Multimedia Tools and Applications (2023) 82:879–898
https://doi.org/10.1007/s11042-022-12720-7

A coarse-to-fine temporal action detection method combining light and heavy networks

Fan Zhao[1] · Wen Wang[1] · Yu Wu[1] · Kaixuan Wang[1] · Xiaobing Kang[1]

Received: 21 December 2020 / Revised: 17 May 2021 / Accepted: 21 February 2022 /
Published online: 10 June 2022

Abstract

Temporal action detection aims to judge whether there existing a certain number of action instances in a long untrimmed videos and to locate the start and end time of each action. Even though the existing action detection methods have shown promising results in recent years with the widespread application of Convolutional Neural Network (CNN), it is still a challenging problem to accurately locate each action segment while ensuring real-time performance. In order to achieve a good tradeoff between detection efficiency and accuracy, we present a coarse-to-fine hierarchical temporal action detection method by using multi-scale sliding window mechanism. Since the complexity of the convolution operator is proportional to the number and the size of the input video clips, the idea of our proposed method is to first determine candidate action proposals and then perform the detection task on these candidate action proposals only with a view to reducing the overall complexity of the detection method. By making full use of the spatio-temporal information of video clips, a lightweight 3D-CNN classifier is first used to quickly determine whether the video clip is a candidate action proposal, avoiding the re-detection of a large number of non-action video clips by the heavyweight deep network. A heavyweight detector is designed to further improve the accuracy of action positioning by considering both boundary regression loss and category loss in the target loss function. In addition, the Non-Maximum Suppression (NMS) is performed to eliminate redundant detection results among the overlapping proposals. The mean Average Precision (mAP) is 40.6%, 51.7% and 20.4% on THUMOS14, ActivityNet and MPII Cooking dataset when the Intersection-over-Union (tIoU) threshold is set to 0.5, respectively. Experimental results show the superior performance of the proposed method on three challenging temporal activity detection datasets while achieving real-time speed. At the same time, our method can generate proposals for unseen action classes with high recalls.

Keywords Untrimmed video · Temporal action detection · Convolutional neural network (CNN),·action proposal · Detector · Classifier

✉ Fan Zhao
vau@xaut.edu.cn

[1] Department of Information Science, Xi'an University of Technology, Xi'an 710054, China

1 Introduction

As one of the most challenging sub-tasks in visual understanding system, the purpose of action detection is not only to capture the activity, but also to accurately locate the start time and end time of each activity in an untrimmed video. This task is of vital value in the fields of robotics, video analysis and public security [29, 31, 52]. It has largely attracted the attention of many academics and industrial researchers, and many promising results have been achieved in recent years. However, due to unclear action boundaries, big difference of action duration, as well as the diversified shooting environment and editing technology, successful temporal action localization is still a very challenging task.

In the past several years, most approaches address this challenging issue by classifying temporal segments generated in the form of sliding windows with hand-crafted features [9, 17, 29, 31, 45]. Due to the limitation of manual features, the performance of most methods that rely on hand-crafted features still requires much improvement. Compared with traditional hand-crafted features, the deep features extracted from videos based on deep learning methods are more descriptive. In recent years, with the renaissance of convolutional neural networks (CNNs), many deep learning-based methods have achieved impressive improvements in temporal action localization, and these methods can be divided into one-stage localization and two-stage localization. Inspired by a set of object detection methods - single shot detection models such as SSD [27] and YOLO [33], the one-stage localization methods [1, 24, 28, 34] aim to determine the boundaries and categories of multiple action instances in time direction. With the corresponding models of temporal convolution networks, the proposals and categories of temporal actions are achieved simultaneously. However, due to the variety of action types and scales, as well as the structural limitations of the corresponding CNN models, these methods cannot effectively and accurately deal with action segments with arbitrary length.

Although there are many ways taking these two stages jointly, most detection methods still carry out these two stages separately. The two-stage action location methods [2, 5–8, 10–15, 17, 19, 22, 23, 25, 26, 30, 36–38, 41, 46–50, 54, 55] are generally based on the paradigm of proposal generation and then category recognition or boundary regression. Existing methods for action proposal generation can be roughly categorized into two main types: sliding windows based methods [2, 7, 11, 37] and probability distribution based methods [15, 25, 28, 46, 55]. The former methods first generate proposals via pre-defined multi-scale sliding windows and then train a model to evaluate proposals, and good results can be achieved by uniformly covering all the segments in videos. However, the proposals generated by sliding windows are not flexible enough, and increasing the number of sliding windows will lead to the high computation cost. The latter methods generate proposals by grouping continuous regions with high actionness confidence. This type of methods generates more flexible proposals and shows higher performance than sliding-window based methods. However, it still retains some shortcomings, such as producing proposals with high scores but low overlap. Methods [10, 22, 54] usually uses a regression mechanism with classification function to refine the proposal segments. However, when the boundaries of background and action segments are not obvious, low-overlap proposals are still generated.

Our main contributions can be summarized in the following aspects:

(1) In order to achieve a good tradeoff between detection efficiency and accuracy, this paper proposes a coarse-to-fine temporal action detection scheme, which combines a light-weight classifier and a heavyweight detector. As many frames in the untrimmed video are

empty from actions, the classifier avoids running the CNN-based detector on these regions without actions, which reduces the whole complexity. By considering both boundary regression loss and category loss in the detection loss function, the heavy-weight detector further improves the accuracy of action positioning.

(2) Before running the CNN-based detector on the candidate proposal, in order to further improve the accuracy of the temporal action detection, a certain amount of context information is added around the candidate as the input of the detector.

(3) Almost all sliding window based detection methods may generate multiple proposals with different temporal overlap around a ground truth action instance. In order to obtain higher recall with fewer proposals, non-maximum suppression (NMS) is performed at frame-level rather than proposal-level to remove redundant detection results.

(4) Extensive experimental results demonstrate that the proposed method is competitive with state-of-the-art methods on three challenging temporal activity detection datasets such as THUMOS14, ActivityNet and MPII Cooking, while achieving real-time speed.

The rest of this paper is organized as follows. We introduce the related work in Section 2, and describe the details of the proposed method in Section 3. Then, we present experimental results in Section 4. Finally, we conclude the results in Section 5.

2 Related work

Recently, many methods have been proposed to deal with different challenges in temporal action detection and recognition in undivided video, such as low resolution, camera motion, occlusion, background clutter, and scale and temporal variations. Although action detection and recognition has been studied for many years, they are still in the testing stage of laboratory datasets, and are far from practical application and industrialization. The locating task of temporal actions when they occur in a video is still very challenging. In this section, we will review the related techniques about the processing of temporal actions.

2.1 Action recognition

With the availability of efficient hardware and the development of deep learning, convolutional networks has been widely used in many works, such as facial expression recognition [21], place classification [51], object-tracking [53] and gesture/action recognition [4, 20, 39, 42, 44]. Action recognition is an important branch of video related research areas and has been extensively studied. Earlier methods such as improved Dense Trajectory (iDT) [44] mainly adopt hand-crafted features such as HOF, HOG and MBH. In recent years, convolutional networks have achieved great performance in action recognition [39, 42]. Typically, two-stream network [39] learns appearance and motion features based on RGB frame and optical flow field separately. C3D network [42] adopts 3D convolutional layers to directly capture both appearance and motion features from raw frames volume. I3D [4] combines two-stream network and three-dimensional convolution.

Action recognition models can be used for extracting frame or snippet level visual features in long and untrimmed videos. In order to quickly determine whether there are actions in a large number of sliding windows, we introduce a ResNet-10 based lightweight 3D-CNN classifier for the coarse proposal detection. The lightweight 3D classifier is chosen here because it not only makes full use of the spatio-temporal information of the video clip, but also has high execution efficiency.

2.2 Object and temporal action detection

Temporal action detection is an extension of object detection in temporal dimension, therefore, it is necessary to study object detection. Object detection is a fundamental step in computer vision and has received increasing attention during decades. The goal of object detection is to classify and locate natural objects in the image, such as people, cars and bicycles. Object detection can be mainly split into the one-stage detectors [27, 33] and the two-stage ones [13–15, 32, 56]. Temporal action detection aims to classify the actions in temporal proposals into correct categories. Inspired by single-pass object detectors such as SSD [27] and YOLO [33], many one-stage action detection methods are proposed in recent years [1, 24, 28, 34, 43]. Although the detection speed of the one-stage methods is fast, most of them still faces a big positioning accuracy challenge due to the fixed scale. Applied to an exhaustive search of the possible object locations, this yields the well-known cascade approach to sliding-window object detection [13, 14, 40, 58]. Among the two-stage target detection methods, RCNN [15] and its variants [13, 14, 52] are very popular models, which use a "detection by classification" framework: first make a proposal, and then classify the proposal. Motivated by the fact that looking for a locally optimal hypothesis at a coarse resolution often predicts well the best hypothesis at the next resolution level, coarse-to-fine cascade design [32] is proposed for fast deformable object detection. A coarse-to-fine adaptation is designed for cross-domain two-stage object detection, where the foregrounds are first figured out from entire images in different domains and the global prototype for each category is then built across domains [56]. Two-stage scheme aims to boost efficiency performance by solving problems through a coarse to fine process, it has also been successfully applied to temporal action detection [2, 5, 6, 10, 11, 19, 22, 23, 25, 26, 36–38, 41, 46, 47, 49, 50, 54, 55]. As the backbone of multi-stage detection, the sliding window mechanism is often used to generate candidate proposal segments [37]. However, multi-scale sliding windows will lead to expensive calculation costs and inaccurate event boundaries. Cascaded Boundary Regression (CBR) [10], Temporal Unit Regression Network (TURN) [11] and Multiscale Proposal Regression Network (MPRN) [55] are used to refine the temporal boundaries of the sliding windows. By adding its context to the candidate proposal, we combine the category decision with boundary regression to refine the action boundary of the candidate action proposals. The detector designed in this way can further improve the positioning accuracy. Boundary Sensitive Network (BSN) [25] adopts "local to global" fashion to locally combine high probability boundaries as proposals and globally retrieve candidate proposals using proposal-level feature. The frame-level fine-grained fashion can generate more flexible action time duration, which makes the performance of BSN superior to other algorithms. Different with BSN, our method directly operates at the window-level, avoiding BSN's frame-level depth feature extraction and temporal evaluation, so can be run with much higher efficiency.

In order to achieve a good tradeoff between detection efficiency and accuracy, we propose a hierarchical temporal action detection method which combines a lightweight network and a heavyweight network. Similar to these coarse-to-fine (CF) schemes [32, 56], our method starts by coarsely locating candidate action proposals and then only propagates higher-scoring hypotheses to the next level of fine-grained detection. However, different from the methods used in the two separate stages in the above literatures, we first take advantage of a low-complexity action classifier to eliminate non interesting video regions (i.e. regions without actions), then use a heavyweight detector on regions of interest to accurately detect action instance. The idea behind our proposed method is to first determine relevant action video clips and then perform the detection task on these video clips only with a view to reducing the overall complexity of the detection method. In addition, before running the CNN-based detection core on the candidate proposal segment, a certain size of context information is added around the candidate to further improve the accuracy of detection. Almost all sliding window based detection methods may generate multiple proposals with different temporal overlap around a ground truth action instance. In order to suppress redundant proposals to obtain higher recall with fewer proposals, NMS is not performed at the proposal-level, but at frame-level in the overlapping proposals. The advantage of this is that fine-grained NMS processing can reduce positioning errors.

3 Coarse-to-fine temporal action detection method

3.1 Architecture

Since the complexity of the convolution operator is proportional to the number and the size of the input video clips, the goal of our framework is to accurately detect action instances while optimizing the execution time; this is done by first taking advantage of a low-complexity coarse-grained classifier to extract candidate action proposals, then using a fine-grained detector on regions of interest for an accurate action detection. The coarse-grained classifier can eliminate CNN-based re-detection of a large number of non-action video segments; The fine-grained detector can locate the action segment delicately to ensure the accuracy.

The architecture of the proposed algorithm is shown in Fig. 1, which is mainly composed of five modules, namely video clip extraction, event and nonevent classification, context extension, detection and post-processing. Firstly, multi-scale sliding window mechanism is used to generate video clips of different lengths from the input original video. Secondly, a lightweight classifier is designed to identify whether the down-sampled video clip belongs to an action or not. There are two reasons why Resnet-10 is chosen as the backbone network of the classifier here. One is that this residual network can produced strong features. The other is that it has good real-time performance, which can quickly judge whether the video segment is an action. Thirdly, after the candidate temporal action proposals are obtained by a watershed threshold segmentation of the action confidences, the corresponding contexts are added to the video clip to get the expanded action proposals. Fourthly, dense frame-level position regression and category prediction are performed on the extended action proposals by a specially designed detector, which takes the heavyweight Resnet-50 [16] as its backbone network. Finally, different post-processing is carried on video clips of different scales. When the candidate action proposal corresponds to small scale, the threshold segmentation and NMS merging are performed only for the frame-level recognition results to locate the action position in the extended candidate action proposal, which is also the strategy adopted by most target detectors. However, when the size of candidate action proposal is large enough, after the recognition result is processed using the same way as to a short video, a similar processing is also performed on the frame-level offsets for the high-scoring proposals. The advantage of doing so is that finer results for long video timing boundaries can be obtained.

In the undivided video, the minimum length of the actual target action is only 1 to 2 s, while the maximum length is several minutes. In order to detect action segments in both of short and long video segments well later, we extract video clips by using multi-scale sliding windows with lengths of (32, 64, 128, 256, 512) and the step size as 1/4 of the sliding window length. After video segments with different sizes are generated by dividing the original video using the corresponding sliding windows, 32 frames are evenly sampled from each video segment to as the input image sequence of the subsequent classifier.

3.2 Extraction of coarse-proposals using a classifier

As the premise of action recognition, temporal action detection from undivided video is a fundamental and critical task in computer vision. With the advance of 3D-CNN [23], many

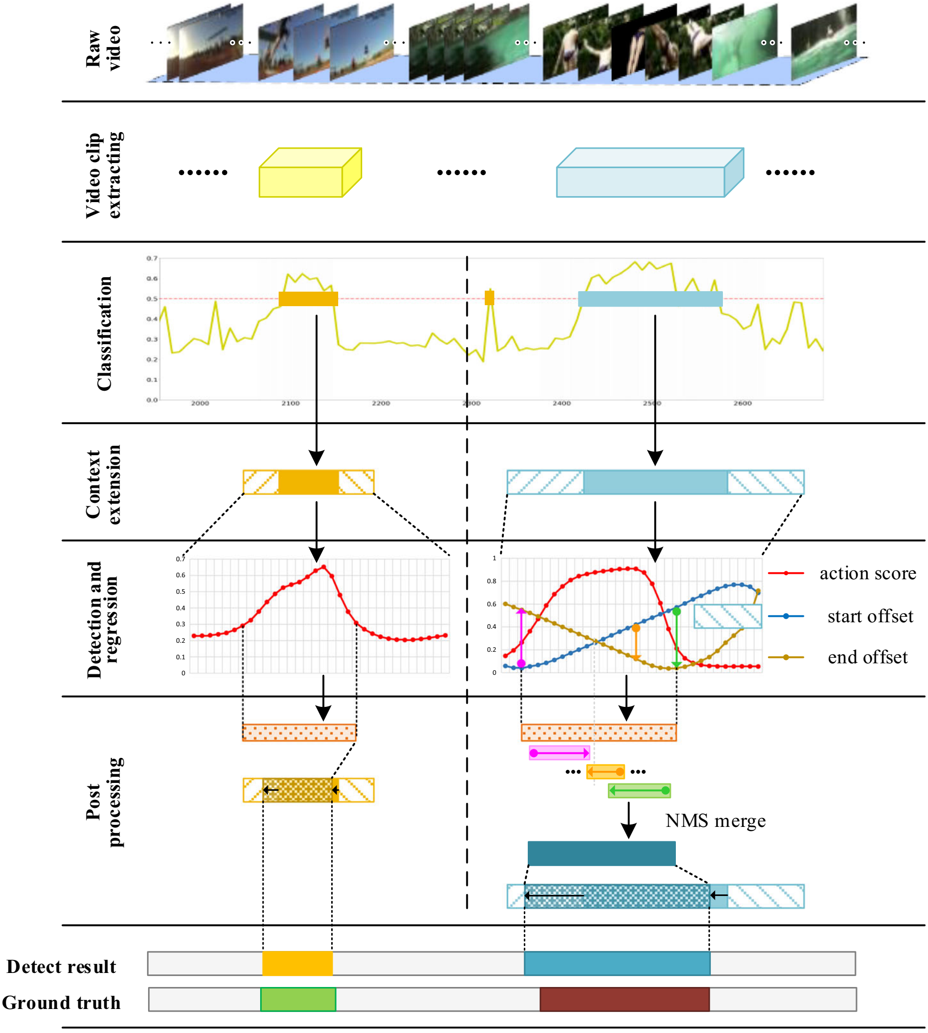


Fig. 1 The pipeline of coarse-to-fine temporal action detection method

temporal action detection methods are proposed in recent years. However, these methods related to 3D-CNN are computationally complex, which limits their application in the actual environment to a certain extent. Therefore, "how to detect arbitrary-length temporal action proposal efficiently and accurately" remains to be solved to a large extent. The basic idea of our proposed method is to first quickly find a continuous region with high action confidence, and then carry out accurate action detection in the extended trusted region.

To make a good trade-off between speed and performance, a light-weight temporal action classifier is used to judge the sampled video clip whether it is temporal action proposal or not. Because it is a lightweight network, ResNet-10 [16] is used in this paper as the backbone network of temporal action classifier. The classifier includes only two classes, one corresponding to action proposals and the other corresponding to non-action proposals. ResNet-10 architecture is constructed with very small feature sizes in each layer as given in Table 1, which results in less than 1 M (≈ 862 K) parameters. The classifier consists of an input layer, 5 convolutional layers, a pooling layer and an output layer. For training the classifier, we assign a binary class label (whether it belongs to an action) to each sampled video clip, and use standard softmax cross entropy loss. After confidence score is obtained by inputting sampled video clip to the classifier, binarization and watershed post-processing are performed to obtain the action candidate proposal result of the given sampled video segment.

The reason for adopting a lightweight 3D-CNN classifier based on ResNet-10 for rough action detection lies in the following two points. On the one hand, its backbone network is light and its execution efficiency is high enough. On the other hand, the 3D classifier makes full use of the temporal and spatial information of the video segments. Therefore, the proposed method can quickly recognize whether the video clip is an action or not.

3.3 Extraction of fine-proposals using a detector

Although the efficiency of the lightweight classifier is high enough, due to the limited accuracy of the lightweight classifier, the time boundary of the candidate action proposals extracted from the raw video is relatively rough. In order to solve this problem, we design a heavyweight detector to refine the time boundary of the candidate action proposal segment. On the one hand, the deeper video features extracted from the fine detector can improve the recognition of action categories; on the other hand, considering both boundary regression loss and category

Table 1 Classifier (ResNet-10) architectures

Layers	Building Blocks	Shape of the input
Conv1	3*7*7,64	3*32*112*112
Maxpool	3*3*3,64	16*16*56*56
Conv2_x	$\begin{bmatrix}3*3*3, 64\\3*3*3, 64\end{bmatrix}*1$	16*16*28*28
Conv3_x	$\begin{bmatrix}3*3*3, 128\\3*3*3, 128\end{bmatrix}*1$	16*16*28*28
Conv4_x	$\begin{bmatrix}3*3*3, 256\\3*3*3, 256\end{bmatrix}*1$	32*8*14*14
Conv5_x	$\begin{bmatrix}3*3*3, 512\\3*3*3, 512\end{bmatrix}*1$	64*4*7*7
FC	Global Average Pooling, Fc Layer with Softmax	128*2*4*4

loss in the target loss function can improve the accuracy of candidate proposal action positioning. This specially designed detector includes three steps: (i) the extended action proposal is first obtained by concatenating candidate action proposal and its temporal contexts; (ii) for the extended candidate action proposal, frame-level action recognition results, start and end offsets of extended action proposal are predicted through the heavyweight action detection network; (iii) different post-processing are performed on specific video clips with different scales, That is, when the candidate action proposal corresponds to small scale, the threshold segmentation and NMS merging is performed only for the frame-level recognition result to locate the action position in the extended candidate action proposal. However, when its size is large enough, two-stage cascading segmentation and NMS processing are performed on the outputs of the detector, that is, the recognition result is processed first, and then the offset results with the higher recognition score are processed by the same way. The reason why short video clips and long video clips are processed differently lies in the following considerations. The contribution of each frame in a short video to whether the video clip is an action segment cannot be ignored. However, in a long video, even if the action recognition score of a certain frame is high, the probability of the video segment being an action is not necessarily great. When there is a large deviation between a certain frame and its adjacent frames in the temporal action position estimation, the influence of this frame on the action detection can be reduced accordingly. Therefore, this paper considers both the recognition score and the time offset in the action detection of long video clips. Before running the CNN-based detector on the candidate proposals, we add a certain amount of context information around the candidate to further improve the accuracy of detection.

The network structure of the detector is shown in Fig. 2. As fine-grained information needs to be represented by more discriminative depth features, Resnet-50 is chosen as the backbone network of our action detector. The candidate action proposal is fed to the Resnet-50 network as input, and a 1024*N-dimensional feature map is obtained. Output layer is designed next to the feature layer, which consists of two branches, one is the classification branch and the other is the time offset branch, both for frame-level predictions. The classification branch is used to predict the frame level action probability, and the time offset branch is used to predict the distance of each frame relative to the start frame and the end frame of an independent action segment.

For training the detector, we first assign a binary classification label, a start boundary offset label and an end boundary offset label to each frame of the candidate action proposals. The classification label indicates whether the processed frame belongs to an action segment, being 1 for positive sample and 0 for negative sample. The start and the end boundary offset correspond to the Euclidean distance between the current frame and the start time and end time of the actual action segment, respectively.

A multitask loss function is designed to jointly train classification and time boundary regression.

$$L = L_{reg} + L_{cls}\cdot\lambda \tag{1}$$

Here$L_{reg}$is the loss for time boundary regression, $L_{cls}$is the loss for action / non-action classification, and$\lambda$is a hyperparameter. Since we need to generate accurate time boundary predictions for long-term and short-term action segments indiscriminately, the regression loss should be scale-invariant. Directly using L1 or L2 loss for regression would guide the loss bias towards longer action proposals, so we adopt the tIoU (temporal Intersection over Union) loss due to its invariance to objects of different scales.

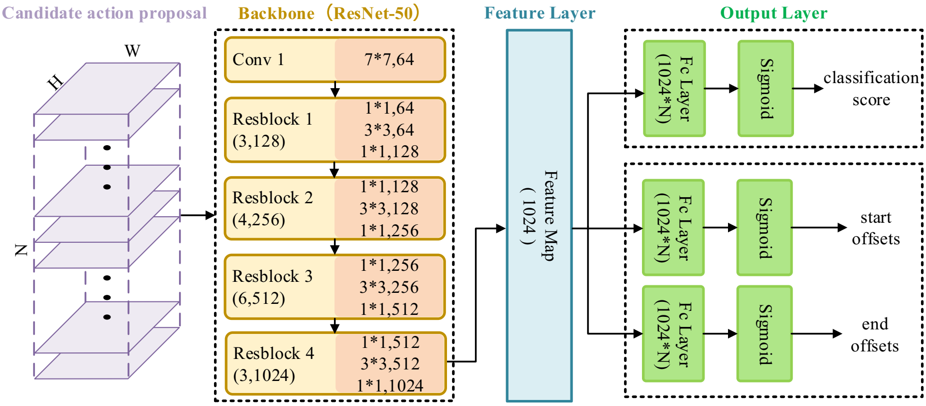


Fig. 2 The pipeline of coarse-to-fine temporal action detection method

$$L_{reg} = -\log tIoU\left(\widehat{R}, R^{*}\right) = -\log\frac{\left|\widehat{R}\cap R^{*}\right|}{\left|\widehat{R}\cup R^{*}\right|} \tag{2}$$

where$\widehat{R}$represents the predicted time boundary offset and$R^{*}$is the corresponding true value. The union part is:

$$\left|\widehat{R}\cup R^{*}\right| = \left|\widehat{R}\right| + \left|R^{*}\right| - \left|\widehat{R}\cap R^{*}\right| \tag{3}$$

3.4 Redundant proposals suppression post-processing

Around a ground truth action instance, multiple proposals may be generated with different temporal overlap. In order to accurately locate the temporal regions where actions take place in the untrimmed video, we need to delete the redundant action clips among the overlapping proposals.

By inputting the candidate action video clip to the heavyweight detector, a dense prediction channel of action/non-action is outputted to indicate the frame-wise confidence of being action or not. If NMS is only performed according to action score in a single candidate proposal, the action positioning error will inevitably occur when merging overlapping proposal segments. In order to reduce the action detection error in proposal-level action merging, when the overlap tIoU with two proposals is greater than a pre-fixed threshold, all frames in the two proposals are firstly filtered based on the action confidence score, and then locality-aware non-maximum suppression (NMS) [57] is performed on the filtered frames to suppress redundant results to obtain higher recall with fewer proposals. This step is recursively applied to the remaining proposals to generate final proposal set.

4 Experiments

To verify the effectiveness of our method, we tested the algorithm from four aspects: candidate action proposal generation, action detection, generalization and running efficiency. In the experiments, we call two models as coarser-classifier and finer-detector respectively.

4.1 Datasets and evaluation metrics

We conduct experiments on three large public benchmarks of temporal action localization: ActivityNet [3], THUMOS-14 [18] and MPII Cooking [35].

ActivityNet dataset is a large-scale action understanding data-set for proposal generation, temporal detection and action recognition. It contains 19,994 temporally annotated untrimmed videos with 200 action categories, which are collected from YouTube. The entire dataset is divided into training, validation and testing sets by the ratio of 2:1:1.

THUMOS-14 dataset is widely used in action detection and action recognition tasks. It contains 413 temporally annotated untrimmed videos with 20 action categories, and the average number of action instances per video is 15.5. Compared with ActivityNet dataset, THUMOS-14 dataset is smaller in scale, but its annotation is more accurate and intensive. In addition, the duration of action instance is from a few seconds to more than one hour, which makes it more challenging than ActivityNet dataset.

MPII Cooking [35] is a large, fine-grained cooking activities dataset. It records 44 videos with a total length of more than 8 h (881,755 frames) of 12 participants performing 65 different cooking activities, such as cut slices, pour, and spice. It contains 7 splits after performing leave-one-person-out cross validation. Each split uses 11 subjects for training, leaving one for validation.

In temporal action proposal generation task, average recall (AR) is usually used as evaluation metric. In addition, in order to evaluate the relationship between recalls and the number of proposals, the curve of average recalls with different average number of proposals (AR-AN) is drawn as an evaluation metric. Only when the prediction has the correct category and the tIoU with ground truth instance is greater than the tIoU threshold, it will be marked as correct. For temporal action detection task, mean Average Precision (mAP) is usually used as evaluation metrics. mAP computes the precision of each action class respectively. On ActivityNet dataset, we set tIoU thresholds {0.5, 0.75, 0.95}. On THUMOS-14 dataset, since the goal of temporal action detection task is to more accurately locate the action proposals and classify the action categories, we adopt the IoU thresholds {0.4, 0.5, 0.6, 0.7} for calculation of mAP. On MPII Cooking dataset, by following the same experimental setting reported in the literatures, the overlap threshold of tIoU is set to 0.5 for calculation of mAP.

4.2 Implementation details

4.2.1 Training

In the lightweight action classification network, we use the ResNet-10 to extract the frame-level action confidence score. In the generation process of video clip, multiscale sliding windows are used on the undivided video, and 32 frames are then evenly sampled from the extracted video clips as a training sample of the classifier. The sample is labeled according to its temporal overlap rate with the ground truth. When tIoU is greater than 0.7, the sample is labeled as 1 (being action clip), otherwise labeled as 0 (being background). In the training process, the batch size is 32, and a total of 3000 iterations are required. We use stochastic gradient descent (SGD) as the optimizer to train the classifier network with momentum as 0.9. On the THUMOS-14 dataset, the basic learning rate of the network is set to 0.01, the decay rate is set to 0.1, and the learning rate drops once every 850 iterations. On ActivityNet dataset, the base learning rate of the network is 0.01, the decay rate is 0.1, and the learning rate decreases once every 1500 iterations. On the MPII Cooking dataset, the basic learning rate of the network is set to 0.01, the decay rate is set to 0.1, and the learning rate drops once every 850 iterations.

In heavyweight action detection network, the feature extraction is based on ResNet-50. In addition to the class labels similar to the classifier, the training data labels also include start offset label and end offset label. Start offset is the absolute difference in the number of frames between each frame and the true start frame of the action segment. The end offset is the absolute difference in the number of frames between each frame and the true end frame of the action segment. We use SGD with momentum 0.9. On THUMOS-14 dataset, the network is trained for 6 K iterations with the learning rate of 0.01 and scaled down by 0.1 every 1.5 K iterations until the learning rate is less than 10^{-5}. On ActivtyNet dataset, the network is trained for 11 K iterations with the learning rate of 0.01 and scaled down by 0.1 every 2.6 K iterations until the learning rate is less than 10^{-5}. On MPII Cooking dataset, the network is trained for 6 K iterations with the learning rate of 0.01 and scaled down by 0.1 every 1.5 K iterations until the learning rate is less than 10^{-5}. The proposed method is implemented on PyTorch.

4.2.2 Inference

During testing phase, the initial proposals are generated by action classifier. The action classification network is used for generating the coarse proposals, which are sent to action detection network to carry out the boundary regression. To achieve precise localization results with higher tIoU thresholds, we empirically set the watershed threshold to 0.55 in action classifier and the NMS threshold to 0.35 in action detector on THUMOS-14 dataset. We empirically set the watershed threshold to 0.37 in action classifier and the NMS threshold to 0.25 in action detector on ActivtyNet dataset. And we empirically set the watershed threshold to 0.50 in action classifier and the NMS threshold to 0.40 in action detector on MPII Cooking dataset.

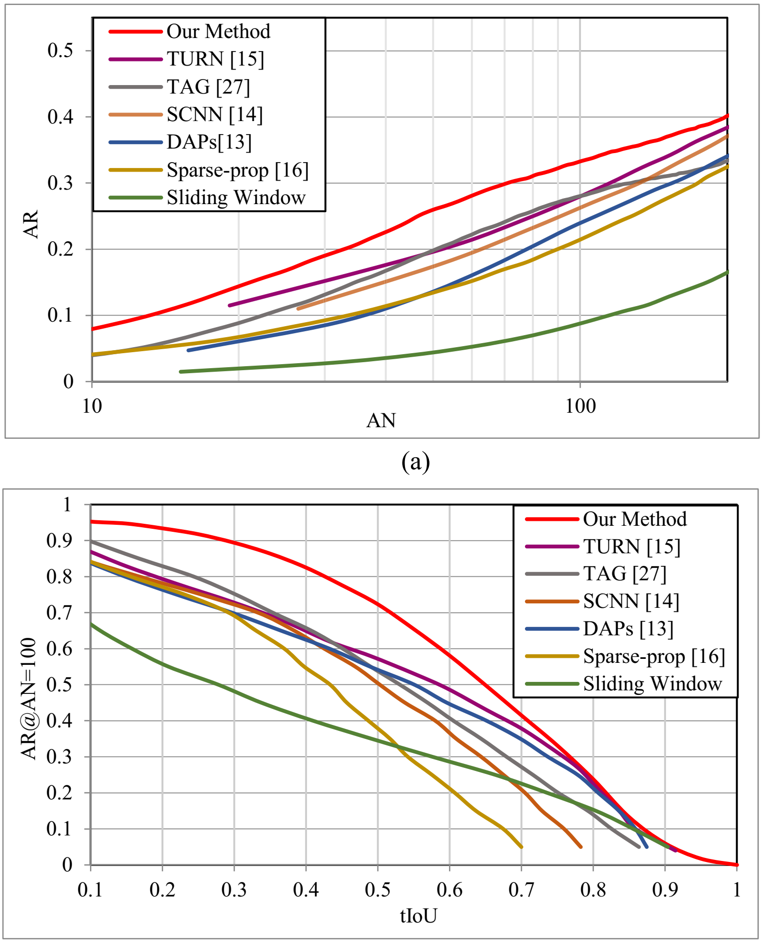
4.3 Action proposal generation

4.3.1 Compared with other proposal generation methods

The goal of our framework is to accurately detect action clips while optimizing the execution time; this is done by first taking advantage of a low-complexity a lightweight candidate action proposal extraction method to process the whole untrimmed video and eliminating non interesting regions (i.e. regions without actions), then using a CNN-based detector on regions of interest for an accurate action detection. For keeping the performance similar to that of the original CNN-based detection method, the probability of a real video clip being roughly classified as an action proposal should be as large as possible, and the number of generated action proposals should be as small as possible. In order to verify the effectiveness of our method, we compare our method with other proposal generation methods on average recall rate against average selected number of proposals (AR@AN), which is shown in Fig. 3(a) on THUMOS-14 dataset. As same in reference [2], the recall is averaged over multiple tIoU thresholds from 0.5 to 1. It can be seen from Fig. 3(a) that our method achieves the best AR@AN performance compared to several other action proposal generation methods. When AN is not greater than 100, the improvement of AR performance compared to other methods is particularly significant. This is because the lightweight classifier effectively deletes a large number of non-action video clips, after adding context information, the remaining small amount of video clips are only allowed to enter a heavyweight detector, thereby obtaining accurate action boundary. The AR-AN curve proves that although the number of retrieved proposals extracted by our algorithm is small, it maintains a fairly high average recall performance.

When AN is set to a fixed value of 100, the AR curve at different tIoU thresholds is drawn in Fig. 3(b), in which tIoU covers the range from 0.1 to 1.0. As can be seen from Fig. 3(b), the area under the AR@AN = 100 curve by our algorithm is the largest in the all methods. That is, our algorithm has the best AR performance under each tIoU. For example, when tIoU equals to 0.5, the AR of our algorithm is 0.72, which is 0.14, 0.17, 0.17, 0.22, 0.32, 0.37 higher than TURN [11], DAPs [7], TAG [47], SCNN [37], Sparse-prop [2] and Sliding-window method, respectively. The AR@AN = 100 curve further verifies the superior AR performance of our coarse-to-fine hierarchical temporal action detection when AN is set as a fixed value.

When AR is set to a fixed value, the greater the value of tIoU is, the higher the accuracy of the action positioning is, and vice versa. When the AR value is set to a fixed value of 0.5, the tIoU value between the retrieved proposals and the ground truth are 0.65, 0.59, 0.55, 0.53, 0.50, 0.43 and 0.28 by our method, TURN [11], DAPs [7], TAG [47], SCNN [37], Sparse-



(a)

(b)

Fig. 3 Comparison of our proposal generation method with other temporal proposal generation methods on THUMOS-14 dataset under two metrics: AR-AN and AR@AN-tIoU, (a) AR-AN, (b) AR@AN-tIoU

prop [2] and Sliding-window method, respectively. The maximum value of tIoU at a given AR verifies the superiority of the accuracy of the proposed method over other algorithms.

4.3.2 Generalization ability of proposals

Another important feature of temporal action proposal generation method is the ability to work on unseen action classes. We use the lightweight actionness classifier trained on ActivityNet to generate proposals on THUMOS-14 and ActivityNet. We report the average recalls with 10 and 100 seen classes and unseen classes on THUMOS-14 dataset and ActivityNet dataset, respectively. As shown in Table 2, The average recalls of unseen action classes on THUMOS-14 dataset and ActivityNet dataset is about 2.6% and 4.8% lower than that of seen action classes, respectively. This shows that our proposal generation method also works well on unseen activity classes. The experimental results clearly demonstrate the generalization capacity of our proposal scheme.

4.4 Action detection

After the candidate action proposals are generated by the lightweight classifier, we run the heavyweight detector on the candidates, which outputs all the action classes and the start time and end time of the candidate action clip; and then we execute NMS to delete the redundant action clips among the overlapping sliding windows, so as to accurately locate the temporal regions where actions take place in the untrimmed video. We utilize mean Average Precision (mAP) as the criterion to evaluate the performance of our proposed model on the task of temporal action detection.

Table 3 shows the comparison of our method with the state-of-the-art temporal action detection methods on THUMOS-14 dataset, which are evaluated via mAP under the tIoU ranging from 0.4 to 0.7. It can be seen from Table 3 that our method outperforms other methods except that it is slightly worse than BSN. After frame-level visual feature is extracted by a two-stream network, BSN adopts “local to global” fashion to generate proposals. It is precisely because of this “local-to-global” frame-level fine-grained fashion that more flexible action time duration can be generated, which makes the performance of BSN more superior than other algorithms. However, the proposed method directly operates at the window-level,

Table 2 Comparison of our proposal generation method with other proposal generation methods on the seen action classes and unseen action classes. We report the average recalls with 100 proposals (AR@100) on THUMOS-14 dataset and ActivityNet dataset

THUMOS-14 dataset		
Method	Seen(10 classes)	Unseen(10 classes)
TAG [47]	46.6	28.3
MPRN [55]	50.12	42.34
Ours	**53.06**	**48.23**
ActivityNet dataset		
Method	Seen(100 classes)	Unseen(100 classes)
CTAP [12]	74.06	72.51
MPRN [55]	75.86	73.07
Li et al. [23]	76.42	74.15
Ours	**76.25**	**74.19**

avoiding BSN's frame-level depth feature extraction and temporal evaluation, so can be run with much higher efficiency.

Table 4 shows the action detection comparison results on validation set of ActivityNet in terms of mAP when tIoU equals 0.5, 0.75 and 0.95, respectively. As shown in Table 4, our method obtains competitive temporal action detection results compared with other methods. Especially when tIoU is set to 0.5, the average mAP performance of our method is improved by 5.2% compared with the BSN method. As the overlap threshold increases, the advantages of our algorithm will decrease to a certain extent. This is because our fast detection mechanism adopts window-level coarse granularity, which inevitably leads to the decline of the accuracy of action detection. Bur fortunately, our method achieves a good tradeoff between performance and efficiency, and the positioning accuracy can be improved by extending the range value of the window scale.

On the MPII Cooking dataset, we follow the same experimental setting of tIoU as that in [34, 40, 43, 58], which is set to 0.5. The temporal action localization results are listed in Table 5. Among the comparison algorithms, the sliding window method is treated as the baseline method. As can be seen in Table 5, our method improves the baseline by 12.5%, and performs at least 4.7% better than the state-of-the-art methods. A series of mechanisms, such as the generation of video clips with different lengths, the context extension of candidate proposals, and the frame-level NMS merging between overlapping proposals, make our method perform well on fine-grained (MSRII Cooking) for temporal action localization.

In Fig. 4, we show the average classification accuracy (Average Precision, AP) and mAP of 20 categories of the proposed algorithm and the other two algorithms in the THUMOS14 dataset when overlap threshold tIoU equals 0.5. As shown in Fig. 4, our method is better than the other two methods in most of the behavior categories. The behavior proposal generation method in this paper can produce more accurate proposals and thus improve the accuracy of

Table 3 Action detection results on THUMOS-14 in terms of mAP when overlap threshold tIoU equals 0.4, 0.5 and 0.6, 0.7, respectively

Methods	0.4	0.5	0.6	0.7
Wang et al. [45]	11.7	8.3		
Oneata et al. [31]	20.8	14.4	8.5	3.2
Richard et al. [34]	23.2	15.2	–	–
Dong et al. [6]	28.0	20.5	11.7	6.0
SCNN [37]	28.7	19.0	10.3	5.3
CDC [38]	29.4	23.3	13.1	7.9
TURN [11]	33.2	25.6	14.6	7.7
TAG [47]	39.8	28.2	–	–
CBR [10]	41.3	31.0	19.1	9.9
BSN [25]	**45.0**	**36.9**	**28.4**	**20.0**
Li et al. [22]	38.7	28.7	18.9	8.1
Liu et al. [26]	40.6	30.1	–	–
Zheng et al. [55]	37.6	29.9	22.9	14.7
Wu et al. [46]	29.7	20.6	–	–
Kim et al. [19]	39.4	27.7	16.3	9.7
Yao et al. [50]	40.1	29.6	–	–
Song et al. [41]	32.6	26.4	–	–
Li et al. [23]	–	32.8	–	–
Our Method	40.6	34.1	26.4	18.11

Table 4 Action detection results on validation set of ActivityNet in terms of mAP when tIoU equals 0.5, 0.75 and 0.95, respectively

Methods	0.5	0.75	0.95
Wang et al. [45]	42.5	2.9	0.1
BSN [25]	46.5	**29.9**	**8.0**
TAG [47]	44.1	24.1	5.0
Shen et al. [36]	36.9	23.1	3.4
Li et al. [22]	42.3	24.9	5.2
Yao et al. [50]	37.9	–	–
Our Method	**51.7**	27.8	3.2

behavior recognition. But for the long jump video, the performance of our algorithm is weaker than SCNN [37], which is due to the following points. In the long jump video, in addition to the athletes, there are also judges, coaches and many audiences appearing in the scene at the same time. Athletes may maintain a state of preparation for a period of time before taking off, resulting in a rather vague behavior boundary. The fine-stage proposal detector is based on coarse-stage proposal classifier, so our method does not work well for the proposals with vague boundary because of the non-frame level granularity of the coarse classifier. Fortunately, the tradeoff between efficiency and performance can be achieved by further adjusting the granularity size.

In addition, we further visualize the proposals detected by our method on THUMOS-14 dataset. We randomly selected one video from it and retrieved proposals with the top-1 predicted scores from videos. Figure 5 provides an example of the generated proposal by our method, in which the red box, blue line and green line represent the ground truth, first-level proposal and the second-level proposal, respectively. It can be seen from Fig. 5 that although the proposal obtained by the coarse classifier is somewhat different from the ground truth, it is completely contained within the time period of the ground truth, which proves the classifier a certain effect on generating the action proposals. After further refinement of the first-level action proposal by fine detector, the final generated action proposal is basically the same as the ground truth, which verifies a superior proposal-retrieval performance of the detector.

4.5 Execution efficiency

We define the complexity of the CNN-based detection and our two-stage method, denoted by C_{dec} andC_{our}, respectively, which are as

$$C_{dec} = \sum_{r=1}^{R} \left(L/32^{\mathrm{r}}\right) \times o_{\mathrm{Res50}} \tag{4}$$

$$C_{our} = \sum_{r=1}^{R} \left(L/32^{\mathrm{r}}\right) \times p \times \left(o_{\mathrm{Res10}} + o_{\mathrm{Res50}}\right) \tag{5}$$

Here, $L$is the total number of frames of the video to be processed, $r$and$R$is the number of scale levels and the total scale levels in generating of video clips by sliding-window mechanism, respectively. $o_{\mathrm{Res50}}$and o_{Res10} is the complexity of executing the detector and the coarse

Table 5 Action detection results on the MPII Cooking dataset in terms of mAP when tIoU equals 0.5

Model	0.5
Sliding Window	7.9
Van et al. [43]	13.1
Richard et al. [34]	14.0
Zhu et al. [58]	14.9
Song et al. [40]	15.7
Our Method	**20.4**

classifier, respectively.$p$is the ratio of candidate action proposals in the total video segments, which equals to $1 - tIOU$.

Our aim is to choose a low computational complexity algorithm for relevant video clips extraction so that$C_{our}$is (much) lower thanC_{dec}while keeping the same performance as the original CNN-based detection method. For reducing the overall complexity of the temporal action location algorithms, first-stage classifier is firstly proposed to determine relevant video clips and second-stage detector is then used to perform the detection task on these video clips.

Normally, the parameter FLOPs (floating point operations) is used to measure the complexity of the CNN network model. In our hierarchical mechanism, Single forward$o_{\mathrm{Res10}}$and$o_{\mathrm{Res50}}$are calculated with FLOPs metric, which is about 90.07 FLOPs and 160.21 FLOPs, respectively. In the experiment, when $tIOU$equals 0.7, the complexity of performing action detection on each sliding window is saved by 85.13 FLOPs by using the two-stage scheme.

We further investigate the execution performance of our two-level temporal action detection method. Experimental tests are performed on NVIDIA 1660ti. The extraction of first-level action proposals runs on average at 70fps, and the execution cost includes the reading video stream with sliding windows of five-scales and running the ResNet10-based lightweight classifier. Using the ResNet50 as the backbone network of the proposal detector, the generation of second-level action proposals runs on average at 43fps, including post-processing.

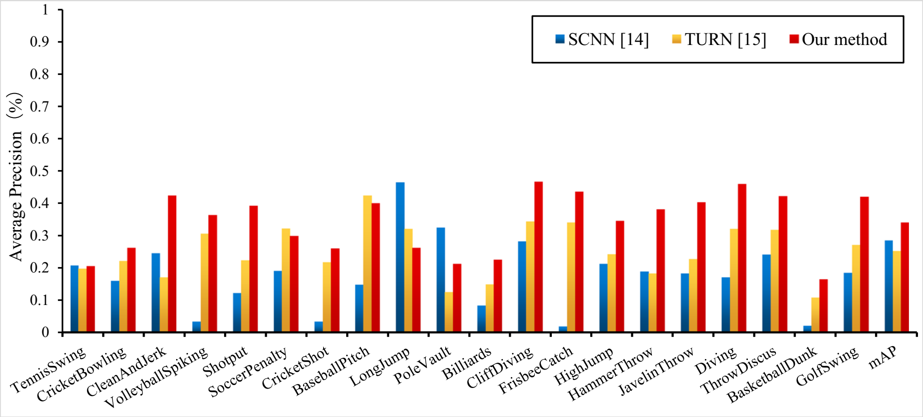


Fig. 4 The average precision for each action category on THUMOS-14. The results are calculated with the official toolkit

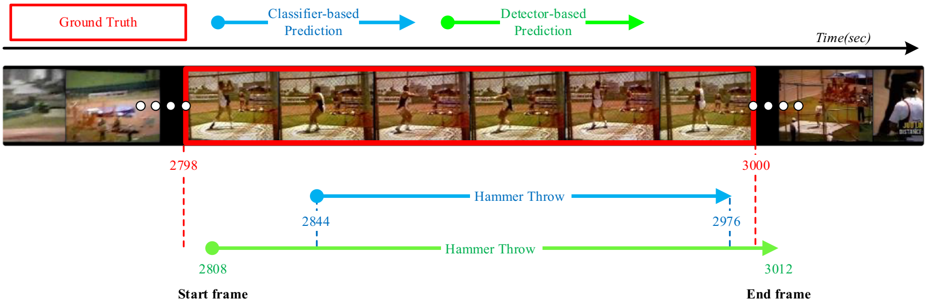


Fig. 5 Qualitative results on THUMOS-14

5 Conclusions

In this paper, we present a coarse-to-fine temporal action detection method, which progressively localizes action instances occurring at an untrimmed video. The proposed architecture consists of two models: (1) A classifier which is a lightweight CNN architecture to determine whether the sampled video clip may be a candidate action proposal and (2) a detector which is a deep CNN to regress the boundaries and identify the action category simultaneously. Experimental results show that the proposed method has achieved a detection performance that is competitive with the state-of-the-arts on three challenging data sets while maintaining real-time performance.

Although different from the typical "proposal generation first and then classification" method, as a "classification first and then detection" method, our method still requires two separate stages to obtain the final action instances. The future study is to streamline the experimental steps and try to obtain more accurate timing boundaries.

Author contributions Fan Zhao directed the experiment, conceived the study, wrote and revised the manuscript. Wen Wang and Yu Wu wrote the program code and analyzed the experimental data. Kaixuan Wang collected the data and verified experiments. Xiaobing Kang proofread the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding This work was supported by Natural Science Basic Research Project of Shaanxi Province of China (Grant No. 2021JLM-59), National Natural Science Foundation of China (Grant No. 61671376).

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Buch S, Escorcia V, Ghanem B, Li F, Niebles J (2017) End-to-end, single-stream temporal action detection in untrimmed videos. In Proceedings of the British Machine Vision Conference
2. Caba F, Carlos J, Ghanem B (2016) fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1914-1923
3. Caba Heilbron F, Escorcia V, Ghanem B, Carlos J (2015) Activitynet: a large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–970
4. Carreira J, Zisserman A (2017) Quo vadis, action recognition? A new model and the kinetics dataset. In: proceedings of the IEEE conference on computer vision and pattern recognition, pp 6299-6308
5. Chen G, Zhang C, Zou Y (2020) AFNet: temporal locality-aware network with dual structure for accurate and fast action detection. IEEE Trans Multimedia 23:2672–2682
6. Dong P, Zhu L, Zhang Y (2019) Category-level multi-attention based boundary refinement for action detection. IEEE Int Conf Image Process. 230-234
7. Escorcia V, Heilbron F, Niebles J, Ghanem B (2016) Daps: deep action proposals for action understanding. In: European conference on computer vision, pp. 768–784
8. Fayyaz M, Gall J (2020) SCT: set constrained temporal transformer for set supervised action segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 501–510
9. Gaidon A, Harchaoui Z, Schmid C (2013) Temporal localization of actions with actoms. In: Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 2782–2795
10. Gao J, Yang Z, Nevatia R (2017) Cascaded boundary regression for temporal action detection. In: Proceedings of the British Machine Vision Conference
11. Gao J, Yang Z, Chen K, Sun C, Nevatia R (2017a) Turn tap: temporal unit regression network for temporal action proposals. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3628–3636
12. Gao J, Chen K, Nevatia R (2018) Ctap: complementary temporal action proposal generation. In: Proceedings of the European conference on computer vision, pp. 68–83
13. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 144–1448
14. Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Patt Anal Mach Intell:1137–1149
15. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587
16. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778
17. Jain M, Gemert J, Jegou H, Bouthemy P, Snoek C. (2014) Action localization with tubelets from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 740–747
18. Jiang Y, Liu J, Roshan A, Toderici G, Laptev I, Shah M, Sukthankar R (2014) THUMOS Challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14
19. Kim J, Heo J (2019) Learning coarse and fine features for precise temporal action localization. IEEE Access 7:149797–149809
20. Kim J, Hong G, Kim B, Dogra D (2018) DeepGesture: deep learning-based gesture recognition scheme using motion sensors. Displays. 38-45
21. Kim J, Kim B, Roy P, Jeong D (2019) Efficient facial expression recognition algorithm based on hierarchical deep neural network structure. IEEE Access. 41273-41285
22. Li N, Guo H, Zhao Y. (2018) Active temporal action detection in untrimmed videos via deep reinforcement learning. IEEE Access. 59126-59140
23. Li T, Bing B, Wu X (2020) Boundary discrimination and proposal evaluation for temporal action proposal generation. Multimed Tools Appl 80(2):2123–2139
24. Lin T, Zhao X, Shou Z (2017a) Single shot temporal action detection. In: proceedings of the 25th ACM international conference on multimedia. ACM. 988-996
25. Lin T, Zhao X, Su H, Wang C, Yang M (2018) Bsn: boundary sensitive network for temporal action proposal generation. In: proceedings of the European conference on computer vision (ECCV), pp 3-19
26. Liu J, Wang C, Liu Y (2019) A novel method for temporal action localization and recognition in untrimmed video based on time series segmentation. IEEE Access. 135204-135209
27. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg C (2016) SSD: single shot multiBox detector. In: Proceedings of the European Conference on Computer Vision, pp. 21–37

28. Long F, Yao T, Qiu Z, Tian X, Luo J, Mei T (2019) Gaussian temporal awareness networks for action localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 344–353
29. Mettes P, Van Gemert J, Cappallo S, Mensink T, Snoek C (2015) Bag-of-fragments: selecting and encoding video fragments for event detection and recounting. In: proceedings of the 5th ACM on international conference on multimedia retrieval, pp 427-434
30. Nguyen P, Liu T, Prasad G, Han B (2018) Weakly supervised action localization by sparse temporal pooling network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6752–6761.
31. Oneata D, Verbeek J, Schmid C (2013) action and event recognition with fisher vectors on a compact feature set. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1817-1824
32. Pedersoli M, Vedaldi A, Gonzalez J, Roca X (2015) A coarse-to-fine approach for fast deformable object detection. Pattern Recogn 48:1844–1853
33. Redmon J, Divvala, S, Girshick, R, Farhadi, A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788
34. Richard A, Gall J (2016) Temporal action detection using a statistical language model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3131–3140
35. Rohrbach M, Amin S, Andriluka M, Schiele B (2012) A database for fine grained activity detection of cooking activities. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1194–1201
36. Shen Z, Wang F, Dai J (2020) Weakly supervised temporal action localization by multi-stage fusion network. IEEE Access. 17287-17298
37. Shou Z, Wang D, Chang S (2016) Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1049–1058
38. Shou Z, Chan J, Zareian A, Miyazawa K, Chang S (2017) Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5734–5743
39. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos arXiv: 1406.2199
40. Song H, Wu X, Zhu B, Wu Y, Chen M, Jia Y (2019) Temporal action localization in untrimmed videos using action pattern trees. IEEE transactions on multimedia. 717-730
41. Song H, Tian L, Li C (2020) Action temporal detection method based on confidence curve analysis. Multimed Tools Appl 79:34471–34488
42. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4489–4497
43. Van G, Jain M, Gati E, Snoek C (2015) APT: action localization proposals from dense trajectories. In British Machine Vision Conference
44. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision, pp. 3551–3558
45. Wang L, Qiao Y, Tang X (2014) Action recognition and detection by combining motion and appearance features. THUMOS14 Action Recogn Chall. 1(2):2
46. Wu Y, Yin J, Wang L, Liu H, Dang Q, Li Z, Yin Y(2018) Temporal action detection based on action temporal semantic continuity. IEEE Access, pp 31677-31684
47. Xiong Y, Zhao Y, Wang L, Lin D, Tang X (2017) A pursuit of temporal accuracy in general activity detection. arXiv:170302716
48. Xu M, Gao M, Chen Y, Davis L, Crandall D (2018) Temporal recurrent networks for online action detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5532–5541
49. Yang X, Yang X, Liu M, Xiao F, Davis L, Kautz J (2019) STEP: Spatio-temporal progressive learning for video action detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, In, pp 264–272
50. Yao G, Lei T, Liu X, Jiang P (2018) Temporal action detection in untrimmed videos from fine to coarse granularity. Appl Sci 8
51. Yeo W, Heo Y, Choi Y, Kim B (2020) Place classification algorithm based on semantic segmented objects. Appl Sci 10(24):9069
52. Yeung S, Russakovsky O, Mori G, Fei L (2016) End-to-end learning of action detection from frame glimpses in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2678–2687

53. Yuan Y, Chu J, Leng L, Miao J, Kim B (2020) A scale-adaptive object-tracking algorithm with occlusion detection. EURASIP J Image Video Process 2020:1–15
54. Zhao Y, Xiong Y, Wang L, Wu Z, Tang X, Lin D (2017) Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2914–2923
55. Zheng J, Chen D, Hu H (2019) Multi-scale proposal regression network for temporal action proposal generation. IEEE Access 7:183860–183868
56. Zheng Y, Huang D, Liu S, Wang Y (2020) Cross-domain object detection through coarse-to-fine feature adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 13766–13775
57. Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, Liang J (2017) East: an efficient and accurate scene text detector. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 5551–5560.
58. Zhu Y, Newsam S (2017) Efficient action detection in untrimmed videos via multi-task learning. In 2017 IEEE winter conference on applications of computer vision. 197-206

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.