PAPER

TIG: A Multitask Temporal Interval Guided Framework for Key Frame Detection

Shijie WANG[†*], **Xuejiao HU**[†*], **Sheng LIU**[†], **Ming LI**[†], **Yang LI**[†a)], ***Nonmembers***, ***and*** **Sidan DU**[†b)], ***Member***

SUMMARY Detecting key frames in videos has garnered substantial attention in recent years, it is a point-level task and has deep research value and application prospect in daily life. For instances, video surveillance system, video cover generation and highlight moment flashback all demands the technique of key frame detection. However, the task is beset by challenges such as the sparsity of key frame instances, imbalances between target frames and background frames, and the absence of post-processing method. In response to these problems, we introduce a novel and effective Temporal Interval Guided (TIG) framework to precisely localize specific frames. The framework is incorporated with a proposed Point-Level-Soft non-maximum suppression (PLS-NMS) post-processing algorithm which is suitable for point-level task, facilitated by the well-designed confidence score decay function. Furthermore, we propose a TIG-loss, exhibiting sensitivity to temporal interval from target frame, to optimize the two-stage framework. The proposed method can be broadly applied to key frame detection in video understanding, including action start detection and static video summarization. Extensive experimentation validates the efficacy of our approach on action start detection benchmark datasets: THUMOS'14 and Activitynet v1.3, and we have reached state-of-the-art performance. Competitive results are also demonstrated on SumMe and TVSum datasets for deep learning based static video summarization.

key words: *key frame detection, action start detection, action recognition, video summarization, video understanding*

1. Introduction

Nowadays, as the amount of video data is increasing on streaming media, detecting key frames becomes a challenging task attracting broad attention in multimedia application. For instances, abnormal events are more obvious when checking video surveillance system; viewers can quickly jump to the clips they are interested in when watching videos according to predicted key frames; highlight moments will be quickly generated after a sport match. Hence, the task holds practical application value, and there exist several downstream tasks that can be practically derived from it. However, as the key frames are sparse compared with the whole video frames, it is difficult to precisely understand the semantic feature for the model. In this work, we propose a novel key frame detection framework.

Key frame detection aims to precisely extract specific frames for further application in realms like video understanding and video analysis. To demonstrate the effectiveness of our framework, we focus on two downstream tasks: action start detection and static video summarization. As illustrated in Fig. 1 (a), action start detection is a point-level task towards detecting the start frames of an action instance in a video along with their category. Ljung et al. [1] have endeavored in this direction, albeit with limited success. Static video summarization extract several key frames representing important content of the raw video, as demonstrated in Fig. 1 (b). However, static video summarization is treated as clustering problem by many previous works [2], [3] and there is a lack of deep learning method. For this reason, we apply our framework on the static task and provide the competitive results.

Numerous challenges persist in the execution of this task. (1) Label data is very unbalanced. As for the task of action start detection, action start frames are quite few throughout the entire videos, there may be just less than ten frames of action start out of thousands of frames. A number of previous works have made efforts to overcome such difficulties. Shou et al. [4] adopt Generative Adversarial Network (GAN) to generate hard negative samples around the start point, but it may let the model make ambiguous judgments during inference stage. Gao et al. [5] use reinforcement learning techniques to predict the start probability at each time, but the long-term reward produces very little effect. (2) Background frames are far more than target frames, so that it is difficult for model to learn the feature of target frames. For feature learning, most previous works have considered recurrent neural network (RNN) and variants, Gate Recurrent Unit (GRU) [6] and Long Short Term Memory (LSTM) [7] that pay more attention to temporal feature. With Transformer [8] emerging, many works [9], [10] utilize it as backbone. However, Joungbin et al. [11] demonstrates that RNN is more adequate for action related task than Transformer. Inspired by them, we utilize LSTM with convolution projection layers as backbone. (3) There is a lack of post-processing methods to refine predicted candidates for point-level task. Works of video summarization utilize knapsack algorithm to deal with different time span of shots, but it fail to meet the requirement of selecting frames. We propose a point-level non-maximum suppression algorithm based on Soft-NMS [12]. The confidence score decay function is related to temporal interval instead of Intersection of Union (IoU).

To address aforementioned challenges, we propose an effective and universal Temporal Interval Guided (TIG)

Manuscript received February 5, 2024.
Manuscript revised April 11, 2024.
Manuscript publicized May 17, 2024.

[†]School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China.
[*]Shijie Wang and Xuejiao Hu contributed equally to this work.
a) E-mail: yogo@nju.edu.cn
b) E-mail: coff128@nju.edu.cn

DOI: 10.1587/transinf.2024EDP7031

Copyright © 2024 The Institute of Electronics, Information and Communication Engineers

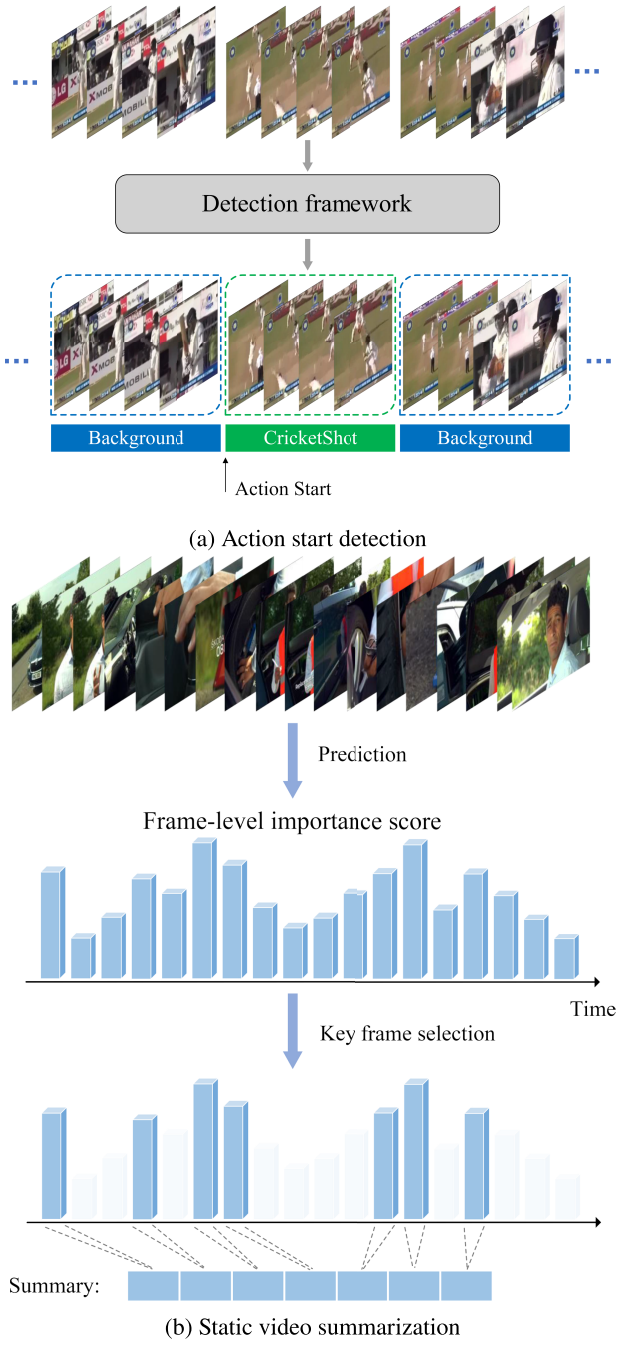


Fig. 1 Illustration of downstream tasks of key frame detection. (a) Action start detection endeavors to identify start point of action instances and output the action category. (b) Static video summarization extract several key frames representing important content of the raw video.

framework in this work, it contains two sub-networks: Classification network and Temporal regression network. The two-stage framework enhances the ability of simultaneously performing classification and localization task. Both sub-networks utilize LSTM with convolution projection layers as backbone for spatial and temporal feature learning. The convolution projection layers facilitate the aggregation of local feature, thereby augmenting the capability of spatial feature and semantic information learning. In addition, we propose a TIG-loss for Temporal regression network for concentrating on predicting temporal interval, which can let the network better cope with challenge of temporal localizing. At last, PLS-NMS is designed to solve problems of candidate overlapping and select more precise final results, it is utilized to further refine the candidates produced by results of two sub-networks.

In summary, we make following contributions:

(1) We introduce a universal and effective two-stage Temporal Interval Guided (TIG) framework for key frame detection. The incorporation of a spatio-temporal feature learning module coupled with local projection layers significantly enhances the capability of semantic information understanding. It is worth mentioning that a structured TIG-loss is proposed for optimization during temporal regression stage.

(2) We propose a novel post-processing strategy PLS-NMS, which is suitable for point-level detection task. The strategy can effectively solve the problems of candidate adjacent and overlapping, contributing to more precise detection of the key frames.

(3) Sufficient experiments provide competitive results on the two downstream tasks of action start detection and static video summarization, and demonstrate the universality of our approach for point-level key frame detection task. Specifically, our approach represents its superiority over previous state-of-the-art (SOTA) methods on THUMOS'14 and Activitynet v1.3 datasets for action start detection.

2. Related Work

Temporal Action Detection. TAD is oriented towards detecting the whole action instances along with their corresponding categories in untrimmed videos. The main framework is divided into two sheets: proposal-based methods [13]–[15] and proposal-free methods [16], [17]. Proposal-based methods like U-BlockConvCaps [18] builds a Capsules Boundary Network to avoid some limitations of the invariance caused by pooling and inability. Proposal-free methods like Shou et al. [19] propose CDC network with CDC filter to abstract action semantics, while boosting prediction of per-frame action and localization of temporal boundaries. Additionally, Li et al. [20] introduce a coarse-to-refine framework for weakly-supervised TAD which only needs action label for supervision.

Online Action Detection. OAD is first present by De Geest et al [21] and treated as per-frame labeling task. Given the absence of visibility into information beyond the current frame during inference, RNN and the variants have been prominently utilized for the task. Huang et al. [22] propose a RNN based network to depict the spatial-temporal semantic information of actions; Liu et al. [23] incorporates both RGB and skeleton information for a multi-modality recurrent neural network to get more precise detection results. The advent of attention mechanisms [8] has given rise to Transformer-based methodologies for OAD, such as [24],

[25]. E2E-LOAD [26] proposed an end-to-end Transformer-based model to address long-term understanding and efficient online reasoning problems.

Online Detection of Action Start. ODAS closely aligns with our task objectives, aiming to identify the action start frame and corresponding action category in an online manner. The task is first introduced by Shou et al. [4] and a GAN based model is proposed as baseline concerning the sparsity of action start label. In consideration of good performance of recurrent network, Wang et al. [27] both utilize recurrent modeling backbones with short-term looking back policy to focus on grasping local semantic context information. Gao et al. [5] propose a two-stage framework leveraging a policy gradient method to enhance attention for both classification and localization. DABR [28] establishes probability density function near action boundaries to reduce penalty for frames near ground-truth boundary points. In addition, there is also work like [1] have tackled the task in an offline manner. They train the RNN-based model with a structured loss function and propose a new Mouse Reach Dataset for dedicated research in this domain.

Video Summarization. Video summarization aims to generate a concise summary of original video for viewers to fast browse and better understand it. There are two main approaches for video summarization: dynamic and static. Dynamic video summarization focus on detecting video shots of high importance. Supervised methods like PGL-SUM [29] utilize attention mechanism by combining global and local multi-head attention to address the limitations of RNN models, MSVA [30] fuse extracted motion features and static visual feature to better learn representation of video feature. Unsupervised methods, including [31]–[33], generate summaries without annotated importance score. GL-RPE [33] utilizes global and local relative position embedding to capture both local and global interdependencies between video frames. Static video summarization lay emphasis on frame-level detection, and utilize conventional methods like clustering algorithm to solve the task. Yasmin et al. [2] utilize similarity-based agglomerative clustering algorithm to cluster the frames into different groups for further summarizing. Bhattacharjee et al. [3] propose the Artificial Bee Colony optimization algorithm to optimize shot length for better extracting key frames. In addition, there are also works like A2summ [34] focusing on multimodal summarization which generating summary with matched text and videos.

3. Method

In this section, we first present the problem definition of key frame detection, and subsequently elaborate the framework of our model. Notably, we introduce an effective Temporal Interval Guided (TIG) loss, which actively contributes to the model training during the temporal regression stage. Finally, we propose a post-processing algorithm, PLS-NMS, to further refine predicted candidates and produce the ultimate results.

3.1 Problem Definition

The input of our model is a sequence of T video frames noted as $\{x_1, x_2, \cdots, x_T\}$, where x_t represents the frame of time t, and the overall duration T varies among videos. In accordance with standard practice, we adopt a pre-trained feature extractor to obtain the feature vector F_t from each video clip at time t. Our objective is to predict whether the frame at time t belongs to key frame, specifically for action start detection, signifying whether the frame is an action and represents the beginning of the action. The output prediction is defined as $\{y_1, y_2, \cdots, y_T\}$, y^t can be expressed as $\{c_t, d_t\}$, with $c_t \subseteq \{1, \cdots, N\}$ representing action label obtained from the probability distribution across the N action categories, d_t represents the temporal interval between current time point and nearest ground-truth action start point. The unit of measurement for d_t is frame. For static video summarization, output y_t is made up of $\{s_t, d_t\}$, where s_t represents importance score. It is noteworthy that the output is in a downsampled dimension, the final summary is generated by selecting frames corresponding to the positions of predicted key frames after upsampling. Structured output prediction poses a challenge for the task.

3.2 Framework Demonstration

We propose a novel Temporal Interval Guided (TIG) framework as illustrated in Fig. 2. The framework comprises contains two sub-networks: Classification network and Temporal regression network. Classification network focus on predicting frame-level classification. Raw video frames are organized into several clips, and feature extractors are employed to convert the video clips into 1D RGB and optical flow feature vectors respectively. Then the features are concatenated to form two-stream feature (TS feature). Classification network take TS feature as input and output probability distribution serving as confidence score throughout action labels. The final result is determined by selecting the label with the highest score. The aggregation of TS feature and intermediate feature produced by Classification network is fed into Temporal regression network to fulfill the task of frame-level regression. The network outputs the temporal interval from key frame point. Finally, the results from both two sub-networks are combined to construct final results.

Classification network. On account of notable performance of recurrent networks in the domain of video understanding, especially in action related fields, we utilize the variant LSTM to construct Classification network. At each time step t, it uses previous hidden state $h_{t-1,c}$, previous cell $c_{t-1,c}$ and current TS feature f_t as input to update hidden state $h_{t,c}$ and cell $c_{t,c}$. The output of LSTM is intermediate feature $f_{I,c}$. The formulation of is expressed as follows:

$$h_{t,c}, c_{t,c}, f_{I,c} = LSTM(h_{t-1,c}, c_{t-1,c}, f_t) \quad (1)$$

Drawing inspiration from [10], we utilize convolution

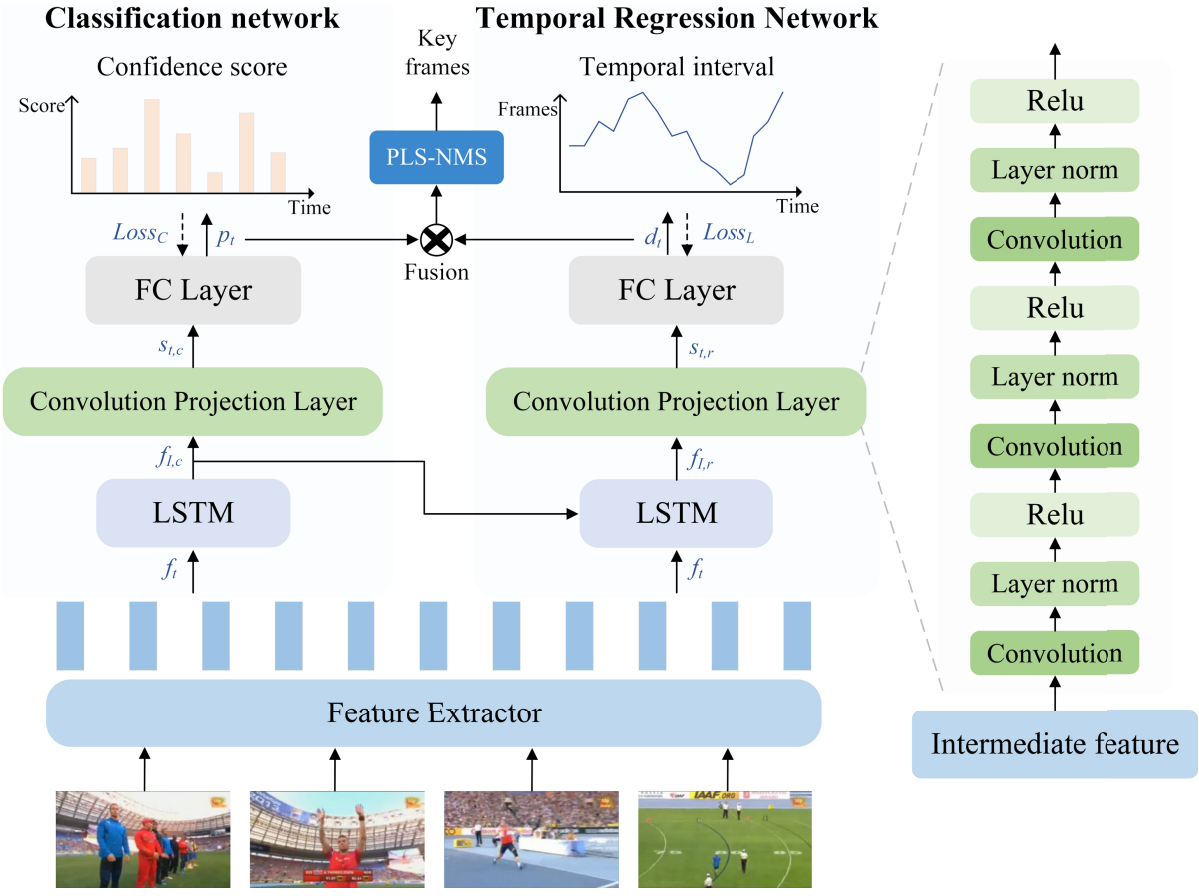


Fig. 2 Overview of TIS framework. The framework comprises two sub-networks: classification network and temporal regression network. Classification network focus on frame-level action classification, while temporal regression network predicts temporal interval for each frame. Both sub-networks utilize LSTM with convolution projection layer as backbone. PLS-NMS is applied to refine the preliminary results and produce final key frame points.

projection layer to strengthen local feature learning. Convolution projection layer is realized through three convolution blocks, each comprising a 1D convolution layer, layer normalization, and ReLU activation. Finally, a fully-connected layer with softmax activation is used as the classification head. The ultimate probability distribution of action label is derived through Eqs. (2) and (3):

$$s^i_{t,c} = ReLU(LN(Conv(s^{i-1}_{t,c}))) \tag{2}$$

$$p^k_t = softmax(W_c s_{t,c} + b_c) \tag{3}$$

Where $s^i_{t,c}$ denotes the output of $i-th$ convolution block, p^k_t represents predicted probability of action label k, W_c and b_c are parameters of fully-connected layer in Classification network. Notably, output of Classification network for static video summarization is importance score.

Temporal regression network. Temporal regression network has the similar structure with Classification network: a convolution layer comprised of three convolution blocks and a regression head of a fully-connected layer. The output of temporal interval can be obtained by following equations:

$$h_{t,r}, c_{t,r}, f_{I,r} = LSTM(h_{t-1,r}, c_{t-1,r}, (f_t + f_{I,c})) \tag{4}$$

$$s^i_{t,r} = ReLU(LN(Conv(s^{i-1}_{t,r}))) \tag{5}$$

$$d_t = W_r s_{t,r} + b_r \tag{6}$$

Where $h_{t,r}, c_{t,r}$ express hidden state and cell of LSTM, $s^i_{t,r}$ denotes the output of $i-th$ convolution block, d_t is the temporal interval from nearest action start point at time t, W_r and b_r are parameters of fully-connected layer in Temporal regression network.

3.3 Loss Function

Classification loss function. As for action start detection, to learn the class label for each frame, we choose cross entropy as loss function for optimization. This loss function quantifies the disparity between the predicted probability distribution and the ground-truth one-hot label. Mathematically, it can be expressed as:

$$Loss_{C,asd} = -log\frac{e^{p^k_{t,gt}}}{\sum_{k=1}^{K} e^{p^k_{t,pre}}} \tag{7}$$

Where $p^k_{t,gt}$ is ground-truth probability, $p^k_{t,pre}$ is predicted probability, K is the number of classes.

For static video summarization, we utilize Mean Square Error (MSE) loss function for training which is described as:

$$Loss_{C,svs} = \frac{1}{T}\sum_{t=1}^{T}(s_{t,gt} - s_{t,pre})^2 \tag{8}$$

Where $s_{t,gt}$ denotes ground-truth importance score, $s_{t,pre}$ represents predicted importance score, and T is total length of the video.

Temporal regression loss function. In the context of frame-level temporal interval regression, the absence of a suitable existing loss function prompts us to introduce an effective Temporal Interval Guided (TIG) loss for optimization during the temporal regression stage. This loss function is designed to emphasize the temporal interval from the nearest key frame point for each frame, ensuring that the calculated value remains within a range conducive to effective network learning. The function is represented as follow:

$$Loss_R = |1 - \frac{d_{t,pre} + 1}{d_{t,gt} + 1}| \tag{9}$$

Where $d_{t,pre}$ denotes predicted temporal interval, and it is a non-negative floating number; $d_{t,gt}$ represents the ground-truth temporal interval, and its actual value is non-negative integer.

Finally, the total loss function for network joint training. Both values of $Loss_C$ and $Loss_R$ varies from 0 to 1 with different magnitude. The joint loss is expressed as:

$$Loss = Loss_C + Loss_R \tag{10}$$

3.4 Post Processing

During inference stage, the predicted action label and temporal interval are combined to determine final key frame point. Specifically, we consider points with predicted temporal interval $d_{t,pre} \leq d_{thd}$ as candidates, where d_{thd} is a hyper-parameter. Then we propose PLS-NMS to further process the candidates. PLS-NMS is more suitable for point-level task as its confidence score decay function is related to temporal interval instead of IoU, meanwhile the problem of candidates overlapping in short time intervals is better solved.

Similar to Soft-NMS, we first sort all candidate points by confidence score s_n ($n \in N$, N denotes the number of candidates). Point with highest score input into final result box R, while the remaining points are placed into candidate box C. Then we calculate the decayed confidence score s_j ($j \in J$, J represents the number of remaining candidates) for points in box C. This is achieved by employing a decay function, which takes into account the temporal interval and the point most recently placed into box R. Points with scores less than the threshold s_{thd} are then eliminated. This process is repeated until there are no points left in box C, signifying that the points in box R constitute the final key frame points.

Algorithm 1 PLS-NMS

Input: $C = \{c_1, c_2, ...c_n\}$, c_i is cadidate point,
$S = \{s_1, s_2, ...s_n\}$, s_i is confidence score,
$D = \{d_1, d_2, ...d_n\}$, d_i is predicted temporal distance.
$R \leftarrow \{\}$
while $C \neq empty$ **do**
  $m \leftarrow argmax\ S$
  $R \leftarrow R \cup m, C \leftarrow C - m$
  for c_i in C **do**
    $s_i \leftarrow f(s_i, d_i, d_m)$
    if $s_i \leq s_{thd}$ **then**
      $C \leftarrow C - c_i$
    end if
  end for
end while
return R, S

Fig. 3 The pseudo-code of proposed PLS-NMS post-processing algorithm, where f is confidence score decay function and output R contains final results, with corresponded confidence score in S.

The confidence score decay function is presented as follows:

$$\hat{s_j} = \frac{s_j}{1 + e^{|d_j - d_m|}} \tag{11}$$

Where s_j denotes the raw confidence score, d_j expresses predicted temporal interval of candidate j, d_m is predicted temporal interval of latest point put into box R. The whole process algorithm is formally described in Fig. 3.

4. Experiments

To demonstrate the universality, effectiveness of proposed model, we conduct sufficient experiments on THUMOS'14 [35], Activitynet v1.3 [36], TVSum [37], and SumMe [38] datasets.

4.1 Datasets

THUMOS'14 [35] is a popular dataset for action detection which contains 20 types of action. The total duration of the dataset is more than 20 hours. Following [4], [5], [27], [28], we train our model with validation dataset of 200 untrimmed videos and test them with test dataset of 213 untrimmed videos.

ActivityNet v1.3 [36] is one of the largest datasets for action detection which contains approximately 15K untrimmed videos in total and 200 action classes are annotated. Following [4], [5], [27], [28], we train our models on the train set and test them on the validation set.

SumMe [38] is one of the benchmark datasets for video summarization. It contains 25 videos covering both first-person and third-person view. Every video is annotated by 18 users for subjective summary. Moreover, frame-level ground-truth importance score that adopted by averaging user summaries per frame is provided for training.

TVSum [37] is another benchmark dataset for video

Table 1 Comparisons with state-of-the-art methods using p-mAP(%) at depth *Rec*=1.0 under different offset thresholds on THUMOS'14.

Offsets (second)	1	2	3	4	5	6	7	8	9	10	average
Shou et al. [4]	3.1	4.3	4.7	5.4	5.8	6.1	6.5	7.2	7.6	8.2	5.9
Wang et al. [27]	10.0	13.5	15.0	16.2	16.8	17.8	18.2	18.4	18.9	19.0	16.4
Startnet [5]	19.5	27.2	30.8	33.9	36.5	37.5	38.3	38.8	39.5	39.8	34.2
DABR [28]	**26.7**	**38.4**	43.3	45.9	47.8	49.5	50.4	51.1	51.7	52.1	45.7
Iljung et al. [1]	9.0	14.0	16.0	17.0	18.0	19.0	19.0	20.0	20.0	20.0	17.2
TIG (Ours)	17.2	33.6	**45.0**	**51.8**	**56.4**	**59.4**	**61.4**	**62.5**	**63.3**	**63.8**	**51.4**

summarization. It is composed of 50 videos and annotated by 20 users for user summary. Frame-level ground-truth importance score is also provided for training.

4.2 Evaluation Protocols

For action start detection, we assess the performance of our model with point-level mAP (p-mAP) introduced by [4]. Confidence scores for each action class are sorted firstly and then calculate by order. A prediction is classified as positive when its action class is correct and temporal interval from a ground-truth point is smaller than a specified offset threshold. It is essential to emphasize that calculation for the same ground-truth point is not allowed. Under each action class point-level average precision (p-AP) is calculated firstly, and p-mAP is subsequently obtained by averaging p-AP throughout action classes. Furthermore, we adopt another metric AP depth at recall X% proposed by [4]. The p-AP on the Precision-Recall curve with the recall rate is averaged from 0% to X%. The p-mAPs under different offset thresholds are then averaged to obtain the final average p-mAP at each depth. Temporal offset thresholds are varied from 1s to 10s.

For static video summarization, as there is no standard evaluation protocols, we use the same metric as employed in dynamic video summarization, as adopted in previous studies [29], [30], [34]. We compare similarity between the generated summary and user annotated summary by F1-Score which is obtained by precision and recall. The computed F1-Scores for all users are then averaged to obtain the final F1-Score. Additionally, we consider Spearman's ρ and Kendall's τ correlation coefficients proposed by [39] to cover the shortage of F1-Score. We conduct our experiments on five random divided splits of training and testing datasets provided by previous work [40]. The reported results are obtained by averaging results across the five splits. Notably, MSVA [30] provide five new non-overlap splits to rectify issues related to video dropped or duplicated in previous splits. We also report our results on the non-overlap splits.

4.3 Implementation Details

Feature description. For action start detection, following [5], [41], [42], we downsample the videos into 24 fps and set every 6 frames as a chunk for feature extraction. We utilize TSN [43] model to extract both RGB and flow features. For THUMOS'14 dataset, we adopt the same metric as in [42] where RGB feature are extracted with model of Resnet-50 [44] as backbone and flow feature are extracted with BN-Inception [45]. For Activitynet v1.3, both features are extracted by Resnet-50 pretrained on Kinetics-400. TS feature are obtained by concatenating RGB and flow feature.For static video summarization, we follow previous work [29] with the same setting for both SumMe and TVSum datasets. Videos are downsampled to 2 fps, and features are pre-extracted by GoogleNet [46].

Parameter settings. Hidden size of LSTM for Classification network and Temporal regression network are set to 4096 and 1024 respectively. Convolution projection layer contains three 1D convolution layers with kernel size=3 and padding size=1. We utilize stochastic gradient descent (SGD) optimizer with learning rate=5e-3 and momentum=0.9 to train our model. Threshold of temporal interval d_{thd} of PLS-NMS is set to 11 on THUMOS'14 and 8 on Activitynet v1.3 respectively, while 3 for static video summarization. Confidence score threshold s_{thd} is set to 0.2 for action start detection and 0.5 for static video summarization.

4.4 Comparison with SOTA Methods

We compare our results with SOTA methods on action start detection, and reproduce several deep learning methods of dynamic video summarization for comparision of static video summarization. We use evaluation metrics p-mAP and average p-mAP proposed by [4] on the two popular datasets THUMOS'14 and Activitynet for action start detection. In the context of static video summarization, we adopt F1-Score, Spearman's ρ and Kendall's τ correlation coefficients on two benchmark datasets TVSum and SumMe.

Results on THUMOS'14. As presented in Table 1, we compare our results with both online methods (*Shou et al.* [4], Startnet [5], *Wang et al.* [27], DABR [28]) and offline method (Iljung et al. [1]). The focus of the action start detection task is on evaluating the average performance across various temporal offset thresholds. Notably, our method significantly outperforms other SOTA methods under most offset thresholds. What is worth mentioning that we outperform DABR [28] by 5.7% p-mAP on average across offsets. Remarkably, our results are more than twice as effective as Iljung et al. [1] as both offline methods. Table 2 illustrates the comparison on different recall depths. It can be seen that we have reached SOTA performance under most depths of

Table 2 Comparisons with state-of-the-art methods using average p-mAP(%) at different depths on THUMOS'14.

Depth Rec.	@0.1	@0.2	@0.3	@0.4	@0.5	@0.6	@0.7	@0.8	@0.9	@1.0
Shou et al. [4]	42.7	27.3	19.8	14.9	11.8	10.0	8.5	7.4	6.6	5.9
Wang et al. [27]	49.0	44.3	40.8	36.8	31.7	27.5	23.7	20.7	18.4	16.4
Startnet [5]	77.4	70.2	64.5	59.1	54.2	49.3	45.1	41.2	37.6	34.2
DABR [28]	**90.6**	**83.9**	**78.9**	74.6	69.5	64.7	60.0	55.4	50.5	45.7
Iljung et al. [1]	64.0	57.0	55.0	54.0	54.0	53.0	53.0	53.0	53.0	**53.0**
TIG (Ours)	83.6	80.1	77.3	**74.7**	**71.8**	**68.6**	**65.2**	**61.1**	**56.6**	51.4

Table 3 Comparisons with SOTA methods using p-mAP(%) at depth *Rec*=1.0 under different offset thresholds on Activitynet v1.3.

Offsets (second)	1	2	3	4	5	6	7	8	9	10	average
SceneDetect [47]	-	-	-	-	-	-	-	-	-	4.7	-
ShotDetect [48]	-	-	-	-	-	-	-	-	-	6.1	-
Shou et al. [4]	-	-	-	-	-	-	-	-	-	8.3	-
Startnet [5]	8.1	10.2	11.8	13.3	14.4	15.3	16.1	16.7	17.4	18.0	14.1
DABR [28]	**8.5**	**12.4**	15.0	17.0	18.8	20.0	21.1	21.9	22.7	23.5	18.1
TIG (Ours)	6.1	12.2	**16.0**	**18.7**	**20.5**	**22.0**	**23.3**	**24.2**	**25.2**	**25.9**	**19.4**

Table 4 Comparisons with baseline methods using F_1 score, Kendall τ and Spearman ρ correlation coefficients metrics on SumMe and TVSum. Noted that F_1 reports experiment results conducted on previous random splits and F_1^* reports experiment results conducted on new non-overlap splits provided by MSVA [30].

	SumMe				TVSum			
	F_1	F_1^*	Kendallτ	Spearmanρ	F_1	F_1^*	Kendallτ	Spearmanρ
A2Summ [34]	21.4	-	0.095	0.104	18.1	-	0.091	0.103
PGL-SUM [29]	22.5	-	0.123	0.134	18.2	-	0.113	0.127
TIG (ours)	**23.5**	-	**0.153**	**0.168**	**20.3**	-	**0.142**	**0.159**
MSVA [30]	-	23.5	0.120	0.131	-	13.3	0.020	0.022
TIG (ours)	-	**23.7**	**0.138**	**0.150**	-	**17.3**	**0.064**	**0.071**

Table 5 Ablation study evaluated using p-mAP(%) at depth *Rec*=1.0 under different offset thresholds on THUMOS'14. The first line presents results of model training without convolution projection layer, the second line presents results of model training with L1-loss, and results of candidates post-process with NMS are shown in the third line.

conv	TIS-loss	PLS-NMS	Offsets(second)										
			1	2	3	4	5	6	7	8	9	10	average
	✓	✓	14.6	32.2	43.2	48.8	52.5	54.9	56.4	57.8	58.4	58.9	47.8
✓		✓	14.4	26.9	35.4	39.4	42.4	43.7	45.3	46.2	46.9	47.4	38.8
✓	✓		15.6	32.7	41.9	47.3	51.0	52.9	54.4	55.2	55.9	56.2	46.3
✓	✓	✓	**17.2**	**33.6**	**45.0**	**51.8**	**56.4**	**59.4**	**61.4**	**62.5**	**63.3**	**63.8**	**51.4**

recall, and outperform other methods by a substantial margin. This comprehensive evaluation affirms the effectiveness of our model across various evaluation scenarios.

Results on Activitynet v1.3. The comparative analysis of results of results on Activitynet v1.3 is presented in Table 3. Given the vast scale of this dataset, the challenge of learning video features poses a significant hurdle, leading to generally unsatisfactory results across all methods. Despite these challenges, our method demonstrates remarkable performance, achieving competitive results and surpassing previous methods across a majority of temporal offsets.

Results on SumMe and TVSum. For fair comparison, we have reproduced several recent static video summarization methods with shared source code and evaluate under metrics of static video summarization as baselines. For other methods, predicted frame-level importance scores are first sorted, then summaries are generated by selecting top 15% frames. In contrast, our method generates summaries using the output of PLS-NMS, ensuring that the length does not exceed 15% of the original videos. We compare with

A2Summ [34] and PGL-SUM [29] on previous splits, and compare with MSVA [30] on new splits they provide. As depicted in Table 4, our method outperforms all other methods under all evaluation metrics on both datasets. This comprehensive evaluation underscores the effectiveness and superior performance of our proposed approach in the realm of static video summarization.

In general, our method consistently surpasses the performance of previous approaches, achieving SOTA performance on action start detection and also provide competitive results on static video summarization. This consistent superiority substantiates the universality and effectiveness of our method in the domain of key frame detection.

4.5 Ablation Study

To demonstrate effectiveness of each part of the proposed method, we conduct ablation study on THUMOS'14 dataset within realm of action start detection.

Effectiveness of convolution projection layer. We conducted a comparative analysis to assess the performance of our model with and without the convolution projection layer, results are presented in first and last line of Table 5. It demonstrates that the inclusion of the convolution projection layer enhances the learning capabilities of the network, particularly by emphasizing local features.

Effectiveness of TIG loss. We compare the proposed TIG-loss with L1-loss. Results of network training with L1-loss are shown in second line of Table 5. Network training with TIS loss outperforms the other significantly by around 14% p-mAP on average. TIS loss controls the loss value within the range conducive to effective network learning, rendering the network more sensitive to temporal interval.

Effectiveness of PLS-NMS. To demonstrate the superiority of PLS-NMS, we compare it with traditional NMS, which results are shown in third line of Table 5. The performance of PLS-NMS exceeds NMS by 5% p-mAP on average, presenting suitability of PLS-NMS for point-level tasks and its effectiveness in addressing challenge of candidates overlapping.

Ablation study of hyper-parameters. For choosing the optimal hyperparameters in PLS-NMS, we first fix the time threshold to a value within an appropriate range, then obtain the optimal score threshold through ablation study. After that, the score threshold is fixed and time threshold is adjusted. Results of ablation study of hyper-parameters as depicted in Fig. 4. Figure 4 (a) presents results when time threshold is fixed to 11 and score threshold varies from 0.1 to 0.9. Figure 4 (b) presents results when score threshold is fixed to 0.2 and time threshold varies from 5 to 15. This experimentation provides insights into the selection of hyper-parameters in PLS-NMS. Remarkably, there is only a minor discrepancy in results when the score threshold varies from 0.1 to 0.4 and the time threshold varies from 9 to 15. This study demonstrates the stability and insensitivity of our method to changes in hyper-parameters when they fall within an appropriate range.

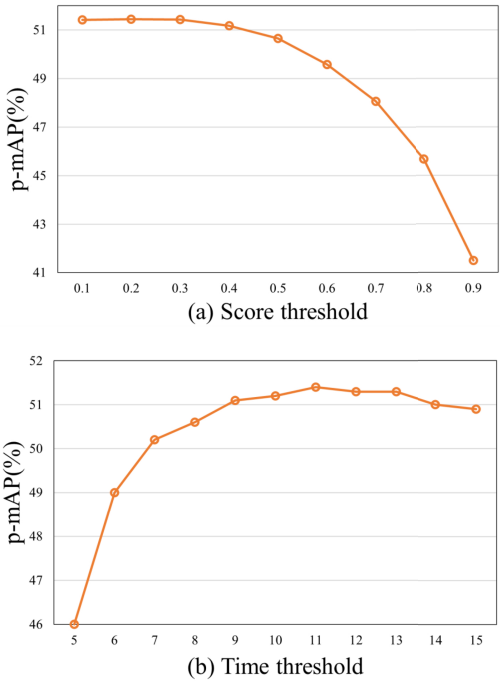


Fig. 4 Ablation study of hyper-parameters in PLS-NMS. Averaged p-mAP across 1–10s offsets is reported for every threshold. (a) Time threshold is fixed to 11 and score threshold varies from 0.1 to 0.9; (b) score threshold is fixed to 0.2 and time threshold varies from 5 to 15.

5. Visualization Results

In this section, we present visualizations of example results generated by our model, as illustrated in Fig. 5. We choose examples of BaseballPitch and FrisbeeCatch from THUMOS'14 dataset to demonstrate results of action start detection. The example of BaseballPitch is relatively easier for model to predict action start point as the athlete making up a large part of the video frame and the color of athlete's clothes contrasting sharply with the background. However, the example of FrisbeeCatch is not that easy. There are a lot of people in the video, so it is challenging for the model to detect when the action actually starts. In both examples, our model exhibits exceptional performance, accurately predicting action categories and ensuring that the temporal distances between the predicted action start frames (marked in red) and the ground-truth frames (marked in blue) are less than 1s.

Furthermore, we choose 'video_20' and 'video_42' from TVSum dataset to demonstrate performance of our model in static video summarization. The gray bars represent annotated importance scores, while blue bars are predicted importance scores of our selected key frames. Notably, the majority of our predictions align accurately with the ground-truth, and the margin of error is within a few frames. Other than action start, key frame is a more ambiguous concept for both human and model. Therefore, though there are some instances where we fall short of predicting high ground-truth values, the visualization results still strongly demonstrate the effectiveness of our model.

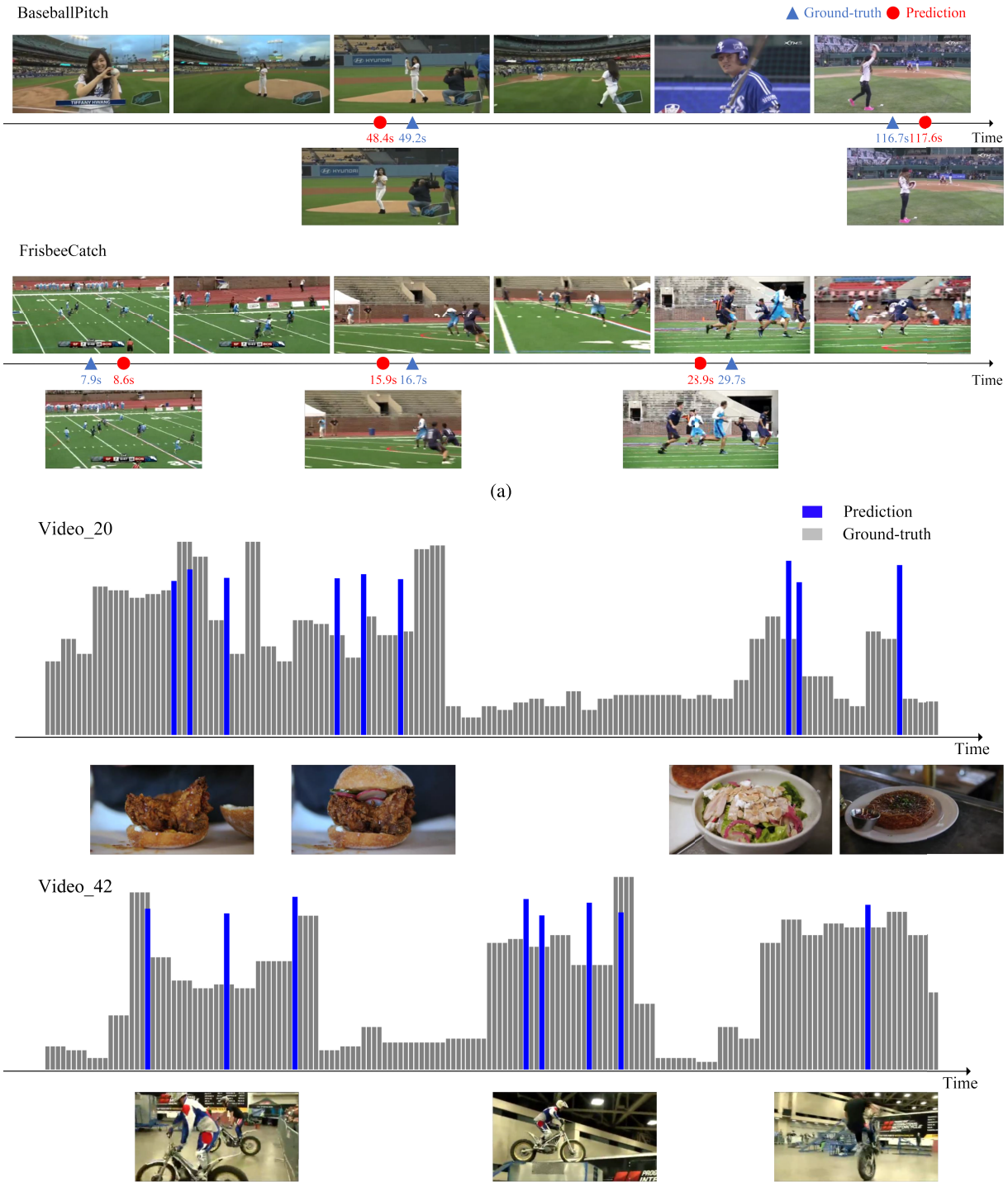


Fig. 5 Visualization results of examples from THUMOS'14 and TVSum dataset. (a) Results of action start detection. Ground-truth action start frames are labels with blue and our predictions are labeled with red; (b) gray bars in the background are ground-truth importance score and predicted importance score of key frames are labeled with blue.

6. Conclusion

In this work, we introduce a novel and effective Temporal Interval Guided (TIG) framework designed for key frame detection. Our two-stage framework is complemented by spatio-temporal feature learning module with convolution projection layer, a structured TIG-loss and a post-processing strategy PLS-NMS, demonstrating a highly competitive performance in both action start detection and static video sum-

marization. Sufficient experiments have proved the universality and effectiveness of our approach, it significantly outperforms previous works and have reached SOTA performance on datasets of action start detection. Moreover, our approach has the potential of extending to other temporal point-level detection tasks, such as anomaly detection and temporal signal detection. In future work, we will further investigate multimodal key point detection like combining text and video context.

References

[1] I.S. Kwak, J.-Z. Guo, A. Hantman, K. Branson, and D. Kriegman, "Detecting the starting frame of actions in video," 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pp.478–486, 2020.

[2] G. Yasmin, S. Chowdhury, J. Nayak, P. Das, and A.K. Das, "Key moment extraction for designing an agglomerative clustering algorithm-based video summarization framework," Neural Computing and Applications, vol.35, no.7, pp.4881–4902, 2023.

[3] T. Bhattacharjee, S. Saha, A. Konar, and A.K. Nagar, "Static video summarization using artificial bee colony optimization," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp.777–784, 2018.

[4] Z. Shou, J. Pan, J. Chan, K. Miyazawa, H. Mansour, A. Vetro, X. Giro-i-Nieto, and S.-F. Chang, "Online detection of action start in untrimmed, streaming videos," Proc. European Conference on Computer Vision (ECCV), vol.11207, pp.534–551, Sept. 2018.

[5] M. Gao, M. Xu, L. Davis, R. Socher, and C. Xiong, "Startnet: Online detection of action start in untrimmed videos," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp.5541–5550, 2019.

[6] H. Eun, J. Moon, J. Park, C. Jung, and C. Kim, "Learning to discriminate information for online action detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.806–815, 2020.

[7] R. De Geest and T. Tuytelaars, "Modeling temporal structure with LSTM for online action detection," 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp.1549–1557, 2018.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in Neural Information Processing Systems, vol.30, 2017.

[9] F. Cheng and G. Bertasius, "Tallformer: Temporal action localization with a long-memory transformer," European Conference on Computer Vision (ECCV), vol.13694, pp.503–521, 2022.

[10] C.-L. Zhang, J. Wu, and Y. Li, "Actionformer: Localizing moments of actions with transformers," European Conference on Computer Vision (ECCV), vol.13664, pp.492–510, 2022.

[11] J. An, H. Kang, S.H. Han, M.-H. Yang, and S.J. Kim, "Miniroad: Minimal RNN framework for online action detection," 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp.10307–10316, 2023.

[12] N. Bodla, B. Singh, R. Chellappa, and L.S. Davis, "Soft-nms — improving object detection with one line of code," 2017 IEEE International Conference on Computer Vision (ICCV), pp.5562–5570, 2017.

[13] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," European Conference on Computer Vision (ECCV), Cham, vol.11208, pp.3–21, 2018.

[14] J. Tan, J. Tang, L. Wang, and G. Wu, "Relaxed transformer decoders for direct action proposal generation," Proc. IEEE/CVF International Conference on Computer Vision (ICCV), pp.13526–13535, Oct. 2021.

[15] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional module for temporal action localization in videos," IEEE Trans. Pattern Anal. Mach. Intell., vol.44, no.10, pp.6209–6223, 2022.

[16] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Learning salient boundary feature for anchor-free temporal action localization," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.3319–3328, 2021.

[17] X. Liu, S. Bai, and X. Bai, "An empirical study of end-to-end temporal action detection," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.19978–19987, 2022.

[18] Y. Chen, B. Guo, Y. Shen, W. Wang, W. Lu, and X. Suo, "Capsule boundary network with 3d convolutional dynamic routing for temporal action detection," IEEE Trans. Circuits Syst. Video Technol., vol.32, no.5, pp.2962–2975, 2022.

[19] 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," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1417–1426, 2017.

[20] B. Li, R. Liu, T. Chen, and Y. Zhu, "Weakly supervised temporal action detection with temporal dependency learning," IEEE Trans. Circuits Syst. Video Technol., vol.32, no.7, pp.4473–4485, 2022.

[21] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, "Online action detection," European Conference on Computer Vision (ECCV), Cham, vol.9909, pp.269–284, 2016.

[22] J. Huang, N. Li, T. Li, S. Liu, and G. Li, "Spatial–temporal context-aware online action detection and prediction," IEEE Trans. Circuits Syst. Video Technol., vol.30, no.8, pp.2650–2662, 2020.

[23] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng, "Multi-modality multi-task recurrent neural network for online action detection," IEEE Trans. Circuits Syst. Video Technol., vol.29, no.9, pp.2667–2682, 2019.

[24] X. Wang, S. Zhang, Z. Qing, Y. Shao, Z. Zuo, C. Gao, and N. Sang, "Oadtr: Online action detection with transformers," IEEE/CVF International Conference on Computer Vision (CVPR), pp.7565–7575, 2021.

[25] M. Xu, Y. Xiong, H. Chen, X. Li, W. Xia, Z. Tu, and S. Soatto, "Long short-term transformer for online action detection," Advances in Neural Information Processing Systems, pp.1086–1099, 2021.

[26] S. Cao, W. Luo, B. Wang, W. Zhang, and L. Ma, "E2e-load: End-to-end long-form online action detection," 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp.10388–10398, 2023.

[27] T. Wang, Y. Chen, H. Lv, J. Teng, H. Snoussi, and F. Tao, "Online detection of action start via soft computing for smart city," IEEE Trans. Ind. Informat., vol.17, no.1, pp.524–533, 2021.

[28] X. Hu, S. Wang, M. Li, Y. Li, and S. Du, "Distribution-aware activity boundary representation for online detection of action start in untrimmed videos," IEEE Signal Process. Lett., vol.31, pp.765–769, 2024.

[29] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, "Combining global and local attention with positional encoding for video summarization," 2021 IEEE International Symposium on Multimedia (ISM), pp.226–234, 2021.

[30] J.A. Ghauri, S. Hakimov, and R. Ewerth, "Supervised video summarization via multiple feature sets with parallel attention," 2021 IEEE International Conference on Multimedia and Expo (ICME), pp.1–6s, 2021.

[31] Y. Jung, D. Cho, D. Kim, S. Woo, and I.S. Kweon, "Discriminative feature learning for unsupervised video summarization," Proc. AAAI Conference on Artificial Intelligence, vol.33, no.1, pp.8537–8544, 2019.

[32] Y. Yuan and J. Zhang, "Unsupervised video summarization via deep reinforcement learning with shot-level semantics," IEEE Trans. Circuits Syst. Video Technol., vol.33, no.1, pp.445–456, 2023.

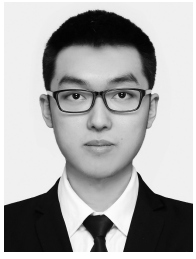
[33] Y. Jung, D. Cho, S. Woo, and I.S. Kweon, "Global-and-local relative position embedding for unsupervised video summarization," Euro-

pean Conference on Computer Vision (ECCV), vol.12370, pp.167–183, 2020.
[34] B. He, J. Wang, J. Qiu, T. Bui, A. Shrivastava, and Z. Wang, "Align and attend: Multimodal summarization with dual contrastive losses," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.14867–14878, 2023.
[35] 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.
[36] F.C. Heilbron, V. Escorcia, B. Ghanem, and J.C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.961–970, 2015.
[37] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5179–5187, 2015.
[38] M. Gygli, H. Grabner, H. Riemenschneider, and L.V. Gool, "Creating summaries from user videos," European Conference on Computer Vision (ECCV), vol.8695, pp.505–520, 2014.
[39] M. Otani, Y. Nakashima, E. Rahtu, and J. Heikkilä, "Rethinking the evaluation of video summaries," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.7588–7596, 2019.
[40] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," European Conference on Computer Vision (ECCV), Cham, vol.9911, pp.766–782, 2016.
[41] M. Xu, M. Gao, Y.-T. Chen, L. Davis, and D. Crandall, "Temporal recurrent networks for online action detection," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp.5531–5540, 2019.
[42] Y. Zhao and P. Krähenbühl, "Real-time online video detection with temporal smoothing transformers," European Conference on Computer Vision (ECCV), vol.13694, pp.485–502, 2022.
[43] 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," European Conference on Computer Vision (ECCV), Cham, vol.9912, pp.20–36, 2016.
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, 2016.
[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International Conference on Machine Learning, pp.448–456, 2015.
[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1–9, 2015.
[47] https://github.com/Breakthrough/PySceneDetect.
[48] https://github.com/johmathe/Shotdetect.

Shijie Wang received the B.S. degree in School of electronic science and engineering from Nanjing University, Nanjing, China, in 2022. He is currently working towards M.S. degree in electronic science and technology with the Nanjing University. His research interests include video understanding, computer vision and artificial intelligence.

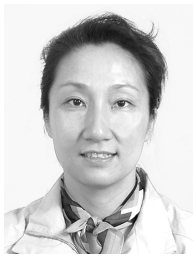
Xuejiao Hu received the B.S. degree in Computer Science and Technology from Binjiang college, Nanjing University of information science & Technology, Nanjing, China, in 2017, and the M.S. degree from School of Information Sciences and Technology, Nanjing Agriculture University, Nanjing, China, in 2019. She is currently pursuing the Ph.D. degree in electronic science and technology with the Nanjing University, Nanjing, China. Her research interests include computer vision, machine learning.

Sheng Liu received the B.S. degree in School of electronic science and engineering from Nanjing University, Nanjing, China, in 2021. He is currently working towards Ph.D. degree in electronic science and technology with the Nanjing University. His research interests include image processing, computer vision and artificial intelligence.



Ming Li received the B.S. degree in School of electronic science and engineering from Nanjing University, Nanjing, China, in 2017. He is currently working towards Ph.D. degree in electronic science and technology with the Nanjing University. His research interests include image processing, computer vision and artificial intelligence.

Yang Li received the B.S. and M.S. degrees in mechanical engineering from Southeast University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electronics engineering from Nanjing University, Nanjing, China, in 2006. He is an Associated Professor with School of Electronic Science and Engineering, Nanjing University, China, since 2009. His research interests include digital signal processing and computer vision.



Sidan Du received the B.S. and M.S. degrees in electronic engineering from Xidian University, Xian, China in 1984 and 1987, respectively, and the Ph.D. degree in physics from Nanjing University, Nanjing, China, in 1997. She is currently a Professor in the school of Electronic Science and Engineering, Nanjing University. Her research interests include computer vision and signal processing.