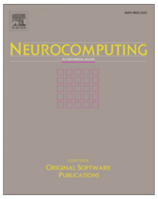
Neurocomputing 524 (2023) 106–116

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Temporal action detection with dynamic weights based on curriculum learning

Yunze Chen [a,b], He Jiang [a,b], Junrui Xiao [a,b], Ding Li [a,b], Qingyi Gu [a,*]

[a] *Institute of Automation, Chinese Academy of Sciences, East Zhongguancun Road, Haidian District, Beijing, China*
[b] *School of Artificial Intelligence, University of Chinese Academy of Sciences, Jingjia Road, Huairou District, Beijing, China*

ARTICLE INFO

Article history:
Received 27 September 2022
Revised 28 November 2022
Accepted 18 December 2022
Available online 27 December 2022
Communicated by Zidong Wang

Keywords:
Video understanding
Video classification
Action segmentation in videos

ABSTRACT

To enable temporal action localization, the computer needs to recognize the locations and classes of action instances in a video. The main challenge to temporal action detection is that the videos are often long and untrimmed, consisting of varying action content. Existing temporal action detection frameworks exhibit a gap between the training and testing phases, which is detrimental to model performance. Specifically, all positive samples are trained identically in the training phase. By contrast, in the testing phase, the positive samples with the best classification and localization scores are selected, while all others are suppressed. To mitigate this issue, we build an auxiliary branch to unify the training and testing procedures. In the construction of the auxiliary branch, we design a dynamic weighting strategy based on curriculum learning, where the weights of training samples are a combination of their classification and localization scores. Motivated by the speculation of curriculum learning, we emphasize the importance of classification and localization scores in different training stages. The classification score accounts for a higher proportion of the combined score in the early stages of the training process. As the epoch increases, the localization score gradually increases in proportion as well. The experimental results demonstrate that our methodology of curriculum-based learning enhances the performance of current action localization techniques. On THUMOS14, our technique outperforms the existing state-of-the-art technique (57.6% vs 55.5%). And the performance on ActivityNet v1.3 (mAP@Avg) reaches 35.4%.

1. Introduction

As an important direction of computer vision, the interpretation of human behavior from raw video data has a wide range of applications in video recommendation, security surveillance, human behavior analysis, and other fields [1]. Video-based action classification [2,3] has recently flourished owing to the development of deep learning and the accessibility of a variety of media resources. However, although the action classification process trims short videos by default (each video contains only one action clip), practical applications frequently feature long untrimmed videos that may include numerous active periods of arbitrary lengths. Therefore, in this study, we primarily concentrate on temporal action detection [4]. Temporal action detection can be applied in many areas such as the analysis of video content and video recommendations. For a untrimmed video, temporal action localization addresses two tasks: localization and recognition. Specifically, 1) locate when the action occurs, i.e., the start time and end time of the action. 2) Identify the category of the action (e.g., diving, playing billiards). In summary, the goal of temporal action detection is to locate the start and end times of each action instance in a long untrimmed video and predict the corresponding label. Since an untrimmed video may contain numerous active periods of arbitrary lengths, temporal action detection is a challenging task in video analysis.

Over the past several years, numerous studies have been conducted on temporal action localization. Conventionally, researchers first employ an action recognition network [2,3] to extract features from short videos. To determine an action's temporal boundaries, some algorithms [5–7] predict the probability of each frame is the start and end boundaries of the action. Consequently, the highest probability points are concatenated as the start and end times of the action. However, these methods apply a separate classification network for actions and and overlook the benefits from classification information. To integrate classification and localization models into a single end-to-end framework, some

[*] Corresponding author.
E-mail addresses: chenyunze2018@ia.ac.cn (Y. Chen), Jianghe2019@ia.ac.cn (H. Jiang), xiaojunrui2020@ia.ac.cn (J. Xiao), liding2019@ia.ac.cn (D. Li), qingyi.gu@ia.ac.cn (Q. Gu).

https://doi.org/10.1016/j.neucom.2022.12.049
0925-2312/© 2022 Elsevier B.V. All rights reserved.

approaches [8,9] develop one- or two-stage procedures with reference to object detection. One-stage techniques [10,8] split each video into an equal number of segments, and then predict the labels and boundary offsets of the anchors. In contrast, two-stage methods [4,9] first develop a set of action proposals, and then apply classification and boundary regression to each proposal individually. Rather than constructing action proposals, [11] performs classification and regression for each frame and achieves state-of-the-art (SOTA) performance.

Despite their mutual differences, all existing temporal action detection frameworks employ two phases: training and testing. Similar to other deep-learning-based methods [10], localization information is used to distinguish between positive and negative proposals (IoU between the anchor and ground truth) in the training process. In contrast, the testing process locates reliability proposals for final evaluation. That is, both classification information and localization information need to be emphasized. Because a solid prediction in evaluation corresponds to high classification and localization scores, a proposal with a higher degree of agreement between the classification and localization heads must be assigned a higher weight in the training phase. During the training phase, however, positive samples are selected solely using localization information, and are taken into account equally without regard to quality [6,7]. These issues create a gap between the training and testing processes. This gap reduces the quality of the selected positive samples, i.e., it increases the number of false-positive samples. Additionally, background frames in the video could also be marked as positive samples whenever they exhibit transitions. The phenomenon also raises the number of false positive samples.

As illustrated in Fig. 1, the prior methods select positive proposals based on the localization score (tIoU score higher than 0.5). All positive proposals are weighted equally without considering their quality. And in our approach, we give higher weights to the proposal with higher classification and localization scores. In addition, the background (camera turns to the audience) is also labeled as diving, because it occurs in the action of diving. To address the aforementioned issue, we develop an auxiliary branch. It combines the classification and localization branch to differentiate between the foreground and background (actionness score in Fig. 1). Specifically, when constructing the auxiliary branch, we combine the classification and localization scores with reference to the dynamic weight assignment [12,13] in object detection. The combined scores are then trained as the auxiliary branches' targets. Using the auxiliary branch, as well as the dynamic weight assignment technique, we aim to reduce the probability of false positives, and close the gap between training and testing.

However, existing weight assignment methods do not account for a reasonable combination of classification and detection scores. Instead, they use hyperparameters to aggregate the two scores, making the model sensitive to the hyperparameters' values [12]. To get rid of the dependency of sensitive hyperparameters, we apply a heuristic weight assignment strategy. Specifically, for the task of temporal action detection, researchers believe that the localization branch has a greater impact on model performance than the classification branch [14]. However, localization scores in the early training period are unreliable. According to this prior, we build a weight assignment paradigm with curriculum learning as the goal of the auxiliary branch. Our objective is to place more importance on reliable classification branches in the early training stages, and on localization scores that are crucial to the model in the latter training stages. We establish a parameter α that grows proportionately to the epoch. Early in the training process, categorization scores make up a greater proportion of the combined scores. As α increases in the latter rounds of training, the localization scores gradually increase in proportion as well. Based on the combined scores, we construct an auxiliary branch that seeks to bridge the gap between training and testing. We apply our approach to the anchor-free method AFSD [11], and test its efficacy on the widely-used datasets THUMOS14 [15] and ActivityNet v1.3 [16].

1. We design a new auxiliary branch to synchronize the network's training and testing procedures.
2. When building the auxiliary branch, a dynamic weight assignment paradigm based on curriculum learning is employed to guide the network's training.
3. We achieve state-of-the-art performance based on AFSD [11]. Performance improved by 57.6% on THUMOS14, and 35.4% on ActivityNet v1.3.

The remainder of this paper is organized as follows. Section 2 provides background review on action recognition, object detection, and temporal action detection. The proposed temporal action detection framework is described in Section 3. In Section 4, we experimentally validate the proposed methodology and compare it with earlier studies. Finally, Section 5 summarizes the paper and provides an overview of future directions.

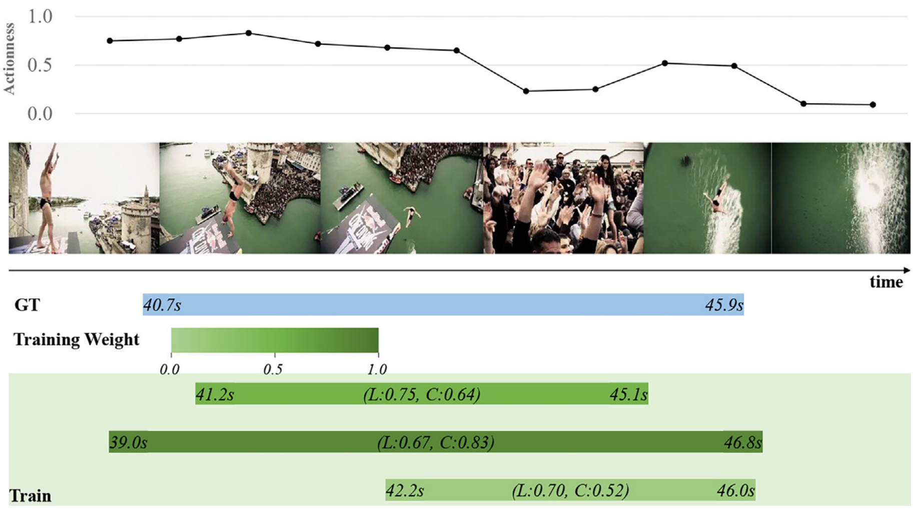


Fig. 1. An illustration of diving with background (camera turns to the audience). L and C represent localization and classification scores, respectively.

2. Related Works

2.1. Action Recognition

As an important branch of video content analysis, action recognition aims to classify trimmed videos into specific categories. Early action recognition techniques first extract manual features [17,18] (e.g., HOG, HOF, and MBH), and then use classifiers to categorize the activities. The most prominent early technique, iDT [17,18], comprises three components: dense sampling, trajectory tracking, and trajectory-based feature extraction. Based on the extracted features, iDT performs feature encoding and classification.

With the advent of deep learning, model performance has become increasingly linked to dataset size. As numerous large-scale action categorization datasets (e.g., UCF-101 [19], Sports-1 M [20], and Kinetics [21]) have been published, deep learning methods have achieved advanced performance. Specifically, [22] uses RGB frames and stacked optical flow vectors to learn the appearance features and motion information of objects, respectively. The C3D network [23] extracts spatial and temporal information from the original video using a series of 3D convolutional kernels. Equipped with two-stream networks and 3D convolution, I3D [24] constructs a two-stream model based on 3D convolution (Two-Stream Inflated 3D ConvNets), which provides pre-training strategies. To ensure effective and efficient learning over an entire video, TSN [25] combines a sparse temporal sampling method with video-level supervision for long-range temporal modeling. These action recognition networks are commonly utilized as feature extractors for event captioning, action segmentation, and temporal action localization.

2.2. Object Detection

The goal of object detection is to categorize and individually locate every target in an image. To accomplish this, the integration of training and testing processes is critical. Several studies have been conducted as attempts to reduce the gap between training and inference. To overcome the rigid label assignment strategy during training, MetaAnchor [26] assigns training anchors adaptively by anticipating the anchor distribution, GuidedAnchor [27] predicts the shape of anchor points by semantic feature maps, and FreeAnchor [28] selects the best anchor point based on the loss function to improve the accuracy of the match between the anchor point and the target. In addition, researchers [29,30] argue that since predicted proposals have different qualities in inference, samples should be treated differently in the training phase. Noisy Anchor [29] generates soft labels to reweight the training samples, whereas Generalized Focal Loss [30] assigns each anchor a soft weight by combining the categorization and localization scores.

The aforementioned approaches combine classification and localization scores using hyperparameters, which require manual modification. However, it is challenging for hyperparameter-based techniques to be effective in temporal action detection. In this task, the length of the videos varies and there are significant differences between action categories. To mitigate the effect of hyperparameters, we design an auxiliary branch that merges classification and localization features automatically.

2.3. Curriculum Learning

Curriculum learning is a training strategy which allows the model to learn the easy and the hard samples sequentially, and thus mimic the order in which humans learn. The concept of curriculum learning is introduced in the first place by [31]. In short, curriculum learning means learning from simpler data to more difficult. More specifically, this method trains the model first with a simpler subset, and then slowly increases the complexity of data, until the whole training data is trained. Curriculum learning strategies show a great power to improve generalization and convergence rates of different models in wide ranges of applications, including computer vision, and natural languages. For example, [32] reduces the training time by 70% and improves the performance by 2.2% compared with the base. The key challenge in learning curriculum learning is to determine the difficulty level of each case. In computer vision, [33] uses distance to classification boundary to indicate a sample's difficulty. [34] devises a new formulation, called Self-Paced Learning (SPL), in which the less lossy samples are regarded as easier and emphasized during the training process. [35] proposes an adaptive function to determine the difficulty level of the sampling. This method emphasizes easier samples during early training and hard samples during later training. In this paper, different from the approach above to define difficulty for each sample, we define difficulty for each branch. The localization branch is more difficult than the classification branch and have a larger impact on the performance of the model [14]. In our curriculum-learning framework, the emphasis is more on the simpler classification branch during the initial training phase, and more on the harder localization branch in later training phases.

2.4. Temporal action localization

The objective of temporal action detection is to identify the classes of actions, and corresponding start and end times, in long untrimmed videos. This process employs an action recognition network for feature extraction, and draws inspiration from object detection. Significant progress has been made in temporal action detection in recent years. To predict an action's temporal boundaries, SSN [5] divides the proposal into three sections: starting, actionness, and ending. The model then integrates the three sections' features into two classifiers, which attempt to categorize the action and determine whether it is complete. BSN [6] predicts the probability that each frame is the start or end frame of the action, and subsequently concatenates the time points corresponding to high probabilities to get the proposal. Based on BSN, BMN [7] creates a boundary-matching infographic to acquire better proposals. However, the aforementioned techniques can only generate temporal boundary predictions and must rely on separate models for classification. Thus, the localization and classification models are separate and unable to exchange information.

To combine classification and localization models into a single training and testing framework, some methods are constructed in an end-to-end object detection framework. Drawing inspiration from SSD [36,10] generates a one-dimensional temporal convolution to produce multiple temporal action anchors. SSTAD [37] employs a recurrent neural network (RNN) architecture to carry out proposal generation and classification simultaneously. Decouple-SSAD [8] uses two branches for regression and classification, respectively, to produce reliable proposed boundary and classification results. Inspired by the end-to-end framework Faster R-CNN [4], TAL-Net [9] integrates contextual information. Likewise, with reference to the anchor-free approach, AFSD [11] provides an effective spatio-temporal localization methodology that yields state-of-the-art results.

However, the gaps remain between the training and testing procedures of these techniques. Specifically, the testing procedure selects proposals with high localization and classification scores. In the training phase, however, positive samples are selected based solely on the localization score, and are considered equally with respect to each other. To minimize the gap between training and testing, our framework prioritizes proposals with high localization and classification scores in both phases. Based on the significance of location scores in temporal action detection, we construct a weight assignment paradigm based on curriculum learning. This makes our assignment criterion suitable for the temporal localization problem.

3. Method

3.1. Overview

This section presents our curriculum-based structure, and its improvements over the base framework in Fig. 2. First, we define temporal action detection, extract base features, and generate proposals in accordance with standard procedures. The training and testing processes for the fundamental architecture are then discussed, along with the problems inherent to the basic framework. To overcome these problems, we present our curriculum learning structure. In particular, we discuss how our framework improves upon the basic model by minimizing the incongruence between training and testing. Specifically, our network is divided into two sections: the base network, which is based on traditional architecture, and our auxiliary branch. In the base network, we extract and connect spatial and temporal feature vectors using the feature extractor module. We input the connected vectors into the temporal convolution to create the localization and classification features, respectively. The localization branch outputs the distance between the current frame and the start as well as the end action boundaries. The classification branch, on the other hand, chooses which action category the present segment belongs to. Equipped with the base framework, we combine the characteristics of the classification and localization branches to create an auxiliary branch. The section that follows provides further details.

3.2. Notation and Preliminaries

Problem Definition. For an untrimmed video X, the temporal proposal annotation is $\Psi_g = \{\varphi_i = [t_{s,i}, t_{e,i}]\}_{i=1}^{N_g}$, where N_g denotes the number of ground truths. $[t_{s,i}, t_{e,i}]$ is the start and end time of the action instance φ_i. Temporal action detection aims to predict the set of candidate proposals $\Psi_p = \{\varphi_i = [t_{s,i}, t_{e,i}, s_i]\}_{i=1}^{N_p}$ to cover Ψ_g with high recall and high overlap, where s_i is the predicted confidence score of φ_i that will be used for proposal ranking.

Base Feature Extraction. Because untrimmed videos typically range up to several minutes in duration, it is difficult to input them directly into the visual coder for feature extraction, as computational resources are typically limited. Instead, videos are often split into smaller equally-sized segments. Thus, an input video with frame l can be divided into l_s segments by time interval σ. These segments can be formulated as $\{s_n\}_{n=1}^{l_s}$.

$$S = \{s_n\}_{n=1}^{l_s}, \quad l_s = \frac{l}{\sigma} \tag{1}$$

Subsequently, each segment is fed into a pre-trained visual coding system, such as two-stream [22] or I3D [24], which extracts spatial and temporal feature vectors. Finally, these features are linked so that they can be processed together.

Proposal Generation. By convention, we input the joint features into the temporal convolution to acquire the localization and classification features. The localization branch outputs the distance between the start and end boundaries for each frame of the video $(\hat{d}_i^s, \hat{d}_i^e)$, while the classification branch produces a classification score y_i. Thus, each position i outputs a proposal (s_i, e_i, y_i), where:

$$\begin{aligned} s_i &= i - \hat{d}_i^s \\ e_i &= i + \hat{d}_i^e \end{aligned} \tag{2}$$

3.3. Basic Training and Inference Processes

3.3.1. Training

We apply classification loss ℓ_{cls} and localization loss ℓ_{loc} to train the classification branch and localization branch, respectively. ℓ_{cls} is softmax focal loss between classification prediction p_i and ground truth label l_i:

$$\begin{aligned} L_{cls} &= \frac{1}{N}\sum_i \ell_{\text{focal}}(p_i, l_i) \\ &= \frac{1}{N}\left[\sum_{i=1}^{N_{\text{pos}}} \alpha_f (1-p_i)^{\gamma} \log p_i + \sum_{i=1}^{N_{neg}} (1-\alpha_f) p_i^{\gamma} \log(1-p_i)\right] \end{aligned} \tag{3}$$

where α_f and γ are the hyper-parameters introduced in [38]. $N \in \{N_{pos}, N_{neg}\}$. N is number of training samples in each batch. N_{pos}, N_{neg} are the number of positive and negative samples for predictions, individually. For the localization loss:

$$\ell_{loc} = \frac{1}{N_{pos}} \sum_i \left(1 - \frac{|\hat{\phi}_i \cap \phi_i|}{|\hat{\phi}_i \cup \phi_i|}\right) \tag{4}$$

ℓ_{loc} is a tIoU loss between boundaries $\hat{\phi}_i = (s_i, e_i)$and the corresponding ground trurh ϕ_i. We can therefore use objective function ℓ to optimize the model, which is composed of ℓ_{cls} and ℓ_{loc}:

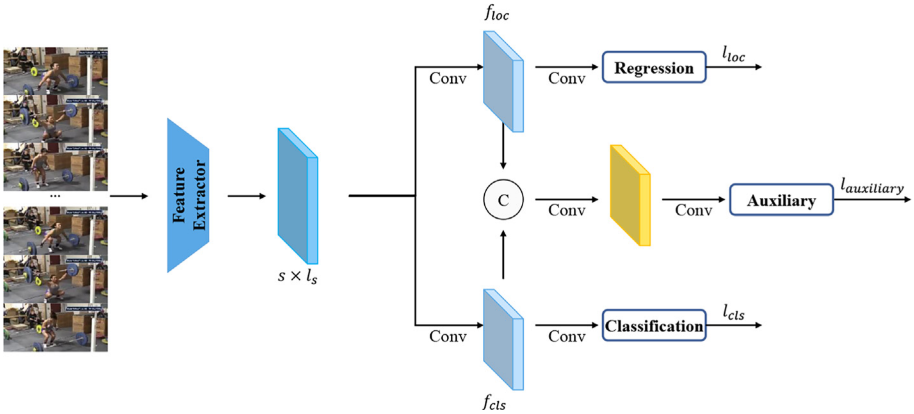


Fig. 2. Summary of our framework.

$$\ell = \ell_{cls} + \ell_{loc} \tag{5}$$

3.3.2. Inference

In inference, the classification score, formulated as $s = y_i$, is used to determine the score of each prediction segment $\psi = (s_i, e_i, s)$. All predictions are subsequently combined and entered into Soft-NMS [39] to eliminate redundant proposals.

3.3.3. Problem

However, the aforementioned basic framework causes a mismatch between training and inference. In particular, evaluation metrics in temporal action detection suggest that a good prediction should yield not only a high classification score, but also accurate localization [29]. Accordingly, proposals with greater consistency between classification and localization heads must be assigned higher weights throughout the training phase. However, in the basic training phase, all samples are treated equally, creating a gap between training and inference. Furthermore, the existing label assignment strategy marks the background as positive (transitions in Fig. 1), which may cause additional performance degradation.

3.4. Our Proposals

3.4.1. Network

To minimize the impact of the transition problem, we devise an auxiliary task to estimate the probability that each frame belongs to the foreground or the background. As shown in Fig. 2, the auxiliary branch is created by integrating classification and localization features. More specifically, we concatenate two features and use 1×1 convolution to reduce the channel dimension. Then, the 3×3 convolution is applied to the reduced features to generate the auxiliary branch. In the inference phase, the prediction scores of the auxiliary task are employed for non-maximum suppression (NMS) [39]. Unlike typical methods that employ classification results as ranking criteria for NMS, we bridge the gap between training and testing by relying on the auxiliary score. The pseudo-code for our framework is in Algorithm 1.

Algorithm 1: Our Dynamic Weighting Framework Based on Curriculum Learning.

```
Input: $\alpha_0, s, p, t, T, N_{pos}, N_{neg}$
$\alpha_0$ is a hyperparameter for adjusting the classification and
   localization weights,
$s$ is the localization score predicted by the model,
$p$ is the classification score predicted by the model,
$t$ is the current number of epochs,
$T$ is the total number of network training epochs,
$N_{pos}$ is the number of predicted positive samples,
$N_{neg}$ is the number of predicted negative samples.
Output: Losses for classification, localization, and auxiliary
   branches $\ell_{cls}$,$\ell_{loc}$,$\ell_{auxiliary}$
for $b_i \in N_{pos}$ do
    $\alpha = (1.0 - \alpha_0) \cdot e^{-t/\sqrt{T}} + \alpha_0$        ◁ Eq. 9
    $w_i = (1 - \alpha) \cdot s_i + \alpha \cdot p_i$        ◁ Eq. 8
    Calculate localization loss $L_{loc}$        ◁ Eq. 7
end for
for $b_i \in N_{pos}$ or $b_i \in N_{neg}$ do
    Calculate classification loss $L_{cls}$        ◁ Eq. 6]
    Calculate auxiliary loss $L_{auxiliary}$        ◁Eq. 10
end for
return $L_{cls}, L_{loc}, L_{auxiliary}$
```

3.4.2. Training

As discussed in [29], training samples should not be weighted equally. In particular, learning from high-quality samples improves the detector's performance, whereas learning from low-score samples reduces detection efficiency as a result of noise. To simplify the learning process, we reweight the positive samples based on their prediction scores, where:

$$L_{cls} = \frac{1}{N}\left[\sum_{i=1}^{N_{\text{pos}}} w_i \ell_{\text{focal}}(p_i, l_i) + \sum_{j=1}^{N_{neg}} \ell_{\text{focal}}(p_j, l_j)\right] \tag{6}$$

$$\ell_{loc} = \frac{1}{N_{pos}} \sum_i w_i \cdot \left(1 - \frac{\left|\hat{\phi}_i \cap \phi_i\right|}{\left|\hat{\phi}_i \cup \phi_i\right|}\right) \tag{7}$$

$$w_i = (1 - \alpha) \cdot s + \alpha \cdot p \tag{8}$$

Compared to Eq. 3 and Eq. 4, Eq. 6 and Eq. 7 add dynamic weights w_i (Eq. 8). The dynamic weight w_i, used to narrow the gap between the classification and localization branches, is the combination of the localization score s and classification score p. Existing dynamic weighting strategies [12,13] combine these scores through the manual modification of hyperparameters. Instead, we use a prior [14] to integrate the two scores. Although the localization score is more significant than the classification score in optimal models, it is difficult to converge in the early stages of training. According to this prior, we design an optimization criterion based on curriculum learning to determine the hyperparameter α:

$$\alpha = (1.0 - \alpha_0) \cdot e^{-t/\sqrt{T}} + \alpha_0 \tag{9}$$

where t is the current training epoch, T is the total number of epochs, and α_0 is a hyperparameter. As illustrated in Eq. 9, the classification score accounts for a larger portion of the final score in the early stages of training. In the latter stages, the proportion of localization scores gradually increases with α.

Fig. 3 displays the weight changes for classification and localization scores under our approach compared to the baseline.In the training process, we contrast the weight changes of the localization and classification branches in the base with our curriculum learning approach. In basic operation, classification and localization branches are treated equally. The weight selection for the prior approach is kept constant during training. In contrast, during the early training phase of our technique (curriculum learning approach), the categorization branch dominates. The weight of the localization score gradually rises as the epoch increases.

Based on the cumulative final score, the auxiliary branch is trained to discriminate between foregrounds and backgrounds. Specifically, each sample predicts a score c_i with a target w_i, which is a continuous value ranging from 0 to 1. We utilize the fused score w_i as the learning objective without any transformations. Therefore, the auxiliary loss can be calculated as:

$$L_{\text{auxiliary}} = \frac{1}{N}\left[\sum_{i=1}^{N_{\text{pos}}} |c_i - w_i| BCE(c_i, w_i) + \sum_{i=1}^{N_{neg}} c_i BCE(c_i, 0)\right] \tag{10}$$

where BCE represents the binary cross-entropy loss. $N \in \{N_{pos}, N_{neg}\}$. N is number of training samples in each batch. N_{pos}, N_{neg} are the number of positive and negative prediction samples, respectively.

3.4.3. Inference

The test process is intended to select proposals with highest scores, which will be produced into action clips. We use the prediction of the auxiliary branch as the final score for the proposal, which is different from traditional methods based on classification

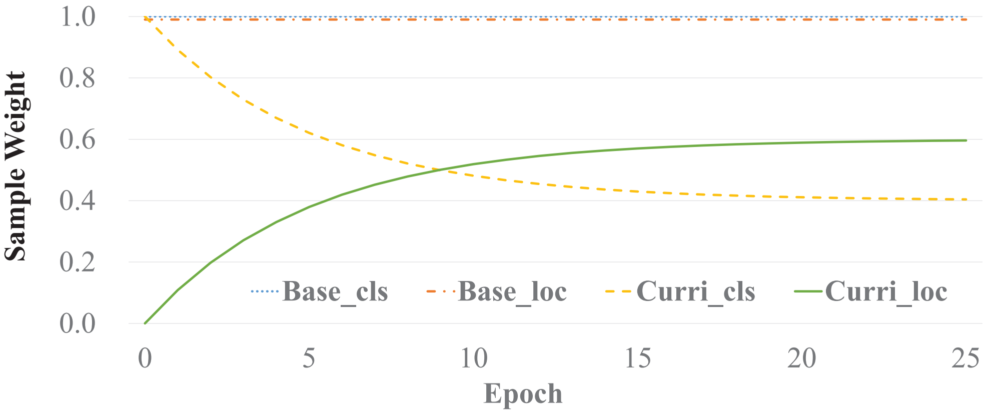


Fig. 3. Weight changes in classification and localization branches of the base and in our curriculum-learning method..

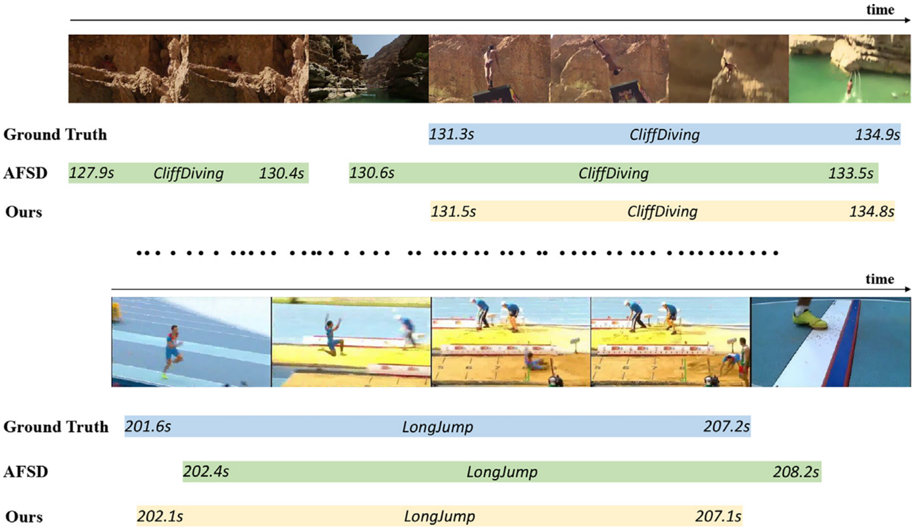


Fig. 4. Qualitative results on THUMOS14 dataset.

scores. The auxiliary score integrates classification and localization features, automatically. Then, the proposals are sorted out by the auxiliary scores and sent to the Soft-NMS [39] to remove the redundant proposals from the list. The remaining proposals will be output as a clip of action.

4. Experiments

4.1. Experimental Settings

Datasets. We run studies on the THUMOS14 [15] and ActivityNet v1.3 [16] datasets, which are both frequently used datasets. On THUMOS14, the average number of action clips in each video is 15, while the length of the videos ranges from a few seconds to over an hour. For these reasons, it is challenging to perform temporal action detection on THUMOS14. According to tradition, our model is tested on 213 test movies after being trained on a validation set of 200 temporal annotations with 20 categories. With regard to large-scale datasets ActivityNet v1.3, it contains 200 action categories, 10024 training, 4926 validation, and 5044 test videos. We conduct training on the training set and testing on the validation set in accordance with standard procedure.

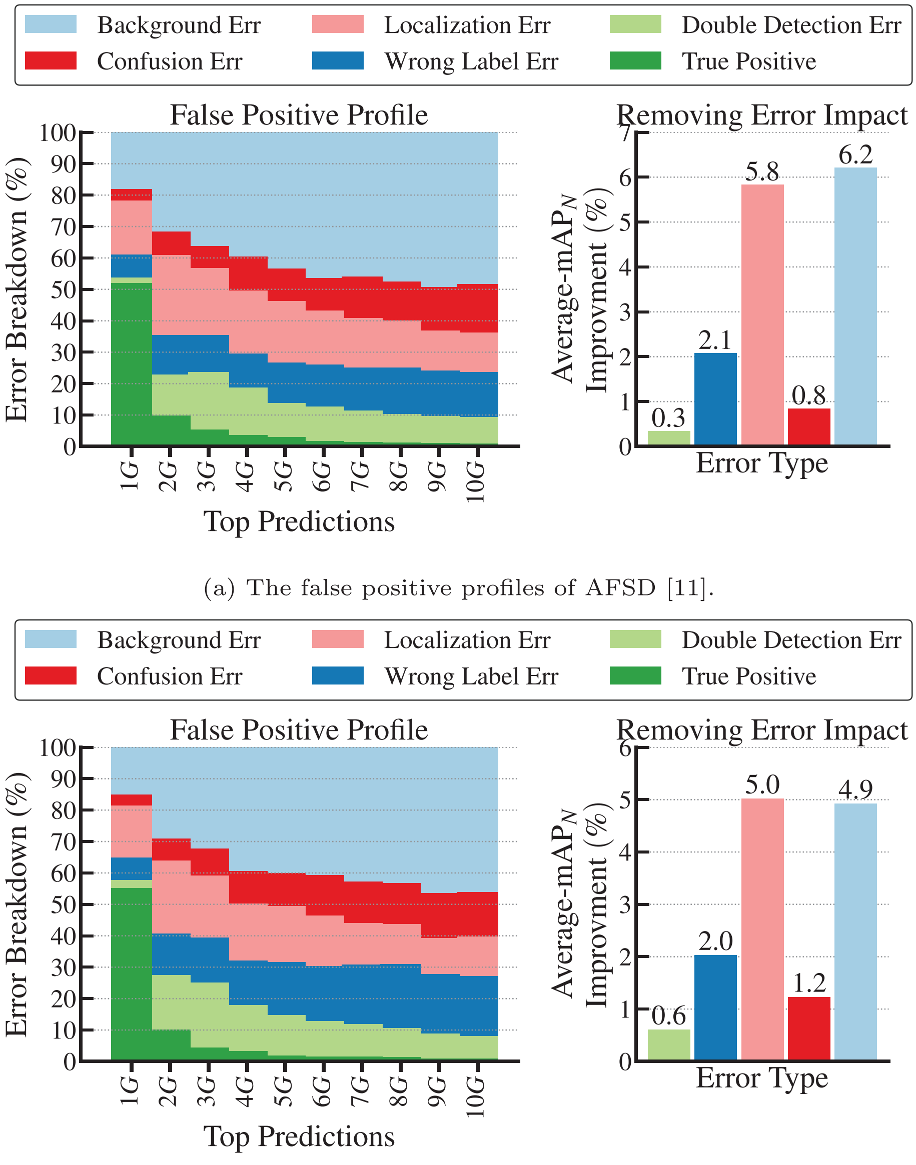
Evaluation Metrics. We use the mean Average Precision (mAP) as a measurement of assessment. Specifically, a proposal is deemed to be correct if the predicted category matches the ground truth instance, and its tIoU in relation to the ground truth instance exceeds a predetermined threshold. The tIoU thresholds on THUMOS14 are $\{0.3, 0.4, 0.5, 0.6, 0.7\}$, while the tIoU thresholds on ActivitiyNet v1.3 are $\{0.50, 0.75, 0.95\}$.

Implementation Details. We utilize the suggested technique with the anchor-free SOTA methodology AFSD [11]. In the study, we maintain their default settings to provide a fair comparison. Following AFSD [11], we fine-tune the I3D [24] model that has been pre-trained on Kinetics to extract the features of the video. During the training phase, our model is trained by Adam [40] with a learning rate of 10^{-5} and a weight decay of 10^{-3}. In testing, the final localization and classification scores are calculated by averaging the outcomes of the RGB and optical flow frames. In addition, we employ random cropping and horizontal flipping to enhance the data. The experiments are carried out on a server using two GPUs (NVIDIA GeForce GTX 3080Ti, 12 GB memory) and Intel(R) Core(TM) i7-6700 K CPU @ 4.00 GHz. It takes about 15 h and 26 h to complete training for THUMOS14 and ActivityNet v1.3, respectively.

4.2. Ablation Study

4.2.1. Visualization Experiments

Qualitative Visualization. We visualize the qualitative results of THUMOS14 in Fig. 4. In these examples, the baseline method



(a) The false positive profiles of AFSD [11].

(b) The false positive profiles of our method.

Fig. 5. A false positive (FP) analysis of the results on THUMOS14 using the tool from [14]. Left: calculate FP errors by considering predictions for top 10 ground truth instances (G). Right: The impact of the error type on average mAP_N, which is the improvement obtained by removing the prediction of each error type.

(AFSD [11]) can correctly predict the class of action. But it is vulnerable to interference by background frames, and it is difficult to accurately predict the position of the actions. By adopting the curriculum-based method, we reduce the influence of background and predict temporal boundary accurately.

Quantitative Visualization. To better understand errors and their types, we use the tools described in [14] for the quantitative analysis of false positive samples. As shown in Fig. 5, the most important impact is the location and the background errors. Localization error leads to degradation of 5.8% average-mAP, while the background error causes a degradation of 6.2% average-mAP. By adding our curriculum-learning approach, localization and background errors are significantly reduced. The effects of localization and background errors are reduced to 5.0% and 4.9%, respectively. Therefore, we can effectively reduce the impact of localization and background errors through the construction of auxiliary branches and the curriculum learning approach.

4.2.2. Hyper-parameter Experiments

The Effectiveness of α_0. We evaluate the model's performance under varying α_0 values to assess the impact of the hyperparameter. As shown in Fig. 6 and Table 1, models with α_0 values of 0.2 and 0.4 outperform other models significantly. Although the early stages of training are dominated by the classification score, emphasis shifted to the detection score in the latter epochs. This phenomenon is consistent with the prior emphasis on classification scores at the beginning of training and on detection scores in the later training stages [14]. Additionally, it is inefficient to distribute weights using easy-to-train classification scores during the overall training process. For instance, performance is 56.2% at mAP@0.5 in THUMOS14 and 33.4% at mAP@Avg in ActivityNet when only the classification score is used to assign dynamic weights. In this paper, we set 0.4 as the value of alpha, which yields a performance of 57.6% at mAP@0.5 in THUMOS14 and 35.4% at mAP@Avg in ActivityNet v1.3.

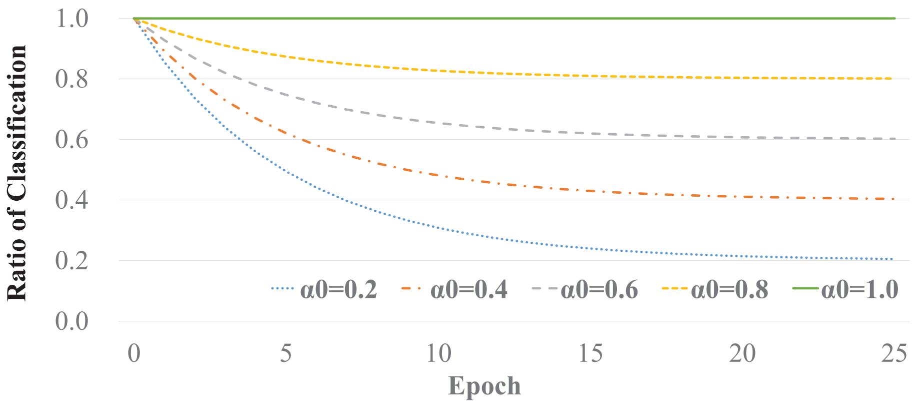


Fig. 6. The ratio of classification score in the final score (ratio of classification) with different α_0 values during the training process.

Table 1
The performance of the model in THUMOS14 and ActivityNet v1.3 at various α_0 settings. The mAP of tIoU is calculated by using a 0.5 threshold (mAP@0.5) in THUMOS14. The average mAP (mAP@Avg) is calculated in [0.50:0.05:0.95] on ActivityNet v1.3.

α_0	THUMOS14(mAP@0.5)	ActivityNet(Avg)
0.2	57.3	35.2
0.4	**57.6**	**35.4**
0.6	56.8	35.1
0.8	56.6	33.9
1.0	56.2	33.4

Table 2
The performance of the model in THUMOS14 and ActivityNet v1.3 at various γ settings.

α_f	γ	THUMOS14(mAP@0.5)	ActivityNet(Avg)
0.25	1.0	57.4	**35.4**
0.25	2.0	**57.6**	**35.4**
0.25	5.0	56.8	34.9

Table 3
The performance of the model in THUMOS14 and ActivityNet v1.3 at various α_f settings.

α_f	γ	THUMOS14(mAP@0.5)	ActivityNet(Avg)
0.10	2.0	57.5	35.1
0.25	2.0	**57.6**	**35.4**
0.50	2.0	56.6	34.2
0.75	2.0	56.1	33.6
0.90	2.0	56.0	33.5
0.99	2.0	49.7	32.7

Table 4
Comparison of the speed of the inference, the size of the model and the complexity of our method and our baseline AFSD [11]

Method	Runtime	#Params	Flops	mAP@0.5 (%)
AFSD [11]	55.7 ms	44.7 M	84.4G	55.5
Ours	57.3 ms	45.2 M	84.5G	57.6

The Effectiveness of α_f and γ. We also evaluate the hyperparameter α_f and γ in focal loss [38]. We control variates to verify the effects of α_f and γ in Table 2 and Table 2, respectively. When the α_f value is 0.25 and the γ value is 2.0, the model achieve the best results on both the THUMOS14 and the ActivityNet v1. 3 datasets.3.

Table 5
The effectiveness of each component of our approach using THUMOS14. The performance of the fundamental framework is 55.5% for mAP@0.5. The performance rises to 57.6% when the three suggested methods are used.

Method	Performance				
dynamic weighting	×	✓	✓	✓	✓
auxiliary branch	×	×	✓	×	✓
curriculum learning	×	×	×	✓	✓
mAP@0.5(%)	55.5	56.1	56.7	57.1	**57.6**

Table 6
The analysis for the auxiliary branch. Auxiliary (classification) represents delete the auxiliary branch, but add the auxiliary loss to the classification branch.

Setting	mAP@0.5 (%)
w/o auxiliary branch	57.1
w/ auxiliary (classification)	57.4
w/ auxiliary branch	**57.6**

Table 7
Performance of the model on AFSD and PGCN in THUMOS14.

Type	Model	THUMOS14	
		mAP@0.5 (%)	↑
Anchor-free	AFSD [11]	55.5	-
	Ours	57.6	2.1
Two-stage	PGCN [41]	49.1	-
	Ours	50.9	1.8

4.2.3. Comparison of Model Parameters and Inference Speed

We compare the time of inference and complexity of the model of our curriculum-Learning method and the baseline AFSD in Table 4. Specifically, we inference the model for 25 times on a Nvidia Geforce GTX 3080 Ti, and we report an average inference time. In addition, we calculate FLOPs and the parameters using the Python tool Opcounter. As can be observed in Table 4, compared with the baseline AFSD, our method yields relatively small amounts of additional time and complexity of the model, but it can significantly improve the performance of the model.

4.2.4. The Effectiveness of Each Module

We apply our algorithm to the state-of-the-art method [11] to validate the performance of each component: dynamic weighting, auxiliary branch, and curriculum learning. The dynamic weights of training samples are the sums of their classification and localization scores, the auxiliary branch bridges the gap between training

Table 8
Performance comparison with state-of-the-art methods on THUMOS14 and ActivityNet1.3. The performance is measured by mAP at different IoU thresholds. The average mAP is calculated in [0.3:0.1:0.7] on THUMOS14 and [0.50:0.05:0.95] on ActivityNet v1.3, respectively. Ours denotes the performance of our proposed method based on AFSD [11].

Method	Publication	THUMOS14(mAP@IoU)					ActivityNet-1.3(mAP@IoU)			
		0.3	0.4	0.5	0.6	0.7	0.50	0.75	0.95	Average
SSN [5]	ICCV2017	51.9	41.0	29.8	-	-	43.3	28.7	5.6	28.3
TAL-Net [9]	CVPR2018	53.2	48.5	42.8	33.8	20.8	38.2	18.3	1.3	20.2
BSN [6]	ECCV2018	53.5	45.0	36.9	28.4	20.0	46.5	30.0	8.0	30.0
BMN [7]	ICCV2019	56.0	47.4	38.8	29.7	20.5	50.1	34.8	8.3	33.9
MGG [42]	CVPR2019	53.9	46.8	37.4	29.5	21.3	-	-	-	-
GTAN [43]	CVPR2019	57.8	47.2	38.8	-	-	52.6	34.1	8.9	34.3
G-TAD [44]	CVPR2020	54.5	47.6	40.2	30.8	23.4	50.4	34.6	9.0	34.1
BC-GNN [45]	ECCV2020	57.1	49.1	40.4	31.2	23.1	50.6	34.8	9.4	34.3
BU-TAL [46]	ECCV2020	53.9	50.7	45.4	38.0	28.5	43.5	33.9	9.2	30.1
A2Net [47]	TIP2020	58.6	54.1	45.5	32.5	17.2	43.6	28.7	3.7	27.8
BSN++ [48]	AAAI2021	59.9	49.5	41.3	31.9	22.8	51.3	35.7	8.3	34.9
TVNet [49]	VISIGRAPP2022	64.7	58.0	49.3	38.2	26.4	51.4	35.0	**10.1**	34.6
DCAN [50]	AAAI2022	68.2	62.7	54.1	43.9	**32.6**	51.8	36.0	9.5	35.4
AFSD [11]	CVPR2021	67.3	62.4	55.5	43.7	31.1	52.4	35.3	6.5	34.4
Ours	-	**68.8**	**64.9**	**57.6**	**44.6**	31.9	**52.8**	**36.2**	7.3	**35.4**

and testing by combining classification and localization branches, and curriculum learning ensures that weight assignment adheres to Eq. 8 and Eq. 9. As shown in Table 5, when assigning dynamic weights during the training phase, the performance of the model improves from 55.5% to 56.1% for THUMOS14 (mAP@0.5). Subsequently, performance further reaches 56.7% and 57.1% by including the auxiliary branch and curriculum learning paradigms, respectively. Finally, by merging the three aforementioned modules, our technique outperforms [11] by 2.1% on mAP@0.5. These experiments illustrate the effectiveness of each component of our algorithm. In particular, the curriculum learning approach improves the model performance by one percent, from 56.1% to 57.1%. In addition, we attempt to analyze the auxiliary branches. As shown in Table 6, if we delete the auxiliary branch but add the auxiliary loss to classification branch, the performance of the model improves from 57.1% to 57.4% The combination of classification and localization scores is the target for training the auxiliary/actionness scores (Eq. 10). The performance of this model is slightly lower than adding the auxiliary branch (57.6%).

4.2.5. The Effectiveness under Different networks

To verify the effective application of our methodology to different network, we use our methods on the anchor-free method AFSD [11] and two-stage method PGCN [41], respectively. As shown in Table 7, the performance of AFSD and PGCN is enhanced by the application of our methods. The performance of AFSD improves by 2.1% on THUMOS14. And the performance of PGCN improves by 1.8% on THUMOS14.

4.3. Comparison with State-of-the-art Methods

Table 8 compares the proposed strategy's performance with that of the most recent SOTA approaches. We implement our proposed technique on the anchor-free framework AFSD [11], which is included in CVPR2021. Compared with AFSD, our method exhibits improved performance on both the THUMOS14 and ActivityNet v1.3 datasets, especially with the mAP@0.5 (THUMOS14) and mAP@Avg (ActivityNet v1.3) metrics. On the THUMOS14 dataset, our method yields a performance improvement from 55.5% to 57.6% for mAP@0.5. For ActivityNet v1.3, the average mAP increases from 34.4% to 35.4% under our strategy. Table 8 lists recently developed SOTA methods for temporal action detection, along with the conferences where they were published. The SSN [5] in ICCV 2017 and TAL-Net [9] in CVPR2018 to TVNet [49] in VISIGRAPP 2022 and DCAN [50] in AAAI 2022 are all covered by the approaches we have listed. Our method maintains a competitive performance with that of existing SOTA methods. Specifically, we achieve top results in several metrics, including mAP [0.3:0.1:0.7] on THUMOS14, as well as mAP@0.50, mAP@0.75, mAP@0.95, and mAP@Avg on ActivityNet v1.3.

5. Conclusion

This paper proposes an auxiliary branch that combines the features of classification and localization to unify the training and testing procedures. The auxiliary branches are constructed by dynamic weighting based on curriculum learning, in accordance with the following criterion: Early training stages rely on classification scores, whereas late training stages emphasize localization scores. We successfully reach a performance of 57.6% on THUMOS14 (mAP@0.5), and 35.4% on ActivityNet v1.3 (mAP@Avg), using our curriculum-based learning approach. Future research directions include: (1) Verifying our method's accuracy with additional approaches, particularly the most recent transformer method. (2) Attempting to merge classification and localization branches in a more logical manner, such as through attention mechanisms.

CRediT authorship contribution statement

Yunze Chen: Conceptualization, Writing - original draft. **He Jiang:** Methodology. **Junrui Xiao:** Visualization. **Ding Li:** Writing - review & editing. **Qingyi Gu:** Supervision, Funding acquisition.

Data availability

The data that has been used is confidential.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the Scientific Instrument Developing Project of the Chinese Academy of Sciences under Grant YJKYYQ20200045; in part by the National Natural Science Foundation of China under Grant 62276255. Declaration of Competing Interest The authors declare that they have no known com-

peting financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Fan, W. bing Huang, C. Gan, S. Ermon, B. Gong, J. Huang, End-to-end learning of motion representation for video understanding, IEEE Conference on Computer Vision and Pattern Recognition (2018) 6016–6025.

[2] T.C. Sparks, N. Storer, A. Porter, R. Slater, R. Nauen, Insecticide resistance management and industry: the origins and evolution of the insecticide resistance action committee (irac) and the mode of action classification scheme, Pest Management Science 77 (2021) 2609–2619.

[3] Y. Tan, Y. Hao, X. He, Y. wei Wei, X. Yang, Selective dependency aggregation for action classification, Proceedings of the 29th ACM International Conference on Multimedia.

[4] S. Ren, K. He, R.B. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2015) 1137–1149.

[5] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, D. Lin, Temporal action detection with structured segment networks, IEEE International Conference on Computer Vision (2017) 2933–2942.

[6] T. Lin, X. Zhao, H. Su, C. Wang, M. Yang, Bsn: Boundary sensitive network for temporal action proposal generation, Proceedings of the European Conference on Computer Vision.

[7] T. Lin, X. Liu, X. Li, E. Ding, S. Wen, Bmn: Boundary-matching network for temporal action proposal generation, IEEE International Conference on Computer Vision (2019) 3888–3897.

[8] Y. Huang, Q. Dai, Y. Lu, Decoupling localization and classification in single shot temporal action detection, IEEE International Conference on Multimedia and Expo (2019) 1288–1293.

[9] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D.A. Ross, J. Deng, R. Sukthankar, Rethinking the faster r-cnn architecture for temporal action localization, IEEE Conference on Computer Vision and Pattern Recognition (2018) 1130–1139.

[10] T. Lin, X. Zhao, Z. Shou, Single shot temporal action detection, Proceedings of the 25th ACM international conference on Multimedia.

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

[12] C. Feng, Y. Zhong, Y. Gao, M.R. Scott, W. Huang, Tood: Task-aligned one-stage object detection, IEEE International Conference on Computer Vision (2021) 3490–3499.

[13] K. jik Kim, H.S. Lee, Probabilistic anchor assignment with iou prediction for object detection, Proceedings of the European Conference on Computer Vision.

[14] H. Alwassel, F.C. Heilbron, V. Escorcia, B. Ghanem, Diagnosing error in temporal action detectors, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 256–272.

[15] H. Idrees, A.R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, M. Shah, The thumos challenge on action recognition for videos "in the wild", Computer Vision and Image Understanding 155 (2017) 1–23.

[16] F.C. Heilbron, V. Escorcia, B. Ghanem, J.C. Niebles, Activitynet: A large-scale video benchmark for human activity understanding, IEEE Conference on Computer Vision and Pattern Recognition (2015) 961–970.

[17] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, IEEE Conference on Computer Vision and Pattern Recognition (2011) 3169–3176.

[18] H. Wang, C. Schmid, Action recognition with improved trajectories, IEEE International Conference on Computer Vision (2013) 3551–3558.

[19] K. Soomro, A.R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, IEEE Conference on Computer Vision and Pattern Recognition.

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, IEEE Conference on Computer Vision and Pattern Recognition (2014) 1725–1732.

[21] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, A. Natsev, M. Suleyman, A. Zisserman, The kinetics human action video dataset, IEEE Conference on Computer Vision and Pattern Recognition.

[22] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, Conference and Workshop on Neural Information Processing Systems.

[23] D. Tran, L.D. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, IEEE International Conference on Computer Vision (2015) 4489–4497.

[24] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, IEEE Conference on Computer Vision and Pattern Recognition (2017) 4724–4733.

[25] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L.V. Gool, Temporal segment networks: Towards good practices for deep action recognition, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 20–36.

[26] T. Yang, X. Zhang, W. Zhang, J. Sun, Metaanchor: Learning to detect objects with customized anchors, Conference and Workshop on Neural Information Processing Systems.

[27] J. Wang, K. Chen, S. Yang, C.C. Loy, D. Lin, Region proposal by guided anchoring, IEEE Conference on Computer Vision and Pattern Recognition (2019) 2960–2969.

[28] X. Zhang, F. Wan, C. Liu, X. Ji, Q. Ye, Learning to match anchors for visual object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2022) 3096–3109.

[29] H. Li, Z. Wu, C. Zhu, C. Xiong, R. Socher, L.S. Davis, Learning from noisy anchors for one-stage object detection, IEEE Conference on Computer Vision and Pattern Recognition (2020) 10585–10594.

[30] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, Conference and Workshop on Neural Information Processing Systems.

[31] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: ICML '09, 2009.

[32] E.A. Platanios, O. Stretcu, G. Neubig, B. Poczos, T.M. Mitchell, Competence-based curriculum learning for neural machine translation, ArXiv abs/1903.09848.

[33] G. Alain, A. Lamb, C. Sankar, A.C. Courville, Y. Bengio, Variance reduction in sgd by distributed importance sampling, ArXiv abs/1511.06481.

[34] V. Cirik, E.H. Hovy, L.-P. Morency, Visualizing and understanding curriculum learning for long short-term memory networks, ArXiv abs/1611.06204.

[35] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, F. Huang, Curricularface: Adaptive curriculum learning loss for deep face recognition, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5900–5909.

[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, Proceedings of the European Conference on Computer Vision.

[37] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, J.C. Niebles, End-to-end, single-stream temporal action detection in untrimmed videos, British Machine Vision Conference.

[38] T.-Y. Lin, P. Goyal, R.B. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2020) 318–327.

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

[40] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, ICLR (Poster).

[41] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, C. Gan, Graph convolutional networks for temporal action localization, IEEE International Conference on Computer Vision (2019) 7093–7102.

[42] Y. Liu, L. Ma, Y. Zhang, W. Liu, S.-F. Chang, Multi-granularity generator for temporal action proposal, IEEE Conference on Computer Vision and Pattern Recognition (2019) 3599–3608.

[43] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, T. Mei, Gaussian temporal awareness networks for action localization, IEEE Conference on Computer Vision and Pattern Recognition (2019) 344–353.

[44] M. Xu, C. Zhao, D.S. Rojas, A.K. Thabet, B. Ghanem, G-tad: Sub-graph localization for temporal action detection, IEEE Conference on Computer Vision and Pattern Recognition (2020) 10153–10162.

[45] Y. Bai, Y. Wang, Y. Tong, Y. Yang, Q. Liu, J. Liu, Boundary content graph neural network for temporal action proposal generation, Proceedings of the European Conference on Computer Vision.

[46] P. Zhao, L. Xie, C. Ju, Y. Zhang, Y. Wang, Q. Tian, Bottom-up temporal action localization with mutual regularization, Proceedings of the European Conference on Computer Vision.

[47] L. Yang, H. Peng, D. Zhang, J. Fu, J. Han, Revisiting anchor mechanisms for temporal action localization, IEEE Transactions on Image Processing 29 (2020) 8535–8548.

[48] H. Su, W. Gan, W. Wu, J. Yan, Y. Qiao, Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation, Association for the Advancement of Artificial Intelligence.

[49] H. Wang, D. Damen, M. Mirmehdi, T. Perrett, Tvnet: Temporal voting network for action localization, International Conference on Computer Vision Theory and Applications.

[50] G. Chen, Y.-D. Zheng, L. Wang, T. Lu, Dcan: Improving temporal action detection via dual context aggregation, Association for the Advancement of Artificial Intelligence.

Yunze Chen received the B.Sc. degree from the Harbin Institute of Technology, Weihai, China, in 2018. He is currently pursuing the Ph.D. degree with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include computer vision and video processing.

He Jiang received the B.Sc. degree in Electronic Information Science and Technology from Nankai University, Tianjin, China, in 2019. He is currently pursuing the master's degree at the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His current research interests include computer vision and object detection.

Junrui Xiao received the B.Sc. degree from Xidian University, Shaanxi, China, in 2020. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His current research interests include computer vision and model compression.

Ding Li received the B.E. degree in electrical engineering from Wuhan University, Wuhan, Hubei, China, in 2016. He is currently pursuing the Ph.D. degree in the Institute of Automation, Chinese Academy of Sciences, and University of Chinese Academy of Sciences. His research interests include machine learning, neural networks, and computer vision.

Qingyi Gu (M'13) received the B.E. degree in Electronic and Information Engineering from Xi'an Jiaotong University, China, in 2005. He received the M.E. degree, and Ph.D. degree in Engineering, Hiroshima University, Japan, in 2010, and 2013 respectively. He is currently a professor in Institute of Automation, Chinese Academy of Sciences, China. His primary research interest is high-speed image processing, and applications in industry and biomedicine.