Exploiting Informative Video Segments for Temporal Action Localization

Che Sun, Hao Song, Xinxiao Wu, *Member, IEEE,* Yunde Jia, *Member, IEEE,* and Jiebo Luo, *Fellow, IEEE*

***Abstract*—We propose a novel method of exploiting informative video segments by learning segment weights for temporal action localization in untrimmed videos. Informative video segments represent the intrinsic motion and appearance of an action, and thus contribute crucially to action localization. The learned segment weights represent the informativeness of video segments to recognizing actions and help infer the boundaries required to temporally localize actions. We build a supervised temporal attention network (STAN) that includes a supervised segment-level attention module to dynamically learn the weights of video segments, and a feature-level attention module to effectively fuse multiple features of segments. Through the cascade of the attention modules, STAN exploits informative video segments and generates descriptive and discriminative video representations. We use a proposal generator and a classifier to estimate the boundaries of actions and classify the classes of actions. Extensive experiments are conducted on two public benchmarks: THUMOS2014 and ActivityNet1.3. The results demonstrate that our proposed method achieves competitive performance compared with the state-of-the-art methods. Moreover, compared with the baseline method that equally treats video segments, STAN achieves significant improvements with the mAP increased from 30.4% to 39.8% on the THUMOS2014 dataset and from 31.4% to 35.9% on the ActivityNet1.3 dataset, demonstrating the effectiveness of learning informative video segments for temporal action localization.**

***Index Terms*—Temporal Action Localization, Informative Video Segments, Supervised Temporal Attention Network, Attention Mechanism.**

I. Introduction

TEMPORAL action localization in untrimmed videos aims to analyze whether a specific action occurs in videos and determine the temporal boundaries (the start time and the end time) of the action simultaneously. There has been much work on temporal action localization in untrimmed videos [1], [2], [3], [4], [5], but it remains challenging due to the cluttered background, large variances of appearance and motion, and low resolution. Moreover, the same action may occur several times in a video and the durations of action instances with the same class may vary from a few seconds to a few minutes, which further makes it extremely difficult to localize actions in untrimmed videos.

To tackle these problems, many methods based on deep neural networks have been proposed and achieved remarkable progress in temporal action localization, thanks to the successes of deep learning on various visual tasks [6], [7], especially video analysis [8], [9], [10], [11]. Some of the prominent methods [2], [12] resort to sliding windows to produce temporal boundaries of actions and many other methods [13], [14], [15], [16] generate proposals as candidate action instances for localization. These deep methods equally treat each video segment within the sliding windows or proposals and directly aggregate the video segments for temporal action localization. In practice, different segments embody diverse information in a video. Some segments contain the intrinsic motion and appearance of an action, which will play a vital role on action localization. Taking the triple jump action as an example, the segment of jumping action is obviously more important than other segments to localize the triple jump in a video, since the jumping motion reflects the essence characteristics of the triple jump. Therefore, it is necessary to exploit the informative video segments to represent the intrinsic motion and appearance information.

In this paper, we propose a novel method that exploits informative video segments by learning video segment weights for temporal action localization in untrimmed videos. The learned weights represent the importance of the corresponding video segments to recognizing actions and benefit predicting temporal boundaries to localize actions, as shown in Fig. 1. We build a supervised temporal attention network (STAN) that includes three modules: a segment-level attention module, a feature-level attention module, and a localization module. The segment-level attention module is designed to dynamically learn the weights of video segments with supervised attention mechanism. With the learned weights, the segments are fed into a long short-term memory (LSTM) model to capture the temporal relationships between them. The feature-level attention module is introduced to softly aggregate the static appearance and dynamic motion features of each segment by computing the weights of these two features. Through a cascade of the segment-level attention module and the feature-level attention module, STAN exploits the informative video segments and generates video representation with superior descriptive and discriminative ability. Moreover, the localization module is designed to classify the action classes and determine the temporal action boundaries for the input videos, consisting of a proposal generator and a classifier. The proposal generator is used to identify the input video as either background proposal or action proposal, and the classifier is used to classify the action classes of the identified action proposal. Finally, a non-maximum suppression (NMS) strategy is employed to remove the videos with small classification scores and produce the temporal boundaries of action instances. Fig. 2 shows the

C. Sun, H. Song, X. Wu, and Y. Jia are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 10081, P.R. China, e-mails: {sunche, songhao, wuxinxiao, jiayunde}@bit.edu.cn.

J. Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14627. E-mail: jluo@cs.rochester.edu.

Corresponding author: Xinxiao Wu.

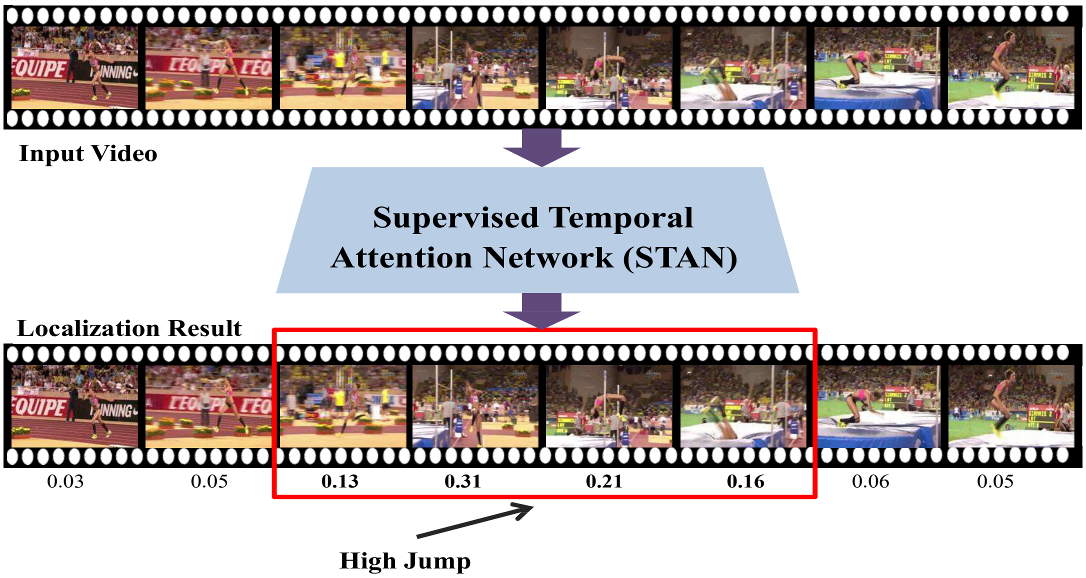


Fig. 1. An illustration of using the proposed STAN to temporally localize an action in a video. The input video with any temporal length is split into a series of segments with equal temporal length. STAN learns the weights of the segments, recognizes the action categories and estimates the boundaries of actions.

architecture of the STAN.

The contributions of this paper are summarized as follows.

- We propose a novel method for temporal action localization by exploiting informative segments in untrimmed videos. These informative segments reflect the intrinsic motion and appearance characteristics of actions, thus contributes a lot to the action localization.
- We build a supervised temporal attention network (STAN) to dynamically learn the weights of video segments via a supervised attention mechanism for representing the importance of different segments.
- We design dual attention blocks to refine and encode features of local segments with the consideration of global context information, where the first attention block learns the local video segments measurement and the second attention block learns the globally context-aware video segments measurement.
- Experiment results on two challenging datasets of THUMOS2014 and ActivityNet1.3 demonstrating the effectiveness of learning informative video segments for temporal action localization.

II. Related Work

A. Temporal Action Localization

Early methods of temporal action localization use sliding windows to sample candidate video segments with multiple temporal scales, and then adopt classifiers to classify the segments. Karaman *et al.* [17] proposed a saliency based pooling method to improve the fisher vector encoding [18] of the improved dense trajectory (iDT) [19], and then fused the frame-level CNN features for action classification. Wang *et al.* [20] fused the features of iDT and CNN to design an action recognition and detection system. They also used a post-processing method to boost the localization performance. Xu *et al.* [21] extracted CNN features and improved dense trajectories by using the vector of locally aggregated descriptors encoding method [22] to recognize and localize the action in videos. Shou *et al.* [2] built a three-stage framework for temporal action localization with an overlap loss function. In [23], a multi-task learning framework is proposed, which consists of three highly related steps: generating action proposals, recognizing actions and refining action localization. Zhao *et al.* [24] used a structured temporal pyramid to model the temporal structure of each action instance, where the context information of an action instance is explored to generate features for temporal action localization. These methods equally treat each video segment within the sliding windows. Different from them, our method dynamically learns the weights of video segments to discover the informative segments which contains the intrinsic motion and appearance information of actions for temporal action localization.

Much recent work tries to extract action proposals from videos and then classifies the proposals into action classes. They often use different aggregation methods to combine representations of segments or frames in a video for action localization by learning action prototypes and actions jointly. Buch *et al.* [25] employed the temporal segment network (TSN) [10] and the recurrent sequence encoder to aggregate video segments for generating action proposals. Gao *et al.* [13] used a cascaded boundary regression model to produce class-agnostic proposals and detect specific actions by using the pooling aggregation method. Xu *et al.* [14] applied the region-based method to temporal action localization and generated candidate temporal regions containing actions by performing temporal convolutions. Based on the work of [14], Chao *et al.* [26] improved receptive field alignment to exploit the temporal context of actions for generating proposals and classifying actions. Gao *et al.* [27] presented a temporal unit regression network to classify the action and regress the

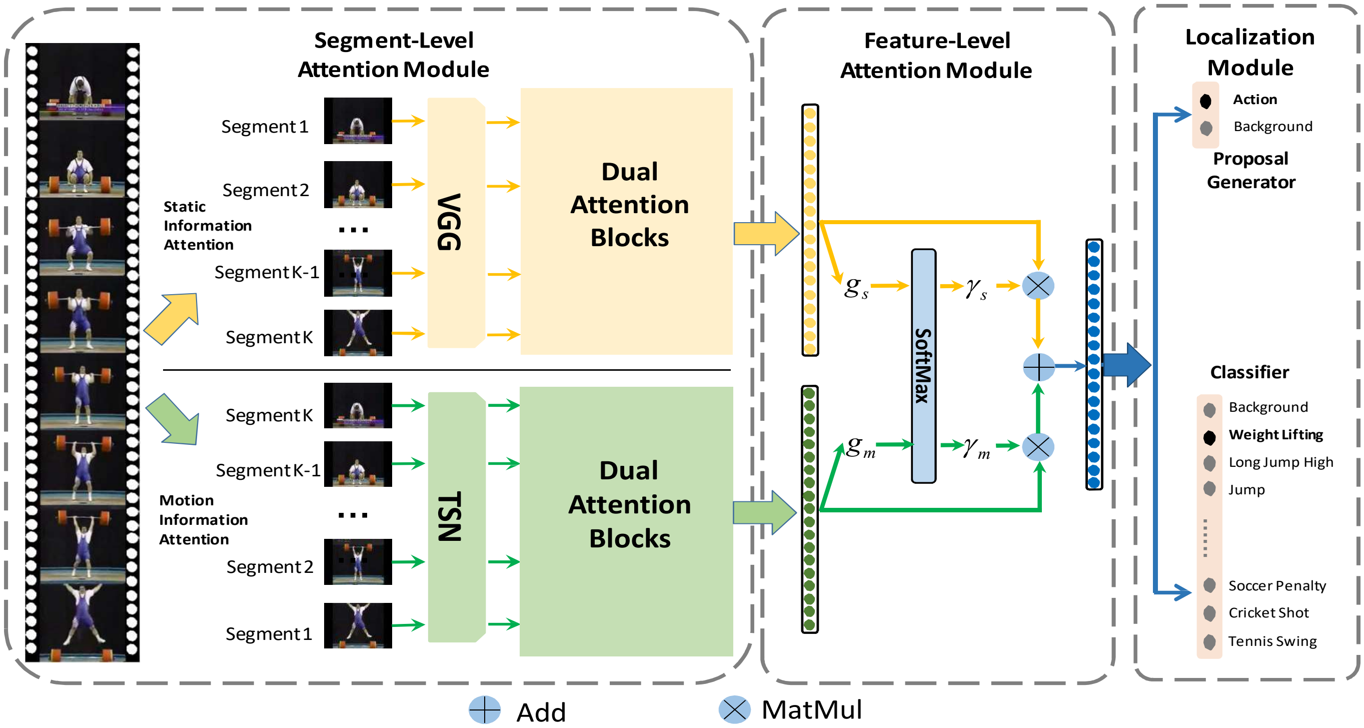


Fig. 2. The architecture of the STAN. It includes three modules: a segment-level attention module, a feature-level attention module, and a localization module. The segment-level attention module learns the weights of video segments with dual attention blocks. The feature-level attention module combines the appearance and dynamic features of each segment by computing the weights of these two features. Through a cascade of the segment-level attention module and the feature-level attention module, STAN learns informative video segments. Moreover, the localization module is designed to classify the action classes and determine the temporal action boundaries for the input videos, including a proposal generator and a classifier.

boundaries. Different from these methods that equally treat each video segment or frame within a video, our method dynamically learns the weight of each segment to effectively eliminate the influence of background and fully exploit action informativeness in a video. Closely related to our work is [28] that used semantically-constrained recurrent memory modules to selectively aggregate relevant context for action localization. The one-way chained structure of the recurrent memory module weighs the contributions of most segments in the local context. In contrast, we design dual attention blocks of the segment-level attention module to learn segment weights over the entire action video at the same time, which is beneficial for exploiting each informative segment with the consideration of the global context for action localization.

B. Attention Mechanism

Inspired by the successes of attention mechanism in natural language processing [29], [30], [7], many researchers have applied the attention mechanism to computer vision. Mnih *et al.* [31] first used the attention mechanism with recurrent neural networks to locate the highlight regions for image classification. Ba *et al.* [32] proposed deep recurrent neural networks trained with reinforcement learning and attention mechanism to find the most relevant regions of the image for object recognition.

Recently, attention mechanism has also been introduced to video analysis [33], [34], [35], [36]. Wang *et al.* [33] presented hierarchical attention networks to combine the spatial information and the temporal information for action understanding. Shi *et al.* [37] used the attention-based LSTM to capture the long-term dependence and find the salient portions. Nguyen *et al.* [34] used the attention mechanism to find the background or action segments for weakly supervised temporal action localization. Li *et al.* [36] used the spatial and temporal attention mechanism and fused video features of multiple modalities for action recognition. These methods use attention mechanisms to capture more important parts, and then generate a more discriminative representation for a video analysis task. However, these methods calculate the attention weights without regard to the temporal structure of the entire action video, which may focus more on the importance of a single part and ignore the context information of the entire action. Different from existing attention-based methods, we use a segment-level attention module to learn the weighted segments considering the temporal context over the entire action video, which is beneficial for capturing correlations among segments to represent intrinsic motion and appearance of an action instance.

III. Method

A video is usually split into a series of segments with equal temporal length to deal with actions with any temporal length. A common strategy is using average pooling or max pooling on these segments to generate a feature representation of the entire video from these segments for temporal action

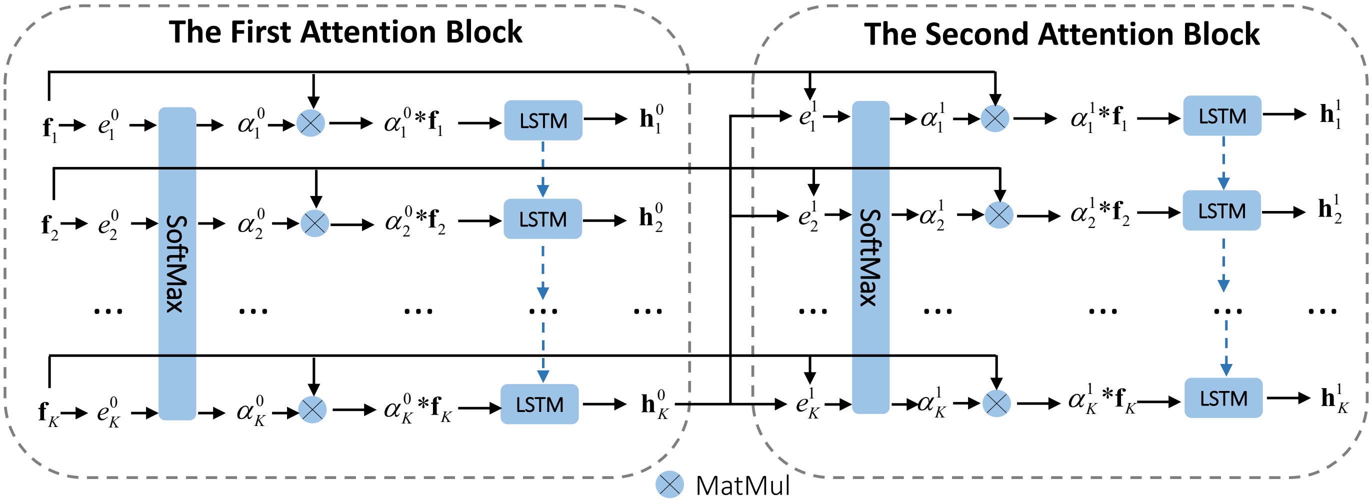


Fig. 3. The architecture of the dual attention blocks. f_i represents the feature vector of the i-th segment. α_i^0 and α_i^1 denote the attention weights of the i-th segment in the first attention block and the second attention block, respectively. h_i^0 and h_i^1 are the outputs of the corresponding LSTM blocks. The first attention block generates the video representation h_k^0 with context information that are used to select context-aware segments in the second attention block.

localization. Feature encoding methods, such as the fisher vector and the vectors of locally aggregated descriptors, are also extensively used in previous work to generate video representations. In these methods, the video segments are usually equally treated without considering their informativeness, and the temporal relationship between segments are not effectively investigated. Therefore, we propose to exploit informative video segments to represent the intrinsic motion and appearance information for temporal action localization.

Encouraged by the successes of the attention mechanism on various applications [38], [31], [39], [40], we build a supervised temporal attention network (STAN) to exploit the informative video segments by learning video segment weights. As shown in Fig. 2, the STAN includes three modules: a segment-level attention module, a feature-level attention module, and a localization module.

A. Segment-Level Attention Module

In the segment-level attention module, we design dual attention blocks to refine and encode features of local segments with the consideration of global context information, where the first attention block learns the universal video segments measurement and the second attention block learns the context-aware video segments measurement, as shown in Fig. 3. In each attention block, we use the long short-term memory (LSTM) model to aggregate all weighted segments for capturing the temporal relationships. Furthermore, we add a supervised constraint to the second attention block to eliminate the influence of background segments. The supervised constraint ensures that the learned weighted segments cover the complete action durations.

1) The First Attention Block: Given an input video v and its action class label y, the video v is split into K non-overlap segments, denoted by $\{s_1, s_2, \cdots, s_K\}$. Let $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K\}$ be the feature vectors of the segments and $\{\alpha_1^0, \alpha_2^0, \cdots, \alpha_K^0\}$ be the weights of the segments in the first attention block. We build an attention layer that filters the feature vectors $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K\}$ by taking the inner product to obtain the corresponding encodings $\{e_1^0, e_2^0, \cdots, e_K^0\}$ by

$$e_t^0 = \mathbf{u}^{0\top} \cdot \mathbf{f}_t, \tag{1}$$

where $\mathbf{u}^0$ is the parameter of the first attention layer with the same size of the feature vector, and $\mathbf{f}_t$ refers to the feature vector of the t-th segment. Then the encodings $\{e_1^0, e_2^0, \cdots, e_K^0\}$ are passed to a softmax operator to calculate the positive weights $\{\alpha_t^0\}$ with the constraint of $\sum_{t=1}^K \alpha_t^0 = 1$ by

$$\alpha_t^0 = \frac{\exp(e_t^0)}{\sum_{j=1}^K \exp(e_j^0)}. \tag{2}$$

Different from the existing attention models [14], [41], [24] that use average pooling or concatenation operation, we aggregate the weighted segments using a LSTM model to generate the video representations by capturing temporal information. The weighted segments are calculated by $\mathbf{x}_t = \alpha_t^0 * \mathbf{f}_t$, which are treated as the input of the LSTM model. We calculate the last hidden state $\mathbf{h}_K^0$ as the feature representation of the input video by

$$\mathbf{h}_K^0 = LSTM(\alpha_t^0 * \mathbf{f}_t, \mathbf{V}^0), \tag{3}$$

where $\mathbf{V}^0$ refers to the set of parameters of the LSTM.

2) The Second Attention Block: In the first attention block, the process of calculating attention weights α_t^0 does not take the context information into consideration. Intuitively, weighting a video segment can benefit from other segments, where the segments are often correlated but temporally very separated. This correlation reflects the informativeness of segments, which may play an important role on action localization. Thus, we introduce the second attention block to select context-aware segments that are more discriminative. The weight of a segment in the second attention block is learned by the current segment representation $\mathbf{f}_K$ and the entire video representation $\mathbf{h}_K^0$, which takes the context information into consideration.

Let $\mathbf{u}^0$ be the parameter of the first attention layer, and $\mathbf{h}_K^0$ be the learned feature representation, where $\mathbf{h}_K^0$ is computed by $\mathbf{u}^0$ using Eqs. (1)(2)(3). The parameter of the second attention layer $\mathbf{u}^1$ is calculated by a transfer layer with the input $\mathbf{h}_K^0$:

$$\mathbf{u}^1 = \tanh(\mathbf{W}^1\mathbf{h}_K^0 + \mathbf{b}^1), \tag{4}$$

where $\mathbf{W}^1$ and $\mathbf{b}^1$ are the weight matrix and the bias vector, respectively. $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ imposes the hyperbolic tangent nonlinearity. We replace $\mathbf{u}^0$ by $\mathbf{u}^1$, and then reuse Eqs. (1)(2)(3) with another set of parameters to generate the output of the second attention block $\mathbf{h}_K^1$. The parameters $\{\mathbf{u}^0, \mathbf{V}^0, \mathbf{W}^1, \mathbf{b}^1, \mathbf{V}^1\}$ are all trainable in the segment-level attention module, where $\mathbf{V}^0$ and $\mathbf{V}^1$ stand for the parameters of the LSTMs in the first and second attention block, respectively.

3) The Supervised Constraint: The segment-level attention module with the dual attention blocks captures the informative segments of an input video for action localization, but the background segments in the sliding window are non-negligible noises. The background segments usually have unique features that may get a higher attention weight under the conventional unconstrained method, but they contain less action information in essence. In order to eliminate these noises, we impose a supervised constraint on the segment-level attention module to filter out the background segments and retain the meaningful action segments. According to the ground-truth action boundaries, we assign an "actionness" label to each segment as the supervised information to guide the learning of segment weights. The "actionness" label represents whether the segment contains an action frame or not. In practice, we relax the supervised constraint in the learning progress to fully exploit the ability of the attention mechanism. We use a multi-class loss function as the supervised constraint to train the attention module, which will be discussed in Section III-D.

Through the supervised learning, the segment-level attention module not only distinguishes the action segments from the background segments, but also captures the informative segments covering the complete action in the input video for action localization. The impact of useless segments will be reduced to produce more effective representations.

B. Feature-Level Attention Module

In temporal action localization, appearance features from each frame and motion features from each video are both helpful to improving the localization accuracy. So we extract appearance feature $\mathbf{f}_t^s$ and motion feature $\mathbf{f}_t^m$ of the t-th video segment to describe a video from spatial and temporal viewpoints, respectively, and build a feature-level attention module to weigh multiple features for fusion.

Using the LSTM model in the segment-level attention module, the learned appearance and motion feature representations of an entire video are represented as $\mathbf{h}_K^s$ and $\mathbf{h}_K^m$, respectively. We introduce an attention layer into the network to dynamically fuse the appearance and motion features of the video. Specifically, the attention layer with the trainable parameter $\mathbf{q}$ encodes the feature $\mathbf{h}_K^s$ and $\mathbf{h}_K^m$, and outputs $\mathbf{g}^s$ and $\mathbf{g}^m$ by

$$\begin{aligned} \mathbf{g}^s &= \mathbf{q}^\top \cdot \mathbf{h}_K^s, \\ \mathbf{g}^m &= \mathbf{q}^\top \cdot \mathbf{h}_K^m. \end{aligned} \tag{5}$$

The weights γ^s and γ^m of $\mathbf{h}_K^s$ and $\mathbf{h}_K^m$ are adaptively computed with $\gamma^s + \gamma^m = 1$ by

$$\begin{aligned} \gamma^s &= \frac{\exp(\mathbf{g}^s)}{\exp(\mathbf{g}^s) + \exp(\mathbf{g}_m)}, \\ \gamma^m &= \frac{\exp(\mathbf{g}^m)}{\exp(\mathbf{g}^s) + \exp(\mathbf{g}^m)}. \end{aligned} \tag{6}$$

The combined feature representation $\mathbf{h}_K$ of the video v is given by

$$\mathbf{h}_K = \gamma^s \cdot \mathbf{h}_K^s + \gamma^m \cdot \mathbf{h}_K^m. \tag{7}$$

C. Localization Module

The localization module aims to infer the action boundaries and complete action classification. This module includes a proposal generator and a classifier. The proposal generator generates video proposals that contain action instances. The classifier classifies the generated video proposal into a specific class. There are totally $N+1$ classes, including the background class and N action classes.

1) Proposal Generator: The proposal generator generates potential proposal video clips with respect to the video representation produced by sliding windows and outputs a binary label to represent whether the generated proposal contains an action instance or not. Moreover, we adopt the boundary regression method [27] to accurately locate the boundary of the action.

Given an input video v, its representation $\mathbf{h}_K$ is learned via the segment-level attention module and the feature-level attention module, given by $\mathbf{h}_K = \text{Attention}(v)$. $\mathbf{h}_K$ is then fed into the proposal generator to compute a binary score p and a relative offset $\{s_i, e_i\}$. The binary score p shows that the corresponding proposal is an action or background, i.e., $p = 1$ for an action and $p = 0$ for background. If the proposal is an action (i.e., $p = 1$), then $\{s_i, e_i\}$ is the start and end offset of the action segment in the input video. We use two fully connected layers to construct the proposal generator.

The training samples are selected using the following strategy. For the untrimmed videos, we only select the segments from the ground truth as positive samples. The negative samples consist of background segments that are randomly sampled from the background videos. The temporal Intersection-over-Union (tIoU) between the training video and its groundtruth is the main criterion: (1) if the tIoU of the video is larger than 0.7, a positive label is assigned according to its action class; (2) if the tIoU of the video is smaller than 0.3, we treat the video as background. We train the proposal generator with a positive/negative ratio of 1:1.

2) Classifier: After eliminating background videos using the proposal generator, we train the classifier for $N+1$ classes, containing N action classes and background class. Similar to the proposal generator, the classifier consists of two separate fully connected layers to output action scores and the relative offset. Both the proposal generator and the classifier are built on the segment-level and feature-level attention modules with the same structure but non-shared parameters. For training the classifier, we follow the similar training dataset construction strategy to the proposal generator. The differences are: (1) we explicitly set the action class label $y \in \{1, 2, \cdots, N\}$ when assigning a label for the positive training sample; (2) we train the classifier with a positive/negative ratio of 1:3.

D. Objective Function

The objective function of our network includes three parts: the classification loss, the regression loss, and the supervised attention loss. We use the softmax cross-entropy loss function for classification and the smooth L1 loss function [42] for regression. The supervised attention loss is used to train the segment-level attention module so that the attention module is able to effectively select the "actionness" information from video segments containing actions. We treat the supervised attention learning as a multi-class classification, and use the sigmoid cross-entropy loss to constrain the attention module.

The classification loss is given by

$$L_{cls} = \frac{1}{N_t} \sum_i -y_i \ln(p_i), \tag{8}$$

where y_i is an one-hot encoding label of the action class, N_t stands for the batch size, and p_i is the prediction score calculated by the proposal generator or classifier after the softmax layer.

The regression loss is formulated as

$$L_{reg} = \frac{1}{N_{pos}} \sum_i l_i^* (\|s_i - s_i^*\|_1^{smooth} + \|e_i - e_i^*\|_1^{smooth}), \tag{9}$$

where N_{pos} stands for the number of positive samples in a batch. s_i and e_i are the predicted start and end offset, respectively, calculated by the proposal generator or classifier. s_i^* and e_i^* are the groundtruth start and end offset, respectively. $\|\cdot\|_1^{smooth}$ represents the smooth L1 loss function. l_i^* is the actionness label, i.e., $l_i^* = 1$ for positive samples, and $l_i^* = 0$ for negative samples.

The supervised attention loss is expressed as

$$\begin{aligned} L_{sat} =& \frac{1}{N_{pos}} \sum_i \frac{1}{N_{seg}} \sum_j l_i^* \Big[y_{ij}^s \ln \frac{1}{1 + \exp(-\log e_{ij}^1)} \\ &+ (1 - y_{ij}^s) \ln \frac{\exp(-\log e_{ij}^1)}{1 + \exp(-\log e_{ij}^1)} \Big], \end{aligned} \tag{10}$$

where N_{seg} stands for the number of segments in each video. y_{ij}^s is the label of the j-th segment in the i-th training sample. If the j-th segment contains any action frame, y_{ij}^s is set to 1; otherwise, y_{ij}^s is set to 0. e_{ij}^1 represents the attention encoding of the j-th segment in the i-th training sample in the second attention block. The supervised attention loss is utilized to force the attention encoding to contain more "actionness" information.

The overall objective function is defined as

$$L = L_{cls} + \lambda_1 L_{reg} + \lambda_2 L_{sat}, \tag{11}$$

where λ_1 and λ_2 are the trade-off parameters. λ_1 is set to 1. As for λ_2, we set an initial value of λ_2 to 0.95 and then decreases its value with iterations to relax the constraint. We find that the best models are obtained when λ_2 is multiplied by 0.95 after 1K iterations.

IV. Experiment

A. Datasets

To evaluate the effectiveness of our method, we conduct experiments on two challenging datasets: THUMOS2014 [43] and ActivityNet1.3 [44].

The THUMOS2014 dataset contains videos from 20 specified classes. Since the training subset is constructed by the UCF101 dataset [45] which consists of many trimmed videos, we use 200 and 213 annotated untrimmed videos from the validation and test subsets for training and testing, respectively. The validation subset consists of 3007 action instances and the test subset consists of 3358 action instances. Each video in the validation and test subsets contains more than 15 action instances on average.

The ActivityNet1.3 dataset includes about 19994 videos with 200 classes. It is divided into three subsets: the training subset of 10024 videos, the validation subset of 4926 videos and the test subset of 5044 videos. Each video contains 1.5 action instances on average. Compared with the THUMOS2014 dataset, the ActivityNet1.3 dataset is more complex due to that the action instances in videos usually last for more than 15 seconds.

B. Evaluation Metric

We adopt the conventional evaluation strategy in THUMOS Challenge and calculate the temporal Intersection over Union (tIoU) with the groundtruth. A localization is marked as correct only when it has a correct action class prediction and has a tIoU higher than a threshold. We report the mean Average Precision (mAP) at different tIoU thresholds as the evaluation metric. On the ActivityNet1.3 dataset, the tIoU thresholds are set to $\{0.5, 0.75, 0.95\}$. On the THUMOS2014 dataset, the tIoU thresholds are set to $\{0.1, 0.2, 0.3, 0.4, 0.5\}$.

C. Experiment Setup

1) Implementation Details: We split the untrimmed video into short segments with equal temporal length. The length of video segments is set to 15 frames for the THUMOS2014 dataset and 75 frames for the ActivityNet1.3 dataset. To reduce the computation cost and improve the training efficiency, we set the maximum length of the sliding window to 32 segments on the THUMOS2014 dataset and 64 segments on the ActivityNet1.3 dataset. For the THUMOS2014 dataset, the sliding window of 480 frames ($32 \times 15 = 480$) is able to completely cover 98.9% action instances. For the ActivityNet1.3 dataset, the sliding window of 4800 frames ($64 \times 75 = 4800$) can completely cover 93.5% action instances.

For the THUMOS2014 dataset, we only use the validation dataset to train our proposed STAN. For the ActivityNet1.3

dataset, we use the training set to train the STAN and the validation dataset for testing.

We extract the appearance and motion features of the short segments using VGG-16 [46] and Temporal Segment Networks (TSN) [10], respectively. TSN is constructed by two convolutional neural networks: spatial stream ConvNets and temporal stream ConvNets, both of which adopt the BN-Inception architecture [47]. The two-stream networks are trained within multiple snippets in a video and then fused by segmental consensus modules for action recognition, so the extracted TSN features are more likely to represent the dynamic motion information. In our work, TSN is trained by the ActivityNet1.3 dataset under the experiment setup in [10]. Moreover, we need extra spatial features from each single frame to enhance the performance of our model. We adopt VGG-16 network which takes a single 224×224 RGB image as input and train VGG-16 by the ILSVRC-2012 dataset [6]. The feature extraction part of VGG-16 and TSN is implemented by using the Caffe toolkit [48].

The outputs of the fc-4096 layer of VGG-16 network are treated as the appearance features of segments. As for the motion features, we follow the operation in [49] and extract the 400-dimensional feature vectors from TSN per five frames. All the segment features are normalized using L2-normalization. The segment scales are set to [1, 2, 3, 4, 5, 6, 8, 11, 16, 24, 32] on the THUMOS2014 dataset and [1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 20, 24, 28, 32, 40, 56, 64] on the ActivityNet1.3 dataset. The overlap segment of sliding windows with different scales is set to [0, 1, 2, 3, 4, 5, 6, 8, 12, 16, 24] and [0.6, 1, 2, 3, 4, 5, 7, 8, 10, 12, 14, 16, 20, 24, 28, 32, 40, 48, 56] on the THUMOS2014 and ActivityNet1.3 datasets, respectively. We sample a single frame in the middle of a segment to extract the VGG-16 feature for the segment. We cascade the TSN features of every five frames in the segment as the motion representation.

The VGG-16 and TSN features are fed into dual attention blocks of the same structure. Before the segment-level attention module, we reduce the feature vector dimension to 1024 by a fully connected layer. In the first attention block of dual attention blocks, the attention weight α_i^0 is calculated from a 1024×1 fully connection layer followed by a softmax layer. Then α_i^0 is dot-multiplied by the i-th segment. The dimension of the hidden state in the LSTM model of the first attention block is set to 1024. In the second attention block, the dimension of the hidden state in the LSTM model is also set to 1024. The kernel size of the feature fusion layer in the feature-level attention module is set to 1024×1. Then the fused features are utilized for temporal action localization.

2) Post-processing: During the test procedure, we first generate videos with different temporal lengths using sliding windows. Then we use the proposal network in STAN to remove the background videos and adjust the boundary of positives samples according to the results of boundary regression. These positives proposals may highly overlap with each other, so we adopt a soft non-maximum suppression (soft-NMS) [59] method to eliminate highly overlapping. The soft-NMS threshold is set to 0.8 for the ActivityNet1.3 dataset and 0.65 for the THUMOS2014 dataset. We keep the top 300 proposals after the soft-NMS for action classification. Subsequently, the classifier accepts these processed proposals to produce the prediction scores and refine the temporal boundaries of the action instances. Finally, we conduct greedy non-maximum suppression (NMS) to remove redundant localization results and set the overlap threshold of NMS to $\alpha - 0.1$ in this paper, where α is the mAP threshold in evaluation.

TABLE I
RESULTS ON THE THUMOS2014 DATASET WITH VARIED TIOU THRESHOLD α. WE USE THE MEAN AVERAGE PRESISION (MAP) (%) AS THE LOCALIZATION RESULTS. THE HIGHEST TWO SCORES ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY.

	α				
	0.1	0.2	0.3	0.4	0.5
The Handcrafted Features					
Karaman *et al.* [17]	1.5	0.9	0.5	0.3	0.2
Wang *et al.* [20]	19.2	17.8	14.6	12.1	8.5
Oneata *et al.* [50]	39.8	36.2	28.8	21.8	14.3
Heilbron *et al.* [51]	36.1	32.9	25.7	18.2	13.5
Deep Neural Networks					
Shou *et al.* [2]	47.7	43.5	36.3	28.7	19.0
Yeung *et al.* [52]	48.9	44.0	36.0	26.4	17.1
Zhu *et al.* [12]	47.7	43.6	36.2	28.9	19.0
Lin *et al.* [53]	50.1	47.8	43.0	35.0	24.6
Shou *et al.* [54]	-	-	40.1	29.4	23.3
Buch *et al.* [28]	-	-	45.7	-	29.2
Xu *et al.* [14]	54.5	51.5	44.8	35.6	28.9
Yuan *et al.* [41]	51.0	45.2	36.5	27.8	17.8
Dai *et al.* [55]	-	-	-	33.3	25.6
Zhao *et al.* [24]	*66.0*	*59.4*	51.9	41.0	29.8
Gao *et al.* [13]	60.1	56.7	50.1	41.3	31.0
Qiu *et al.* [56]	-	-	48.2	42.4	34.2
Kong *et al.* [57]	54.7	53.0	48.5	41.3	32.5
Alwassel *et al.* [58]	-	-	51.8	42.4	30.8
Liu *et al.* [16]	-	-	*56.0*	47.4	38.8
Zeng *et al.* [5]	**69.5**	**67.8**	**63.6**	**57.8**	**49.1**
STAN (ours)	56.9	55.7	52.8	*47.5*	*39.8*

TABLE II
AVERAGE PRECISION (AP)(%) FOR EACH CLASS OF TEMPORAL ACTION LOCALIZATION ON THE THUMOS2014 DATASET. WE SET OVERLAP THRESHOLD α TO 0.5 FOR EVALUATION. THE HIGHEST TWO SCORES ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY.

Method	[50]	[52]	[2]	[14]	STAN
BaseballPitch	8.6	14.6	14.9	**26.1**	*18.7*
BasketballDunk	1	6.3	20.1	**54.0**	*52.6*
Billiards	2.6	*9.4*	7.6	8.3	**10.9**
CleanAndJerk	13.3	**42.8**	24.9	27.9	*42.7*
CliffDiving	17.7	15.6	27.5	*49.2*	**71.3**
CricketBowling	9.5	10.8	15.7	**30.6**	*18.1*
CricketShot	2.6	3.5	*13.8*	10.9	**15.9**
Diving	4.6	10.8	17.6	*26.2*	**36.3**
FrisbeeCatch	1.2	*10.4*	5.1	**20.1**	2.2
GolfSwing	*22.6*	13.8	18.2	16.1	**32.7**
HammerThrow	34.7	28.9	19.1	*43.2*	**62.4**
HighJump	17.6	*33.3*	20	30.9	**59.6**
JavelinThrow	22	20.4	18.2	*47.0*	**68.3**
LongJump	47.6	39.0	34.8	*57.4*	**88.7**
PoleVault	19.6	16.3	32.1	*42.7*	**83.0**
Shotput	11.9	16.6	12.1	*19.4*	**32.0**
SoccerPenalty	8.7	8.3	**19.3**	*15.8*	13.5
TennisSwing	3	5.6	**19.4**	16.6	*18.1*
ThrowDiscus	*36.2*	29.5	24.4	29.2	**46.7**
VolleyballSpiking	1.4	5.2	4.6	*5.6*	**22.2**
mAP	14.4	17.1	19.0	*28.9*	**39.8**

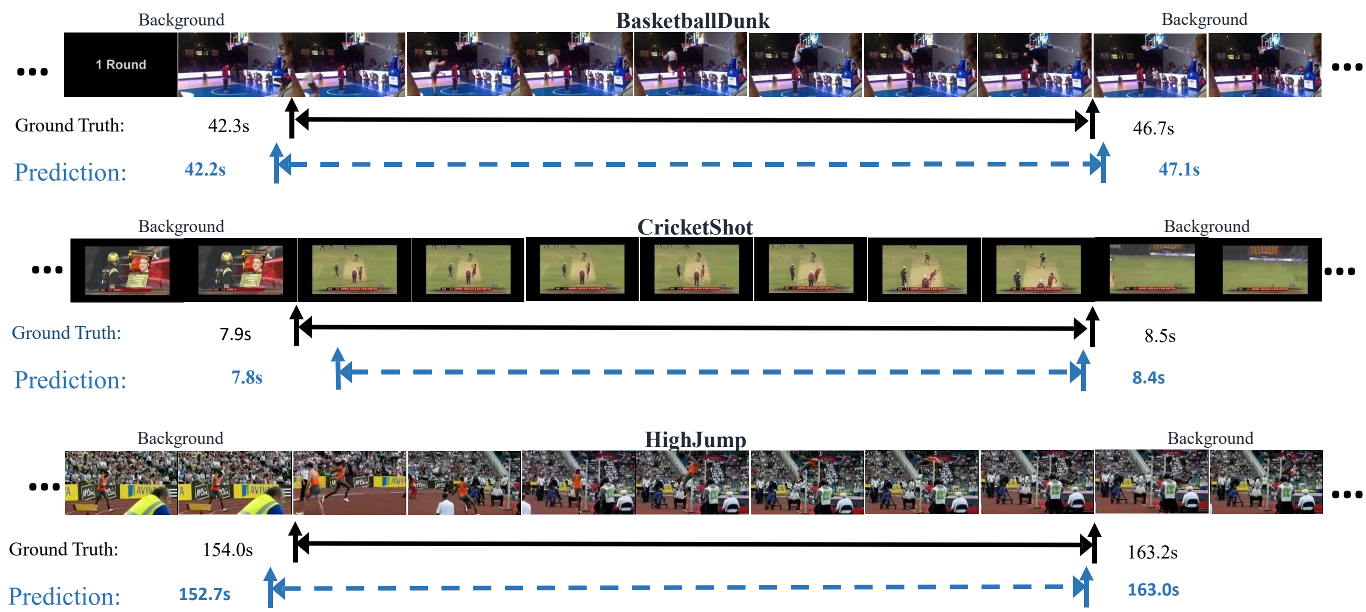


Fig. 4. The prediction results of three action instances on the THUMOS2014 test dataset. The ground truth and the prediction results are shown below the image sequences. The three action classes are "BasketballDunk", "CrieckShot" and "HighJump".

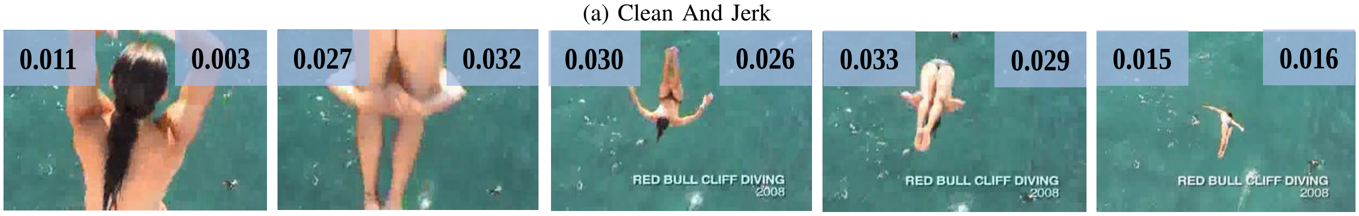
D. Results on the THUMOS2014 Dataset

1) mAP Results: We report the comparison results between our method and the state-of-the-art methods in Table I. From Table I, we can observe that: (1) STAN outperforms most existing methods especially when α is greater than 0.3, which demonstrates that our method localizes the action boundaries with higher accuracy in more difficult situations. (2) The method of [5] performs much better than all other methods with the mAP (α=0.5) of 49.1%, probably because they use a proposal post-processing method P-GCN with BSN proposals [49] as extra input for temporal action localization, which will highly improve the performance. (3) STAN achieves better results than the top four methods using handcrafted features and feature vector encoding, which obviously verifies the superiority of STAN on simultaneously learning discriminative video representation and accurate localization model. Compared to these handcrafted features with the feature vector encoding or other encoding methods, our network can produce more discriminative video representations with the attention mechanism. (4) Compared with the middle methods based on deep neural networks, our method still performs better than most methods in most cases. Concretely, our STAN outperforms the RNN-based methods [52], [28], [56], since the STAN effectively couples the attention mechanism and the LSTM model in dual attention blocks, which exploits the informative video segments for temporal modeling of the entire video to further enhance the action localization. (5)The STAN also works better than the methods [13], [56], [57] that use average pooling or concatenation operations to generate final video representations. This proves that our method can generate more descriptive and discriminative video representations by learning weighted video segments.

Table II shows the comparison results of per-class AP between our method and [50], [52], [52], [14] on the THUMOS2014 dataset. It is interesting to notice that our method achieves improvements on some challenging classes such as "CliffDiving", "LongJump", "PoleVault", and performs more stable on different action classes. The result of our method is not ideal for locating the action of "FrisbeeCatch", probably because there is no difference between action and background of "FrisbeeCatch", except whether the frisbee is flying in the air or not. In other words, the "FrisbeeCatch" action has no clear decomposition structure, and our segment-based method can not accurately predict the action boundary.

2) Qualitative Results: Fig. 4 shows some examples of the prediction results on the THUMOS2014 dataset: "BasketballDunk", "CricketShot", and "HighJump". Several video frames are sampled from video segments to represent the entire action instance. The temporal boundary of each localized action instance is measured in seconds. Each prediction duration with the highest classification score is associated with the nearest ground truth annotation. We observe that the temporal boundary of actions estimated by our method has high tIoU with the corresponding ground truth. For the examples of "BasketballDunk" and "CricketShot", our method localizes the action instance accurately. For the "HighJump" example, the predicted action start is a little earlier than the ground truth, because it is difficult to determine the boundary between the preparation and the start of "HighJump". We also show several segment snapshots with their attention weights of the second attention layer in Fig. 5. As shown in Fig. 5 (b), the video segments in the middle columns represent the important sub-actions of cliff diving, whose weights are obviously larger than other segments. It means that the segments in the middle columns are more informative than other segments, which is also in line with human's perceptions.

(a) Clean And Jerk



(b) Cliff Diving

(c) ThrowDiscus

Fig. 5. Typical examples showing the weights of segments in the segment-level attention module on the THUMOS2014 dataset. The action classes are "Clean And Jerk"(a), "Cliff Diving"(b), and " ThrowDiscus"(c). We only display 5 segments from 32 segments of each video clip and each segment is represented by only one frame. The values on the top-left of each frame represent the weights of each segment with the motion features. The values on the top-right of each frame represent the weights of each segment with the appearance features.

E. Results on ActivityNet Dataset

1) mAP Results: We also compare our STAN with the existing methods on the more complex ActivityNet1.3 dataset with various action lengths. From Table I and Table III, we find that our method does not perform as well on the ActivityNet1.3 dataset as it does on the THUMOS2014 dataset compared with several existing methods [55], [54], probably due to that the segment length on the ActivityNet1.3 dataset is much longer than that on the THUMOS2014 dataset (75 frames vs 15 frames), so our segment-level sliding window-based method may regress unclear frame-level action boundaries on the ActivityNet1.3 dataset. Specifically, both our method and the work of [55] adopt segment-level sliding windows, so our method achieves comparable results compared with [55] on the ActivityNet1.3 dataset at thresholds of 0.5 and 0.75. Our method performs a little worse than [55] at a threshold of 0.95, probably due to that [55] uses an extra longer context window to ensure that the boundaries of long action instances are captured. Our method works a little worse than [54], because [54] conducts frame-level predictions rather than segment-level predictions to generate proposals, which is more suitable for action localization on the ActivityNet1.3 dataset. Nevertheless, our method yields a higher mAP at a threshold of 0.95 than [54], which indicates that our method locates the action boundaries more accurately especially on the more difficult scenarios. Furthermore, though the average mAP of our method is 4% worse than that of [54] on the ActivityNet1.3 dataset, our method achieves a significant improvement over [54] on the THUMOS2014 dataset, and the mAP at the threshold of 0.5 has increased from 23.3% to 39.8%.

TABLE III
TEMPORAL ACTION LOCALIZATION RESULTS (MAP) (%) ON THE ACTIVITYNET1.3 DATASET. THE HIGHEST TWO SCORES ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY.

Model	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 0.95$	Average
Singh *et al.* [60]	34.5	-	-	-
Li *et al.* [61]	30.4	-	-	-
Shou *et al.* [54]	**45.3**	**26.0**	0.2	**23.8**
Xu *et al.* [14]	26.8	-	-	12.7
Dai *et al.* [55]	*36.4*	21.1	**3.9**	-
STAN (ours)	35.9	*21.3*	*1.7*	*19.8*

2) Qualitative Results: We report several localization results on the ActivityNet1.3 dataset in Fig. 6, and show some segment snapshots with the attention weights of the second attention layer in Fig. 7. These weights reflect the importance of different segments for action classification and localization. For example, in Fig. 7 (c), we observe that a person is playing guitar and there is no obvious difference among these segments, so their weights are almost the same.

F. More Evaluation of the Proposed Method

1) Ablation Study: Table IV shows the efficacy of different individual components on the action localization. "VGG" represents that only VGG features are fed into the segment-level attention module to generate final video representation for action localization, where the feature-level attention module is removed. "TSN" means that the VGG features are replaced by the TSN features and other experiment settings are the same as "VGG". "w/o attention" means removing attention mechanism, where all the weights $\{\alpha_t^0, \alpha_t^1, \gamma_s, \gamma_m | t = 1, 2, \cdots, K\}$

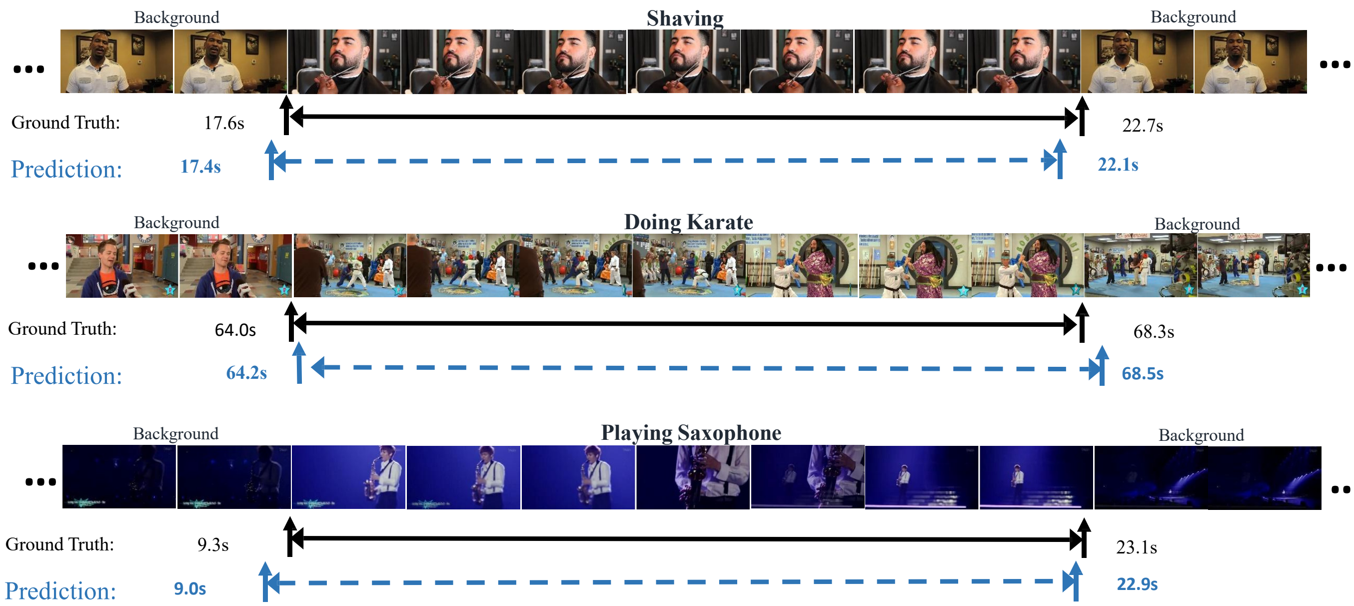


Fig. 6. The prediction results of five action instances on the ActivityNet 1.3 validation dataset. The ground truth and the prediction results are shown below the image sequences. The three action classes are "Shaving", "Doing Karate" and "Playing Saxophone".

are set to the fixed value of 1 during training and testing. "w/o feature-level attention" refers to removing the feature-level attention module and "w/o second attention" indicates removing the second attention block in the segment-level attention module. "w/o supervised attention" means that the supervised attention loss $\mathcal{L}_{sat}$ is removed during training. "w/o relaxation" means the network is trained in a supervised way without relaxing the supervised attention loss, where λ_2 in Eq. (11) is fixed to 0.95.

It is interesting to observe that: (1) The motion information and the appearance information of videos are complementary. Both the TSN and VGG features can contribute to producing informative features with video segments. Moreover, the TSN feature is more effective than the VGG feature for action localization of videos. (2) The attention mechanism is useful in generating informative features of videos for temporal action localization, with the mAP gains of 9.4% and 4.5% on the THUMOS2014 and ActivityNet1.3 datasets, respectively, when compared with "w/o attention". (3) when TSN and VGG features are treated equally ("w/o feature-level attention"), the experiment result is even worse than that only using TSN features, possibly because the VGG feature misleads action localization in some cases. Therefore, it is useful to use feature-level attention to dynamically weight different features. (4) The experiment result of "w/o second attention" also shows the effectiveness of learning the globally context-aware video segments measurement. (5) The supervised attention learning can benefit discarding the negative segments and the relaxation can improve the performance of the attention mechanism.

2) Evaluation of Different Segment Lengths: The length of video segments will influence the overall performance, because we use segment-level feature vectors for frame-level boundary regression. We conduct experiments to compare the results of different segment lengths on the THUMOS2014 dataset,

TABLE IV
TEMPORAL ACTION LOCALIZATION RESULTS (MAP) (%) OF DIFFERENT COMPONENTS IN THE STAN ($\alpha = 0.5$).

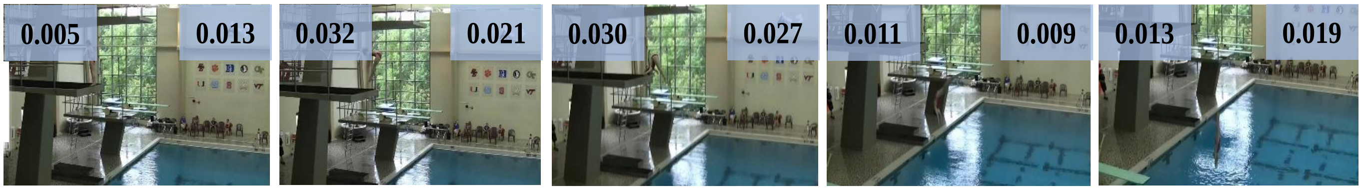
	THUMOS2014	ActivityNet1.3
VGG	29.1	26.5
TSN	35.4	32.3
w/o attention	30.4	31.4
w/o feature-level attention	34.4	28.5
w/o second attention	36.0	34.2
w/o supervised attention	37.2	33.6
w/o relaxation	38.6	33.3
STAN (ours)	39.8	35.9

TABLE V
TEMPORAL ACTION LOCALIZATION RESULTS (MAP) (%) OF DIFFERENT SEGMENT LENGTH (FRAMES) IN THE STAN ($\alpha = 0.5$).

Segment Length	10	15	30	45	60
mAP	38.2	39.8	36.8	34.5	33.7

as shown in Table V. When the segment length is set to 15 frames, our method achieves the highest mAP at a threshold of 0.5. The possible reason is that longer segments fail to locate accurate frame-level action boundaries while shorter segments can not cover most action instances on the THUMOS2014 dataset.

3) Speed Comparison: As shown in Table VI, we make a comparison of the inference speed. We choose two segment-level sliding window-based methods [2], [62] and two frame-level proposal-based methods [54], [14] for comparison, and the inference speed is directly copied from their original papers, except that the speed of S-CNN [2] is reported from from [62]. Our method achieves 203 FPS using a single NVIDIA GTX1080Ti GPU with pre-trained TSN and



(a) Plataform Diving

(b) Javelin Throw

(c) Playing Guitarra

Fig. 7. Typical examples showing the weights of segments in the segment-level attention module on the ActivityNet1.3 dataset . The action classes are "Plataform Diving"(a), "Javelin Throw"(b), and " Playing Guitarra"(c). We only display 5 segments from 64 segments of each video clip and each segment is represented by only one frame. The values on the top-left of each frame represent the weights of each segment with the motion features. The values on the top-right of each frame represent the weights of each segment with the appearance features.

VGG features, which is faster than the segment-level sliding window-based methods [2], [62] that use the end-to-end method to process original high-dimensional video data. Our method is slower than the frame-level proposal-based methods [54], [14] that perform fully convolutional operations on the frame level, probably due to the recurrent architectures of LSTMs for segment-level prediction in our model.

TABLE VI
COMPARISON THE ACTION DETECTION SPEED FOR TEST.

	Method	FPS
Segment-Level	S-CNN [2]	60
	DAP (TiTan X) [62]	135
Frame-Level	CDC (TiTan X) [54]	500
	R-C3D (TiTan Xm) [14]	569
	R-C3D (TiTan Xp) [14]	1030
	Ours (1080Ti)	203

4) Weight Analysis of Different Features: As shown in Table VII, we compare the mean and standard deviation (std. dev.) of weights of different features γ_s and γ_m in Eq. (6) On both the THUMOS2014 dataset and ActivityNet1.3 dataset, the mean value of γ_s is much small than that of γ_s, and their std. dev. values are small. This indicates that in most cases, the importance of motion features (TSN) is much greater than the static features (VGG) for action localization, which is consistent with the results of our ablation study.

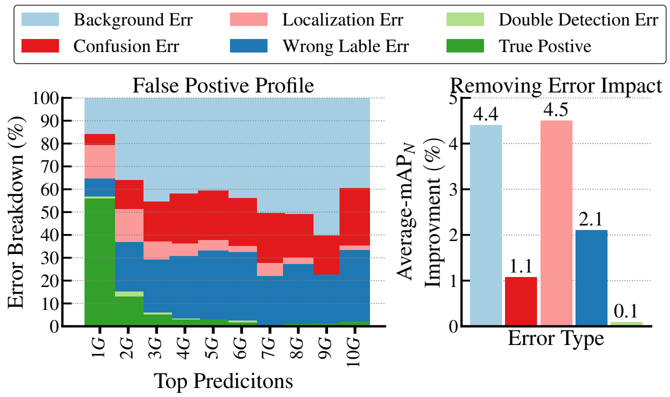
5) DETAD Analysis: To further evaluate our method, we conduct the DETAD analysis [63] on the THUMOS2014 dataset, including false positive analysis, average-mAP$_N$ sensitivity, and false negative analysis.

TABLE VII
THE MEAN AND STD. DEV. OF γ_s AND γ_m ON THE THUMOS2014 AND ACTIVITYNET1.3 DATASETS

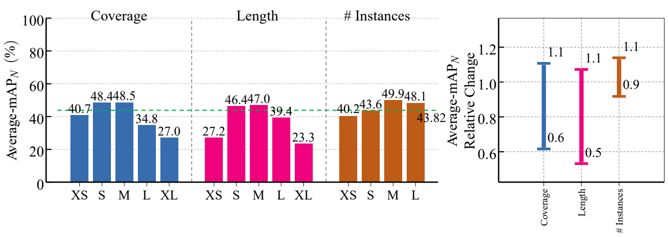
Method	THUMOS2014		ActivityNet1.3	
	mean	std. dev.	mean	std. dev.
γ_s	0.129	0.029	0.095	0.011
γ_m	0.871	0.029	0.905	0.011

False Positive Analysis. Fig. 8 (a) shows the false positive profiles and the impact of error types on the average-mAP$_N$ of our method. We observe that the true positive rate is high in top-1G and top-2G predictions, meaning that our method scores the true predictions higher and the wrong predictions lower. This verifies that our method achieves good action localization results with fewer predictions. The background error of our method is high, mainly because we retain more sliding windows for higher recall rates when performing action proposals. The impact of error types on the average-mAP$_N$ shows that eliminating more backgrounds and regressing action boundaries more accurately are two important ways to improve the performance of our method.

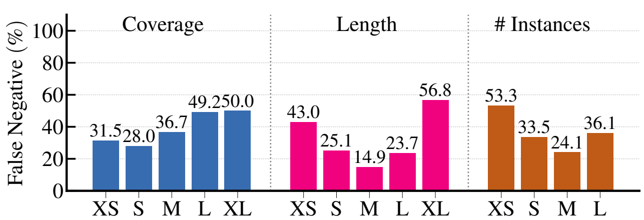
Average-mAP$_N$ Sensitivity. Fig 8 (b) shows the sensitivity of our method mAP$_N$ (0.5 tIoU) to action characteristics of the coverages, lengths, and numbers of instances. The dashed line represents the overall performance. We find that our method achieves higher mAP on the small (S) and medium (M) durations of videos as well as lengths of action instances. The sensitivity profile also shows that the performance of our method is related to the length of the video and the length of the action instances, probably because our method adopts



(a)



(b)



(c)

Fig. 8. Illustration of the three types of analyses that the diagnostic tool on the THUMOS2014 dataset. (a) The false positive profiles of our methods and the impact of error types on the average-mAP$_N$ (0.5 tIOU). (b) The average-mAP of our method for different characteristics and the sensitivity profile. The dashed line is the overall performance. (c) The average false negative rate of our method for characteristics of the coverages, lengths, and numbers of instances.

segment-level LSTMs to aggregate sliding windows, which may be highly influenced by the temporal information. It is interesting to notice that our method does not show a strong sensitivity to the characteristic of the total count of instances in videos, which verifies that our method can effectively find multiple instances per video.

False Negative Analysis. Fig. 8 (c) illustrates the false negative rate for three pairs of characteristics. We observe that the results are inverse to that of the average-mAP$_N$ sensitivity shown in Fig. 8 (b), and our method prefers to find multiple instances per video.

V. Conclusions

We have presented a novel method of exploiting informative video segments by learning segment weights for temporal action localization in untrimmed videos. The learned weights can effectively capture the informativeness of video segments to representing the intrinsic motion and appearance of an action. The method is implemented through a Supervised Attention Temporal Network (STAN) consisting of a cascade attention module for temporal action localization. With the supervision of "actionness" information, the segment-level attention module can dynamically learn the weights of video segments to represent their contributions to action localization. The feature-level attention module can learn the weights of multiple segment features for combination to further boost the localization performance. Extensive experiments on commonly used public datasets show the superior performance of STAN on temporally localizing actions in untrimmed videos. We believe that STAN is a general solution to capturing the intrinsic motion and appearance information in videos, and plan going to apply it to other video analysis tasks.

Acknowledgment

This work was supported in part by the Natural Science Foundation of China (NSFC) under grants No. 61673062.

References

[1] J. Yuan, Y. Pei, B. Ni, P. Moulin, and A. Kassim, "Adsc submission at thumos challenge 2015," in *CVPR THUMOS Workshop*, vol. 1, 2015.

[2] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[3] Q. Liu and Z. Wang, "Progressive boundary refinement network for temporal action detection."

[4] H. Song, X. Wu, B. Zhu, Y. Wu, M. Chen, and Y. Jia, "Temporal action localization in untrimmed videos using action pattern trees," *IEEE Transactions on Multimedia*, vol. PP, no. 99, pp. 1–1, 2018.

[5] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7094–7103.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" *Computer Vision and Image Understanding*, vol. 163, pp. 90–100, 2017.

[8] M. Hasan and A. K. Roy-Chowdhury, "A continuous learning framework for activity recognition using deep hybrid feature models," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1909–1922, 2015.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[10] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.

[11] J. Hou, X. Wu, Y. Sun, and Y. Jia, "Content-attention representation by factorized action-scene network for action recognition," *IEEE Transactions on Multimedia*, vol. 20, no. 6, pp. 1537–1547, 2017.

[12] Y. Zhu and S. Newsam, "Efficient action detection in untrimmed videos via multi-task learning," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 197–206.

[13] J. Gao, Z. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," *British Machine Vision Conference*, 2017.

[14] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 6, 2017, p. 8.

[15] D. Guo, W. Li, and X. Fang, "Fully convolutional network for multiscale temporal action proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3428–3438, 2018.

[16] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S.-F. Chang, "Multi-granularity generator for temporal action proposal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3604–3613.

[17] S. Karaman, L. Seidenari, and A. Del Bimbo, "Fast saliency based pooling of fisher encoded dense trajectories," vol. 1, no. 6, 2014.

[18] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European conference on computer vision*. Springer, 2010, pp. 143–156.

[19] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[20] L. Wang, Y. Qiao, and X. Tang, "Action recognition and detection by combining motion and appearance features," *THUMOS14 Action Recognition Challenge*, vol. 1, p. 2, 2014.

[21] Z. Xu, L. Zhu, Y. Yang, and A. G. Hauptmann, "Uts-cmu at thumos 2015," in *CVPR workshop*, vol. 6, no. 8, 2015.

[22] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[23] Y. Zhu and S. Newsam, "Efficient action detection in untrimmed videos via multi-task learning," *arXiv preprint arXiv:1612.07403*, 2016.

[24] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, D. Lin, and X. Tang, "Temporal action detection with structured segment networks," *arXiv preprint arXiv:1704.06228*, 2017.

[25] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "Sst: Single-stream temporal action proposals," in *Computer Vision and Pattern Recognition*, 2017, pp. 6373–6382.

[26] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1130–1139.

[27] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," *arXiv preprint arXiv:1703.06189*, 2017.

[28] 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 *Proceedings of the British Machine Vision Conference (BMVC)*, vol. 1, 2017, p. 2.

[29] W. Yin, S. Ebert, and H. Schütze, "Attention-based convolutional neural network for machine comprehension," *arXiv preprint arXiv:1602.04341*, 2016.

[30] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European Conference on Computer Vision*. Springer, 2016, pp. 451–466.

[31] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[32] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014.

[33] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li, "Hierarchical attention network for action recognition in videos," *arXiv preprint arXiv:1607.06416*, 2016.

[34] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," *arXiv preprint arXiv:1712.05080*, 2017.

[35] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based lstm and semantic consistency," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2045–2055, 2017.

[36] 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, 2019.

[37] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Joint network based attention for action recognition," *arXiv preprint arXiv:1611.05215*, 2016.

[38] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[39] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention." in *ICML*, vol. 14, 2015, pp. 77–81.

[40] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua, "Neural aggregation network for video face recognition," *arXiv preprint arXiv:1603.05474*, 2016.

[41] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng, "Temporal action localization by structured maximal sums," *arXiv preprint arXiv:1704.04671*, 2017.

[42] R. Girshick, "Fast r-cnn," in *IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[43] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "THUMOS challenge: Action recognition with a large number of classes," http://crcv.ucf.edu/THUMOS14/, 2014.

[44] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[45] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[46] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[48] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[49] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[50] D. Oneata, J. Verbeek, and C. Schmid, "The lear submission at thumos 2014," *Computer Vision and Pattern Recognition [cs.CV]*, 2014.

[51] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1914–1923.

[52] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2678–2687.

[53] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 988–996.

[54] 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 1417–1426.

[55] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen, "Temporal context network for activity localization in videos," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5727–5736.

[56] H. Qiu, Y. Zheng, H. Ye, Y. Lu, F. Wang, and L. He, "Precise temporal action localization by evolving temporal proposals," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. ACM, 2018, pp. 388–396.

[57] W. Kong, N. Li, S. Liu, T. Li, and G. Li, "Blp-boundary likelihood pinpointing networks for accurate temporal action localization," *arXiv preprint arXiv:1811.02189*, 2018.

[58] H. Alwassel, F. Caba Heilbron, and B. Ghanem, "Action search: Spotting actions in videos and its application to temporal action localization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 251–266.

[59] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5561–5569.

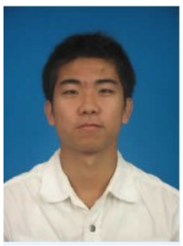
[60] G. Singh and F. Cuzzolin, "Untrimmed video classification for activity detection: submission to activitynet challenge," *arXiv preprint arXiv:1607.01979*, 2016.

[61] J. Li, X. Liu, Z. Zong, W. Zhao, M. Zhang, and J. Song, "Graph attention based proposal 3d convnets for action detection."

[62] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Daps: Deep action proposals for action understanding," in *European Conference on Computer Vision*. Springer, 2016, pp. 768–784.

[63] H. Alwassel, F. Caba Heilbron, V. Escorcia, and B. Ghanem, "Diagnosing error in temporal action detectors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 256–272.

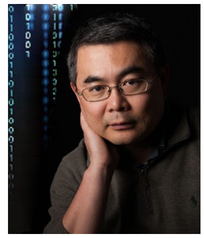
Che Sun received the B.S. degree from Beijing Institute of Technology(BIT), Beijing, China, in 2017. He will pursue the Ph.D. degree at the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology. His research interests include computer vision, machine learning.



Hao Song received the B.S. degree from North China Electric Power University (NCEPU), Baoding, China, in 2012 and the Ph.D. degree from the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, in 2018. His research interests include computer vision, machine learning, and video retrieval.

Xinxiao Wu (M'09) is an Associate Professor in the School of Computer Science at the Beijing Institute of Technology. She received the B.A. degree in computer science from the Nanjing University of Information Science and Technology in 2005 and the Ph.D. degree in computer science from the Beijing Institute of Technology in 2010. She was a post-doctoral research fellow at Nanyang Technological University, Singapore, from 2010 to 2011. Her current research interests include machine learning, computer vision and video analysis and understanding.

Yunde Jia (M'11) received the B.S., M.S., and Ph.D. degrees from the Beijing Institute of Technology (BIT) in 1983, 1986, and 2000, respectively. He was a visiting scientist with the Robotics Institute, Carnegie Mellon University (CMU), from 1995 to 1997. He is currently a Professor with the School of Computer Science, BIT, and the team head of BIT innovation on vision and media computing. He serves as the director of Beijing Lab of Intelligent Information Technology. His interests include computer vision, vision-based HCI and HRI, and intelligent robotics.



Jieno Luo (S93, M96, SM99, F09) joined the Department of Computer Science, University of Rochester, in 2011, after a prolific career of over 15 years with Kodak Research. He has authored over 400 technical papers and holds over 90 U.S. patents. His research interests include computer vision, machine learning, data mining, social media,and biomedical informatics. He has served as 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.