\]

(5)

where \( L_{\text{coarse}} \) and \( L_{\text{fine}} \) denote the loss function for CPM and BTM respectively. \( \alpha, \beta \) are weighted parameters, which are empirically set to 1.

Concretely, \( L_{\text{coarse}}, L_{\text{fine}} \) are formulated as follows:

\[
L_{\text{coarse}} = \frac{1}{N} \sum_k [L_{\text{cls}}(p_k, p_k^*) + \sum_k [p_k^* \geq 1] L_{\text{reg}}(o_k, o_k^*)]
\]

(6)

where \( N \) represents the total number of training samples, \( L_{\text{cls}} \) and \( L_{\text{reg}} \) are the classification loss and regression loss respectively. \( p_k, p_k^* \) are the predicted category score and corresponding ground truth. \( o = (\Delta c, \Delta t) \) represents the predicted offset of temporal localization and duration, \( o^* \) presents the corresponding ground truth.

\[
L_{\text{fine}} = \frac{1}{N} \sum_k [p_k^* \geq 1] L_{\text{loc}}(b_k, b_k^*)
\]

(7)

where \( L_{\text{loc}} \) denotes the binary logistic loss function. \( b_k = (\Delta S_k, \Delta E_k) \) is the predicted offsets to the starting and ending boundary points for each action candidate \( k \), and \( b_k^* \) is the corresponding ground truth label.

3) INFERENCE

During inference, all prediction results except those from the last decoder layer are ignored. Score fusion and redundant candidate suppression are performed to obtain final results. Technically, to take full advantage of various confidence scores for each refined proposal \( p \), the final confidence score is calculated by multiplying scores generated by CPM and BTM, namely:

\[
p = p_1 \ast p_s \ast p_e
\]

(8)

IV. EXPERIMENTS AND RESULTS

A. CONFIGURATION AND IMPLEMENT DETAILS

1) DATASET

For the task of temporal action detection, THUMOS’14 [41] provides 1010 videos for validation with 220 temporal annotated videos and 1574 videos for testing with 212 temporal annotated videos from 20 action categories. Following common practice, the proposed network is trained on validation set and evaluated on test set. The ActivityNet-v1.3 [42] is classified into training, validation and test subsets, which contain 10024, 4926 and 5044 videos from 200 categories, respectively. Following the standard evaluation protocol, the network is trained on test subset and the evaluation results are reported on validation subset.

2) EVALUATION METRICS

For quantitative evaluation of the coarse-to-fine network, the metric of mean average precision (mAP) is employed with varying IoU thresholds. We directly adopt the official toolkit to compute the mAP on THUMOS’14, while following the standard protocol provided by the competition authorities on ActivityNet, the IoU threshold varies in the range \([0.5 : 0.05 : 0.95]\).

3) IMPLEMENT DETAILS

During training process, temporal overlapped clips are retained to enlarge the training sets. The clips which have at least a complete action instance are retained for training. In the test stage, raw input videos are segmented into clips without temporal intersection. All the clips are used for evaluation. In THUMOS’14, both the RGB and optical flow frames are sampled at a frame rate of 10 fps (10 frames per second). The temporal interval of each clip \( L \) is set to 256 frames (i.e., roughly about 25.6 seconds), which is longer than most of other action instances in the dataset. Different from THUMOS’14, ActivityNet-v1.3 is a much large scale benchmark, the frame rate is only set to 3 fps. Correspondingly, we set \( L \) as 768, thus, roughly covering about 256 seconds of a video. Such a temporal duration is overwhelming longer than most of the action instances in training subset. The query number is set to 30, 10 in THUMOS’14 and ActivityNet-v1.3 respectively. The transformer encoder layer \( L_E \) is set to 2 and the transformer decoder layer \( L_D \) is set to 4. The learning rate is initialized to \( 2 \times 10^{-4} \) and reduced by a coefficient of 10 after some epochs. The model is trained for 60, 15 epochs on THUMOS’14 and ActivityNet-v1.3 separately, meanwhile, the learning rate is decreased after 50, 12 epochs. The batch size is set to 16. The model is implemented on Ubuntu 16.04 server with a GeForce RTX 2080ti GPU.

B. PARAMETER SETTINGS OF THE COMPARED METHODS

The following state-of-the-art methods are compared for the task of temporal action localization. The parameter settings of these algorithms are briefly described as follows:

(1) The LEAR submission at THUMOS14 [43] adopts Fisher vector to encode dense trajectory features. They used a sliding window to slide over the video with a stride of 10 frames. The window size is set to 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 and 150 frames respectively. After scoring the windows, they applied non-maximum suppression algorithm to filter out overlapped proposals.

(2) Yeung et al. [44] exploited a recurrent neural network-based agent to observe moments and refine hypotheses in video. They used VGG-16 to extract FC7-layer features and embedded them into a 1024-d observation vector. The recurrent network is set to a 3-layer LSTM with 1024 hidden units in each layer. Sampling rate of frame is 5 fps and 1 fps in THUMOS14 and ActivityNet respectively. The agent is given...
6 observations for each sequence. All other hyperparameters are learned by cross-validation.

(3) S-CNN [45] exploits three cascaded networks to address the task of temporal action localization, namely proposal network, classification network and localization network. The whole model is trained with momentum of 0.9, learning rate of 0.0001 and weight decay factory of 0.0005. During training, the learning rate is divided by a factor of 10 for each 10K iterations.

(4) CDC [24] conducts upsampling in time and downsampling in space to predict temporal dynamics and action categories at a fine granularity. The model is optimized with stochastic gradient descent. The learning rate, momentum and weight decay are set to 0.00001, 0.9 and 0.005 respectively. To accelerate convergence, the CDC is pre-trained on Sports-1M.

(5) SSAD [13] uses both two-stream network and C3D network to extract snippet-level action score features. The SSAD network is optimized by adaptive moment estimation algorithm with multi-task learning. During training, the window length is set to 512 and the whole network is trained for 30 epochs with a learning rate of 0.0001.

(6) R-C3D [10] devises a fully convolutional network to generate temporal regions and classifies these selected regions into specific action instances. The R-C3D is initialized with C3D weights and finetuned on UCF101 with a fixed learning rate of 0.0001.

(7) CBR [21] builds a two-stage pipeline with CBR model to refine the temporal boundary. The CBR is trained with the Adam optimizer. The batch size and learning rate are set as 128 and 0.005 respectively.

(8) BSN [23] first generates action proposals locally, then evaluates whether a proposal embodies an action globally. They used two-stream network to encode features. The learning rate is set to 0.001 for 10 epochs, then decreased by 10% for another 10 epochs. For Soft-NMS, the threshold $\theta$ is set as 0.65 and 0.8 on THUMOS14 and ActivityNet respectively.

(9) GTAN [6] learns temporal structure by exploiting gaussian kernels. They employed stochastic gradient descent algorithm to optimize the network with 0.9 momentum. The original learning rate is set as 0.001. The weight decay parameter and mini-batch size are set to 0.0001 and 16 respectively.

(10) BMN [25] evaluates confidence score of proposals through boundary matching mechanism. To train BMN from scratch, the learning rate, epoch number and batch size are set to 0.001, 10 and 16 respectively.

(11) TAL-Net [19] is inspired by the Faster R-CNN framework, thus the parameter configuration of TAL-Net follows the Faster R-CNN implementation to a large extent. In proposal stage, the anchor scale is set as $\{1, 2, 3, 4, 5, 6, 8, 11, 16\}$.

(12) BSN++ [46] takes advantage of complementary information between the starting and ending boundaries to generate temporal proposals. The network is trained by Adam optimizer from scratch. The initial learning rate and batch size are set to 0.001 and 16 respectively.

(13) TCANet [47] introduces temporal context and complementarity from local and global characteristics to produce high-quality action proposals. The batch size, learning rate and epoch number are set to 16, 0.0004 and 5, respectively.

(14) RTD-Net [48] proposes a Transformer-like structure to directly produce action proposals. The network is trained with AdamW from scratch. The batch size and learning rate are set to 32 and 0.0001, respectively.

### C. COMPARISON TO STATE-OF-THE-ART SOLUTIONS

The performance comparison of the proposed CFNet to prior solutions on THUMOS’14 is presented in Table 1, where the experimental results are obtained after fusing two-stream scores. As we can see that CFNet is superior to the prior solutions with a significant mAP improvement on all IoU thresholds. In particular, CFNet obtains remarkable improvement of 2.1% at $tIoU = 0.5$ compared to the suboptimal TCANet (from 44.6% to 46.7%), which is an advanced pipeline in THUMOS’14. It is observed that the previous works (such as [43], [44] and [45]) show relatively weak performance, because they neglect the hierarchical action granularity. In contrast, the proposed CFNet takes full advantage of three cascaded steps to encode discriminative features. Specifically, the CPM first explores three different measures to learn actionness. Then, the features of these ordinary proposals are strengthened by FB module, which can highlight salient action characteristics. Next, based on these enhanced features, the BTM is exploited to learn long-term temporal dependency at a fine granularity. Finally, matching strategy is utilized to refine temporal action boundary as well as update the confidence scores of corresponding proposals. As expected, the experimental results reflect that our model can predict proposals with high scores, and the boundaries of action instances are more precise.

The detection results on ActivityNet-v1.3 are presented in Table 2. Similarly, although the frame rate on ActivityNet is set as only 3 fps, CFNet outperforms the prior solutions on all preset IoU thresholds. Figure 4 shows some qualitative examples, the ground truth temporal action interval and predicted

| IoU | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|-----|-----|-----|-----|-----|-----|
| Oneata et al. [43] | 28.8 | 21.8 | 15.0 | 8.5 | 3.2 |
| Yeung et al. [44] | 36.0 | 26.4 | 17.1 | - | - |
| S-CNN [45] | 36.3 | 28.7 | 19.0 | 10.3 | 5.3 |
| CDC [24] | 40.1 | 29.4 | 23.3 | 13.1 | 7.9 |
| SSAD [13] | 43.0 | 35.0 | 24.6 | - | - |
| R-C3D [10] | 44.8 | 35.6 | 28.9 | - | - |
| CBR [21] | 50.1 | 41.3 | 31.0 | 19.1 | 9.9 |
| BSN [23] | 53.5 | 45.0 | 36.9 | 28.4 | 20.0 |
| GTAN [6] | 57.8 | 47.2 | 38.8 | - | - |
| BMN [25] | 56.0 | 47.4 | 38.8 | 29.7 | 20.5 |
| TAL-Net [19] | 53.2 | 48.5 | 42.8 | 33.8 | 20.8 |
| BSN++ [46] | 59.9 | 49.5 | 41.3 | 31.9 | 22.8 |
| TCANet [47] | 60.6 | 53.2 | 44.6 | 36.8 | 26.7 |
| RTD-Net [48] | 53.9 | 48.9 | 42.0 | 33.9 | 23.4 |
| CFNet | 61.2 | 54.5 | 46.7 | 37.5 | 27.3 |
FIGURE 4. The detection results of two qualitative examples from THUMOS’14 and ActivityNet-v1.3 separately. The CPM is a simple detector which is just equipped with coarse proposal module. # denotes the frame number of the action boundary.

TABLE 2. Comparison to the prior solutions on activitynet-v1.3 with varying tIoU in terms of mAP.

| tIoU | 0.5 | 0.75 | 0.95 | Average |
|------|-----|------|------|---------|
| R-3D [10] | 26.80 | - | - | 12.70 |
| TAL-Net [19] | 38.23 | 18.30 | 1.30 | 20.22 |
| CDC [24] | 43.83 | 25.88 | 0.21 | 22.77 |
| BSN [23] | 46.45 | 29.96 | 8.02 | 30.03 |
| BMN [25] | 50.07 | 34.78 | 8.29 | 33.85 |
| GTAN [6] | 52.61 | 34.14 | 8.91 | 34.31 |
| BSN++ [46] | 51.3 | 35.7 | 8.3 | 34.9 |
| TCA-Net [47] | 51.9 | 34.9 | 7.5 | 34.4 |
| RTD-Net [48] | 46.4 | 30.5 | 8.6 | 30.5 |
| CFNet | 53.2 | 35.9 | 9.6 | 35.7 |

interval by CPM and CFNet are all given in the figure. Here, CPM denotes a simple detector which just uses coarse proposal module to predict action interval. As illustrated in the figure, CPM combines three different actionness classifiers to produce ordinary action intervals. In contrast, since CFNet uses FB module to strengthen features and further improve action boundary through the BTM module, more accurate action intervals are obtained.

D. INFERENCE SPEED

The comparison of detection speed between the proposed CFNet and previous prior solutions is presented in Table 3. It is noted that CFNet is evaluated on Ubuntu 16.04 server with a GeForce RTX 2080ti GPU. It is obviously showed that the proposed coarse-to-fine network can run a high speed of 1260 fps, and outperform other approaches by a big margin. The significant efficiency gain of the CFNet mainly attributes to the following two aspects: (1) CFNet falls in the category of one-stage pipeline and eliminates the time-consuming proposal step used in the counterpart networks; (2) CFNet integrates three modules into a unified architecture, thus can be well trained in an end-to-end fashion.

E. ABLATION STUDY

To investigate the effect of different parts and settings in the proposed CFNet, we perform a set of ablation tests on THUMOS’14 at threshold 0.5. The results are reported with the input of RGB images and optical flow separately. It is observed that different thresholds possess a consistent performance trend.

1) CASCADED DETECTION MODULES

Considering that the features are progressively enhanced through three modules, we explore how the network performance improves with the change of module configuration. At the same time, the confidence score of each proposal is progressively updated by multiplying scores from different modules. The experimental results are exhibited in Table 4. It is clearly showed that the sequentially embedded modules can substantially enhance the accuracy of the generated action proposals, which demonstrate the effectiveness of different components.

2) EFFECTIVENESS OF TRANSFORMER LAYERS

Generally, utilizing multiple layer transformers can achieve better performance, which have been demonstrated in several studies, e.g., object detection [49] and NLP [32]. We carry out ablation experiments to investigate the number of different layers in BTM. The experimental results are presented in
Table 5. Investigate the number of transformer layers in BTM.

| layer | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|-------|-----|-----|-----|-----|-----|
| 1     | 60.4| 53.6| 45.1| 36.3| 25.8|
| 2     | 61.2| 54.5| 46.7| 37.5| 27.3|
| 6     | 60.7| 53.2| 44.9| 36.6| 25.2|

Table 5. It is clearly displayed that, the proposed CFNet obtains the best accuracy when setting the number of transformer layer as 3. However, building transformer structure of more than three layers does not lead to significant accuracy gain. On the contrary, it makes the whole network more complex and difficult to train.

V. CONCLUSION

In this paper, we propose an elegant coarse-to-fine network (CFNet) for temporal action localization, which progressively refines action boundaries through three stages, namely, coarse proposal module (CPM), fusion block (FB) and boundary transformer module (BTM). Specifically, CPM firstly outputs coarse action candidates by exploiting pointwise, pairwise and gated actionness. Then, FB is devised to generate discriminative frame-level features by fusing the features of coarse proposals and corresponding raw frame input. Afterward, the action boundaries are estimated with a transformer structure by learning long-term temporal dependency. Finally, proposal matching mechanism is used to update confidence score and further refine the boundary of each action candidate. Thus, more precise temporal action interval can be obtained in a fine granularity. Benefited from the hierarchical architecture, progressive learning manner and end-to-end training, CFNet obtains significant performance gain over the prior solutions on THUMOS’14 and ActivityNet-v1.3 dataset. In the future, we plan to implement the well-designed coarse-to-fine pipeline for action detection in complex industrial environment, that will be a challenging work.

REFERENCES

[1] X. Yan, H. Gong, Y. Jiang, S. T. Xia, F. Zheng, X. You, and L. Shao, “Video scene parsing: an overview of deep learning methods and datasets,” Comput. Vis. Image Understanding, vol. 201, Dec. 2020, Art. no. 103077.
[2] Y. Liu, C. Shen, C. Yu, and J. Wang, “Efficient semantic video segmentation with per-frame inference,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Aug. 2020, pp. 352–368.
[3] R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen, “VRSTC: Occlusion-free video person re-identification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7183–7192.
[4] S. Sudhakaran, S. Escalera, and O. Lanz, “LSTA: Long short-term attention for egocentric action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9954–9963.
[5] Z. Tu, H. Li, D. Zhang, J. Dauwels, B. Li, and J. Yuan, “Action-stage emphasized spatiotemporal VLAD for video action recognition,” IEEE Trans. Image Process., vol. 28, no. 6, pp. 2799–2812, Jan. 2019.
[6] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, “Gaussian temporal awareness networks for action localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 344–353.
[7] L. Yang, H. Peng, D. Zhang, J. Fu, and J. Han, “Revisiting anchor mechanisms for temporal action localization,” IEEE Trans. Image Process., vol. 29, pp. 8535–8548, 2020.
[8] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, Nov. 2015.
[9] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, “Single shot video object detector,” IEEE Trans. Multimedia, vol. 23, pp. 846–858, 2020.
[10] H. Xu, A. Das, and K. Saenko, “R-C3D: Region convolutional 3D network for temporal action detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5783–5792.
[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “SSD: Single shot MultiBox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Oct. 2016, pp. 21–37.
[12] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[13] T. Lin, X. Zhao, and Z. Shou, “Single shot temporal action detection,” in Proc. ACM Int. Conf. Multimedia, Oct. 2017, pp. 988–996.
[14] Y. Huang, Q. Dai, and Y. Lu, “Decoupling localization and classification in single shot temporal action detection,” in Proc. IEEE Int. Conf. Multimed. Expo. (ICME), Jul. 2019, pp. 1288–1293.
[15] H. Xu, A. Das, and K. Saenko, “Two-stream region convolutional 3D network for temporal action detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 10, pp. 2319–2332, Jun. 2019.
[16] Y. Bai, H. Xu, K. Saenko, and B. Ghanem, “Contextual multi-scale region convolutional 3D network for action detection,” arXiv:1801.09184, Sep. 2018, pp. 256–272.
[17] C. Lin, J. Li, Y. Wang, Y. Tai, D. Luo, Z. Cui, and R. Ji, “Fast learning of temporal action proposal via dense boundary generator,” in Proc. Conf. AAAI Artif. Intell., Feb. 2020, pp. 11499–11506.
[18] 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 Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 1130–1139.
[19] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, “Turn tap: Temporal unpleasantness regression network for temporal action proposals,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 3628–3636.
[20] J. Gao, Z. Yang, and R. Nevatia, “Cascaded boundary regression for temporal action detection,” 2017, arXiv:1705.01180.
[21] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S. F. Chang, “Multi-granularity generator for temporal action proposal,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3604–3613.
[22] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, “BSN: Boundary sensitive network for temporal action proposal generation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 3–19.
[23] 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 Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5734–5743.
[24] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, “BMN: Boundary-matching network for temporal action proposal generation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3889–3898.
[25] J. Gao, K. Chen, and R. Nevatia, “CTAP: Complementary temporal action proposal generation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 68–83.
[26] H. Qiu, Y. Zheng, H. Ye, Y. Lu, F. Wang, and L. He, “Precise temporal action localization by evolving temporal proposals,” in Proc. ICMR, Jan. 2018, pp. 388–396.
[27] Z. Shou, D. Wang, and S. F. Chang, “Temporal action localization in untrimmed videos via multi-stage CNNs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1049–1058.
[28] K. Yang, P. Qiao, D. Li, S. Lv, and Y. Dou, “Exploring temporal preservation networks for precise temporal action localization,” in Proc. Conf. AAAI Artif. Intell., Feb. 2018, vol. 32, no. 1, pp. 1–8.
[29] T. Lin, Y. Xiong, L. Wei, Z. Wu, Z. Wu, and D. Lin, “Temporal action detection with structured segment networks,” in Proc. IEEE Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2914–2923.
[30] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles, “End-to-end, single-stream temporal action detection in untrimmed videos,” in Proc. BMVC, Jan. 2017, pp. 1–12, doi: 10.5244/C.31.93.
[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., Dec. 2017, pp. 5998–6008.
\[ J. \text{Uszkoreit, and N. Houlsby, “An image is worth } 16 \times 16 \text{ words: Transformers for image recognition at scale,” 2020, arXiv:2010.11929.} \]

\[ R. \text{Girdhar, J. Carreira, C. Doersch, and A. Zisserman, “Video action transformer network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 244–253.} \]

\[ G. \text{Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?” 2021, arXiv:2102.05095.} \]

\[ L. \text{Wang, Y. Qiao, and X. Tang, “Action recognition and detection by combining motion and appearance features,” THUMOS14 Action Recogniz. Challenge, vol. 1, no. 2, p. 2, 2014.} \]

\[ A. \text{K. F. Lui, Y. H. Chan, and M. F. Leung, “Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings,” in Proc. IEEE Int. Conf. Big Data (Big Data), Dec. 2021, pp. 1709–1717.} \]

\[ W. \text{Cao, Z. Zhang, C. Liu, R. Li, Q. Jiao, Z. Yu, and H. S. Wong, “Unsupervised discriminative feature learning via finding a clustering-friendly embedding space,” Pattern Recognit., vol. 129, Sep. 2022, Art. no. 108768.} \]

\[ M. \text{G. Gan and Y. Zhang, “Temporal attention-pyramid pooling for temporal action detection,” IEEE Trans. Multimedia, early access, Apr. 8, 2022, doi: 10.1109/TMM.2022.3166025.} \]

\[ J. \text{Gao, Z. Shi, G. Wang, J. Li, Y. Yuan, S. Ge, and X. Zhou, “Accurate temporal action proposal generation with relation-aware network,” in Proc. Conf. AAAI Artif. Intell., Feb. 2020, pp. 10810–10817.} \]

\[ Y. \text{G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, (2014). THUMOS Challenge: Action Recognition With A Large Number of Classes. [Online]. Available: http://crcv.ucf.edu/THUMOS14/} \]

\[ F. \text{C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, “ActivityNet: A large-scale video benchmark for human activity understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 961–970.} \]

\[ D. \text{Oneata, J. Verbeek, and C. Schmid. The LEAR Submission at Thumos 2014. Accessed: 2014. [Online]. Available: https://hal.inria.fr/hal-01074442/} \]

\[ S. \text{Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, “End-to-end learning of action detection from frame glimpses in videos,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2678–2687.} \]

\[ Z. \text{Shou, D. Wang, and S. F. Chang, “Temporal action localization in untrimmed videos via multi-stage CNNs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1049–1058.} \]

\[ H. \text{Su, W. Gan, W. Wu, Y. Qiao, and J. Yan, “BSN++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation,” 2020, arXiv:2009.07641.} \]

\[ Z. \text{Qing, H. Su, W. Gan, D. Wang, W. Wu, X. Wang, Y. Qiao, J. Yan, C. Gao, and N. Sang, “Temporal context aggregation network for temporal action proposal refinement,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 485–494.} \]

\[ J. \text{Tan, J. Tang, L. Wang, and G. Wu, “Relaxed transformer decoders for direct action proposal generation,” 2021, arXiv:2102.01894.} \]

\[ N. \text{Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Aug. 2020, pp. 213–229.} \]