A Multitemporal Scale and Spatial–Temporal Transformer Network for Temporal Action Localization

Zan Gao, *Member, IEEE*, Xinglei Cui, Tao Zhuo, Zhiyong Cheng, An-An Liu, *Senior Member, IEEE*, Meng Wang, *Fellow, IEEE*, and Shenyong Chen, *Senior Member, IEEE*

***Abstract*—Temporal action localization plays an important role in video analysis, which aims to localize and classify actions in untrimmed videos. Previous methods often predict actions on a feature space of a single temporal scale. However, the temporal features of a low-level scale lack sufficient semantics for action classification, while a high-level scale cannot provide the rich details of the action boundaries. In addition, the long-range dependencies of video frames are often ignored. To address these issues, a novel multitemporal-scale spatial–temporal transformer (MSST) network is proposed for temporal action localization, which predicts actions on a feature space of multiple temporal scales. Specifically, we first use refined feature pyramids of different scales to pass semantics from high-level scales to low-level scales. Second, to establish the long temporal scale of the entire video, we use a spatial–temporal transformer encoder to capture the long-range dependencies of video frames. Then, the refined features with long-range dependencies are fed into a classifier for coarse action prediction. Finally, to further improve the prediction accuracy, we propose a frame-level self-attention module to refine the classification and boundaries of each action instance. Most importantly, these three modules are jointly explored in a unified framework, and MSST has an anchor-free and end-to-end architecture. Extensive experiments show that the proposed method can outperform state-of-the-art approaches on the THUMOS14 dataset and achieve comparable performance on the ActivityNet1.3 dataset. Compared with A2Net (TIP20, Avg{0.3:0.7}), Sub-Action (CSVT2022, Avg{0.1:0.5}), and AFSD (CVPR21, Avg{0.3:0.7}) on the THUMOS14 dataset, the proposed method can achieve improvements of 12.6%, 17.4%, and 2.2%, respectively.**

***Index Terms*—Frame-level self-attention (FSA), multiple temporal scales, refined feature pyramids (RFPs), spatial–temporal transformer (STT), temporal action localization (TAL).**

I. Introduction

In RECENT years, with the emergence of a large number of Internet videos and surveillance videos [12], [19], [20], [28], [29], [54], [57], [60], temporal action localization (TAL) has attracted much attention in academia and industry [42]. It is also widely used in the human–machine interaction domain [7], [18], [32], [45], [58]. As an important branch of video understanding, the goal of TAL is to locate the start and end of each action instance in untrimmed videos and predict its categories.

According to the different processing strategies used, recent TAL methods can be roughly divided into three categories: anchor-based, actionness-guided, and anchor-free methods. Owing to the fixed predefined anchors, anchor-based methods are not flexible enough to handle various action categories. In addition, anchor-based methods are very sensitive to some hyperparameters. Unlike anchor-based methods, actionness-guided approaches do not require predefined anchors to generate action proposals. However, this strategy requires an extra model for action classification, and its computational cost is relatively high. In contrast to the actionness-guided methods, anchor-free methods do not enumerate boundaries, avoiding redundant proposals and reducing the amount of computation.

To localize and classify actions with different temporal scales, the majority of existing anchor-free techniques often predict actions on the feature space of a single temporal scale, i.e., the feature space of each individual pyramid layer. However, the features of a low-level temporal scale lack enough semantics for action classification, while a high-level scale cannot provide the rich details of action boundaries. As a result, it is difficult to consider both the semantics and boundaries for each action instance simultaneously. Moreover, the long-range dependencies

Manuscript received 23 January 2023; revised 6 March 2023; accepted 6 April 2023. Date of publication 5 May 2023; date of current version 8 June 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61872270, Grant 62171145, Grant 61906108, Grant 62020106004, and Grant 92048301, in part by Young Creative Team in Universities of Shandong Province under Grant 2020KJN012, in part by Jinan 20 Projects in Universities under Grant 2020GXRC040, in part by Shandong Project Towards the Integration of Education and Industry under Grant 2022PYI001, Grant 2022PY009, and Grant 2022JBZ01-03, and in part by the Tianjin Education Committee Science and Technology Development Foundation under Grant 2017KJ254. This article was recommended by Associate Editor Z. Yu. *(Corresponding author: Tao Zhuo.)*

Zan Gao is with the Shandong Artificial Intelligence Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China, and also with the Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, Tianjin 300384, China (e-mail: zangaonsh4522@gmail.com).

Xinglei Cui, Tao Zhuo, and Zhiyong Cheng are with the Shandong Artificial Intelligence Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China (e-mail: 15163881521@163.com; zhuotao724@gmail.com; jason.zy.cheng@gmail.com).

An-An Liu is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: anan0422@gmail.com).

Meng Wang is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China (e-mail: eric.mengwang@gmail.com).

Shenyong Chen is with the Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, Tianjin 300384, China (e-mail: sy@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/THMS.2023.3266037.

Digital Object Identifier 10.1109/THMS.2023.3266037

of video frames are often ignored; thus, the existing methods cannot precisely locate the start and end of each instance in untrimmed videos.

To address these issues, in this article, we propose a novel multitemporal-scale spatial–temporal transformer (MSST) network for the TAL task, which can predict actions on a feature space of multiple temporal scales. Specifically, refined feature pyramids (RFPs) with different scales are first employed to pass the semantics from high-level scales to low-level scales. Second, to establish the long temporal scale of the entire video, a spatial–temporal transformer (STT) encoder is utilized to capture the long-range dependencies of video frames. Then, the refined features with long-range dependencies are fed into a classifier for coarse action prediction. Finally, to further improve the prediction accuracy, a frame-level self-attention (FSA) module is used to refine the classification and boundaries of each action instance. Most importantly, these three modules are jointly explored in a unified framework, and MSST has an anchor-free and end-to-end architecture. Extensive experiments on the THUMOS14 and ActivityNet1.3 datasets show that the proposed method is very effective and efficient. The main contributions of this article are summarized as follows.

1) We develop a novel anchor-free MSST network for the TAL task with an end-to-end network architecture. Compared to previous approaches, the proposed method can provide sufficient semantics, rich details of boundaries, and long-range dependencies for robust TAL. In this way, MSST can accurately obtain boundaries and confidence scores.
2) We design an RFP module to mine the semantic information from the high-level temporal scale that can be passed to the low-level temporal scale, and both the semantics and details of actions can be considered simultaneously. Moreover, an STT module is designed to capture the long-range dependencies of video frames. Finally, we refine the classification and boundaries of action instances with an FSA module, which can reduce the noise caused by the background.
3) Extensive experiments on the THUMOS14 and ActivityNet1.3 datasets demonstrate that MSST is very effective and robust, and it can outperform all the state-of-the-art (SOTA) methods on the THUMOS14 dataset and obtain comparable performance on the ActivityNet1.3 dataset.

II. Related Work

In this section, we introduce three types of methods: anchor-based localization, anchor-free methods, and actionness-guided localization. Anchor-based methods [6], [11], [50] generate a set of action proposals with predefined anchors at different temporal scales. Then, the actions are classified, and the boundaries are regressed. For flexible TAL, actionness-guided approaches [21], [23], [26], [36], [38], [51], [53] focus on predicting the confidence scores of the start probability, end probability, and duration of action and then combining them into proposals. For efficient TAL in videos, recent anchor-free strategies [22], [31] only need to generate a proposal at each temporal position, which is achieved by combining the regions from the current position to the start and end positions and does not require predefined anchors. In the following subsections, we will simply introduce them.

A. Anchor-Based Localization

The anchor-based methods rely on predefined anchors of different scales, which are divided into one-stage and two-stage methods. The SSAD [24] and GTAN [30] methods are one-stage methods. The SSAD [24] method utilizes a single-shot structure based on 1-D convolution to generate anchors for TAL, and GTAN [30] uses 3-D ConvNet to extract small segment-level features and uses a temporal Gaussian kernel to generate proposals with different temporal resolutions. Among two-stage methods, R-C3D [50] and TALNet [6] have a similar structure to that of [34]. R-C3D [50] proposes an end-to-end network that combines candidate segment generation and classification to learn features and accepts input videos of any length. TALNet [6] expands the receptive field and extracts the temporal context for features. In addition, late fusion is used for the two-stream architecture. Unlike the above two methods, TURN [11] divides a video into equal-length units, and it includes extracting unit-level features, classifying action instances, and regressing temporal boundaries. Unlike common anchor-based detection techniques, RCL [46] uses continuous anchoring representation to achieve high-quality action detection. Confidence scores are regressed from continuous anchor points, and their confidence scores are jointly determined by video features and temporal coordinates. Although these methods can achieve good results, they are not sufficiently flexible and result in large redundancy.

B. Anchor-Free Methods

Anchor-free methods do not require predefined anchors, and the action proposal is represented by the distances from the current position to the start and end positions. Thus, there is no need for a large number of hyperparameters, and the computational cost is relatively low. For example, CornerNet [17] uses a convolutional network to generate two sets of heatmaps to predict corners for different categories: one for the upper left corner and the other for the lower right corner. It will also find the offset positioned with respect to the corner to make the bounding box more accurate. In TAL tasks, SRF-Net [31] designs a selective receptive field convolution mechanism, which can adaptively adjust the size of the receptive field according to multiple scales of input information at each temporal localization step. The AFSD [22] framework consists of three modules: feature extraction, coarse prediction, and refined prediction. It first extracts the pyramid features from the video, then predicts the boundary and classification of the coarse proposal for each pyramid layer, and finally optimizes the boundaries and classification of each proposal through the refined prediction. In addition, A2Net [52] introduces an anchor-free method to solve the problem of video sequence lengths that are too long or too short, and it can use complementary properties to handle temporal sequences of different lengths. In this article, our method is based on the anchor-free strategy, and we use features of multiple temporal scales for TAL task.

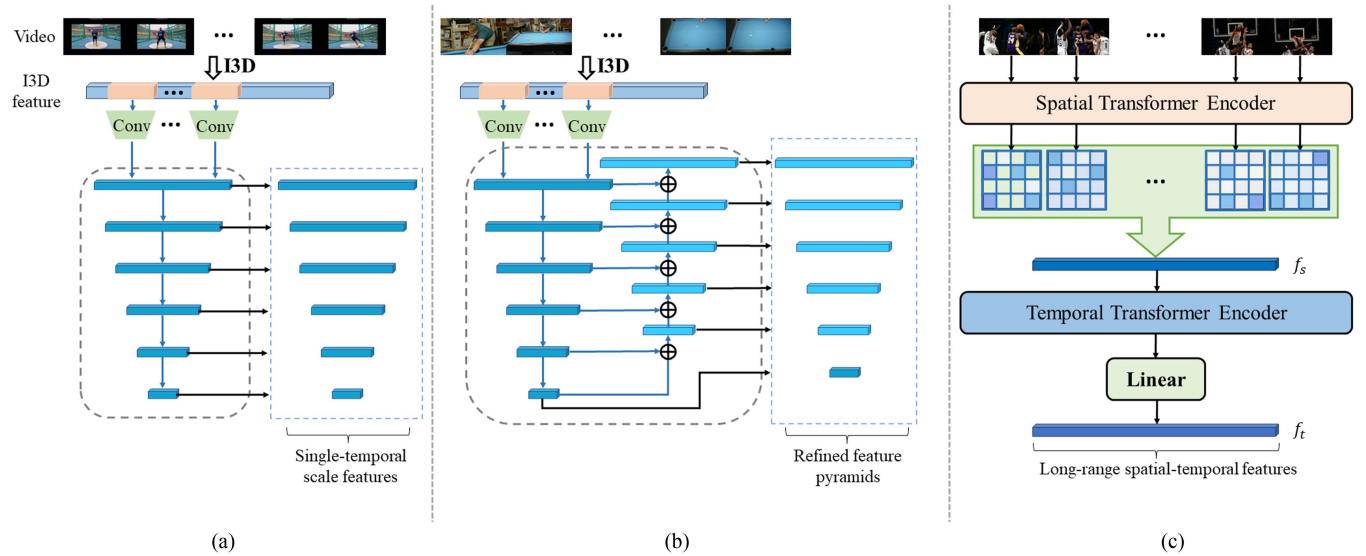

Fig. 1. Difference between our proposed method and others. (a) Original feature pyramids with a single temporal scale. (b) RFPs with multiple temporal scales that pass semantics from high-level to low-level scales. (c) STT for long-range dependencies (ours).

C. Actionness-Guided Localization

This strategy generates proposals by predicting confidence scores for the start probability, end probability, and duration. Unlike the anchor-based approach, actionness-guided localization is more flexible in handling action instances that are too long or too short. TAG [49] and SSN [59] are early methods of this type. TAG scores each sampled snippet, judges whether it is an action or not, and combines the snippets that are actions into proposals. SSN is based on the TAG method of generating proposals and divides each proposal into three stages: start, end, and activity. A pooling operation is performed on each stage, it is determined whether the action is complete, and the action is classified. Soon after, Lin et al. proposed BSN [26], LGN [25], BMN [23], and BSN++ [38]. BSN first locates the boundaries of temporal action segments and directly combines the boundaries. Then, proposal-level features are extracted based on the sequence of action confidence scores. Based on BSN, LGN proposes a "local to global" approach to jointly learn local and global contexts to generate action proposals, locally locate the accurate proposal boundary, and globally evaluate the reliable confidence score. The BSN framework was improved to yield BMN, which densely evaluates the confidence scores of all possible temporal sequences by generating a 1-D boundary probability sequence and a 2-D BM confidence map. Based on the above frameworks, BSN++ uses a boundary complementary classifier to enrich the context information for boundary prediction, and it designs a proposal relationship module that uses channelwise and positionwise global dependencies to model proposal–proposal relationships. In addition, PGCN [53] first uses graph convolutional networks to capture proposal–proposal relationships. Each proposal is represented as a node, and two proposals are represented as an edge. Two types of relationships are used: one for capturing the contextual information of each proposal and the other for describing the association between different actions. Similar to PCGN, GTAD [51] uses three GCNext modules for feature extraction, gradually aggregating temporal information and multilevel semantic information. Then, the extracted features are fed into the SGAlign layer, and the localization module obtains the scores of subgraphs, sorts them, and obtains the final result. Since this method considers all possible combinations of time positions, it incurs high computational costs.

III. MSST: A Novel End-to-End Network

To clearly show the difference between the proposed MSST and other methods, Fig. 1 is presented. Moreover, the framework of the proposed MSST method is illustrated in Fig. 2. As illustrated in Fig. 1(a), a feature space of a single temporal scale is often employed in existing anchor-free methods, but in MSST, as illustrated in Fig. 1(b), to ensure that the feature contains enough semantics and rich details of boundaries, we use nearest-neighbor linear interpolation to merge the semantic information of the high-level temporal features into the lower level. In the RFP module, the semantic information from the high-level temporal scale can be passed to the low-level temporal scale. Then, both the semantics and details of actions can be considered simultaneously. Furthermore, to establish the long temporal scale of the entire video, we propose an STT encoder to capture the long-range dependencies of video frames. As illustrated in Fig. 1(c), the factorized encoder consists of two transformer encoders in series: a spatial transformer encoder that models the latent representation of each video frame and a temporal transformer encoder that models the relationship between frames. These can represent the spatial–temporal context of videos at a long-range scale. In addition, to further enhance the foreground information of video frames, we perform a patch operation on each video frame and use the FSA module to extract the relationship between patches to obtain features with salient foreground information. This can effectively reduce the influence of background noise on action instances and enhance the foreground action features inside videos. Unlike the previous methods [22], [31], [52] that use a convolutional neural network

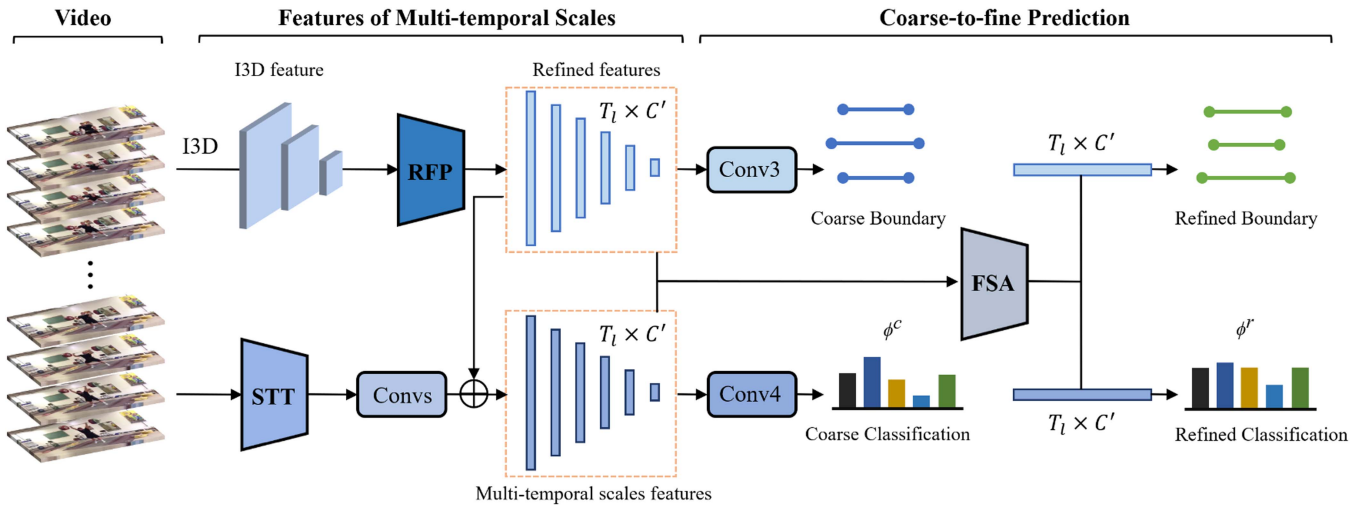

Fig. 2. Overview of the proposed MSST method. In this architecture, the I3D network is used as the feature extractor. The RFPs produce refined features with stronger semantic information at different temporal scales. The STT excavates the spatial–temporal information of videos. The coarse-to-fine prediction is designed to minimize the influence of video background noise, and it consists of an FSA module. Note that $\bigoplus$ is concatenation.

model to exploit the features of a single temporal scale, we propose using features of multiple temporal scales on videos. Based on an anchor-free TAL framework, our method can satisfactorily locate and classify various actions. Next, we describe the main modules of the proposed method.

A. Problem Formulation

Suppose that an untrimmed video $V = \{v_i\}_{i=1}^{T}$ consists of T frames. The set of action instances is represented by $X = \{t_{s,m}, t_{e,m}, \phi_m\}_{m=1}^{M}$, where s_m and e_m represent the start time and end time, respectively. ϕ_m and M indicate the category of the mth action instance and the number of action instances in the video, respectively. Our goal is to predict action segments with the start time, end time, and corresponding action category.

B. Features of Multiple Temporal Scales

The features of multiple temporal scales are generated from an RFP module and an STT module. The details of these two modules are given below.

1) Refined Feature Pyramids: We use the I3D [5] model pretrained on the Kinetics dataset for both RGB frames and optical flows to obtain I3D features. For a video $V \in R^{C\times T\times H\times W}$, C is the number of channels, T is the temporal duration, and H and W are the height and width of the video frame, respectively. The spatial–temporal features are denoted as $\mathbb{F} \in R^{C'\times T'\times H'\times W'}$ through the I3D network, where C', T', H', and W' represent the channel, temporal duration, height, and width, respectively. Then, the spatial–temporal features are converted into a 1-D feature space $\mathbb{F}_{st} \in R^{C'\times T'}$ through 3-D convolutions, and by downsampling with 1-D convolution layers, following [22], feature pyramids with six different temporal scales are generated. The dimension is 512 for the feature pyramids. The first two temporal scales use 3-D convolution downsampling with kernels [1, 6, 6] and [1, 3, 3], respectively. The remaining four temporal scales are downsampled using four identical 1-D convolutions with kernel $=$ 3 and stride $=$ 2. Inspired by the previous method [43], we design an RFP module to compensate for the semantic information in 1-D feature sequences and expand the receptive field. Specifically, we add the semantic information of higher level features to the lower level features. Then, the features of different temporal scales with stronger semantic information can be obtained. In addition, unlike other similar structures, the RFP module uses nearest-neighbor linear interpolation instead of convolution in the RFP process. Thus, there is no need to update the parameters, which reduces the number of trainable parameters and speeds up the calculation. Based on the RFP module, RFPs with stronger semantic information $f \in R^{T_l\times C'}$, $T_l \in \{2, 4, 8, 16, 32, 64\}$ can be generated, where T_i represents the different temporal spans. Moreover, a frame-level feature $f_a^c \in R^{T\times C'}$ is generated by taking the feature of the lowest layer into a feature with temporal span T adopting linear interpolation, which is used for further prediction refinement.

2) Spatial–Temporal Transformer: Long-range dependencies are important for the TAL task. Although we have obtained RFPs at different temporal scales, the long-range spatial–temporal information is still lost because downsampling using convolution has fewer temporal receptive fields obtained from the video than spatial–temporal features. Therefore, we use an STT module to extract the long-range dependencies of video frames.

Inspired by ViViT [1], we extract the long-range spatial–temporal information of videos with a factorized encoder, which consists of a spatial transformer encoder and a temporal transformer encoder in series. The spatial transformer encoder excavates the relationships within video frames from the same temporal index. Then, all the spatial feature outputs after temporal embedding are used as the input for the temporal transformer encoder. The temporal encoder connects frames with different temporal indices to capture the temporal information, and we obtain features with long-range spatial–temporal information from the STT. Therefore, the long-range spatial–temporal information is added to the output of the RFPs.

Specifically, our method employs multiheaded self-attention (MSA) [41] in parallel, and layer normalization (LN) [2] is applied before each MSA block. We first embed the spatial position

encoding for the video $V = \{v_i\}_{i=1}^{T}$ and feed it into the spatial transformer encoder to obtain a feature sequence f_s. Then, we embed the temporal information into this feature sequence and take the feature into the temporal transformer encoder. Finally, the feature $f_t \in R^{T_j \times C'}$ with long-range spatial–temporal information at one temporal scale is produced, where T_j is the temporal scale and C' is the channel. Therefore, this process is denoted as follows:

$$\begin{aligned} f_s &= \mathrm{MSA}\left(\mathrm{LN}\left(v_i\right)\right) \quad i = 1, \ldots, T \\ f_t &= \mathrm{Linear}\left(\mathrm{MSA}\left(\mathrm{LN}\left(f_s\right)\right)\right) \end{aligned} \tag{1}$$

where T represents the number of frames in the video and $\mathrm{Linear}(\cdot)$ represents the fully connected layer. To obtain multiscale features with long-range spatial–temporal information, we downsample f_t through multiple 1-D convolutions to produce feature sequences $f_l \in R^{T_l \times C'}$ at different temporal scales, $T_l \in \{2, 4, 8, 16, 32, 64\}$. Then, we concatenate the feature sequences f obtained by the RFP module and the feature sequences f_l to produce features of multiple temporal scales f_{mts}, which are represented by:

$$f_{\text{mts}} = [f, f_l] \tag{2}$$

where $[\cdot]$ indicates concatenation. We project the features of multiple temporal scales and the RFPs to f_{cls} and f_{loc}, respectively, in two 1-D convolutions at each temporal scale, which is computed as follows:

$$\begin{aligned} f_{\text{loc}} &= \mathrm{Conv1}(f) \\ f_{\text{cls}} &= \mathrm{Conv2}\left(f_{\text{mts}}\right) \end{aligned} \tag{3}$$

where f_{loc} is used for localization and f_{cls} is used for the classification of action instances. The two 1-D convolutions have the same kernel and stride.

C. Coarse-to-Fine Prediction

Based on the refined features of multiple temporal scales, coarse prediction results can be obtained with a classifier. To further improve the performance, we propose an FSA module for both the action category and boundary refinement.

1) Coarse Prediction: Based on f_{loc} and f_{cls}, we use the two 1-D convolutions to produce coarse start and end boundary distances (d_n^s, d_n^e) and a coarse class score ϕ_n^c, respectively, which are computed as follows:

$$\begin{aligned} d_n^s &= \mathrm{Conv3}\left(f_{\text{loc}}\right) \\ d_n^e &= \mathrm{Conv3}\left(f_{\text{loc}}\right) \\ \phi_n &= \mathrm{Conv4}\left(f_{\text{cls}}\right) \end{aligned} \tag{4}$$

where n represents the positions in different temporal scales, d_n^s represents the distance from position n to the start time, and d_n^e represents the distance from position n to the end time. Then, the coarse boundaries $(t_{s,n}^c, t_{e,n}^c)$ can be inferred, where $t_{s,n}^c$ and $t_{e,n}^c$ represent the coarse starting and ending times of the corresponding nth location in the refined pyramid features, $n \in \{0, 1, 2, \ldots, T_l - 1\}$.

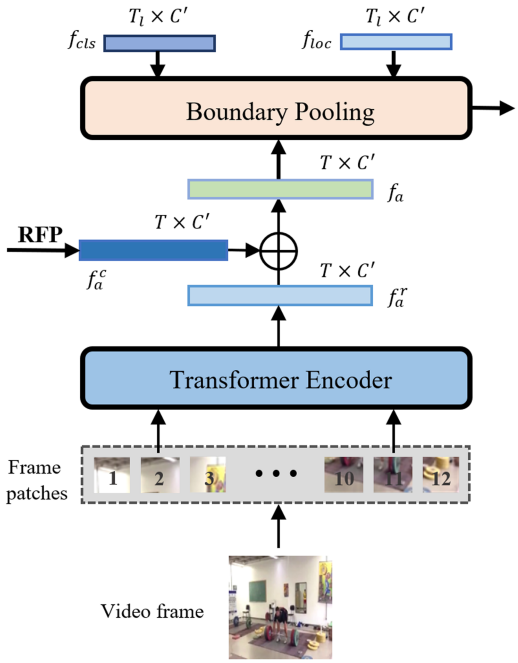

Fig. 3. FSA module. The entire video frame is divided into frame patches, embedded with the position encoding, and fed into the transformer encoder to generate f_a^r. Then, this is combined with f_a^c from the RFP module to produce f_a. Finally, features f_{loc} and f_{cls} from the coarse prediction and f_a are input into the boundary pooling module, and fine-grained features are obtained.

2) Frame-Level Self-Attention: As the features of the pyramid deepen, the temporal dimension will become small, and it will become difficult to find accurate boundaries. Therefore, we use frame-level features for boundary pooling [22]. Since frame-level features are derived from the lowest level of feature pyramids, they ignore the influence of internal background noise in each video frame. Thus, we use a self-attention [41] mechanism to minimize this influence. Inspired by ViT [9], we propose an FSA module to process the image of each frame, which maximizes the separation of foreground and background noises.

As illustrated in Fig. 3, we take all video frames as the input of a transformer encoder and divide each video frame $v_i \in R^{H' \times W' \times C'}$ into frame patches. Then, the relationship between the patches in each frame can be built. Specifically, we divide the frame of $H' \times W' \times C'$ into patches $v_{i,z}$ with a size of $D \times D$, where $z \in \{1, 2, \ldots, Z\}$ and concatenate Z patches into vectors that embed the positional encoding of the patches. Next, the vectors are fed into the transformer encoder, which consists of an MSA and an LN block. Finally, based on the outputs of the transformer encoder, we use the fully connected layer to obtain a frame-level feature to represent the foreground information, which is denoted as

$$\begin{aligned} v_{i,z} &= \mathrm{TP}\left(v_i\right), \quad i = 1, \ldots, T \\ f_a^r &= \mathrm{MSA}\left(v_{i,z} + \mathrm{PE}\left(v_{i,z}\right)\right), \quad z = 1, \ldots, Z \end{aligned} \tag{5}$$

where $\mathrm{PE}(\cdot)$ represents the patch position embedding for each video frame and $\mathrm{TP}(\cdot)$ stands for dividing video frames into

patches. Frame-level features for refined prediction f_a^c are generated by the RFP module, and f_a^r are obtained by FSA. We concatenate f_a^c and f_a^r to obtain the refined frame-level feature f_a. Then, we input the features f_{cls} and f_{loc} in the coarse prediction and the frame-level feature f_a in the fine prediction into boundary pooling [22]. The fine-grained features f_{loc}^r and f_{cls}^r are defined as follows:

$$\begin{aligned} f_{\text{loc}}^r &= \text{BP}\left(f_{\text{loc}}, f_a\right) \\ f_{\text{cls}}^r &= \text{BP}\left(f_{\text{cls}}, f_a\right) \end{aligned} \tag{6}$$

where BP is a boundary pooling method [22]. Fine-grained predictions for f_{loc}^r and f_{cls}^r are produced by two different 1-D convolution layers. Specifically, one convolution layer is used to predict the offsets $(\Delta t_{s,n}^r, \Delta t_{e,n}^r)$ for boundary regression, and the other is used to predict the refined class score ϕ^r. Finally, we add the offsets of the boundaries $(\Delta t_{s,n}^r, \Delta t_{e,n}^r)$ to the coarse boundary and obtain the refined boundaries $(t_{s,n}^r, t_{e,n}^r)$.

D. Training and Inference

1) Training: The set of action instances predicted by coarse prediction and refined prediction contains N samples, which is larger than the maximum number of ground-truth action instances in the dataset. MSST uses multiple losses for coarse boundary regression and classification, and the binary cross-entropy loss is used for the probability of the proposal. The computation of our total loss function can be written as

$$L = L_{\text{cls}}^c + L_{\text{cls}}^r + \alpha\left(L_{\text{loc}}^c + L_{\text{loc}}^r\right) + \beta L_{\text{bce}} \tag{7}$$

where L is the total loss and L_{cls}^c and L_{cls}^r are the losses of the coarse classification and refined classification, respectively. L_{loc}^c and L_{loc}^r are the losses of the coarse boundary regression and refined boundary regression, respectively. L_{bce} is the loss of binary cross-entropy [22]. α and β are hyperparameters. For the coarse classification, focal loss [27] is applied as the constraint because it can not only adjust the weights of positive and negative samples but also control the weights of difficult and easy classification samples; it is computed as follows:

$$L_{\text{cls}}^c = \frac{1}{N^c}\sum_{n=1}^{N^c} \text{L}_{\text{focal}}\left(\phi_n^c, \phi_n\right) \tag{8}$$

where N^c is the number of positive samples in the coarse process and a sample is regarded as positive when it is located among the ground-truth samples. ϕ_n^c is the set of coarse classification results, and ϕ_n is the set of ground-truth labels. For the refined classification, we use a focal loss as follows:

$$L_{\text{cls}}^r = \frac{1}{N^r}\sum_{n=1}^{N^r} \text{L}_{\text{focal}}\left(\phi_n^r, \phi_n\right) \tag{9}$$

where N^r is the number of positive samples when the coarse proposals have a tIoU higher than 0.5 with the ground-truth samples. ϕ_n^r is the refined set of predicted classification results, and ϕ_n is the set of ground-truth labels. We adopt GIoU loss [35] as the constraint for coarse boundary regression, which is computed as follows:

$$L_{\text{loc}}^c = \frac{1}{N^c}\sum_{n=1}^{N^c}\left(1 - \text{GIoU}\left(\psi_n^c, \psi_n\right)\right) \tag{10}$$

where $\psi_n^c = (t_{s,n}^c, t_{e,n}^c)$ is the coarse boundary predicted by the coarse process and $\psi_n = (t_{s,n}, t_{e,n})$ is the corresponding ground truth. For the refined boundary regression, we use the smooth L1 loss [13] as the loss function, which can be calculated as

$$L_{\text{loc}}^r = \frac{1}{N^r}\sum_{n=1}^{N^r}\left(\text{smooth}_{L_1}\left(\hat{\Delta}_n, \Delta_n\right)\right) \tag{11}$$

where $\hat{\Delta}_n = (\Delta t_{s,n}^c, \Delta t_{e,n}^c)$ is the offset between the coarse boundaries and the corresponding ground truth. $\Delta_n = (\Delta t_{s,n}^r, \Delta t_{e,n}^r)$ is the regression target of our refined process. In addition, we use binary cross-entropy loss to suppress proposals with low quality, which is defined as

$$L_{\text{bce}} = \frac{1}{N^r}\sum_{n=1}^{N^r}\text{BCE}\left(\varepsilon_n, \frac{|\psi_n^r \cap \psi_n|}{|\psi_n^r \cup \psi_n|}\right) \tag{12}$$

where BCE is the binary cross-entropy loss. ψ_n^r and ψ_n are the refined boundaries and the corresponding ground truth, respectively. ε_n is the location.

2) Inference: In the inference stage, we use the coarse boundaries (t_s^c, t_e^c), coarse classification results ϕ^c, offsets from the refined process $(\Delta t_s^r, \Delta t_s^r)$, refined classification results ϕ^r, and confidence scores ε obtained by our network. The final prediction for each clip can be computed as follows:

$$\begin{aligned} t_{s,n}^p &= t_{s,n}^c + \frac{1}{2}d_n^c \Delta t_{s,n}^r \\ t_{e,n}^p &= t_{e,n}^c + \frac{1}{2}d_n^c \Delta t_{e,n}^r \\ \phi_n^p &= \frac{1}{2}\left(\phi_n^c + \phi_n^r\right)\varepsilon_n \end{aligned} \tag{13}$$

where $d_n^c = t_{e,n}^c - t_{s,n}^c$. Finally, we adopt Soft-NMS [4] to process all predictions to suppress redundant proposals.

IV. Experiments

A. Datasets

To verify the effectiveness of our proposed method, we strictly follow the latest published references [22], [31], [38] and also conduct experiments on the following two benchmark datasets.

1) THUMOS14 [15]: This dataset contains 1010 validation videos and 1574 testing videos with 101 action categories. We follow [15]: 200 untrimmed videos in the validation set and 213 untrimmed videos in the test set are used for training and testing, respectively. These videos contain 20 categories labeled for TAL. Each video has more than 15 action annotations.

2) ActivityNet1.3 [14]: This dataset contains 19 994 untrimmed videos with 200 action categories. We follow the settings in [14] and divide the dataset into training, testing, and validation sets with a ratio of 2:1:1. There are approximately 1.5 action instances for each video.

TABLE I
PERFORMANCE COMPARISON WITH SOTA METHODS ON THE THUMOS14 DATASET MEASURED BY MAP WITH DIFFERENT TIOU THRESHOLDS

Type	Methods	Backbone	0.1	0.2	0.3	0.4	0.5	0.6	0.7	Avg {0.1:0.5}	Avg {0.3:0.7}
Anchor-based	SSAD [24]	TS	50.1	47.8	43.0	35.0	24.6	-	-	40.1	-
	TURN [11]	C3D	54.0	50.9	44.1	34.9	25.6	-	-	41.9	-
	R-C3D [50]	C3D	54.5	51.5	44.8	35.6	28.9	-	-	43.1	-
	CBR [10]	TS	60.1	56.7	50.1	41.3	31.0	19.1	9.9	47.8	30.3
	TALNet [6]	I3D	59.8	57.1	53.2	48.5	42.8	33.8	20.8	52.3	39.8
	GTAN [30]	P3D	69.1	63.7	57.8	47.2	38.8	-	-	55.3	-
	PCG-TAL [39]	I3D	71.2	68.9	65.1	59.5	51.2	-	-	63.2	-
Actionness	CDC [36]	-	-	-	40.1	29.4	23.3	13.1	7.9	-	22.8
	SSN [59]	TS	60.3	56.2	50.6	40.8	29.1	-	-	47.4	-
	TAG [49]	TS	64.1	57.7	48.7	39.8	28.2	-	-	47.7	-
	BSN [26]	TS	-	-	53.5	45.0	36.9	28.4	20.0	-	36.8
	BMN [23]	TS	-	-	56.0	47.4	38.8	29.7	20.5	-	38.5
	DBG [21]	TS	-	-	57.8	49.4	42.8	33.8	21.7	-	41.1
	GTAD [51]	TS	-	-	54.5	47.6	40.2	30.8	23.4	-	39.3
	BSN++ [38]	TS	-	-	59.9	49.5	41.3	31.9	22.8	-	41.1
	BU-TAL [56]	TS	-	-	53.9	50.7	45.4	38.0	28.5	-	43.3
	TCA-Net [33]	TS	-	-	60.6	53.2	44.6	36.8	26.7	-	44.4
	RTD-Action [40]	TS	-	-	68.3	62.3	51.9	38.8	23.7	-	49.0
	RCL [46]	TS	-	-	70.1	62.3	52.9	42.7	30.7	-	51.7
	DCAN [8]	TS	-	-	68.2	62.7	54.1	43.9	32.6	-	52.3
Others	SCNN [37]	-	47.7	43.5	36.3	28.7	19.0	-	-	19.0	-
	GTAD+PGCN [51]	TS	-	-	66.4	60.4	51.6	37.6	22.9	-	47.8
	ContextLoc [61]	I3D	-	-	68.3	63.8	54.3	41.8	26.2	-	50.9
	Sub-Action [42]	I3D	66.1	60	52.3	43.2	32.9	-	-	50.9	-
	VSGN [55]	TS	-	-	66.7	60.4	52.4	41.0	30.4	-	50.2
Anchor-free	A2Net [52]	I3D	61.1	60.2	58.6	54.1	45.5	32.5	17.2	55.9	41.6
	SRF-Net [31]	C3D	-	-	56.5	50.7	44.8	33.0	20.9	-	41.2
	AFSD [22]	I3D	-	-	67.3	62.4	55.5	43.7	31.1	-	52.0
	MSST (Ours)	I3D	**75.3**	**73.8**	**70.5**	**65.0**	**56.9**	**46.0**	**32.7**	**68.3**	**54.2**

The bold numbers represent the best performance.

B. Experimental Settings

1) Parameters: In our experiments, we follow the experimental setup of [22]. On the THUMOS14 dataset, we sample RGB and optical flow frames using a frame rate of 10 frames/s (FPS) and split the video into clips. For each clip, we set its length T to 256 frames, and adjacent clips have a temporal overlap, which is set to 30 in training and 128 in testing. On the ActivityNet1.3 dataset, the frames are sampled with different FPS, and we guarantee that the number of each video frame is 768. On both the datasets, we set the size of each frame to 96×96 and the size of frame patches D to 24. We also use random cropping and horizontal flipping as data augmentation during training.

We use the Adam [16] optimizer for model training, and the number of epochs is set to 25. In addition, the learning rate is 10^{-5}, the weight decay is 10^{-3}, and the batch size is 1. The hyperparameters are empirically defined as $\alpha = 10$ and $\beta = 1$ on the THUMOS14 dataset. On the ActivityNet1.3 dataset, $\alpha = 1$ and $\beta = 1$. In addition, the tIoU threshold in Soft-NMS [4] is set to 0.3 for THUMOS14 and 0.85 for ActivityNet1.3.

2) Evaluation Metrics: In the TAL task, the mean average precision (mAP) is used as the evaluation metric. We report the mAP for all the experiments. In addition, the tIoU thresholds are [0.1:0.1:0.7] for THUMOS14 and [0.5:0.05:0.95] for ActivityNet1.3.

C. Comparison With SOTA Methods

To verify the effectiveness of the proposed MSST method, we follow AFSD [22], RCL [46], VSGN [55], and RTD-Action [40] and compare the latest TAL methods, which include anchor-based methods, anchor-free methods, and actionness-guided methods. Tables I and II report the comparison results on the THUMOS14 and ActivityNet1.3 datasets, respectively. We discuss the results as follows.

1) Results on THUMOS14: Table I reports the comparison results on the THUMOS14 dataset. For some methods, their tIoU thresholds of 0.1 and 0.2 or 0.6 and 0.7 are not reported. Therefore, we report Avg$\{0.1 : 0.5\}$ and Avg$\{0.3 : 0.7\}$, which represent the average mAP for all tIoU thresholds $\{0.1 : 0.1 : 0.5\}$ and $\{0.3 : 0.1 : 0.7\}$, respectively. Our method outperforms its strong opponents AFSD, RTD-Action, RCL, DCAN, and ContextLoc for all thresholds on Avg$\{0.3 : 0.7\}$ and exceeds 2.3%, 5.2%, 2.5%, 1.9%, and 3.3%, respectively. For Avg$\{0.1 : 0.5\}$, the MSST method outperforms PCG-TAL and A2Net by 5.1% and 12.4%, respectively. These results show that our method significantly outperforms the current SOTA methods, and the MSST method improves from 55.5% to 56.9% when the threshold is 0.5. At a threshold of 0.6, the MSST method exceeds the DCAN method by 2.1%, and the MSST method is slightly improved compared with DCAN for a threshold of 0.7. In particular, when the threshold is 0.4, the MSST method is

TABLE II
PERFORMANCE COMPARISON WITH SOTA METHODS ON THE ACTIVITYNET1.3 DATASET MEASURED BY MAP AT DIFFERENT TIOU THRESHOLDS

Type	Methods	0.5	0.75	0.95	Avg
Anchor-based	R-C3D [50]	26.8	-	-	-
	TALNet [6]	38.2	18.3	1.3	20.2
	GTAN [30]	52.6	34.1	8.9	34.3
	PCG-TAL [39]	44.3	29.9	5.5	28.9
Actionness	CDC [36]	45.3	26.0	0.2	23.8
	SSN [59]	43.2	28.7	5.6	28.3
	TAG [49]	41.1	24.1	5.0	24.9
	BSN [26]	46.5	30.0	8.0	30.0
	BMN [23]	50.1	34.8	8.3	33.9
	GTAD [51]	50.4	34.6	9.0	34.1
	BC-GNN [3]	50.6	34.8	**9.4**	34.3
	BSN++ [38]	51.3	**35.7**	9.0	**34.9**
	BU-TAL [56]	43.5	33.9	9.2	30.1
	RTD-Action [40]	47.2	30.7	8.6	30.8
Others	PGCN [53]	48.3	33.2	3.3	31.1
	ContextLoc [61]	**56.0**	35.2	3.6	34.2
	VSGN [55]	52.3	35.2	8.3	34.7
	Sub-Action [42]	37.1	24.1	5.8	24.1
Anchor-free	A2Net [52]	43.6	28.7	3.7	27.8
	AFSD [22]	52.4	35.3	6.5	34.4
	MSST (Ours)	52.4	34.7	6.0	34.1

more than 2% better than SOTA methods. The MSST method is only 0.4% better than the latest RCL method at a threshold of 0.3. For thresholds of 0.1 and 0.2, the MSST method is more than 3% better than PCG-TAL and more than 13% better than A2Net. Although A2Net and AFSD are anchor-free methods, A2Net leverages the advantages of anchor-based and anchor-free methods, and AFSD focuses on refining the boundaries with a saliency-based refinement module, which ignores the importance of sufficient semantics and long-range spatial–temporal information. In MSST, features of multiple temporal scales with this information are used for TAL. From the above discussion results, we can see that we have the best results for all tIoU thresholds. Therefore, compared with anchor-based and anchor-free methods, the MSST method is better by a large margin for all tIoU thresholds.

2) Results on ActivityNet1.3: Table II shows the comparison results on the ActivityNet1.3 dataset, where Avg indicates the average mAP for all tIoU thresholds {0.5:0.05:0.95}. The MSST method can obtain results comparable to those of other SOTA methods on the ActivityNet1.3 dataset. Because the action instances in the ActivityNet1.3 dataset are long and the scenes of some actions are discontinuous, it is very challenging for anchor-free methods to detect all the action instances. Thus, most TAL methods cannot achieve the best performance on both the datasets simultaneously. For example, RTD-Action can achieve better performance on the THUMOS14 dataset, and the result is 49.0% for Avg$\{0.3:0.7\}$ but only 30.8% on the ActivityNet1.3 dataset. Similarly, BSN++ achieves 34.9% on the ActivityNet1.3 dataset, but it is inefficient on the THUMOS14 dataset, and the result is only 41.1% for Avg$\{0.3:0.7\}$. The reason for this is that the temporal difference between the THUMOS14 dataset and the ActivityNet1.3 dataset is very large. At the same time, TCA-Net achieves the best result on the ActivityNet1.3 dataset, which is 1.4% higher than that of our method but 9.8% lower than that of MSST on the THUMOS14 dataset. Because the feature extraction method we use on the ActivityNet1.3 dataset is different from that of TCA-Net, the feature extraction method of TSN [48] adopted by TCA-Net often performs better on the ActivityNet1.3 dataset than the I3D [5] method. Therefore, some methods can perform well on the THUMOS14 dataset but poorly on the ActivityNet1.3 dataset, and *vice versa*.

TABLE III
COMPARISON OF INFERENCE TIMES ON THE THUMOS14 DATASET

Model	GPU	FPS
SCNN[37]	-	60
CDC[36]	TITAN Xm	500
R-C3D[50]	TITAN Xm	569
R-C3D[50]	TITAN Xp	1030
PBRNet[47]	1080Ti	1488
AFSD[22]	1080Ti	3259
AFSD[22]	V100	4057
MSST (Ours)	V100	1824

From Table II, it can be seen that the actionness-guided methods are more suitable for the ActivityNet1.3 dataset, and they are more likely to obtain better performance. Many proposals are generated, which may include samples of various durations, but for anchor-free methods, the distance to each temporal position can be predicted, which may not cover samples with long durations. For instance, BSN++ and GTAD are actionness-guided methods. They show competitive effects on the ActivityNet1.3 dataset, but the results on the THUMOS14 dataset are not satisfactory. AFSD is an anchor-free method, and its result on the THUMOS14 dataset is very competitive, but the result is not as good as those of the actionness-guided methods on the ActivityNet1.3 dataset. In addition, ContextLoc is a strong opponent on the THUMOS14 dataset, which is 3.3% lower than MSST for Avg$\{0.3:0.7\}$. However, on the ActivityNet1.3 dataset, it is only 0.1% better than ours. Similar to ContextLoc, comparing MSST with GTAD has the same result on the ActivityNet1.3 dataset, but MSST is 14.9% better than GTAD on the THUMOS14 dataset for Avg$\{0.3:0.7\}$. For GTAN, our method is 13% better than on the THUMOS14 dataset for Avg$\{0.1:0.5\}$. However, MSST is only 0.2% worse than GTAN on the ActivityNet1.3 dataset. Thus, although the proposed MSST method does not achieve the best results, its performance is still competitive. From the above experimental analysis, it can be concluded that our method has good generalization ability.

3) Inference Time: We compare the inference speeds of MSST and the other methods on the THUMOS14 dataset. Our model is evaluated on a Tesla V100 GPU, and its inference speed is 1824 FPS. As shown in Table III, MSST is fast, and the inference speed achieved second place. Our method is slower than AFSD for two reasons. First, we use a transformer to excavate spatial–temporal information between frames. Next, we divide each frame of the video into patches and pass each patch through the self-attention mechanism. These require a higher computational cost. Nevertheless, MSST is still efficient.

TABLE IV
BENEFITS OF THE RFP, STT, AND FSA MODULES ON THE THUMOS14 DATASET

Baseline	✓	✓	✓	✓	✓
RFP		✓		✓	✓
STT			✓	✓	✓
FSA					✓
Avg {0.3:0.7}	52.0	52.6	52.8	53.1	**54.2**

TABLE V
IMPACT OF DIFFERENT TIOU THRESHOLDS OF SOFT-NMS ON THE THUMOS14 DATASET

tIoU threshold	mAPs					Avg
	0.3	0.4	0.5	0.6	0.7	
0.2	**70.80**	**65.12**	56.73	45.63	32.10	54.08
0.3	70.50	65.04	**56.89**	46.03	32.71	**54.23**
0.4	70.06	64.74	56.72	46.16	33.11	54.16
0.5	69.56	64.37	56.53	**46.19**	**33.29**	53.99

V. ABLATION STUDY

To verify the effectiveness of each module in our method, we follow the previous methods [22], [31], [43], [44], [52], [53], [56] to conduct an ablation study on the THUMOS14 and ActivityNet1.3 datasets. We mainly focus on the following aspects: 1) the benefits of the RFP, STT, and FSA modules; 2) the impacts of different tIoU thresholds; 3) the visualization results; and 4) the convergence analysis.

A. Benefits of the RFP, STT, and FSA Modules.

To verify the effects of our proposed modules, we evaluate these modules and verify the feasibility of module stacking. In Table IV, the baseline reaches 52.0% on the THUMOS14 dataset. When we add the RFP module to obtain RFPs with semantic information at different temporal scales, the improvement is 0.6%, which shows that the semantic information of the features is indispensable for TAL tasks. In addition, we propose an STT to fully excavate spatial–temporal information and generate features with long-range dependencies in the video, which improves the performance by 0.8%. Then, by combining the RFP and STT modules, the features of multitemporal scales are obtained, and the average mAP is further improved by 1.1%. This shows the effectiveness of the two proposed modules, and features of multitemporal scales are very important for MSST. In addition, we use an FSA module to distinguish the foreground and background, which captures the foreground information inside each frame in the video, reduces noise, and further improves the localization and classification of action instances. By fusing all the three modules, the proposed MSST method achieves the best performance (54.2%) on the THUMOS14 dataset, outperforming the baseline by 2.2%. Based on the above analysis, it can be seen that our three proposed modules are very effective for TAL tasks.

B. Impact of Different tIoU Thresholds

For the MSST method to achieve the best performance, we follow the approach of [44] and [52] and discuss the impact of different tIoU thresholds on the THUMOS14 dataset. As shown in Table V, when the tIoU threshold is 0.2, mAPs of 0.3 and 0.4 yield the best performance. When the tIoU threshold is 0.5, mAPs of 0.6 and 0.7 yield the best performance. When the tIoU threshold is 0.3, only a mAP of 0.5 yields the best performance, but the Avg, which is 0.15% higher than that of the tIoU threshold of 0.2 and 0.21% higher than that of the tIoU threshold of 0.5, is the best result. Although other tIoU thresholds yield the best results for different mAPs, we set the tIoU threshold to 0.3 because it achieves the best performance on average. Setting the tIoU threshold to 0.3 achieves a performance that is as good as possible on all mAPs, which makes the fitting of the MSST method better.

C. Visualization Results

In this section, we visualize the comparison between the predicted results of MSST and the ground truth on the THUMOS14 dataset. Some of the results are shown in Figs. 4 and 5. The coarse boundaries represent the results without FSA refinement, and the refined boundaries represent the results after FSA refinement. Although the coarse results are very close to the ground truth, the refined results are more accurate after FSA refinement. The successful cases are shown in Fig. 4. For the first row, the action instances have similar backgrounds in the same video, and action classification is relatively easy. Therefore, the FSA refines the coarse boundaries and obtains refined boundaries. Some actions in the second row of the video have an error in coarse classification because the action has scenes similar to those of other actions, such as "HighJump" and "BasketballDunk." The coarse classification was refined by FSA to obtain an accurate classification. For the third row, action instances in the same video have different backgrounds, which are very difficult to predict, but FSA can refine the prediction results. The failed cases are shown in Fig. 5. For the first row, the result of the coarse classification is wrong because the background has a greater impact on the action instances, and FSA can minimize the impact of the background on the foreground action so that the final accurate results are obtained. For the second row, the coarse results and refined results are wrong, yet the FSA makes the refined boundaries more precise than the coarse boundaries. This shows that the FSA model is very effective in enabling MSST to refine the results.

D. Convergence Analysis

To further demonstrate the effectiveness of the MSST method, in this section, we analyze the loss convergence of MSST on the THUMOS14 dataset and compare it with other anchor-free SOTA methods, such as AFSD [22] and A2Net [52]. As shown in Fig. 6, we give the RGB stream and optical flow stream loss curves of AFSD and MSST and give the loss curve of A2Net. As seen from these curves, the MSST method, whether in the RGB stream or optical flow stream, can converge in 20–25 epochs, so the convergence speed is very fast. Compared with AFSD, our method converges faster in both the RGB stream and optical flow stream, thus further proving that our method is effective.

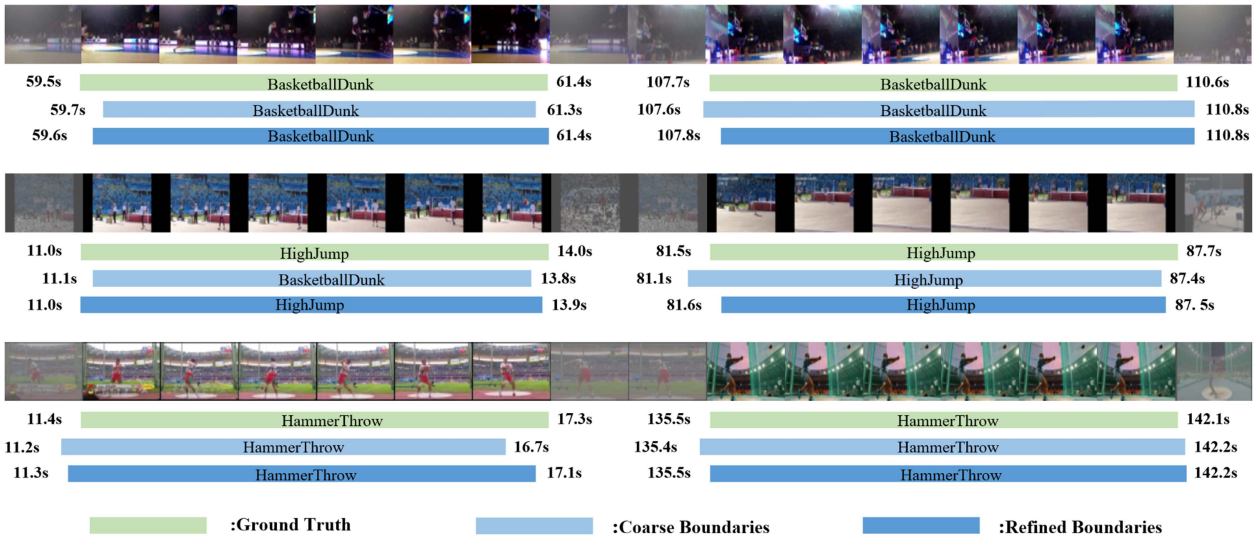

Fig. 4. Visualization examples of the successful results obtained by the proposed MSST on the THUMOS14 dataset.

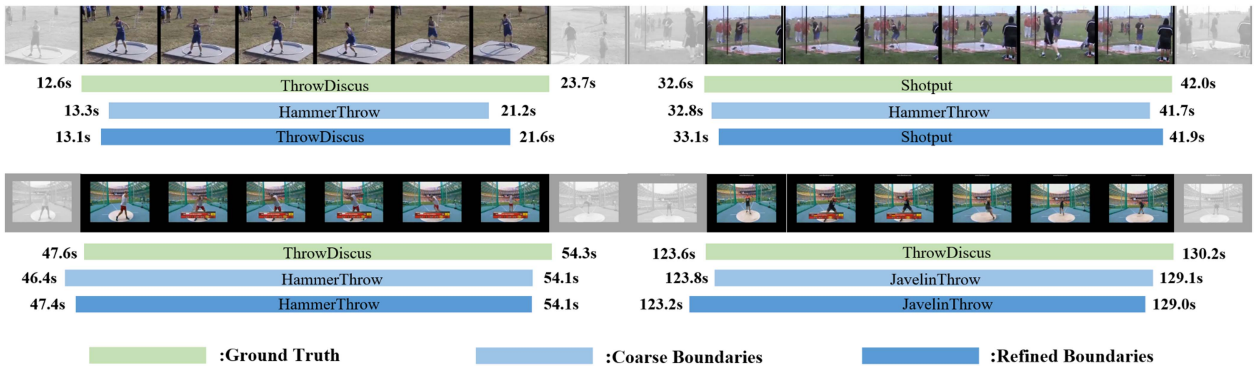

Fig. 5. Visualization examples of the failed results obtained by the proposed MSST on the THUMOS14 dataset.

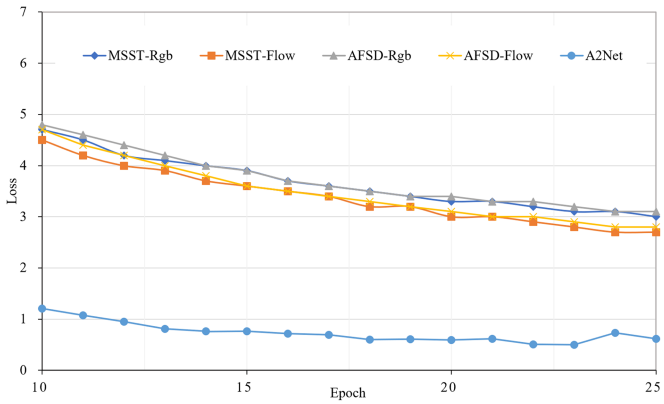

Fig. 6. Convergence curves of MSST on the THUMOS14 dataset, where the x-coordinate denotes the number of epochs and the y-coordinate indicates the loss values.

VI. Conclusion

TAL plays an important role in a large number of practical applications. Examples include video analysis and summarization, human interaction identification, and video caption generation. In this article, we present a novel MSST network for TAL, which predicts actions on a feature space of multiple temporal scales. The results of extensive experiments conducted on two public TAL datasets demonstrate that MSST can significantly outperform SOTA TAL methods on the THUMOS14 dataset and can obtain comparable results on the ActivityNet1.3 dataset. An ablation study also proves that the RFP module can satisfactorily mine semantic information from high to low levels, and the STT module can satisfactorily capture the long-range dependencies of video frames, which can yield accurate boundaries and confidence scores. Then, the FSA module can capture the foreground information of each frame well, which can reduce the influence of background noise. Note that this method can be applied to the capture of exciting moments of sports events, the detection of illegal actions, and the video surveillance of intelligent security. Moreover, the use of TAL not only can reduce labor costs, but also improve the efficiency of capture and detection, and actively promotes the development of artificial intelligence.

The main limitations of this method are that each video frame is divided into patches and many calculations are required, which are difficulties that are also faced by many previous methods. In the future, we will further explore how to mine the temporal relation information and how to reduce the amount of computation needed for dividing patches.

REFERENCES

[1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "VIVIT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6816–6826.

[2] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[3] Y. Bai, Y. Wang, Y. Tong, Y. Yang, Q. Liu, and J. Liu, "Boundary content graph neural network for temporal action proposal generation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 121–137.

[4] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5562–5570.

[5] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4724–4733.

[6] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster R-CNN architecture for temporal action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1130–1139.

[7] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 1, pp. 51–61, Feb. 2015.

[8] G. Chen, Y.-D. Zheng, L. Wang, and T. Lu, "DCAN: Improving temporal action detection via dual context aggregation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 248–257.

[9] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent.*, 2021.

[10] J. Gao, Z. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–11.

[11] 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.

[12] Z. Gao, L. Guo, T. Ren, A.-A. Liu, Z.-Y. Cheng, and S. Chen, "Pairwise two-stream ConvNets for cross-domain action recognition with small data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1147–1161, Mar. 2022.

[13] R. B. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[14] 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.

[15] H. Idrees et al., "The THUMOS challenge on action recognition for videos in the wild," *Comput. Vis. Image Underst.*, vol. 155, pp. 1–23, 2017.

[16] P. D.Kingma, and J. Ba, "ADAM: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.

[17] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 765–781.

[18] C. Li, Y. Hou, P. Wang, and W. Li, "Multiview-based 3-D action recognition using deep networks," *IEEE Trans. Human-Mach. Syst.*, vol. 49, no. 1, pp. 95–104, Feb. 2019.

[19] J. Li, X. Liu, M. Zhang, and D. Wang, "Spatio-temporal deformable 3D convnets with attention for action recognition," *Pattern Recognit.*, vol. 98, 2020, Art. no. 107037.

[20] J. Li, X. Liu, W. Zhang, M. Zhang, J. Song, and N. Sebe, "Spatio-temporal attention networks for action recognition and detection," *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 2990–3001, Nov. 2020.

[21] C. Lin et al., "Fast learning of temporal action proposal via dense boundary generator," in *Proc. 34th Conf. Artif. Intell.*, 2020, pp. 11499–11506.

[22] C. Lin et al., "Learning salient boundary feature for anchor-free temporal action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3320–3329.

[23] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "BMN: Boundary-matching network for temporal action proposal generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3888–3897.

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

[25] T. Lin, X. Zhao, and H. Su, "Joint learning of local and global context for temporal action proposal generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, pp. 4899–4912, Dec. 2020.

[26] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "BSN: Boundary sensitive network for temporal action proposal generation," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 3–21.

[27] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, pp. 318–327, Feb. 2020.

[28] Z. Liu et al., "Temporal feature alignment and mutual information maximization for video-based human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10996–11006.

[29] Z. Liu et al., "Investigating pose representations and motion contexts modeling for 3D motion prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 681–697, Jan. 2023.

[30] 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.

[31] Z. Ou, Y. Luo, J. Chen, and G. Chen, "SRFNet: Selective receptive field network for human pose estimation," *J. Supercomput.*, vol. 78, pp. 691–711, 2022.

[32] A. G. Perera, Y. W. LawT. T. Ogunwa, and J. S. Chahl, "A multiviewpoint outdoor dataset for human action recognition," *IEEE Trans. Human-Mach. Syst.*, vol. 50, no. 5, pp. 405–413, Oct. 2020.

[33] Z. Qing et al., "Temporal context aggregation network for temporal action proposal refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 485–494.

[34] 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.

[35] H. Rezatofighi, N. Tsoi, J. Y. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.

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

[37] 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.

[38] H. Su, W. Gan, W. Wu, Y. Qiao, and J. Yan, "BSN++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 2602–2610.

[39] R. Su, D. Xu, L. Sheng, and W. Ouyang, "PCG-TAL: Progressive cross-granularity cooperation for temporal action localization," *IEEE Trans. Image Process.*, vol. 30, pp. 2103–2113, 2021.

[40] J. Tan, J. Tang, L. Wang, and G. Wu, "Relaxed transformer decoders for direct action proposal generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13506–13515.

[41] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[42] B. Wang, X. Zhang, and Y. Zhao, "Exploring sub-action granularity for weakly supervised temporal action localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2186–2198, Apr. 2022.

[43] C. Wang, H. Cai, Y. Zou, and Y. Xiong, "RGB stream is enough for temporal action detection," 2021, *arXiv:2107.04362*.

[44] H. Wang, D. Damen, M. Mirmehdi, and T. Perrett, "TVNet: Temporal voting network for action localization," in *Proc. 17th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2022, vol. 5, pp. 550–558.

[45] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. O. Ogunbona, "Action recognition from depth maps using deep convolutional neural networks," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 4, pp. 498–509, Aug. 2016.

[46] Q. Wang, Y. Zhang, Y. Zheng, and P. Pan, "RCL: Recurrent continuous localization for temporal action detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13566–13575.

[47] L. Xiao et al., "PBRnet: Pyramidal bounding box refinement to improve object localization accuracy," 2020, *arXiv:2003.04541*.

[48] Y. Xiong et al., "CUHK & ETHZ & SIAT submission to ActivityNet challenge," 2016, *arXiv:1608.00797*.

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

[50] 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. 5794–5803.

[51] M. Xu, C. Zhao, D. S. Rojas, A. K. Thabet, and B. Ghanem, "G-TAD: Sub-graph localization for temporal action detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10153–10162.

[52] L. Yang, H. Peng, D. Zhang, J. Fu, and J. Han, "Revisiting anchor mechanisms for temporal action localization," *IEEE Trans. Image Process.*, vol. 29, pp. 8535–8548, 2020.
[53] R. Zeng et al., "Graph convolutional networks for temporal action localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7093–7102.
[54] J. Zhang et al., "A spatial attentive and temporal dilated (SATD) GCN for skeleton-based action recognition," *CAAI Trans. Intell. Technol.*, vol. 7, no. 1, pp. 46–55, 2022.
[55] C. Zhao, A. K. Thabet, and B. Ghanem, "Video self-stitching graph network for temporal action localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13638–13647.
[56] P. Zhao, L. Xie, C. Ju, Y. Zhang, Y. Wang, and Q. Tian, "Bottom-up temporal action localization with mutual regularization," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 539–555.
[57] Y. Zhao, H. Zhang, Z. Gao, W. Gao, M. Wang, and S. Chen, "A novel action saliency and context-aware network for weakly-supervised temporal action localization," *IEEE Trans. Multimedia*, early access, doi: 10.1109/TMM.2023.3234362.
[58] Y. Zhao et al., "A temporal-aware relation and attention network for temporal action localization," *IEEE Trans. Image Process.*, vol. 31, pp. 4746–4760, 2022.
[59] 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.
[60] X. Zheng, Y. Zhang, Y. Zheng, F. Luo, and X. Lu, "Abnormal event detection by a weakly supervised temporal attention network," *CAAI Trans. Intell. Technol.*, vol. 7, no. 3, pp. 419–431, 2022.
[61] Z. Zhu, W. Tang, L. Wang, N. Zheng, and G. Hua, "Enriching local and global contexts for temporal action localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13496–13505.

Zan Gao (Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2011.

From September 2009 to September 2010, he was with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. From July 2016 to January 2017, he was also with the School of Computing, National University of Singapore, Singapore. From 2011 to 2018, he was with the School of Computer Science and Engineering, Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, Tianjin, China. He is currently a Full Professor with the Shandong Artificial Intelligence Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. He has authored more than 100 scientific articles in international conferences and journals, including IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, CVPR, ACM MM, SIGIR, WWW, and AAAI. His research interests include artificial intelligence, multimedia analysis and retrieval, computer vision, and machine learning.

Xinglei Cui received the bachelor's degree from the Weifang University ,Weifang, China, in 2020. He is currently working toward the master's degree with Shandong Artificial Intelligence Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China.

His research interests include multimedia analysis and retrieval, computer vision, and machine learning.

Tao Zhuo received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, Xi.an, China, in 2016.

From 2016 to 2021, he was a Research Fellow with the National University of Singapore, Singapore. He is currently with Shandong Artificial Intelligence Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His research interests include image/video processing, computer vision, and machine learning.

Zhiyong Cheng received the Ph.D. degree in computer science from Singapore Management University, Singapore.

From 2014 to 2015, he was a Visiting Student with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Full Professor with the Shandong AI Institute, Qilu University of Technology, Jinan, China. He has authored or coauthored papers published in a set of top forums, including ACM SIGIR, MM, WWW, IJCAI, *ACM Transactions on Information Systems*, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON CYBERNETICS. His research interests include large-scale multimedia content analysis and retrieval.

An-An Liu (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees from Tianjin University, Tianjin, China.

He is currently a Professor with the School of Electronic Information Engineering, Tianjin University. He was a Visiting Scholar with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, where he worked with Prof. Take Kanade. His research interests include computer vision and machine learning.

Meng Wang (Fellow, IEEE) received the B.E. and Ph.D. degrees (in the Special Class for the Gifted Young) from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China, respectively.

He is currently a Professor with the Hefei University of Technology, Hefei. His current research interests include multimedia content analysis, search, mining, recommendation, and large-scale computing.

Dr. Wang was the recipient of the Best Paper Awards successively from the 17th and 18th ACM International Conference on Multimedia, the Best Paper Award from the 16th International Multimedia Modeling Conference, Best Paper Award from the 4th International Conference on Internet Multimedia Computing and Service, and Best Demo Award from the 20th ACM International Conference on Multimedia.

Shengyong Chen (Senior Member, IEEE) received the Ph.D. degree in computer vision from the City University of Hong Kong, Hong Kong, in 2003.

From 2006 to 2007, he was with the University of Hamburg, Hamburg, Germany. He is currently a Professor with the Tianjin University of Technology, Tianjin, China, and also with the Zhejiang University of Technology, Hangzhou, China. He has authored more than 100 scientific papers in international journals. His research interests include computer vision, robotics, and image analysis.

Dr. Chen was the recipient of the Fellowship from the Alexander von Humboldt Foundation of Germany and National Outstanding Youth Foundation Award of China in 2013. He is a Fellow of the IET and a Senior Member of the CCF.