Coarse-to-Fine Localization of Temporal Action Proposals

Fuchen Long, Ting Yao, *Member, IEEE*, Zhaofan Qiu, Xinmei Tian, *Member, IEEE*, Tao Mei, *Fellow, IEEE*, and Jiebo Luo, *Fellow, IEEE*

***Abstract*—Localizing temporal action proposals from long videos is a fundamental challenge in video analysis (e.g., action detection and recognition or dense video captioning). Most existing approaches often overlook the hierarchical granularities of actions and thus fail to discriminate fine-grained action proposals (e.g., hand washing laundry or changing a tire in vehicle repair). In this paper, we propose a novel coarse-to-fine temporal proposal (CFTP) approach to localize temporal action proposals by exploring different action granularities. Our proposed CFTP consists of three stages: a coarse proposal network (CPN) to generate long action proposals, a temporal convolutional anchor network (CAN) to localize finer proposals, and a proposal reranking network (PRN) to further identify proposals from previous stages. Specifically, CPN explores three complementary actionness curves (namely pointwise, pairwise, and recurrent curves) that represent actions at different levels for generating coarse proposals, while CAN refines these proposals by a multiscale cascaded 1D-convolutional anchor network. In contrast to existing works, our coarse-to-fine approach can progressively localize fine-grained action proposals. We conduct extensive experiments on two action benchmarks (THUMOS14 and ActivityNet v1.3) and demonstrate the superior performance of our approach when compared to the state-of-the-art techniques on various video understanding tasks.**

***Index Terms*—Action Proposals, Action Recognition, Action Detection, Video Captioning.**

I. Introduction

With the tremendous increase in online and personal media archives, people are generating, storing and consuming large collections of videos. The trend encourages the development of effective and efficient algorithms to intelligently parse video data and discover semantic information [1]–[5]. One fundamental challenge that underlies the success of these advances is action detection from videos, including a temporal aspect or spatio-temporal aspect [6]. However, compared to significant progress in action recognition [7]–[11], the performance of temporal action detection methods remains unsatisfactory. The main bottleneck is the difficulty of localizing high-quality action proposals, which is a crucial step for bridging the performance gap between recognition and detection. The first category of approaches for localizing action proposals is "detection by classification", which employs a temporal sliding window for initial localization followed by a classification stage [12]–[14]. More specifically, the state-of-the-art methods in this category often formulate temporal proposal generation as a binary classification problem (i.e., action or background) within sliding windows [15], [16].

The second category of proposal detection is temporal grouping, which is usually built on 1D actionness signals. The basic idea is to predict the actionness score of each video frame or clip and form an actionness curve over the whole video. The proposals are then generated based on the actionness curve, and the adjacent proposals may be further grouped into a larger proposal. This type of approach can generate high-quality action proposals with fewer, but potentially better instances [17], [18]. However, the actionness grouping methods can hardly detect the fine-grained action proposals (e.g., two "brush painting" clips in the green box in Fig. 1), which are temporally close to each other or covered by a coarser action (e.g., one "painting" clip in the blue box in Fig. 1). The phenomenon is reflected in the generated actionness curve, which has no clear boundaries between the two fine-grained proposals. It is also observed that an action proposal usually has a distinctive temporal structure. Ignoring the structure leads to missing proposals because the coarse proposals cannot yield fine-grained proposals.

Motivated by the above observations, we propose a novel coarse-to-fine temporal proposal (CFTP) approach to localize temporal action proposals by exploring different action granularities. Our proposed CFTP consists of three stages: a coarse proposal network (CPN) to generate long-time action proposals, a temporal convolutional anchor network (CAN) to localize finer proposals, and a proposal reranking network (PRN) to further identify proposals from previous stages. Specifically, CPN first builds the actionness curve by leveraging three actionness measurements, i.e., pointwise, pairwise and recurrent actionness, which are complementary to each other. Then, a

Manuscript received September 18, 2018; revised June 1, 2019 and September 6, 2019; accepted September 9, 2019. Date of publication September 23, 2019; date of current version May 21, 2020. This work was supported in part by the 973 Programme under contract No. 2015CB351803. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mohammed Daoudi. *(Corresponding author: Xinmei Tian.)*

F. Long, Z. Qiu, and X. Tian are with the Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China (e-mail: longfc.ustc@gmail.com; zhaofanqiu@gmail.com; xinmei@ustc.edu.cn).

T. Yao and T. Mei are with the Vision and Multimedia Lab, JD AI Research, Beijing 100105, China (e-mail: tingyao.ustc@gmail.com; tmei@live.com).

J. Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14604 USA (e-mail: jluo@cs.rochester.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2019.2943204

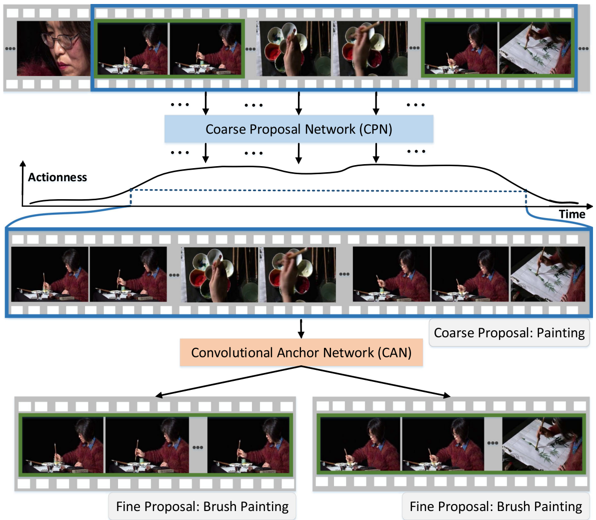

Coarse Proposal Network (CPN)
Actionness
Time
Coarse Proposal: Painting
Convolutional Anchor Network (CAN)
Fine Proposal: Brush Painting
Fine Proposal: Brush Painting

Fig. 1. We investigate action granularities for temporal proposal localization where a coarse action proposal of "painting" in the blue bounding box may consist of two fine action proposals of "brush painting" in the green bounding box (the upper part). The task is to discriminate these two fine-grained "brush painting" proposals. Previous research predominantly focuses on generating a coarse actionness curve that can be used to localize only the coarse action of "painting" because the actionness curve is not that discriminative. Our idea is to further capture hierarchical action granularities and localize fine-grained action proposals in a coarse-to-fine manner. For example, we can first detect a coarse "painting" proposal in the first granularity with a coarse proposal network (the middle part) and further localize two fine-grained "brush painting" proposals in a finer granularity by using a temporal anchor network (the lower part).

watershed actionness grouping method is employed to localize coarse action temporal segments. The long coarse proposals are fed into a fine-grained CAN, which is cascaded by several 1D-convolutional temporal anchors. Each temporal anchor consists of two parts: one convolutional layer for multiscale anchor feature map generation and one predictor with multiple scale ratios at each feature map level. Meanwhile, two types of losses, i.e., binary classification loss and temporal bounding box regression loss, are optimized on top of each anchor module. Finally, a proposal reranking network leverages the proposal scores from previous coarse- and fine-grained networks to identify the final proposals. Unlike the existing techniques, our coarse-to-fine approach is able to progressively localize fine-grained action proposals. We conduct extensive experiments on two action benchmarks (THUMOS14 and ActivityNet v1.3) and demonstrate the superior performances of our approach over the state-of-the-art methods on various video analysis tasks.

The main contributions of this work include 1) analyzing action granularities and 2) localizing temporal action proposals in a coarse-to-fine manner. Specifically, we propose a novel coarse-to-fine temporal proposal architecture to implement this idea and localize more precise action proposals. The remaining sections are organized as follows. Section II describes the related works. Section III details our coarse-to-fine temporal proposal (CFTP) approach. The experimental results of CFTP on temporal action proposal localization are provided in Section IV. Section V and Section VI further present the empirical evaluations of CFTP on the task of action detection and dense video captioning. Finally, Section VII concludes the paper.

II. Related Work

We briefly group the related works into two categories: temporal action detection and temporal action proposal. The former focuses on detecting action clips of known classes, while the latter investigates how to precisely localize video segments that contain interesting activities.

A. Temporal Action Detection

The research in this direction has proceeded along two different dimensions: weakly supervised action detection and supervised action detection. For weakly supervised temporal action detection, the training data contain only video-level category labels but no temporal annotations. Researchers often formulate the problem as a weakly supervised setting and alleviate the problem by using transfer learning [19] or multi-instance learning to learn key evidence for temporal localization [20]. In the direction of supervised temporal action detection, most of the works utilize sliding windows as candidates and focus on designing hand-crafted feature representations for classification [14], [21], [22]. Inspired by the success of deep networks in object detection [23], [24], action detection capitalizes on some deep models to improve performance [18], [25]–[33]. For example, Shou *et al.* developed a two-stage segment-based 3D CNN model (S-CNN) [30] including one proposal network to detect action segments and one classification network to predict segment-level action scores. Similar in spirit, temporal structure segment network detection (TSSND) [18] generates proposals with an actionness grouping algorithm in the first stage and then assigns detection scores via a structure segment network in the second stage. Considering that the separation of two-stage detection may result in suboptimal solutions, one-stage approaches that combine temporal proposal and classification have been further studied. Xu *et al.* proposed an end-to-end region-C3D action detection model (R-C3D) [28] with extended 3D region of interest (RoI) pooling. The single stream temporal action detection (SS-TAD) [27] utilizes a recurrent neural network (RNN)-based architecture to jointly learn action proposals and classification. In addition, with the development of reinforcement learning, Yeung *et al.* [31] explored RNN to learn a glimpse policy for predicting the starting and ending points of actions in an end-to-end manner.

B. Temporal Action Proposal

We summarize the approaches on temporal action proposal mainly into two directions: content-independent proposal and content-dependent proposal. The content-independent algorithms usually generate proposals by uniform sampling or a sliding window [14], [21]. For example, Oneata *et al.* [14] exploited candidate windows of 60 frames and slid the windows in steps of 30 frames. However, both of these methods lead to huge computations for further classification because of the large number of proposals. In contrast, content-dependent proposal methods, e.g., [15], [18], [34]–[36], utilize action proposal labels during training. For instance, Escorcia *et al.* [15] leveraged Long Short-Term Memory cells to learn an appropriate video

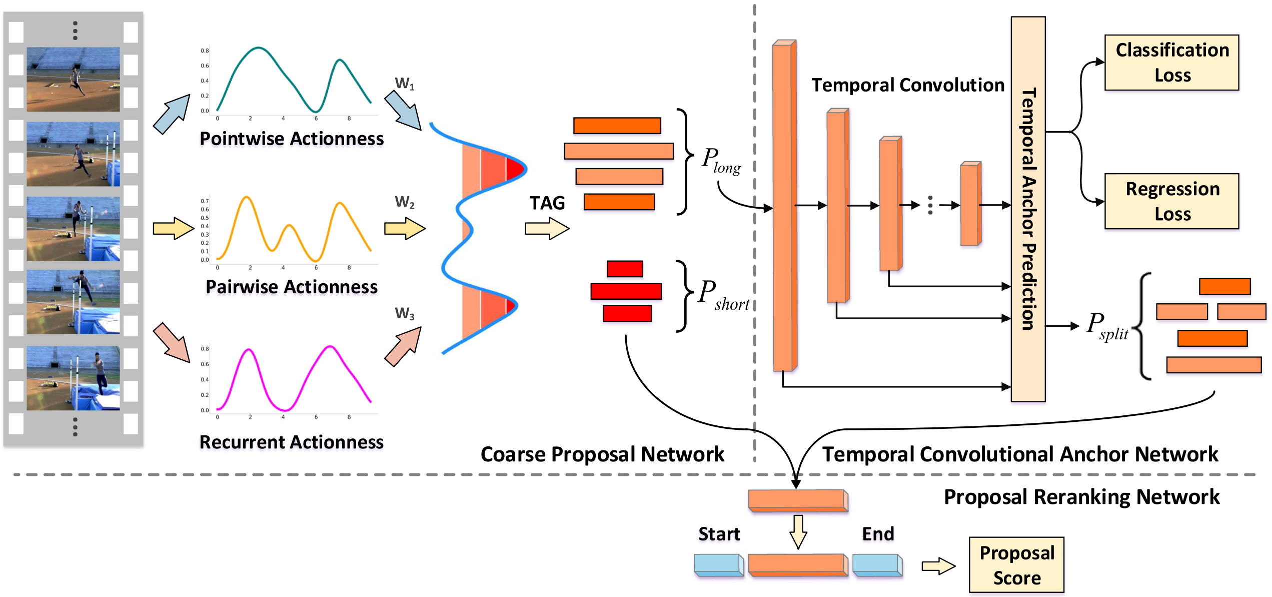


Fig. 2. An overview of our coarse-to-fine temporal proposal (CFTP) architecture (better viewed in color). In the coarse proposal generation stage, proposal candidates are generated by a watershed temporal actionness grouping algorithm (TAG) based on actionness curves. Considering the diversity of action proposals, three actionness measures (namely pointwise, pairwise and recurrent) that are complementary to each other are leveraged to produce the final actionness curve. Next, we feed long proposals P_{long} into the temporal convolutional anchor network for finer proposal P_{split} generation. The temporal convolutional anchor network consists of multiple 1D convolutional layers to generate temporal instances for proposal/background binary classification and bounding box regression. Given the short proposals P_{short} from the coarse stage and fine-grained proposals P_{split} from the temporal convolutional anchor network, a reranking network is utilized for proposal refinement. When further considering video temporal structures, we extend the current part of the proposal with its start and end parts. The duration of the start and end part is half of the current part. The proposal is then represented by concatenating features of each part to exploit the context information.

sequence encoding as a set of discriminative states to indicate proposal scores. Although the method avoids running sliding windows of multiple scales, there is still the need to run an overlapping sliding window, especially when the video duration is long. To address this problem, single stream temporal proposal (SST) [35] generates proposals with only one single pass by utilizing a recurrent GRU-based model, and the recent temporal unit regression network (TURN) [36] builds video units in a pyramid manner to avoid window overlapping. Inspired by the idea of SSD [37], Lin *et al.* [34] utilized a 1D convolution to generate multiple temporal action anchor instances for action proposal and detection. In contrast to the above methods which generate proposals in a fixed multiscale manner, Zhao *et al.* [18] proposed a more flexible actionness grouping method to localize action time intersections in an actionness confidence curve. Nevertheless, the temporal actionness grouping may fail when two action segments are very close or covered by a coarse action instance because the video temporal structure is ignored in the method.

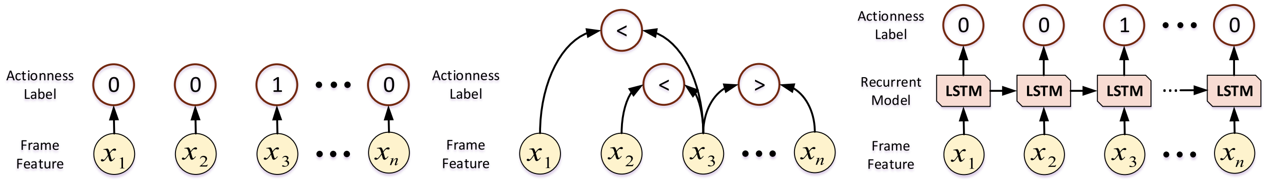
In summary, our approach belongs to content-dependent proposal methods. The aforementioned approaches often ignore the fact that action videos have coarse-to-fine temporal structures that play an important role in proposal generation. Our work in this paper contributes by not only generating more accurate coarse proposals through leveraging different levels of actionness but also elegantly deciding how to split and refine a coarse proposal into more fine-grained proposals through accurate localization of action intersections in videos.

III. Coarse-to-Fine Temporal Proposal (CFTP)

In this section, we present the proposed coarse-to-fine temporal proposal (CFTP) architecture in detail. Fig. 2 shows an overview of our architecture, which consists of three components: a coarse proposal network (CPN) with the fusion of multiple actionness curves to generate coarse action proposals, a temporal convolutional anchor network (CAN) for localizing finer proposals, and a proposal reranking network (PRN) to further refine proposals from previous stages. Specifically, the CPN is designed with multiple actionness grouping at three different levels (pointwise, pairwise and recurrent), while the CAN utilizes 1D-convolutional temporal anchors for fine-grained proposal generation. We discuss CPN and CAN in Section III-A and Section III-B, respectively, followed by presenting PRN in Section III-C.

A. Coarse Proposal Network (CPN)

The basic temporal action proposal methods exploit the sliding window to split a video into multiple clips [13], [14]. However, these types of algorithms tend to produce many redundant and incomplete proposals, impairing the performance of the next step of detection. An alternative direction is to localize the specific clips with high actionness scores, and we follow this recipe in our work. Moreover, considering that the training samples in actionness learning are from different action categories and thus very diverse, we leverage three types of actionness (namely pointwise, pairwise and recurrent) learning for final actionness



(a) Pointwise actionness (b) Pairwise actionness (c) Recurrent actionness

Fig. 3. The overview of three types of actionness learning: (a) Pointwise actionness learning, (b) Pairwise actionness learning, and (c) Recurrent actionness learning.

prediction. The three actionness learning characterize actions at different levels and are complementary to each other.

1) Pointwise Actionness: Pointwise actionness (Fig. 3(a)) is a binary classifier that treats each frame independently. Given the feature set $X = \{x_i | x_i \in \mathbb{R}^D\}_{i=1}^n$ extracted from n video frames, we split it into positive subset X^+ from the ground truth proposal and negative subset X^- from the video background. The binary classifier $\mathcal{M}_b$ is optimized via the following loss function:

$$L_b = -\sum_{x \in X^+} \log(\mathcal{M}_b(x)) - \sum_{x \in X^-} \log(1 - \mathcal{M}_b(x)). \qquad (1)$$

2) Pairwise Actionness: In the action proposal scenario, we also take the actionness degree within a video into account, which represents a relative measure. This can be formulated as a supervised ranking problem. Given a pair of frames, one from the proposal and the other from the background of the same video, we aim to optimize the pairwise classifier (Fig. 3(b)), which outputs a higher score of the frame from the proposal than that from the background. Formally, assume that we have the frame feature pairs $P = \{(x^+, x^-) | x^+ \in X^+, x^- \in X^-\}$, where each pair (x^+, x^-) consists of a positive feature x^+ and a negative feature x^- from an identical video. The loss function of the pairwise actionness classifier $\mathcal{M}_p$ is given by

$$L_p = \sum_{(x^+, x^-) \in P} \max(0, 1 - \mathcal{M}_p(x^+) + \mathcal{M}_p(x^-)). \qquad (2)$$

3) Recurrent Actionness: In addition to pointwise and pairwise learning, recurrent information is additionally explored for actionness prediction. We observe that the action label of one single frame is highly related to the previous frames, which can be formulated as a recurrent prediction problem. Hence, we exploit the recurrent model (Fig. 3(c)) as our third actionness classifier. Specifically, in each iteration, we sample several consecutive frames as the input sequence and the recurrent classifier $\mathcal{M}_r$ with a single-layer LSTM is optimized to predict the actionness label.

4) Actionness Fusion: After training three actionness classifiers $\mathcal{M}_b$, $\mathcal{M}_p$ and $\mathcal{M}_r$, the actionness score S_b, S_p and S_r is produced by each model, and then normalized to [0,1] by a sigmoid operation. We compute the final actionness score by linearly fusing the three scores as

$$S = w_1 S_b + w_2 S_p + w_3 S_r, \qquad (3)$$

where w_1, w_2 and w_3 are fusion weights determined by cross validation. The final score takes advantage of three different aspects and reflects an obvious "peak" in the actionness curve for actions at different levels.

Algorithm 1: Watershed Temporal Actionness Grouping.

```
Input:
  Input video actionness score vector $S = \{s_i\}_{i=1}^n$;
  Watershed level step size $l$;
  Watershed grouping length step size $g$;
  Non-maximal suppression IoU threshold $th$;
Output:
  Video coarse proposal set $\mathbf{P}$;
Initialize watershed level $L = l$, watershed grouping
  length $G = g$, proposal set $\mathbf{P} = \varnothing$;
while $L < 1.0$ do
    while $G < 1.0$ do
        Find the non-overlap consecutive integer interval
          set: $I = \{I_m\}_{m=1}^K = \{[a_m, b_m] | x_i > L, i \in [a_m, b_m]\}_{m=1}^K$, the intervals are in ascending order
          of time;
        Initialize interval index $m = 1$;
        while $m \le K$ do
            Initialize grouping index $z = m$;
            while $(b_z - a_m)/N < G \mathbf{and} z \le K$ do
                $z = z + 1$;
            end while
            Add interval $[a_m, b_z]$ to proposal set $\mathbf{P}$, assign
              average score $P_s = \frac{1}{b_z - a_m + 1} \sum_{i=a_m}^{b_z} s_i$ to the
              proposal;
            $m = z + 1$;
        end while
        $G = G + g$;
    end while
    $L = L + l$;
end while
Apply non-maximal suppression with IoU threshold $th$
  to $\mathbf{P}$;
return $\mathbf{P}$
```

5) Temporal Actionness Grouping: Based on the final actionness scores, we utilize the watershed temporal actionness grouping method to generate action proposals, and the details are given in Algorithm 1. We set the watershed level step size l to 0.085 and watershed grouping length step size g to 0.025. The non-maximal suppression (NMS) is utilized to filter out highly overlapped proposals, and the intersection over union

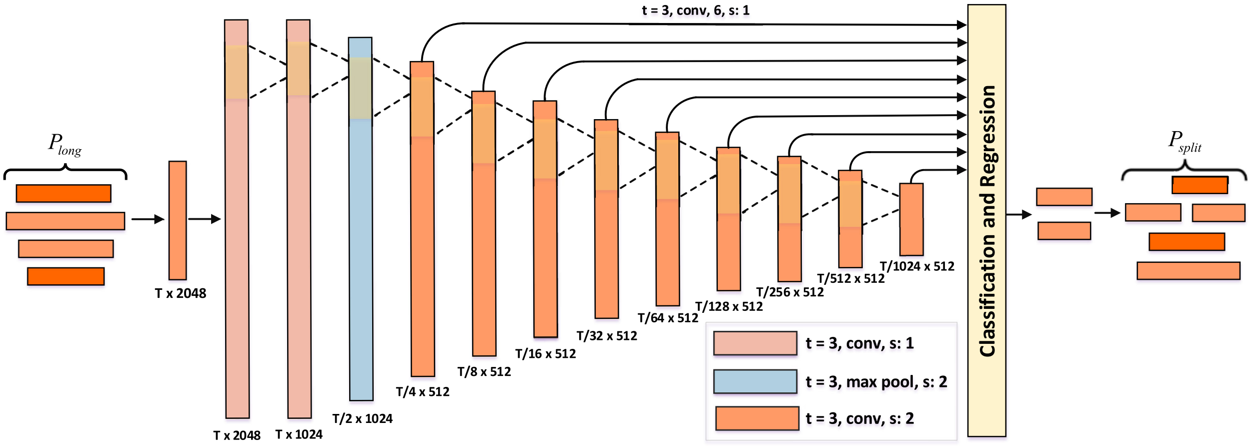


Fig. 4. The detailed network architecture of our temporal convolutional anchor network (CAN). The ReLU active functions after each convolutional layer are not included. ($t = 3$: 1D temporal convolution and temporal dimension is 3; s: stride size.)

(IoU) threshold th of NMS is fixed to 0.95. Because the shot boundaries are not always accurate in the actionness curve, the proposals that are temporally close to each other are easily merged to a longer one by the watershed and temporal group operations. Therefore, we call the proposals in this stage "coarse proposals," which should be further refined.

B. Temporal Convolutional Anchor Network (CAN)

The coarse proposals output from the CPN can be divided into two parts, i.e., P_{long} and P_{short}, according to the duration. Considering that the proposals in P_{long} often contain multiple action instances, the temporal convolutional anchor network (CAN) is then devised and exploited to split and refine the coarse proposals, as shown in Fig. 2. Each time, we feed one coarse proposal from P_{long} into the network, and the fine-grained proposals P_{split} are produced as a more accurate proposal set. Specifically, two main characteristics from region proposal works [23], [37] are integrated into the design of proposal refinement in our CAN: 1) a set of small convolutional filters for generating feature sequences at various time scales; 2) at each time scale, separate predictors on different temporal intervals, namely anchors, are conducted for proposal classification and temporal bounding box regression.

Fig. 4 details the architecture of the CAN. Given one feature set $X_{long} = \{x_i | x_i \in \mathbb{R}^D\}_{i=k}^{k+T}$ extracted from a T-frame proposal (starting from k-th frame) in P_{long}, we first aggregate all the features as one feature map with the size of $T \times D$ and then feed the feature map into two 1D-convolutional layers plus one max-pooling layer to shorten its temporal dimension T or increase the size of the temporal receptive field. Next, nine cascaded 1D temporal convolutional layers are employed to generate a series of output feature maps with different time scales. Each output feature map of the 1D-convolutional layer is further injected into the prediction layer to produce a fixed set of proposals. Specifically, given an output feature map from the j-th layer with temporal length T_j and feature dimension D_j, the basic element (anchor) for predicting the parameters of a proposal is a $3 \times D_j$ feature map cell that outputs a prediction score vector $s_{pred} = (s_{cls}, \Delta c, \Delta w)$ via convolutional layers. $s_{cls} = [s_{ps}, s_{bk}]$ denotes the 2-dimensional classification score for the proposal/background. Δc and Δw are two temporal offsets relative to the default center location a_c and width a_w of this anchor, which are used to adjust its temporal coordinates as

$$\varphi_c = a_c + \alpha_1 a_w \Delta c \quad \text{and} \quad \varphi_w = a_w \exp{(\alpha_2 \Delta w)}, \tag{4}$$

where φ_c and φ_w are the refined center location and width of the anchor, respectively. α_1 and α_2 are utilized to control the impact of the temporal offsets. Furthermore, derived from the idea of anchor boxes in [23], [37], we associate a set of default temporal boundaries with each feature map cell. The multiple temporal scale ratios for these default temporal boundaries are denoted as $R = \{r_d\}_{d=1}^{3} = [1, 1.25, 1.5]$, resulting in a total of $3T_j$ anchors in the j-th layer. For each temporal scale ratio r_d, we achieve one default center location $a_c = (t + 0.5)/T_j$ and width $a_w = r_d/T_j$ of the t-th feature map cell. Finally, 2-dimensional classification scores (s_{ps}, s_{bk}) or offsets $(\Delta c, \Delta w)$ are measured in the classification loss or regression loss layer for each ratio, respectively. As such, we set the output dimension of the convolutional layer (curved arrow) after the j-th feature map for classification or regression to 6, as shown in Fig. 4.

In the training stage, we accumulate all the anchors from the cascaded 1D-convolutional layers and produce the proposal for each anchor through a prediction layer. The overall training objective in our CAN is formulated as a multitask loss by integrating the proposal/background classification loss (L_{cls}) to distinguish proposals from backgrounds and temporal regression loss (L_{reg}) to adjust temporal coordinates of proposals, which is written as

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

where β is the trade-off parameter. Specifically, we measure the classification loss L_{cls} via the softmax loss:

$$L_{cls} = -(1 - y) \log(1 - s_{bk}) - y \log{(s_{ps})}, \tag{6}$$

where $y \in [0, 1]$ represents the binary label of this anchor. We denote g_{iou} as the intersection over union (IoU) between the temporal receptive field of this anchor and its corresponding closest ground truth. If the highest g_{iou} of this anchor is larger than 0.7, we set $y = 1$, otherwise $y = 0$. The temporal regression loss is devised as a smooth L1 loss [38] (S_{L1}) between the predicted proposal with $y = 1$ and the closest ground truth instance of the anchor, which is computed by

$$L_{reg} = S_{L1}(\varphi_c - g_c) + S_{L1}(\varphi_w - g_w), \tag{7}$$

where g_c and g_w represent the center location and width, respectively, of this anchor's closest ground truth instance.

In the prediction stage, the fine-grained proposals in P_{split} consist of the predicted refined center location φ_c, width φ_w and proposal score s_{ps} of each anchor.

C. Proposal Reranking Network (PRN)

Given the short proposals P_{short} with an actionness score for each from the CPN and fine-grained proposals P_{split} with classification probabilities from the CAN, a valid question is how to rank these proposals. Obviously, directly ranking the proposals by sorting different kinds of scores will result in discrepancies in between. As a result, the proposal reranking network (PRN) is exploited to recalculate the score for each proposal, making the proposals comparable. Furthermore, the temporal context information close to the proposal is explored in the PRN to enhance the score prediction.

In general, a proposal can be represented by average pooling the features of all the frames in the proposal. Considering that further taking context information has shown superior effectiveness in representation learning, we build a context-augmented proposal representation by leveraging the information before and after each proposal, as shown in Fig. 2. Assume the input proposal $p = [t_s, t_e] \in P_{short} \cup P_{split}$ has the starting time t_s, ending time t_e and duration $d = t_e - t_s$, we additionally take the start interval $p_s = [t_s - d/2, t_s]$ and end interval $p_e = [t_e, t_e + d/2]$ as the context information. Then, a proposal-level representation f_{p^*} is conducted as $f_{p^*} = [f_{p_s}, f_p, f_{p_e}]$ by concatenating f_{p_s}, f_p and f_{p_e}, which are the average-pooled features in p_s, p and p_e, respectively. Based on this proposal-level feature f_{p^*}, a proposal-level binary classifier is optimized via the following loss function:

$$L_{\text{rank}} = \sum_{p} \max(0, 1 - y_p * \tanh(\boldsymbol{w}_c^T f_{p^*} + b_c)) \text{ s.t. } y_p \in [-1, 1], \tag{8}$$

where $\{\boldsymbol{w}_c, b_c\}$ are linear parameters of the classifier. y_p represents the label of proposal p, which is equal to 1 only if the IoU between the proposal and ground truth is higher than 0.7. After training, the output score $\tanh(\boldsymbol{w}_c^T f_{p^*} + b_c) \in [-1, 1]$ is assigned to each proposal $p \in P_{short} \cup P_{split}$ as the final reranked proposal score. To select the top proposals for evaluation, we apply the standard post-processing techniques [15], [35] with a non-maximum suppression threshold of 0.6 to eliminate near-duplicate detections.

IV. Experiments

We empirically verify the merit of our CFTP by conducting the experiments of video temporal proposal on two popular video recognition benchmarks, i.e., ActivityNet v1.3 [41] and THUMOS14 [42].

A. Datasets

The **ActivityNet v1.3** dataset contains 19,994 videos in 200 classes collected from YouTube. The dataset is divided into three disjoint subsets: training, validation and testing, by 2:1:1. All videos in the dataset have temporal annotations. The labels of the test set are not publicly available, and the performances of the temporal proposal on the ActivityNet dataset are reported on the validation set. The **THUMOS14** dataset has 1,010 videos for validation and 1,574 videos for testing from 20 classes. Among all the videos, there are 220 and 212 videos with temporal annotations in the validation and testing set, respectively. Following [18], we train the model on the validation set and perform the evaluation on the testing set.

B. Experimental Settings

1) Frame Representations: We extract three widely adopted frame representations, i.e., 1,024-way activations from the global_pool layer in the BN-inception model (BN) [39] pretrained on ImageNet ILSVRC12 [43], 4,096-way outputs from the fc7 layer in the 3D convolutional model (C3D) [8] following PCA to 500 dimensions and 2,048-way outputs from the pool5 layer in the Pseudo-3D model (P3D) [40]. The C3D and P3D models are both pretrained on Sport1M [44]. The sample rates of frames used in the three representations are 10, 8 and 8, respectively.

2) Implementations of CPN: In all three actionness learning stages, we utilize one fully-connected layer followed by ReLU to embed input features, and the output dimension is set as the same as the input features. In recurrent actionness learning, we further exploit a one-layer LSTM to model the states of action temporal proposal in each frame and produce the hidden/output representations with the dimension of 2,048. The length of the input sequence to LSTM is fixed to 100, and the sliding window stride is set as 50. In the combination of three actionnesses, the fusion weights w_1, w_2 and w_3 are set as 0.4, 0.4 and 0.2, respectively by cross validation. We extract the coarse proposals P_{long} whose duration is more than 75% of the whole video as the inputs to the temporal convolutional anchor network. The remains are taken as P_{short}.

3) Implementations of CAN: We exploit 9 temporal anchor layers to obtain different proposal anchor instances during training, and the dimension D_j of each layer's output is set as 512. The kernel size of the temporal dimension is 3, and the temporal stride size is 2. We use three temporal scale ratios $R = \{r_d\}_{d=1}^{3} = [1, 1.25, 1.5]$ in each scale of the feature map. For each ratio, 2-dimensional classification scores (proposal/background) or 2-dimensional offsets (center location/width of the anchor) are measured in classification loss or regression loss layer, respectively. As such, we set the output dimension of the convolutional layer before classification/regression loss layer to 6. α_1, α_2, and β are determined on a validation set and finally set to 1.0, 1.0, and 0.5.

4) Training Configuration: We implement the CFTP architecture on the Caffe [45] platform. In all experiments, our networks are trained by utilizing stochastic gradient descent with 0.9 momentum. The initial learning rate is set as 0.001 and decreased by 10% after every 500 iterations on THUMOS14 and 4,000 iterations on ActivityNet. The mini-batch size is 128, and the weight decay parameter is 0.0005.

5) Evaluation Protocols: For quantitative evaluation of our proposed models, we adopt three standard evaluation metrics for action temporal proposals: average recall in different IoU (AR), recall-IoU curve (R-IoU) and the area under the average recall vs. average number of proposals curve (AR-AN Area). The last evaluation metric AR-AN Area was first proposed in the ActivityNet 2017 Challenge in *Action Proposal Task* for evaluation. On both the ActivityNet v1.3 and THUMOS14 datasets, the average recall value of n proposals per video in the AR-AN curve is computed by averaging the recall percentage under each IoU threshold in [0.5:0.05:0.95]. The average recall is calculated by averaging the recall value under each IoU threshold with all proposals. We evaluate performances on the top 100 and top 200 returned proposals in ActivityNet v1.3 and THUMOS14, respectively.

C. Compared Approaches

We compare the following state-of-the-art approaches for action proposal task:

1) Sliding window (SW) is the basic temporal action proposal method that exploits different fixed-scale temporal windows and strides to generate proposals. The number of window scales is set to 20. The window length increases exponentially starting from 2.4 seconds long, and the step size of the window is 0.8 times the window length.
2) Uniformly sample (US) generates proposals with the time center and duration sampled from a uniform distribution of [0, T]. The T represents the whole video duration.
3) The fast temporal proposal (FTP) [16] retrieves high-quality action temporal proposals based on clip-level representations learned by sparse dictionary.
4) The temporal single shot action proposal (TSSAP) [34] is a variant version of the single-shot action detection (SSAD) [34], which generates action proposals with a temporal anchor layer with multiple aspect-ratio predictors.
5) The temporal structure segment network proposal (TSSNP) [18] generates temporal action proposals by grouping only the pointwise actionness curve. The score of each frame is predicted by temporal segment networks [10].
6) The single stream temporal action proposal (SST) [35] builds an RNN-based action proposal network, which could be implemented continuously in a single stream over long video sequences to produce action proposals.
7) The temporal unit regression network (TURN) [36] first decomposes untrimmed video into basic successive short clips (video units). Then, the method jointly predicts the proposal score of the sliding window on video units and refines the temporal boundaries by temporal coordinate regression.
8) The coarse-to-fine temporal proposal (CFTP) is our proposed approach. Specifically, we design three runs: CTP, CFTP$^-$, and CFTP. The CTP consists only of a coarse proposal network (CPN) with a watershed temporal actionness grouping algorithm for proposal generation. The CFTP$^-$ further incorporates temporal convolutional anchor networks (CAN) into the CTP and generates finer action proposals. The run of the CFTP identifies proposals from coarse and fine-grained stages with a reranking proposal network (RPN).

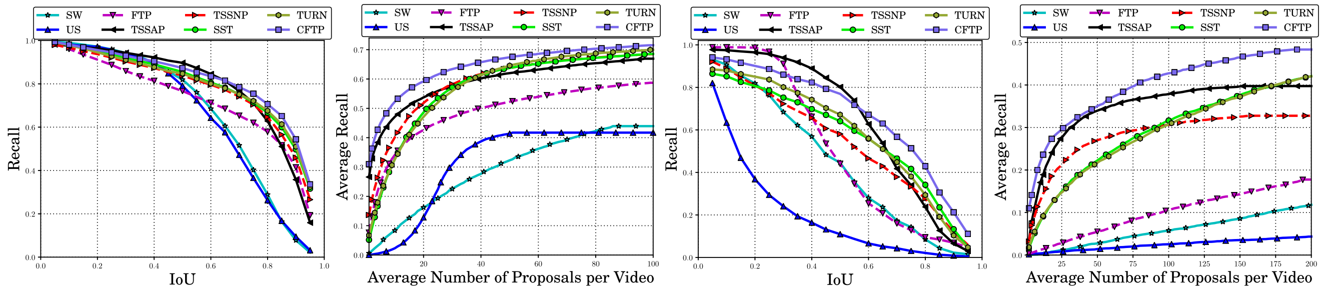
D. Performance Comparison

1) Quantitative Analysis: Table I summarizes the area under average recall vs. average number of proposals curve (AR-AN Area) and average recall (AR) performances by using features extracted from the BN, C3D and P3D models on two action datasets. Overall, the results across different features consistently indicate that our proposed CFTP leads to a performance boost against other baselines. In particular, on the ActivityNet dataset, the area under the AR-AN curve of CFTP achieves 64.52% on the features by the P3D model, making the absolute improvement over FTP, TSSAP, TSSNP, SST and TURN by 15.8%, 5.8%, 7.2%, 4.9% and 4.7%, respectively. As expected, SW and US perform poorly on both datasets because the two simply perform the proposal selection irrespective of video content. CTP, by exploring three actionness curves, improves TSSNP, which capitalizes on only a pointwise actionness curve on all settings. The results basically indicate the advantage of leveraging different measures to characterize actionness, which is very diverse across different actions. There is also a performance gap between TSSAP and our CFTP$^-$. Although both runs involve utilization of temporal convolutional anchor networks, they are fundamentally different in the way that the performance of TSSAP is as a result of taking the original video as the input to the anchor networks, and CFTP$^-$ is by feeding the output long proposals of CPN into the networks. Similarly, compared to SST and TURN, which generates or initializes proposals on original videos, CFTP$^-$ offers coarse but stable proposal candidates from actionness learning in CPN. As indicated by our results, localizing proposals in a coarse-to-fine fashion can constantly lead to better performance. Please also note that TURN controls the size of receptive fields through temporal pyramid with prefixed unit numbers (4, 8, 32, etc.). Instead, CFTP$^-$ employs cascaded 1D temporal convolutions on the input proposals during optimization, and with only once forward operation we can handle multisize receptive fields to refine proposals. As a result, CFTP$^-$ is more efficient at predicting action instances with various scales. In addition, CFTP performs better than CFTP$^-$, which verifies the design of our PRN.

Compared to CTP, CFTP$^-$, which is augmented by further refining the coarse proposals through CAN, exhibits better performance. This basically indicates that relying solely on actionness

TABLE I
AR-AN AREA AND AVERAGE RECALL (AR) COMPARISONS ON ACTIVITYNET V1.3 AND THUMOS14. (BN: BN-INCEPTION [39], C3D: 3D-CONVOLUTION [8], P3D: PSEUDO-3D [40])

Approach	ActivityNet v1.3						THUMOS14					
	AR-AN Area			AR			AR-AN Area			AR		
	BN	C3D	P3D	BN	C3D	P3D	BN	C3D	P3D	BN	C3D	P3D
SW	29.18			43.94			5.75			11.83		
US	31.41			41.72			2.44			4.43		
FTP [16]	44.11	47.14	48.69	55.94	58.16	58.71	7.10	9.71	9.92	14.18	16.68	17.75
TSSAP [34]	56.72	57.99	58.67	64.88	66.21	66.93	30.66	33.48	34.83	39.10	39.53	39.74
TSSNP [18]	56.52	56.88	57.33	68.18	68.48	68.56	25.70	27.73	28.18	32.21	32.65	32.76
SST [35]	57.22	59.43	59.58	69.13	69.81	69.89	25.51	28.94	30.23	41.06	42.08	43.38
TURN [36]	58.03	59.22	59.85	69.54	69.79	69.95	26.01	28.56	30.05	41.12	42.03	43.13
CTP	58.87	59.50	59.78	69.33	69.47	69.73	26.13	27.60	31.95	34.43	35.39	40.78
CFTP$^-$	61.81	62.64	63.29	70.91	72.01	72.66	31.11	32.21	35.41	41.46	43.05	46.33
CFTP	**62.31**	**63.92**	**64.52**	**70.91**	**72.01**	**72.66**	**32.01**	**34.56**	**37.10**	**41.46**	**43.05**	**46.33**



(a) (b) (c) (d)

Fig. 5. (a) Recall-IoU curve and (b) AR-AN curve on ActivityNet v1.3, (c) Recall-IoU curve and (d) AR-AN curve on THUMOS14. The frame representations utilized in this comparison are all extracted by the P3D model [40].

prediction is not discriminative enough to locate fine-grained action proposals. CFTP$^-$, in comparison, benefits from the mechanism of coarse-to-fine localization. The chance that a coarse proposal can be distilled into a finer granularity is better. Another observation is that the performance gain of CFTP$^-$ against CTP tends to be larger on THUMOS14 than ActivityNet dataset. This is also not surprising because the average duration of action proposals in THUMOS14 is only ~4.0 seconds and much smaller than that (~50 seconds) of ActivityNet. In other words, the action proposals in THUMOS14 are very fine-grained. The results again empirically verify the power of our approach. Because our PRN reorders only the ranking position of each proposal in the candidate pool and does not change the proposals, the AR value of CFTP is exactly the same as CFTP$^-$, which is as expected.

2) *Qualitative Analysis:* Fig. 5 further shows the recall-IoU curve and AR-AN curve of different approaches with the frame representations extracted by the P3D model on two datasets. For a fair comparison, the recall values reported in the recall-IoU curve are computed on the same number of returned proposals, i.e., 100 for ActivityNet and 200 for THUMOS14, across different IoU thresholds. In the case of action proposal, the recall on high IoUs is basically more important, as some of the predicted proposals may hit the ground truth by chance with a low IoU. As depicted in the figure, the improvements of our CFTP become obvious when IoU exceeds 0.7 on both datasets. The results indicate that a larger degree of improvement is attained when an action proposal can be correctly localized. In terms of the AR-AN curve, CFTP across different average numbers of proposals per video consistently leads to a performance boost against other baselines on both datasets. Even in the case when fewer than 10 proposals are returned, CFTP still shows apparent improvements, indicating that CFTP benefits from the reranking mechanism and the correct proposals are ranked at the top.

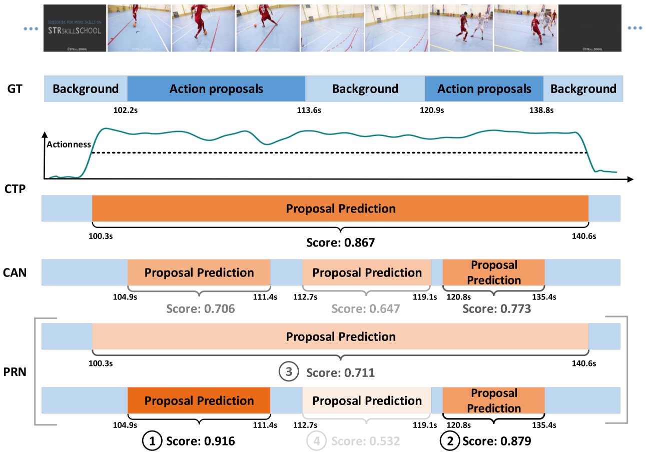


Fig. 6. An illustration of results of each processing step in CFTP.

Fig. 7 shows temporal action proposals on two videos from ActivityNet. The ground truth action proposals and proposals generated by all the compared methods are all given in the figure. As illustrated in the figure, CTP always outputs the correct coarse proposals that contain multiple fine-grained proposals. CFTP further localizes these fine-grained proposals and refines

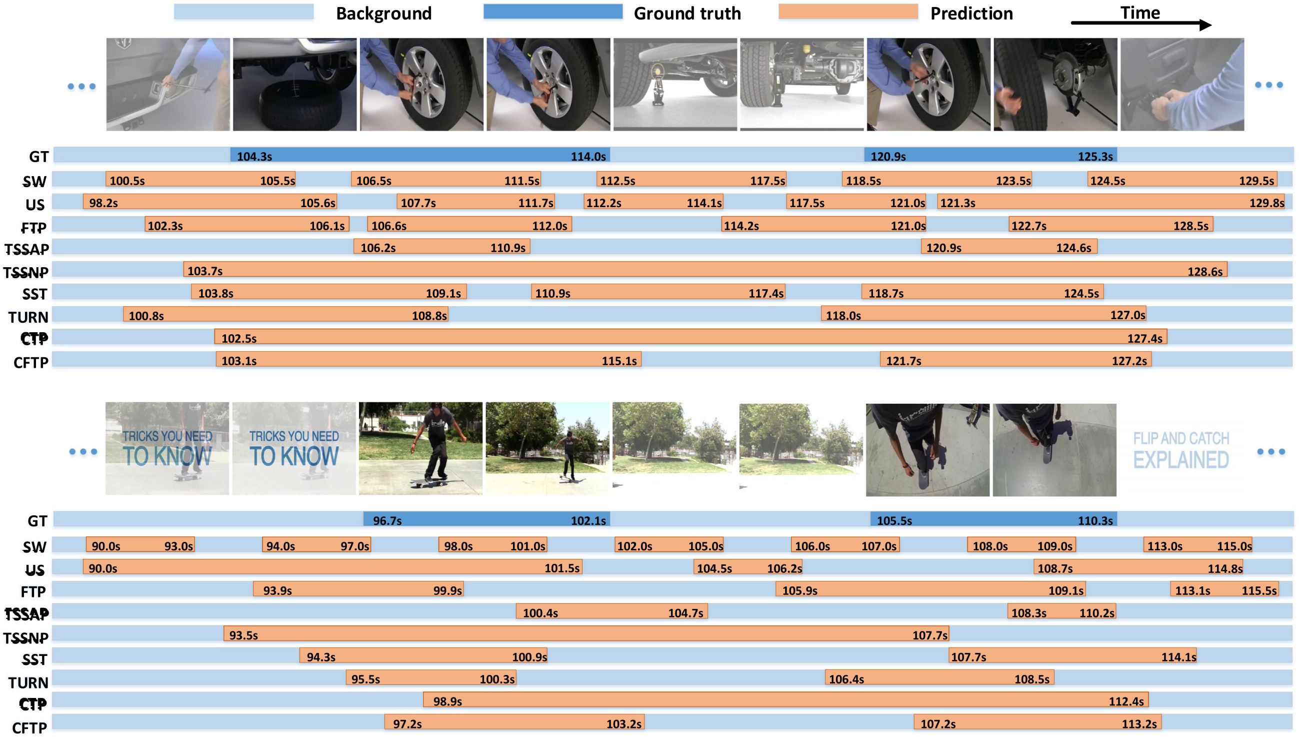


Fig. 7. Examples of temporal action proposals on two videos from ActivityNet v1.3. The boxes in dark blue denote ground truth (GT) action proposals, and the boxes in brown represent the predicted proposals. The proposals generated by CTP and CFTP are given when IoU = 0.7.

TABLE II
GENERALIZABILITY ANALYSIS ON ACTIVITYNET V1.3 DATASETS

Approach	ActivityNet		ActivityNet≤1024	
	AR@50	AR@100	AR@50	AR@100
SST [35]+P3D	22.0	27.4	38.5	42.9
TURN [36]+P3D	23.7	28.1	39.0	43.3
CFTP+P3D	**25.1**	**31.1**	**41.1**	**45.6**

TABLE III
AR-AN AREA AND AVERAGE RECALL (AR) VALUES ON THUMOS14 BY USING DIFFERENT ACTIONNESS IN CPN

Actionness	AR-AN Area	AR
Pointwise	30.73	36.85
Pairwise	31.01	37.75
Recurrent	30.05	34.13
Average Fusion	31.56	39.47
Weighted Fusion	**31.95**	**40.78**

their temporal boundaries in a coarse-to-fine manner. Fig. 6 further illustrates the results of each step in CFTP on one action video. Specifically, CTP computes the actionness curve and generates one coarse action proposal based on the curve, which actually contains two fine-grained proposals. The coarse proposal is further refined by CAN and split into three finer proposals. Because the coarse proposal and finer proposals come from different stages, directly ranking all these proposals on their confidence scores results in discrepancy. As such, we recalculate the scores of all proposals through PRN, making the proposals comparable. Finally, two finer proposals are correctly ranked at the top two positions.

3) Generalizability Analysis: One important property of action proposal approaches is the capability to localize proposals for unseen action categories [15], [35], [36], [46]. Following the similar evaluation protocol in [15], we apply the model trained on THUMOS14 to two sets of ActivityNet v1.3, i.e., the whole validation set of ActivityNet (all 200 classes) and ActivityNet ≤1024 frames (videos from unseen classes with annotations no more than 1024 frames in the validation set). All the comparisons are based on P3D features. As shown in Table II, on the whole validation set, the average recall@100 proposals of CFTP lead to 3.0% and 3.7% improvement over TURN and SST, respectively. On the set of ActivityNet ≤ 1024 frames whose duration statistics on video annotations are similar to that of THUMOS14 but videos are from totally different classes, our CFTP also obtains encouraging performances. The results basically verify the generalizability of CFTP.

4) Effect of the Actionness: Table III details the comparisons by using one of three actionness in the CPN stage to explore the effects of different actionness. Overall, the fusion of three types of actionness leads to better performances against single actionness. The performances of recurrent actionness are lower than those of the other two types of actionness. We speculate that this may be caused by a recurrent model that tends to smooth the actionness curve and thus weakens the responses to severe action changes. The fusion weights are optimally set by cross validation.

5) Effect of the Proposal Duration: To examine how the performance is affected on proposals with different durations, we report the AR-AN Area performances on action instances with respect to different durations in Fig. 8. As expected, CTP can

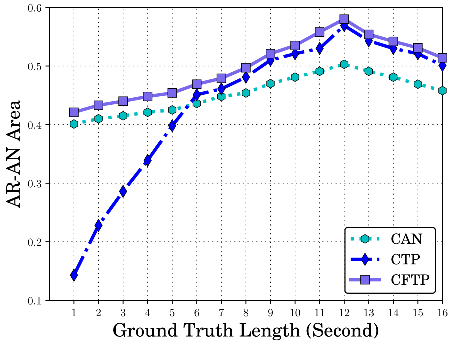


Fig. 8. AR-AN Area performances with different temporal length action ground truth on THUMOS14 (using P3D [40] feature).

TABLE IV
RUN TIME AND THE AREA UNDER AVERAGE RECALL VS AVERAGE NUMBER OF PROPOSALS CURVE PERFORMANCE (AR-AN AREA) OF DIFFERENT VARIANTS OF OUR CFTP (BN: BN-INCEPTION [39]; P3D: PSEUDO-3D [40]). THE RUN TIME ARE REPORTED ON A THREE MINUTES' VIDEO FROM THUMOS14 [42] WITH A REGULAR PC (INTEL DUAL-CORE 3.33 GHZ CPU AND 32GB RAM)

Approach	Time (seconds)		AR-AN Area (%)	
	BN	P3D	BN	P3D
Feature Extraction	19.62	397.61	-	-
CTP	19.73	397.72	26.13	31.95
CFTP$^-$	24.24	402.23	31.11	35.41
CFTP	25.29	403.28	32.01	37.10

TABLE V
THE MAP (IOU = 0.5) PERFORMANCES OF ACTION DETECTION ON THUMOS14

(a) Action Classifier

Method	SVM	S-CNN
DAPs [15]	13.9	-
SST [35]	14.0	23.0
TURN [36]	17.8	25.6
CFTP+C3D	**29.4**	**32.8**

(b) Visual Representations

	SVM	S-CNN
CFTP+BN	28.1	31.5
CFTP+C3D	29.4	32.8
CFTP+P3D	**31.4**	**35.2**

lead to better AR-AN Area performance than CAN on generating long proposals, while CAN outperforms CTP on localizing short/finer proposals whose durations are less than 6 seconds. This confirms that CTP tends to detect long action proposals, but the temporal boundaries are not very accurate. In contrast, CAN feeds the long proposals into multiple temporal 1D-convolutional layers in a cascaded manner and can precisely identify short proposals with clear boundaries. CFTP performing CAN after CTP further improves AR-AN Area values on the proposals of any duration.

Our CFTP may fail in the extreme case when the action proposal is too short, e.g., less than one second. On one hand, CTP cannot detect very short proposals due to the natural continuity of the actionness curve. On the other hand, the initial size of the temporal receptive field in 1D convolutional networks limits the minimum duration of action proposals in CAN. Nevertheless, we can further model the temporal structure and dynamically optimize the temporal scale of each action proposal, which is potentially more effective and will be considered in our future works.

6) Run Time: Table IV lists the detailed run time and the Area under AR-AN curve performances between different variants of our proposed CFTP. The major portion of the run time is consumed by feature extraction, which takes 19.62 and 397.61 seconds by BN-inception [39] and Pseudo-3D [40] architecture, respectively. By performing the coarse proposal network and watershed temporal actionness grouping algorithm, CTP takes another 0.11 seconds on each type of features. By further running the temporal convolutional anchor network and additionally reranking proposals through the proposal reranking network, CFTP$^-$ and CFTP takes 4.51 and 5.56 seconds more on P3D feature, respectively. Meanwhile, CFTP$^-$ and CFTP leads to an improvement over CTP on P3D features by 3.46% and 5.15%, respectively. The run time of CFTP will decrease to approximately 60 seconds on P3D features when testing on a single NVIDIA K40 GPU.

V. CFTP FOR ACTION DETECTION

Next, we examine the impact of CFTP on the action detection task. Without loss of generality, we follow the standard "detection by classification" framework, i.e., first generate proposals by CFTP and then classify proposals. We conduct the experiments from three aspects: 1) exploiting different action classifiers, 2) testing different visual representations utilized in CFTP towards detection performance, and 3) comparing with the state-of-the-art action detection methods. Following the standard measures in the action detection task, we adopt the mAP values computed with the IoU thresholds as the performance metric.

A. Action Classifier

A common need in "detection by classification" methods is an action classifier. We validate two action classifiers on the action proposals produced by our CFTP. One is a one-vs-all linear SVM classifier trained with C = 100 using representations from the C3D model, and the other is the released S-CNN in [30]. Please note that here we also learn CFTP based on representations from C3D for fair comparisons. Table V summarizes the mAP performances with IoU = 0.5 on THUMOS14. The performances of DAPs [15], SST [35] and TURN [36] are directly referred to in the original papers. Performing classification on the proposals by CFTP constantly outperforms other methods across two action classifiers. Compared to TURN, CFTP improves mAP from 17.8% to 29.4% with SVM classifier and from 25.6% to 32.8% with S-CNN. The results indicate the advantage of CFTP on action proposal localization, and thus the chance that an action proposal can be correctly classified is better.

B. Visual Representations

In addition to the evaluations of different visual representations for learning CFTP on action proposals in Table I, we further examine the impact of visual representations exploited

TABLE VI
PERFORMANCE COMPARISONS OF ACTION DETECTION ON THE TEST SETS OF ACTIVITYNET V1.3 AND THUMOS14. (A) AVERAGE MAP WITH IOU THRESHOLDS BETWEEN 0.5 AND 0.95 WITH A STEP SIZE OF 0.05 ON ACTIVITYNET V1.3. (B) MAP WITH IOU = 0.5 ON THUMOS14

(a) ActivityNet v1.3

Method	Average mAP
UTS-D [47]	14.62
MSB-RNN [48]	17.68
OBU-D [49]	17.83
TAG-D [17]	26.05
TSSND [18]	28.28
CFTP	**29.44**

(b) THUMOS14

Method	mAP	Method	mAP	Method	mAP
MAF-D [50]	8.3	S-CNN [30]	19.0	R-C3D [28]	28.9
LEAR [51]	14.4	SST [35]	23.0	SS-TAD [27]	29.2
SLM-D [52]	15.2	CDC [25]	23.3	TSSND [18]	29.8
RLM-D [31]	17.1	SSAD [34]	24.6	CFTP+C3D	**32.8**
PSDF-SVM [22]	18.8	TURN [36]	25.6	CFTP+P3D	**35.2**

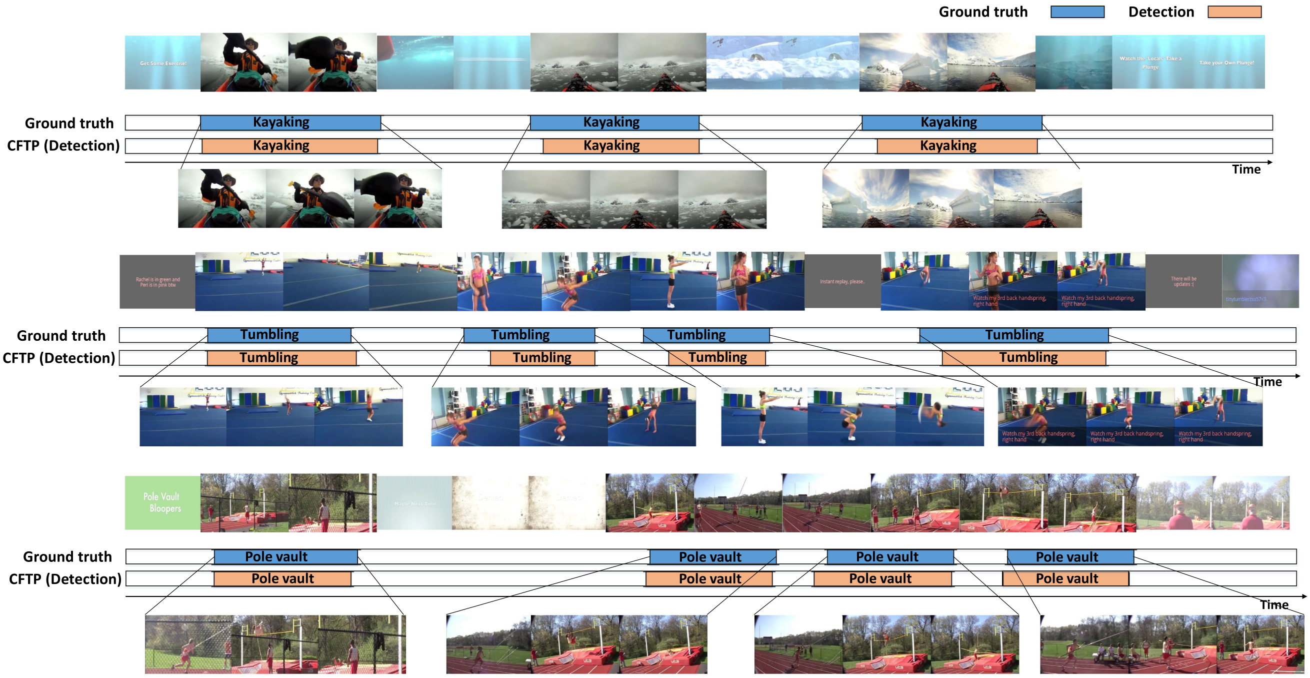


Fig. 9. Examples of three temporal action detection results on ActivityNet v1.3. In each row, a video is shown as a sequence of frames at the top. The blue boxes in the upper bar denote the ground truth proposals, whose sampled frames are illustrated at the bottom. The detection results are shown in the lower bar, where a brown box denotes a predicted proposal from CFTP on the condition of IoU ≥ 0.7.

in CFTP towards detection performance. The performances between the utilization of different visual representations in CFTP are compared on action detection in Table V. As expected, the visual representations extracted by C3D and P3D models that are trained on video data potentially have a higher capability to model temporal structure in videos and lead to more accurate localization of action proposals and better performances on action detection.

C. Comparisons With State-of-the-Art

We compare several state-of-the-art techniques of action detection on both ActivityNet v1.3 and THUMOS14. Following the official evaluation on the ActivityNet test server, we report the average mAP with IoU thresholds between 0.5 and 0.95 (inclusive) with a step size of 0.05 on the ActivityNet v1.3 testing set, and we adopt mAP with IoU $= 0.5$ as a metric on THUMOS14. Please also note that S-CNN released in [30] and fine-tuned on ActivityNet videos is utilized as an action classifier on THUMOS14 and ActivityNet. The performance comparisons are summarized in Table VI. The results across two datasets consistently indicate that our CFTP exhibits better performances than the others. In particular, the average mAP and mAP of CFTP achieves 29.44% and 35.2% on ActivityNet and THUMOS14, respectively, making the improvement over the best competitor TSSND by 1.2% and 5.4%. The performance gain of CFTP tends to be large on THUMOS14, which contains more fine-grained proposals in each video. The results again validate our idea of exploring different granularities to localize fine-grained action proposals. Fig. 9 shows the temporal detection results for three video examples from ActivityNet.

VI. CFTP FOR DENSE VIDEO CAPTIONING

Another task to validate the effectiveness of our temporal action proposal is the task of dense video captioning [53]. The goal of dense video captioning is to localize the action temporal proposals and then describe each proposal with a complete and natural sentence. We conduct our experiments on a recently

TABLE VII
PERFORMANCE COMPARISONS OF DENSE VIDEO CAPTIONING ON THE VALIDATION SET OF ACTIVITYNET CAPTIONS DATASET

Method	METEOR	CIDEr-D
SW	4.51	13.50
US	4.53	13.63
FTP [16]	4.61	14.21
TSSAP [34]	4.76	14.33
TSSNP [18]	4.94	15.16
CFTP	**5.33**	**16.28**

annotated ActivityNet Captions dataset, which provides 3.65 temporally localized sentences on average for each video in ActivityNet v1.3. The standard metrics of METEOR and CIDEr-D are utilized for evaluation in this task.

For a fair comparison, we exploit a popular video captioning system [54] to generate sentences for proposals produced by all the methods. The METEOR and CIDEr-D performances on the validation set of the ActivityNet Captions dataset are reported in Table VII. In general, more accurate temporal proposals lead to better sentence generation performances. Our CFTP improves TSSNP by 1.1% in terms of CIDEr-D, which is considered as a significant progress on this dataset.

VII. CONCLUSIONS

We presented the coarse-to-fine temporal proposal (CFTP) architecture, which explores the hierarchical granularities of actions for the temporal localization of fine-grained action proposals. In particular, we studied the problem of progressively localizing the action proposals in a coarse-to-fine manner. To verify our claim, we devised a coarse proposal network (CPN) and temporal convolutional anchor network (CAN) in our CFTP for this purpose. CPN generates coarse proposals based on three actionness curves, each of which characterizes actions at different level. CAN aims to distill the fine-grained action proposals from the output proposals of CPN through a cascaded temporal anchor network. In addition, a proposal reranking network (PRN) is designed to reorder proposals from CPN and CAN. Experiments conducted on both the ActivityNet and THUMOS14 datasets validated our model and analysis. More remarkably, we achieved superior results over state-of-the-art methods when applying our action proposals to action detection and dense video captioning tasks. Our possible future works include three directions. First, temporal attention will be incorporated into CFTP to further enhance proposal refinement. Second, we will investigate how to end-to-end formulate our action proposal model. Third, we could further model the temporal structure and dynamically optimize the temporal scale of each action proposal.

REFERENCES

[1] C. Xiong, G. Gao, Z. Zha, S. Yan, H. Ma, and T.-K. Kim, "Adaptive learning for celebrity identification with video context," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1473–1485, Aug. 2014.

[2] Z. Qiu, T. Yao, and T. Mei, "Learning deep spatio-temporal dependence for semantic video segmentation," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 939–949, Apr. 2018.

[3] C. Ma, Y. Liu, G. Zhao, and H. Wang, "Visualizing and analyzing video content with interactive scalable maps," *IEEE Trans. Multimedia*, vol. 18, no. 11, pp. 2171–2183, Nov. 2016.

[4] L. Sun, X. Wang, Z. Wang, H. Zhao, and W. Zhu, "Social-aware video recommendation for online social groups," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 609–618, Mar. 2017.

[5] J. Yu, C. Hong, Y. Rui, and D. Tao, "Multitask autoencoder model for recovering human poses," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 5060–5068, Jun. 2018.

[6] G. Yu, N. A. Goussies, J. Yuan, and Z. Liu, "Fast action detection via discriminative random forest voting and top-K subvolume search," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 507–517, Jun. 2011.

[7] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[9] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2011, pp. 3169–3176.

[10] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[11] D. Li, T. Yao, L. Duan, T. Mei, and Y. Rui, "Unified spatio-temporal attention networks for action recognition in videos," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 416–428, Feb. 2019.

[12] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2782–2795, Nov. 2013.

[13] M. Jain, J. van Gemert, H. Jegou, P. Bouthemy, and C. G. Snoek, "Action localization with tubelets from motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 740–747.

[14] D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1817–1824.

[15] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "DAPs: Deep action proposals for action understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 768–784.

[16] F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1914–1923.

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

[18] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2933–2942.

[19] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia, "Temporal localization of fine-grained actions in videos by domain transfer from web images," in *Proc. ACM Multimedia*, 2015, pp. 371–380.

[20] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang, "Video event detection by inferring temporal instance labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2251–2258.

[21] K. Tang, B. Yao, L. Fei-Fei, and D. Koller, "Combining the right features for complex event recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2696–2703.

[22] J. Yuan, B. Ni, X. Yang, and A. A. Kassim, "Temporal action localization with pyramid of score distribution features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3093–3102.

[23] 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.

[24] W. Kuo, B. Hariharan, and J. Malik, "DeepBox: Learning objectness with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2479–2487.

[25] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "CDC: Convolutional-de-convolutional network for precise temporal action localization in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1417–1426.

[26] F. C. Heilbron, W. Barrios, V. Escorica, and B. Ghanem, "SCC: Semantic context cascade for efficient action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1454–1463.

[27] 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. Brit. Mach. Vis. Conf.*, 2017, pp. 93.1–93.12.

[28] H. Xu, A. Das, and K. Saenko, “R-C3D: Region convolutional 3D network for temporal activity detection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5783–5792.

[29] R. D. Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, “Online action detection,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 269–284.

[30] 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.*, 2016, pp. 1049–1058.

[31] S. 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.*, 2016, pp. 2678–2687.

[32] C. Lea, R. V. Michael, D. Flynn, A. Reiter, and G. D. Hager, “Temporal convolutional network for action segmentation and detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 156–165.

[33] 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.*, 2019, pp. 344–353.

[34] T. Lin, X. Zhao, and Z. Shou, “Single shot temporal action detection,” in *Proc. ACM Int. Conf. Multimedia*, 2017, pp. 988–996.

[35] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, “SST: Single-stream temporal action proposals,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6373–6382.

[36] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, “TURN TAP: Temporal unit regression network for temporal action proposals,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3648–3656.

[37] 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.*, 2016, pp. 21–37.

[38] R. Girshick, “Fast R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[39] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[40] Z. Qiu, T. Yao, and T. Mei, “Learning spatio-temporal representation with pseudo-3D residual networks,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5534–5542.

[41] F. 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.*, 2015, pp. 961–970.

[42] Y.-G. Jiang, J. Liu, A. R. Zamir, and G. Toderici, “THUMOS challenge: Action recognition with a large number of classes,” 2014. [Online]. Availiable: http://crcv.ucf.edu/THUMOS14/

[43] O. Russakovsky *et al.*, “ImageNet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[44] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1725–1732.

[45] Y. Jia *et al.*, “Caffe: Convolutional architecture for fast feature embedding,” in *Proc. ACM Multimedia*, 2014, pp. 675–678.

[46] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, “What makes for effective detection proposals?” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 814–830, Apr. 2016.

[47] R. Wang and D. Tao, “UTS at activitynet 2016,” in *Proc. CVPR ActivityNet Challenge Workshop*, 2016, pp. 111–116.

[48] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, “A multi-stream Bi-directional recurrent neural network for fine-grained action detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1961–1970.

[49] G. Singh and F. Cuzzolin, “Untrimmed video classification for activity detection: Submission to ActivityNet challenge,” 2016, *arXiv:1607.01979.*

[50] L. Wang, Y. Qiao, and X. Tang, “Action recognition and detection by combining motion and apperance feature,” in *Proc. ECCV THUMOS Challenge Workshop*, 2014, pp. 17–22.

[51] D. Oneata, J. Verbeek, and C. Schmid, “The LEAR submission at Thumos 2014,” in *Proc. ECCV THUMOS Challenge Workshop*, 2014, pp. 4–10.

[52] A. Richard and J. Gall, “Temporal action detection using a statistical language model,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3131–3140.

[53] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, “Dense-captioning events in videos,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 706–715.

[54] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, “Translating videos to natural language using deep recurrent neural networks,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2015, pp. 1494–1504.

Fuchen Long received the B.E. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2016. He is currently working toward the Ph.D. degree with the School of Information Science and Technology, USTC. His research interests include temporal action proposal and localization, multimedia retrieval and understanding. He has participated several temporal action proposal and detection competitions such as Activity detection in Extended Videos (ActEV-PC) in ActivityNet Challenge 2019, ActivityNet Temporal Action Detection Challenge 2018, and ActivityNet Temporal Action Proposal Challenge 2017.

Ting Yao (M’19) is currently a Principal Researcher with Vision and Multimedia Lab, JD AI Research, Beijing, China. His research interests include video understanding, large-scale multimedia search, and deep learning. Prior to joining JD AI Research, he was a Researcher with Microsoft Research Asia in Beijing, China. He is the Principal Designer of several top-performing multimedia analytic systems in international benchmark competitions such as ActivityNet Large Scale Activity Recognition Challenge 2019–2016, Visual Domain Adaptation Challenge 2018 & 2017, and COCO Image Captioning Challenge. He is the leader organizer of MSR Video to Language Challenge in ACM Multimedia 2017 and 2016, and built MSR-VTT, a large-scale video to text dataset that is widely used worldwide. His works have also led to many awards, including the ACM SIGMM Outstanding Ph.D. Thesis Award 2015 and ACM SIGMM Rising Star Award 2019.

Zhaofan Qiu received the B.E. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2015. He is currently working toward the Ph.D. degree with the Department of Automation, USTC. His research interests include large-scale video classification, semantic segmentation, and multimedia understanding. He has participated in several large-scale video analysis competitions such as ActivityNet Large Scale Activity Recognition Challenge, and THUMOS Action Recognition Challenge. He was awarded the MSRA Fellowship in 2017.

Xinmei Tian (M’13) received the B.E. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2005 and 2010, respectively. She is an Associate Professor with the CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application Systems, USTC. Her current research interests include multimedia information retrieval and machine learning. She was the recipient of the Excellent Doctoral Dissertation of Chinese Academy of Sciences award in 2012 and the Nomination of National Excellent Doctoral Dissertation award in 2013.

Tao Mei (M’07–SM’11–F19) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively. He is a Technical Vice President with JD.com and the Deputy Managing Director of JD AI Research, Beijing, China, where he also serves as the Director of Computer Vision and Multimedia Lab. Prior to joining JD.com in 2018, he was a Senior Research Manager with Microsoft Research Asia, Beijing, China. He has authored or co-authored more than 200 publications (with 12 best paper awards) in journals and conferences, 10 book chapters, and edited 5 books. He holds more than 50 U.S. and international patents (20 granted). He is or has been an Editorial Board Member of IEEE TRANSACTION ON IMAGE PROCESSING, IEEE TRANSACTION ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTION ON MULTIMEDIA, *ACM Transaction on Multimedia, Pattern Recognition*, etc. He was elected as a Fellow of IEEE (2019), a Fellow of IAPR (2016), a Distinguished Scientist of ACM (2016), and a Distinguished Industry Speaker of IEEE Signal Processing Society (2017) for his contributions to large-scale multimedia analysis and applications.

Jiebo Luo (S’93–M’96–SM’99–F09) joined the Department of Computer Science with the University of Rochester in 2011, after a prolific career of more than 15 years with Kodak Research. He has authored more than 400 technical papers and holds more than 90 U.S. patents. His research interests include computer vision, machine learning, data mining, social media, and biomedical informatics. He was the Program Chair of the *ACM Multimedia* 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, and on the Editorial Boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON BIG DATA, *Pattern Recognition, Machine Vision and Applications*, and *ACM Transactions on Intelligent Systems and Technology*. He is also a Fellow of ACM, AAAI, SPIE and IAPR.