Zhu *et al.*
EURASIP Journal on Image and Video Processing (2022) 2022:10
https://doi.org/10.1186/s13640-022-00588-4

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Weakly supervised spatial–temporal attention network driven by tracking and consistency loss for action detection

Jinlei Zhu*, Houjin Chen, Pan Pan and Jia Sun

*Correspondence:
19111077@bjtu.edu.cn

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

Abstract

This study proposes a novel network model for video action tube detection. This model is based on a location-interactive weakly supervised spatial–temporal attention mechanism driven by multiple loss functions. It is especially costly and time consuming to annotate every target location in video frames. Thus, we first propose a cross-domain weakly supervised learning method with a spatial–temporal attention mechanism for action tube detection. In source domain, we trained a newly designed multi-loss spatial–temporal attention–convolution network on the source data set, which has both object location and classification annotations. In target domain, we introduced internal tracking loss and neighbor-consistency loss; we trained the network with the pre-trained model on the target data set, which only has inaccurate action temporal positions. Although this is a location-unsupervised method, its performance outperforms typical weakly supervised methods, and even shows comparable results with some recent fully supervised methods. We also visualize the activation maps, which reveal the intrinsic reason behind the higher performance of the proposed method.

Keywords: Weakly supervised learning, Consistency loss, Spatial attention, Channel attention

1 Introduction

In video processing tasks, motion gives rise to blurring, the camera is often defocused, and the video may be affected by a variety of poses or serious occlusion; therefore, the temporal information plays an important role.

It is especially costly and time consuming to annotate every target location in video frames when the network works with a spatial–temporal attention mechanism. The performance is also important in particular applications. This study proposes a human action tube detection method based on a location weakly supervised spatial–temporal attention mechanism.

As shown in Fig. 1, the source domain has accurate person location and action category labels in the frames, and the target domain only has inaccurate action temporal position labels in the video.

Fig. 1 Source domain has accurate person location and action category labels in the frames, and the target domain only has inaccurate action temporal position labels in the video.

The main contributions of this paper include two points. (1) On the source data set, the manuscript constructs a new multi-loss spatiotemporal attention convolution network based on the source data set, which has target location and classification annotation. (2) In the target domain, the manuscript introduces the internal tracking loss and neighborhood consistency loss. The pre-training model is used to train the target data set, and there are only inaccurate action time positions. Although this is a location-unsupervised classification-supervised method, the mAP performance outperforms typical weakly supervised methods, and even shows comparable results with some recent fully supervised methods.

Since the proposed method uses pre-trained model and amount of weakly labeled data in the target domain, it is a typical weakly supervised learning method. The basic idea of the method is as the follows:

First, we introduce a novel location weakly supervised learning network model with a spatial–temporal attention mechanism for action tube detection. The framework structure clearly differs from the state-of-the-art methods.

Second, we introduce an internal tracking loss and neighbor-consistency loss for weakly supervised learning based on video sequences for which only the action classification temporal label is needed. This is the first study on tracker and consistency loss applied in location weakly supervised situations with a spatial–temporal–attention mechanism for action tube detection.

Third, we also visualize the activation maps, which reveal the intrinsic reason behind the higher performance of the proposed method.

2 Related works

Video processing methods have progressed through high-efficiency coding [1], detecting and object tracking [2], image retrieval [3], image enhancement [4, 5] and image compositing [6] in many applications. Many supervised methods exist in the action detection field. Popular detection methods such as YOLO [7] and SSD [8] are mainly used in representative multi-scale end-to-end models for static images. Considering the importance of temporal information, Trans [9] first proposed a C3D model that introduced local connection and weight sharing features from a 2D convolution to video sequence processing. Although the calculation parameters of U-Net [10] based on 3D-CNN are

relatively large, the performance of video processing was greatly improved compared with R-CNN [11, 12]. I3D [13] uses a dual stream fusion model structure in which 2D and 3D convolutions are fused to implement the migration of Image-Net and other static image data to a 3D video stream processing model. Sun [14] decomposed a 3D convolution into a 2D convolution in the spatial direction and a 1D convolution in the time direction. This notably improves the computational efficiency; however, massive iterative training on video data is still required. To reduce the computational complexity, P3D [15] combines three different module structures. In ResNet(2+1)D [16], new convolution kernels were explored and the C3D model was optimized in terms of parameters and running speed. Video-based 3D multi-scale detection [17, 18] has been widely used in video target recognition, and many open-source projects have validated its performance. Nevertheless, the algorithms are significantly affected by background factors owing to the lack of target focus. With the development of deep learning technology, multi-scale features and attention mechanisms of video were considered in videos for various applications. Popular attention mechanisms [19–21] are particularly important for streaming data processing in the machine-learning field, for example, task-adaptive attention method [22] used in image captioning and self-attention and multi-feature fusion method [23] used in face recognition. Inspired by human vision, the Institute for Human-Machine Communication from Munich University Germany proposed a fast and real-time video action detection method (You Only Watch Once, YOWO) [24], which achieves the highest efficiency at present. It introduced a target attention mechanism based on the video keyframe in the 2D-3D fusion model through a single-stage network. This constitutes the fundamental advantage of previous research results.

There are also some weakly supervised or unsupervised studies in this field. UntrimmedNets [25] introduced a classification module for predicting the classification score for each snippet, and a selection module to select relevant video segments. In addition, STPN [26] added sparsity loss and class-specific proposals. AutoLoc [27] introduced the outer-inner contrastive loss to effectively predict temporal boundaries. W-TALC [28] and Islam and Radke [29] incorporated distance metric learning strategies, and proposed a novel average aggregation module and latent discriminative probabilities to reduce the difference between the most salient regions and the others. TSM [30] modeled each action instance as a multi-phase process to effectively characterize action instances. WSGN [31] assigned a weight to each frame prediction based on both local and global statistics. DGAM [32] used a conditional variational auto-encoder to separate the attention, action, and non-action frames. CleanNet [33] introduced an action proposal evaluator that provides pseudo-supervision by leveraging the temporal contrast in snippets. 3C-Net [34] adopted three loss terms to ensure separability, enhance discriminability, and delineate adjacent action sequences. Moreover, BaS-Net [35] and Nguyen et al. [36] modeled background activity by introducing an auxiliary background class. However, none of these approaches explicitly resolve the issue of modeling an action instance in its entirety. Nanan [37] proposed a spatial-channel filter, and Liu et al. [38] proposed a multi-branch network in which each branch predicts distinctive action parts. HAM-Net [39] hides the most discriminative parts of a video instead of random parts. Our method includes a novel location-interactive weakly supervised learning network model with a spatial–temporal attention mechanism for action tube detection in which

an internal interactive location tracker and consistency loss is used for weakly supervised learning based on video sequence for which only the action classification temporal label is needed.

3 Methodology

3.1 Framework overview

The motivation of this study is to propose an attention network with fewer object bounding box annotations while still achieving comparable results with some recent fully supervised methods. The classification attention maps may be disturbed by the moving background objects, some input data can be predicted well while others are poor, but we cannot decide in advance which video clip or keyframe to choose as input. Therefore, to enhance the robustness of detection, the network need to filter the noise by tracking the objects to see if they exist continuously and always have high confidence value in the previous frames. The overall framework is shown in Fig. 2.

1) Overall framework overview. According to Fig. 2, in source domain, we trained the newly designed multi-loss spatial–temporal attention–convolution network on the first data set, which has both location and classification annotations. In target domain, we introduced an internal tracking loss and neighbor-consistency loss for weakly supervised learning based on video sequence for which only action classification temporal labels are needed and trained the network with the pre-trained model on the second data set, which only has classification annotations. To ensure the continuity of the target in the video sequence, tracking regularization loss is calculated by a tracker between the tracking location and

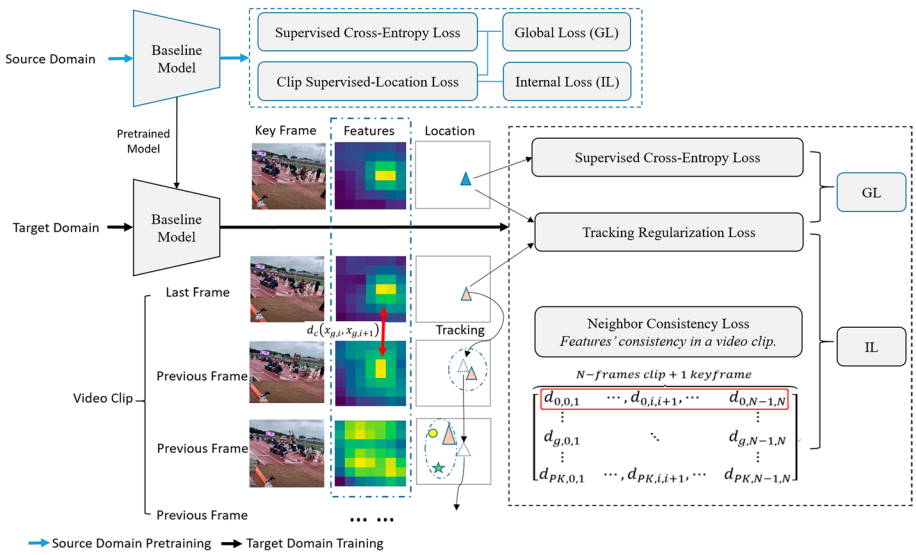

Fig. 2 Overall framework. In source domain, we trained the network on the first data set, which has both location and classification annotations. In target domain, we trained the network with the pre-trained model on the second data set, which only has action classification temporal annotations. To ensure the continuity of the target in the video sequence, Tracking-regularization loss is calculated by a tracker between the tracking location and network's predicted location. The neighbor-consistency loss makes the features of objects more closer between neighbors in the video

network's predicted location. Intuitively, the features' cosine distance between the neighbors is closer in the same video clip, so we introduce neighbor consistency loss in the model.

2) Baseline framework overview. As shown in Fig. 3, there are four branches: the branch no. 2 adopts a spatial attention mechanism for the object location in video frames, and branch no.3 uses a channel attention mechanism to fuse the previous two network branches to obtain the total loss. In branch No.4, The internal loss function focuses on the loss between the network's predicted location and the ground truth based the video sequence.

3.2 Baseline network definition

Referring to Figs. 2 and 3, suppose that a video sequence is an input to the 3D-CNN network, and the original video is sampled in time as

$$X = U\{x(t_0),\ x(t_1),\ \ldots,\ x(t_{N-1})\} \tag{1}$$

where X denotes the clip of video, $x(t)$ is a frame of the video, U means that X consists of the set of frames, and the range of sampling time is $[t_0,\ t_{N-1}]$.

The clips are fed into 3D convolutional network such as *3D-ResNeXt-50* and *3D-ResNeXt-34* [40] and the outputs

$$S_{50} = 3D\_ResNeXt\_34(X) \tag{2}$$

$$S_{101} = 3D\_ResNeXt\_50(X) \tag{3}$$

where the ResNeXt is used to verify our model, and other 3D Convolutional Network backbones can also be used here. Referring to the network branch no. 2 which focuses on the object location in the video sequence, squeezing the tensor S_{50} to the tensor F_{01}:

$$S_{50} \rightarrow F_{00} \in R^{(N\times D^{'})\times H^{'}\times W^{'}} \rightarrow F_{01} \in R^{C^{''}\times H^{'}\times W^{'}} \tag{4}$$

where $(N \times D^{'}) \times H^{'} \times W^{'}$ is the shape of the tensor F_{00} which has N-Frames feature-groups, each group has $D^{'}$ features, and each feature is the size of $H^{'} \times W^{'}$, and $C^{''} \times H^{'} \times W^{'}$ is the shape of the tensor F_{01}.

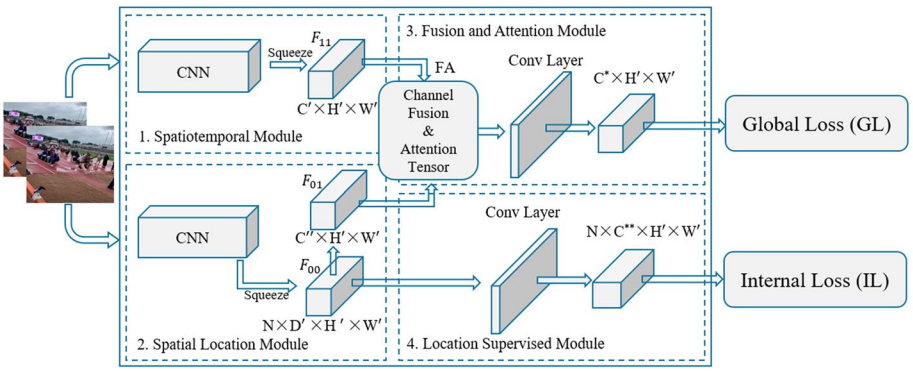

Fig. 3 Baseline framework. The internal loss function focuses on the loss between the network's predicted location and the ground truth locations based on the video sequence

Referring to the network branch no. 1, It is concerned with the classification of action tubes, we further squeeze the tensor S_{101} to the tensor F_{11}:

$$S_{101} \rightarrow F_{11} \in R^{C^{'} \times H^{'} \times W^{'}} \tag{5}$$

where $C^{'} \times H^{'} \times W^{'}$ is the shape of the tensor F_{11}. Since F_{01} and F_{11} have the same feature map dimension $H^{'} \times W^{'}$, so they can be concatenated as the follows:

$$\{F_{01},\ F_{11}\} \rightarrow FA \in R^{\left(C^{'}+C^{''}\right)} \times H^{'} \times W^{'} \tag{6}$$

The network branch no. 2 only focused on the object location in the video sequence, it is referenced by the internal loss function marked as *IL*. The branch no. 1 mainly considers for video object behavior classification, and it is referenced by the global loss function marked as *GL*. Therefore, the network parameters can be learned like:

$$\theta_j = \begin{cases} \theta_j - \alpha(t)\frac{\partial IL(\theta)}{\partial \theta_j}, & \text{if } \theta_j \epsilon \text{Branch 2 or 4} \\ \theta_j - \lambda\alpha(t)\frac{\partial GL(\theta)}{\partial \theta_j}, & \text{if } \theta_j \epsilon \text{Branch 1 or 3} \end{cases} \tag{7}$$

where θ_j denotes the trainable parameter of the network model. We choose different loss functions according to the network branches, $\alpha(t)$ is the learning rate function, and λ is a hyper-parameter.

As shown in Fig. 4, we use the Gram matrix in the neural network to solve the fusion problem. Here, the implementation process of CFAM is simplified as follows:

$$\begin{aligned} &FA \in R^{(C'+C'')\times H'\times W'} \rightarrow FB \in R^{C\times H'\times W'} \\ &\rightarrow FC \in R^{C\times H'\times W'} \rightarrow FD \in R^{C*\times H'\times W'} \end{aligned} \tag{8}$$

where $C = C^{'} + C^{''}$, *FA* is the result of simply concatenating features of network branch No.1 and network branch No.2, *FB* is the mapping feature after 2-layer convolution, the Gram matrix transformer is used between *FB* and *FC*, and *FD* is the mapping feature of *FC* after 2-layer convolution. C^* is the final number of features.

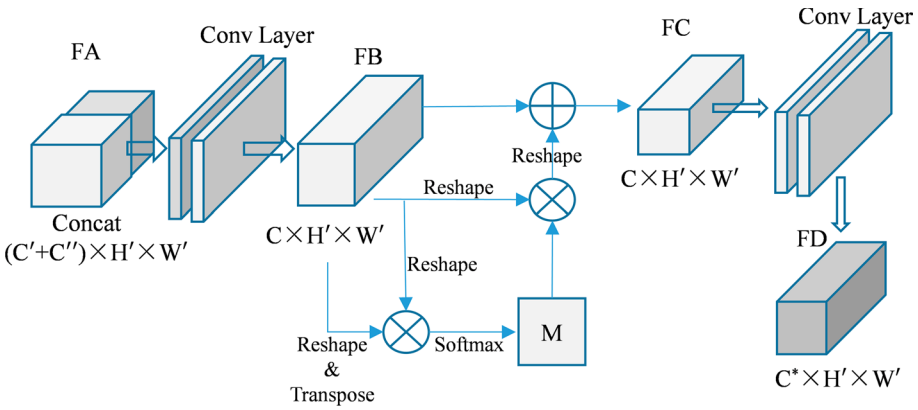

Fig. 4 It is about the details of branch no.3 shown in Fig. 3. We use Gram matrix between FB and FC. This figure corresponds to the formula 8 and 9

By squeezing FB in the directions $H^{'}$ and $W^{'}$, we then obtain the feature FF, $FF \in R^{C\times D}$ where $D = H^{'} + W^{'}$, and the transformer between FB and FC is defined as

$$\begin{aligned} FC &= \beta \cdot \text{reshape}\left(\frac{\exp\left(G_{ij}\right)}{\sum_{j=1}^{C}\exp\left(G_{ij}\right)} \cdot FF\right) + FB \\ &\text{with } G_{ij} = \sum_{k=1}^{D} FF_{ik} \cdot FF_{jk} \end{aligned} \tag{9}$$

where β is a parameter that can be learned by the network. The reshape function transforms the dimension of the value to the same size as FB. In branch no. 3, we obtain the feature FD just before the Softmax function affected by the global loss function.

3.3 Internal and global loss function

In the proposed network model (see Fig. 2), there are two loss functions, namely, the external global loss function and the internal loss function, which can act on the network parameters using the gradient transfer mechanism. We next introduce the internal loss function which focuses on the loss between the predicted key-frame location and the tracking locations; then, the network can be trained under the location weakly supervised attention mechanism, for which only the action classification temporal label is needed.

- Loss function Part A: The action classification loss function is marked as Loss_{cls}.
- Loss function Part B: The location loss function focuses on the location loss of objects in the video clip. The single frame loss is marked as Loss_{loc}, and clip loss is marked as $\text{Loss}_{\text{clip}}$.
- Loss function Part C: The tracker predicted location loss function focuses on the tracking location loss with the previous video sequence, marked asL_{TRB}.
- Loss function Part D: The neighbor consistency loss function focuses on neighbor features's consistency in video sequence, marked asL_{NCB}.

1) Supervised cross-entropy loss

Suppose the image is split by an $S \times S$ grid. We use a cross-entropy function to compute the action classification loss marked as Loss_{cls}:

$$\text{Loss}_{\text{cls}} = -\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{\text{obj}}\left[P_i^j \log\left(\hat{P}_i^j\right) + \left(1 - P_i^j\right)\log\left(1 - \hat{P}_i^j\right)\right] \tag{10}$$

where I_{ij}^{obj}denotes the jth prior box of the ith grid is responsible for the object with the class cls; $I_{ij}^{\text{obj}} = 1$ if the object center exists in the grid; otherwise, $I_{ij}^{\text{obj}} = 0$, S^2 is the total number of grid cells and B denotes the total number of candidate prior boxes. P_i^j and $\hat{P}_i^j$ represent the ground truth and predicted class probability in the grid cell, respectively.

2) Clip supervised-location loss

Suppose a single frame loss function defined as

$$\begin{aligned}\mathrm{Loss}_{\mathrm{loc}} &= \lambda_{\mathrm{co}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{obj}} \left[\left(x_i - \hat{x}_i^j \right)^2 + \left(y_i - \hat{y}_i^j \right)^2 \right] + \\ &\lambda_{\mathrm{co}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i^j} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i^j} \right)^2 \right] \\ &- \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{obj}} \left[C_i^j \log \left(\hat{C}_i^j \right) + \left(1 - C_i^j \right) log \left(1 - \hat{C}_i^j \right) \right]\end{aligned} \tag{11}$$

where I_{ij}^{obj} denotes the jth prior box of the ith grid cell is responsible for the object; $I_{ij}^{\mathrm{obj}} = 1$ if the object center exists in the grid cell; otherwise, $I_{ij}^{\mathrm{obj}} = 0$. S^2 is the total number of grid cells and B denotes the total number of candidate prior boxes, and λ_{co} is an adjustable parameter. The object location $(x_i, y_i, w_i, h_i, C_i^j)$ denotes the (center_left, center_top, width, height, confidence) of the ground-truth box, and $(\hat{x}_i^j, \hat{y}_i^j, \hat{w}_i^j, \hat{h}_i^j, \hat{C}_i^j)$ denotes the location and confidence of the predicted box.

Considering that the video sequence is composed of a series of frames, the video clip loss function can be defined as follows:

$$\mathrm{Loss}_{\mathrm{clip}} = \frac{1}{N} \sum_{k=0}^{N-1} \mathrm{Loss}_{\mathrm{loc}}(k) \tag{12}$$

where N is the number of frames in the video clip.

3) Tacking-regularization-based loss

The tracker location loss function focuses on the loss between the tracker-predicted and network-calculated locations in video frames. We can use KCF [41] as a tracker, and other tracker methods can also be used in this study. The track loss function can be defined as follows:

$$L_{\mathrm{TRB}} = \frac{1}{N} \sum_{i=0}^{N} \mathrm{Loss}_{\mathrm{clip}} \left(\mathrm{Loc}_{\mathrm{clip}}^{i}, \mathrm{Tracker}(\mathrm{Loc}_{\mathrm{clip}}^{i+1}) \right) \tag{13}$$

where $\mathrm{Loc}_{\mathrm{clip}}^{N}$ denotes the target location in keyframe, which comes from output of the network branch no. 4. $\mathrm{Loc}_{\mathrm{clip}}^{i+1}$ denotes the object locations in the ith frame of the clip, $\mathrm{Loc}_{\mathrm{clip}}^{i}$ denotes the object locations in the previous frame. Note that, Since both tracking and attention-based localization are not certain and either cannot be taken as ground truth, this term is more like internal regularization loss, we might as well call it tracking regularization loss here.

4) Neighbor-consistency-based loss

Intuitively, the features' cosine distance between the neighbors is closer in the same video clip, and so we introduce neighbor consistency loss in the model.

$$X_g = \{x_{g,0}, ..., x_{g,i}, ..., x_{g,N}\} \tag{14}$$

$$d_{g,i,j} = d_c(x_{g,i}, x_{g,j}) = f\left(x_{g,i}\right)^T f(x_{g,j}) \tag{15}$$

$$D_N = \begin{bmatrix} d_{0,0,1} & \cdots & d_{0,i,i+1} & \cdots & d_{0,N-1,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{g,0,1} & \cdots & d_{g,i,i+1} & \cdots & d_{g,N-1,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{PK-1,0,1} & \cdots & d_{PK-1,i,i+1} & \cdots & d_{PK-1,N-1,N} \end{bmatrix} \tag{16}$$

where $x_{g,i}$ indicates the ith frame of the video clip, $x_{g,N}$ specially indicates the keyframe, and we normally copy the *(N-1)*th frame of clip as the keyframe. For the gth clip in the batch PK, the cosine distance between all images in X_g is calculated, and $f(\cdot)$ means the target confidence feature of the image.

The distance matrix D_N is adopted to realize neighbor consistency. Intuitively, the distance between $x_{g,i}$ and $x_{g,i+1}$ neighbors should be pulled closer. Besides, to make the closer neighbors get more proportions in NCB loss, a weight w_i that reflects the contribution of the ith neighbor. If the distance between $x_{g,i}$ and $x_{g,i+1}$ is large, then its contribution to $x_{g,i}$ is small:

$$w_i = \begin{cases} \frac{1}{N}\left(1 - \frac{d_c(x_{g,i}, x_{g,i+1})}{\sum_{j=0}^{N-1} d_c(x_{g,j}, x_{g,j+1})}\right), \forall\, i \in \{0, 1, ..., N-2\} \\ d_c\left(x_{g,N-1}, x_{g,N}\right), \quad \text{where } x_{g,N} \text{ is the keyframe} \end{cases} \tag{17}$$

To pull the distance between the $x_{g,i}$ and its neighbors closer, the NCB loss can be formulated as

$$L_{\text{NCB}} = -\sum_{i=0}^{N-1} w_i \log \frac{\exp(d_c\left(x_{g,i}, x_{g,i+1}\right)/\epsilon)}{\sum_{i=0}^{N-1} \exp(d_c\left(x_{g,i}, x_{g,i+1}\right)/\epsilon)} \tag{18}$$

where ϵ is the scaling parameter. NCB loss make the given object frame closer to its neighbors, which can further improve the stability of the model.

5) GL and IL defination

In source domain, we defined *GL* and *IL* as

$$GL = \text{Loss}_{\text{cls}} + \text{Loss}_{\text{loc}}(p\_loc, t\_loc) \tag{19}$$

$$IL = \text{Loss}_{\text{clip}} \tag{20}$$

where *t_loc* is the ground truth location in the keyframe, and *p_loc* is the output location which comes from the output of the network branch no. 3. According to (7), the internal loss function *IL* directly affects the location feature of the sequence. Therefore, we can obtain more attention features of the video sequence to improve the precision of action tube detection.

In target domain, we defined *GL* and *IL* as

$$GL = \text{Loss}_{\text{cls}} + \text{Loss}_{\text{loc}}\left(\text{Loc}_{\text{clip}}^{N-1}, \text{Tracker}\left(\text{Loc}_{\text{clip}}^{N}\right)\right) \tag{21}$$

$$IL = L_{\text{TRB}} + \rho L_{\text{NCB}} \tag{22}$$

where ρ is the hyper-parameter that control the importance of the NCB loss relative to the RAT loss. Note that the data set only has action temporal annotations in target

domain. The location interactive loss is computed by L_{TRB} and L_{NCB} based on the video sequence.

3.4 Parameters about the model

In network branch no. 1, a clip of the video frame sequence is fed into the 3D network as the input and the original video can be sampled in time. The shape of the input data is $[N \times CH \times H \times W]$, where N is the length of the clip, CH is the number of image channels, H is the height of the video image, and W is the width of the video image. If 4 frames of 3-channel RGB images are sampled per second, then a clip consists of 16 frames per 4 seconds, then $N = 16,\ CH = 3$. The tensor S_{101} has a shape $[N^{'} \times C^{'} \times H^{'} \times W^{'}]$, which can be squeezed by setting $N^{'} = 1$, $H^{'} = H/32$, $W^{'} = W/32$. Then, the feature dimension of the *3D-CNN* output is squeezed and transformed into the shape $[C^{'} \times H^{'} \times W^{'}]$. Hence, it is easy to concatenate with the output feature of network branch no. 2, because they have the same single feature map shape $[H^{'} \times W^{'}]$.

In network branch no. 2, we input the same video frame sequence as in network branch no.1, and adopt the *3D-CNN* network to generate the location feature. Given that the two network branches are calculated in parallel, this method does not required additional computing time. The tensor S_{50} has the shape $[N \times D^{'} \times H^{'} \times W^{'}]$. We squeeze the tensor by setting $D^{'} = 1$, $H^{'} = H/32$, $W^{'} = W/32$, and the output shape of network branch no. 2 is $[C^{''} \times H^{'} \times W^{'}]$, where $C^{''} = N \times D^{'} = N$. Let us assume that the learning rate function of branches no. 2 and no. 4 is $\alpha(t)$ acting on the back-propagation process driven by the internal loss function. The learning rate function used in the branches no. 1 and no. 3 is $\lambda\alpha(t)$, where λ is a constant less than 1.

In network branches no. 3 and no. 4, the location regression method partly refers to the idea of YOLO [7]. If the input size is 416×416 and 32 down-sampling is used, then the grid size is 13×13. We also generate a 26×26 feature map with 16 down-sampling, or a 52×52 feature map with 8 down-sampling. Note that the higher the sampling ratio, the larger the feature map. In this process, the k-means method is also used to determine the size of the prior boxes based on the training data set, where k is the selected number. If the number of prior boxes is five, and each box has four position parameters and one confidence parameter, the total number of categories is NumCls, and the dimension of C^{*} is $[5 \times (\mathrm{NumCls} + 4 + 1)]$ in the network branch no.3. The dimension of C^{**} is $[5 \times (4 + 1)]$ in network branch no. 4, because the internal loss does not focus on the classification information. To support multi label objects, a *Softmax* function is used to predict the results.

4 Results and discussion

In this section, we first describe the experimental setup. We then conduct ablation studies and it shows the effectiveness of the different network parts. Next, we provide comparisons with several metric methods for action classification and temporal action location tasks, respectively. Finally, we analyzed the intrinsic reasons of performance improvement, including the working mechanism, attention activation maps and issues that need to be further studied.

4.1 Experimental setup

We first validated our method for action classification on the J-HMDB-21 [42] and UCF101-24 [43] data sets. The UCF101-24 data set contains 24 action classes and 3207 videos, with multiple possible action instances in each video. The J-HMDB-21 data set consists of 928 short videos with 21 action categories in daily life, where each video is trimmed to a single action instance across all frames. Then, we used the THUMOS14 and ActivityNet-v1.3 data sets in the experiment of temporal action localization, where THUMOS14 contains all of the UCF101 actions. THUMOS14 has 13320 trimmed videos for training, and each video includes one action, and the data include UCF101-24 with bounding box annotations. THUMOS14 also has 2500 untrimmed videos for training, each is guaranteed not to include any instance of the 101 actions, and 1010 untrimmed videos for validation.

The image size was 412×412 pixels. In this study, 32 down-sampling was used in the spatial domain to form a 13×13 grid. To improve the generalization ability of the data, a spatial transformer was also used to produce a 0.1 amplitude random shift and 10-degree intermediate random rotation in the spatial domain. A temporal transformer was used for random sequence extraction based on 16 frames. We used the SGD optimizer with the weight decay, in which the momentum parameter, and decay weight were used. The initial value of the learning rate was 0.05, which linearly decreased according to the epoch. The hyper-parameter ρ is 0.2, λ is 0.1 and ϵ is 1.0. For a batch size of 64, at least four TITAN GPU cards or two RTX8000 GPU cards are needed for training.

In the action classification tasks, the indicator Frame-mAP was used as a benchmark. Suppose $x(t_{N-1})$ represents the keyframe of a video clip in (1), the whole video's time range is $[t_0,\ t_{L-1}]$, and X was a clip in the video, then there were $L - N$ clips in the video. The Frame-mAP was the mAP of all of video clips on the validation dataset.

In the action temporal localization tasks, the AP of the temporal action localization mainly considers the localization matching rate between predicted frame localization and ground truth with the same classification label, so the mAP indicator can be defined as follows:

$$\mathrm{AP(cls)} = \frac{\sum_{k=1}^{LN}(\mathrm{Pred}(k) \times \mathrm{real}(k))}{\sum_{k=1}^{LN}\mathrm{real}(k)} \tag{23}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{CLS}}\sum_{\mathrm{cls}=1}^{\mathrm{CLS}}\mathrm{AP(cls)} \tag{24}$$

where Pred(k) indicates the predicted frame localization in all LN video frames, and real(k) indicates the ground truth. AP(cls) means the AP of single category CLS, so the mAP was the mean AP of all categories.

4.2 Ablation study

We performed ablation studies on the UCF101-24 and J-HMDB-21 data sets to prove the effectiveness of each part of the loss. We used 80% of the UCF101-24 data set for training. The Frame-mAP of classification is shown in Table 1. Supposing 20–100%

Table 1 We used 80% of the UCF101-24 dataset for training and 20% for validation

Domain	Mode	20%	30%	50%	70%	100%
Source	Full	80.7	86.8	95.4	96.5	96.7
Target	Weak	93.3	94.9	96.1	96.3	-

The Frame-mAP is shown in the table. We assume 20–100% usage of training data for fully supervised learning with our model in source domain. In target domain, we trained the network with the pre-trained model and the remaining data that only has classification annotations

Table 2 "U-24→J-21" means that UCF101-24 is the source domain used in source domain and J-HMDB-21 is the target domain used in target domain

Methods(Target)	U-24→J-21	J-21→U-24
Baseline+SCEL	61.0	70.3
Baseline+SCEL+TRBL	69.3	79.0
Baseline+SCEL+NCBL	86.3	90.6
Baseline+SCEL+TRBL+NCBL	90.2	94.8

"Baseline + xxx" means that the "xxx" loss function is used upon the baseline model. SCEL stands for supervised cross-entropy loss, TRBL stands for tracking-regularization-based loss, NCBL means neighbor-consistency-based loss

usage of training data for fully supervised learning with the model in source domain. In target domain, we trained the network with the pre-trained model and the remaining data that only had classification annotations.

In Table 2 and Fig. 2, to ensure the continuity of the target in the video sequence, tracking-regularization loss and neighbor-consistency loss were calculated in the video. "U-24→J-21" means that UCF101-24 is used in source domain and J-HMDB-21 is used in target domain. "Baseline + xxx" means that the "xxx" loss function is used upon the baseline model. SCEL stands for supervised cross-entropy loss, TRBL stands for tracking-regularization-based loss, NCBL means neighbor-consistency-based loss.

1) Effectiveness of ALL: In the Table 1, assuming that only 30% of the data have bounding-box annotations, the model can only achieve 86.8% Frame-mAP on the UCF101-24 data set in the source domain. However, if we use the other 70% data without bounding-box annotations to train the network in target domain, then Frame-mAP is 94.9%. It is especially costly and time consuming to annotate every target location in the video frames, and the Track Loss is effective if we only have few data with location labels.

2) Effectiveness of NCBL: When Baseline+SCEL+NCBL use in the training, the Frame-mAP performance achieved 86.3% and 90.6% on the J-HMDB-21 and UCF101-24 data sets, respectively. The NCBL loss is used to pull closer the similar targets within a certain range. Unlike the TRBL loss, the NCBL loss is likely to mine the similarity of targets within the video sequence. It also illustrates the advantage of avoiding completely relying on location labels.

3) Effectiveness of TRBL: The Frame-mAP performance presents no significant difference when we choose different trackers, such as MIL [44], KCF [41] and SRDCF [45]; when the model uses 70% of the data for location unsupervised training, they achieved a performance of 94.1%, 94.9%, 95.1% on the UCF101-24 data set, respectively. This is because the target occupies a large proportion in the image on the target data set, and generally there is no occlusion.

Table 3 Action classification experiment

Method	Mode	J-HMDB-21	UCF101-24
T-CNN [12]	Full	61.3	41.4
ACT [20]	Full	65.7	69.5
STEP [46]	Full	–	75.0
P3D-CTN [15]	Full	71.1	–
I3D [47]	Full	73.3	77.7
ACRN [24]	Full	77.9	80.4
YOWO+LFB [24]	Full	75.7	87.3
3C-Net [34]	Weak	77.9	86.4
HAM-Net [39]	Weak	88.1	92.1
Ours	Weak	90.2	94.8

The table lists the comparison results of Frame-mAP (IOU=0.5, 16 frames clip). We compared with recent fully and weakly supervised methods. Note that, the proposed method is an object location-unsupervised classification-supervised attention network

Table 4 It is about the run time and performance comparison on data set UCF101-24 on a single NVIDIA RTX8000 card with 16-frames video clip

Method	Speed(fps)	Frame-mAP
P3D-CTN	28	–
I3D	30	77.7
3C-Net	45	84.4
HAM-Net	29	92.1
YOWO+LFB	38	86.4
Ours	31	94.8

For our method, *ResNeXt-50* and *ResNeXt-34* are used in its two 3D-CNN backbones

4.3 Experimental results of action classification

We compared the proposed method with state-of-the-art methods on the UCF101-24 and J-HMDB-21 data sets, as shown in Table 3. Using standard metrics, we present the Frame-mAP at IOU threshold 0.5 and 16-frame clips. It can be seen that the proposed method outperforms the state of the art in terms of Frame-mAP, which is improved by 2.1% and 2.7% on the two data sets, respectively. Note that the proposed method used transfer learning, which means training the network on the target domain (UCF101-24) in target domain with the pretrained model trained on the source domain (J-HMDB-21) in source domain, then, obtaining the Frame-mAP of 94.8% on the UCF101-24 data set.

From Table 4, we can see that our method pertains to acceptable performance because of the parallel architecture mechanism consisting of a classification network branch No. 1 and location network branch No. 2. At the same time, because it has two 3D-CNN parallel computing branches, it consumes more computing resources than some state-of-the-art models. The comparison may not be fair without considering computing complexity, but note that, our contribution is introducing tracking loss and neighbor-consistency loss for action detection tasks, if the system needs high real-time performance, it can choose simple backbones.

4.4 Experimental results of temporal action localization

We also conducted experiments on temporal action localization (TAL) using the proposed method. Table 5 summarizes the performance comparisons between the proposed method and state-of-the-art methods. The table lists the comparison results in terms of mAP with state-of-the-art methods (16 frames clips). We compared the typical full and weak methods, T(IOU@0.3) indicates THUMOS14 with IOU@0.3, while T(IOU@0.5) indicates IOU=0.5, and A(IOU@0.5) indicates ActivityNet1.3 with IOU@0.5. Specifically, our proposed method achieves the mAP of 49.6% at IOU threshold 0.5 on the data set THUMOS14. Moreover, our method outperforms the weakly supervised TAL models, and even shows comparable results with some recent fully supervised TAL methods. Note that, we cannot perform the contrast experiment on the data set THUMOS14(UCF101), because only its sub-data set UCF101-24 has object bounding boxes, which means that only about 24% (3207/13320) of the THUMOS14 training data are available for fully supervised pretraining in source domain, and the pretrained model is also used in the experiment based on the ActivityNet1.3 data set.

4.5 Further analysis of the experimental results

We performed experiments for action classification and temporal action localization. The performance was better than recent weakly supervised methods, and even shows comparable results with recent fully supervised methods. We also present the activation maps [50] in Fig. 5, which reveal the intrinsic reason why our method has better attention performance than the state-of-the-art video action tube detection methods. Note the following points:

- The jump action example shows that HAM–NET's attention mechanism is more likely to be disturbed by sudden or rapid object movements such as moving clouds and crowds of people. This is because HAM–NET's attention mechanism is based on the optical flow of the video frames.
- The Walking-With-Dog action example shows that HAM–NET is more likely to ignore important parts of an action such as the presence of a dog in cases where the

Table 5 Temporal action localization experiment

Method	Mode	T(IOU@0.3)	T(IOU@0.5)	A(IOU@0.5)
G-TAD [48]	Full	–	40.2	46.7
P-GCN [49]	Full	63.6	49.1	48.3
Nguyen [36]	Weak	46.6	26.8	–
3C-Net [34]	Weak	40.9	24.6	35.4
WSGN [31]	Weak	42.0	25.1	–
Islam [29]	Weak	46.8	29.6	35.2
BaS-Net [35]	Weak	44.6	27.0	34.5
DGAM [32]	Weak	46.8	28.8	41.0
HAM-Net [39]	Weak	50.3	31.0	41.5
Ours	Weak	64.4	49.6	52.2

The table lists the comparison results of mAP (16 frames clip). We compared with typical fully and weakly supervised methods. T(IOU@0.3) indicates THUMOS14 with IOU@0.3, T(IOU@0.5) indicates IOU=0.5, and A(IOU@0.5) indicates ActivityNet with IOU@0.5. Note that, the proposed method is an object location-unsupervised classification-supervised attention network

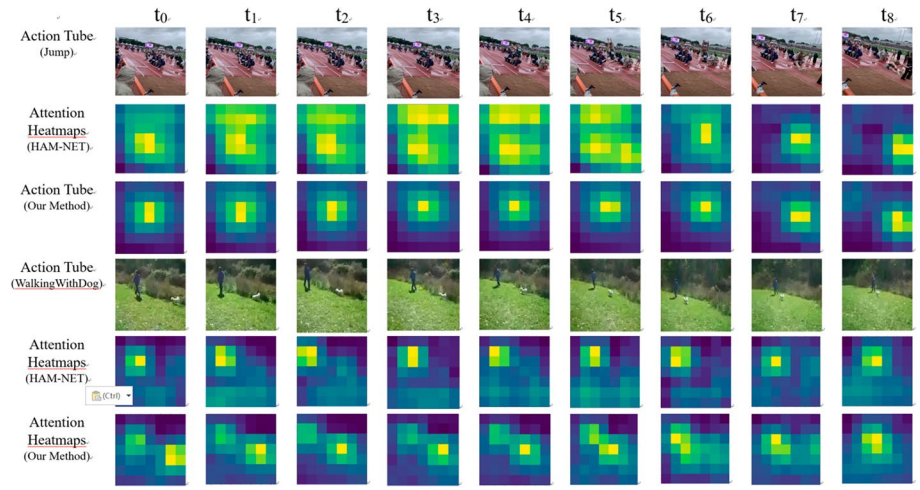

Fig. 5 Activation heat-maps are from the tensors just before the channel fusion network. Jump action example shows that HAM–NET's attention mechanism is more likely be disturbed by sudden or rapid object movements such as moving clouds and crowds of people, because it concerns the optical flow of the video frames. Walking-With-Dog action example shows that HAM–NET is more likely to ignore important parts of an action such as the presence of the dog in cases where the training data set contains a series of similar actions such as in Skiing, Ice-Dancing, Long-Jump. Our method has a higher level of robustness

training dataset contains a series of similar actions, such as in Skiing, Ice-Dancing, and Long-Jump. In the action classification experiment, the keyframe of the video clips is particularly important for HAM–NET to predict the correct results.

- Our method has a higher level of robustness. The internal tracker loss and neighbor consistency loss are more efficient for weakly supervised learning based on video sequences, in which only the action classification temporal labels are needed.

Concerning the experimental results, there are four important points to explain:

First, the classification temporal label is also needed in our method when the object location is achieved through weakly supervised learning; this approach is notably different from other methods.

Second, the performance outperforms typical methods and some recent fully supervised methods because of the spatial-temporal attention mechanism. In other words, the attention mechanism also works well when the object location is based on weakly supervised learning.

Third, although the classification labels of the source domain may be different from that of the target domain, the pretrained model of the source domain can still be transferred, because they are all human actions with the same attention mechanism.

Finally, our contribution is introducing tracking loss and neighbor-consistency loss for action detection tasks. The comparison may not be fair without considering computing complexity, if the system needs high real-time performance, it can choose simple backbones.

However, There is a issue that still requires further study. The tracker needs a few previous target locations of the frames. This means, if some initial locations are wrongly

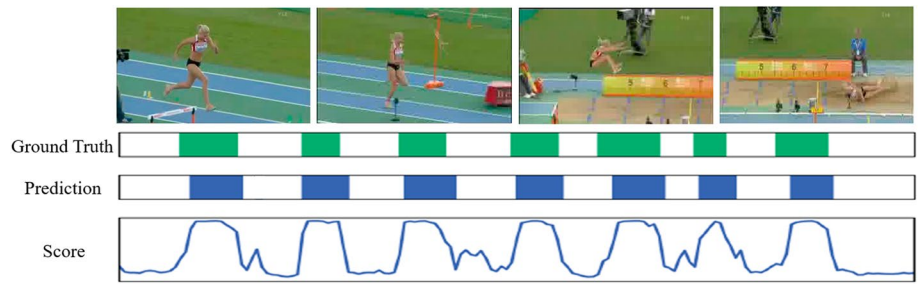

Fig. 6 Temporal action localization experiment on THUMOS14. The horizontal axis denotes time, we sequentially plot the ground truth, predicted localization, and prediction score

predicted by the network, it may affect the following loss to some extent during training. We can skip a few initial frames when training if the length of the video is not very short. Nevertheless, the shortage will affect the mAP to some extent for the temporal action localization tasks. As shown in Fig. 6, the starting point may be wrongly predicted or delayed by several frames. Moreover, if the training data are randomly mixed with reverse-time video clips by a ratio of 1:1, the mAP can be further improved by 0.5% on the THUMOS14 data set. Although the training process may be affected by the initial action frames, the proposed method outperforms the state-of-the-art methods.

In short, the ablation study shows the effectiveness of different parts of the proposed method. In classification tasks, the proposed method outperforms the state-of-the-art in terms of Frame-mAP, which is improved by 2.7% and 2.1% on data sets UCF101-24 and J-HMDB-21, respectively. In action temporal localization tasks, the proposed method achieved higher mAP than the current best scores on the data set THUMOS14. Moreover, the proposed method outperforms the weakly supervised TAL models, and even shows comparable results with some recent fully supervised TAL methods. Concerning the experimental results, we analyzed the intrinsic reasons of performance improvement, including working the mechanism, attention activation maps and the issues that need to be further studied.

5 Conclusions

We introduced a novel location-weakly supervised learning method with a spatial–temporal attention mechanism for action tube detection. The novelty is remarkable compared with previously reported methods. An internal interactive location tracker loss and neighbor consistency loss for weakly supervised learning are designed, in which only the classification temporal label is needed. This is the first study in location weakly supervised situation with a spatial–temporal–attention mechanism for action tube detection. Although this is a location-weakly supervised classification-supervised method, the mAP performance is better than that of typical weakly supervised methods, and even shows comparable results with some recent fully supervised methods.

Abbreviations

TRBL	Tracking-regularization-based loss
NCBL	Neigbor-consistency-based loss
SCEL	Supervised cross-entropy loss

3D-CNN	3-Dimension convolutional neural network
TAL	Temporal action location
SOTA	State-of-the-arts

Acknowledgements
No additional acknowledgements

Author contributions
Jinlei Zhu waw primary author of the methodology, software. Houjin Chen mentioned the conceptualization of this study. Pan Pan was the designer of action classification experiment. Jia Sun was the designer of temporal action location experiment. All authors read and approved the final manuscript.

Author's information
Jinlei Zhu is currently pursuing the Doctor degree with Beijing Jiaotong University, his research interests include machine learning and image analysis, and he currently works in Synthesis Electronic Technology Co.,Ltd. Houjin Chen is currently a Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. Pan Pan is currently pursuing the Doctor degree in communication engineering with the School of Electronic and Information Engineering. Jia Sun is currently pursuing the Doctor degree in communication engineering with the School of Electronic and Information Engineering. School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China.

Funding
This work was partly supported by the Key Research Project of Shandong Province of China (No.2019TSLH0206) and Industry Leading Talent Project of Jinan City of China (No.00982019010).

Availability of data and materials
The public data set can be downloaded from the official website.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 26 April 2021 Accepted: 27 June 2022
Published online: 18 July 2022

References

1. J. Zhu, H. Chen, P. Pan, A novel rate control algorithm for low latency video coding base on mobile edge cloud computing. Comput. Commun. **187**, 134–143 (2022)
2. Q. Zheng, Y. Chen, Interactive multi-scale feature representation enhancement for small object detection. Image Vis Comput (2021). https://doi.org/10.1016/j.imavis.2021.104128
3. C. Yan, Y. Wei, Y. Gao, Deep multi-view enhancement hashing for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **43**(4), 1445–1451 (2020)
4. C. Yan, Y. Zhang, Y. Liu, Y. Zhang, Depth image denoising using nuclear norm and learning graph model. ACM Trans. Multimed. Comput. Commun. Appl. **16**(4), 1–17 (2020)
5. C. Yan, T. Teng, Y. Zhang, H. Wang, Precise no-reference image quality evaluation based on distortion identification. ACM Trans. Multimed. Comput. Commun. Appl. **17**(3s), 1–21 (2021)
6. B. Yu, Z. Xie, D. Huang, Stacked generative adversarial networks for image compositing. EURASIP J Image Video Process **1**, 1–20 (2021). https://doi.org/10.1186/s13640-021-00550-w7
7. J. Redmon, A. Farhadi, Yolov3: An incremental improvement. arXiv preprint (2018). arXiv:1804.02767
8. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, Ssd: Single shot multibox detector. arXiv preprints (2016). arXiv:1512.02325
9. D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks. arXiv preprint (2015). arXiv:1412.0767v4
10. Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, 3D u-net: Learning dense volumetric segmentation from sparse annotation. arXiv preprint (2016). arXiv:1606.06650
11. S. RenK. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
12. R. Hou, C. Chen, M. Shah, Tube Convolutional Neural Network (T-CNN) for action detection in videos. In: IEEE International Conference on Computer Vision, vol. 28, pp. 5822–5831 (2017)
13. C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)
14. L. Sun, K. Jia, D. Y. Yeung, B. E. Shi, Human action recognition using factorized spatio-temporal convolutional networks. In: IEEE International Conference on Computer Vision, pp. 4597–4605 (2015)
15. J. Wei, H. Wang, Y. Yi, Q. Li, D. Huang, P3D-CTN: Pseudo-3D convolutional tube network for spatio-temporal action detection in videos. In: IEEE International Conference on Image Processing, pp. 300–304 (2019)
16. H. Kataoka, T. Wakamiya, K. Hara, Y. Satoh, Would mega-scale datasets further enhance spatiotemporal 3D CNNS. arXiv preprint (2020). arXiv:2004.04968
17. K. Hara, H. Kataoka, Y. Satoh, Can spatiotemporal 3D CNNS retrace the history of 2D CNNS and Imagenet. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6546–6555 (2018)
18. O. Köpüklü, N. Kose, A. Gunduz, G. Rigoll, Resource efficient 3d convolutional neural networks. In: IEEE/CVF International Conference on Computer Vision Workshop, pp. 1910–1919 (2019)

19. E. H. P. Alwando, Y. T. Chen, W. H. Fang, CNN-based multiple path search for action tube detection in videos. In: IEEE Transactions on Circuits and Systems for Video Technology, pp. 104–116 (2018)
20. V. Kalogeiton, P. Weinzaepfel, V. Ferrari, C. Schmid, Action tubelet detector for spatio-temporal action localization. In: IEEE International Conference on Computer Vision, pp. 4415–4423 (2017)
21. W. Wang, D. Liu, X. Liu, L. Pan, Online real-time multiple spatiotemporal action localisation and prediction. In: IEEE International Conference on Computer Vision, pp. 3657–3666 (2017)
22. C. Yan, Y. Hao, et al., Task-adaptive attention for image captioning. In: IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, pp. 43–45 (2021)
23. C. Yan, L. Meng, et al., Age-invariant face recognition by multi-feature fusion and decomposition with self-attention. In: ACM Transactions on Multimedia Computing Communications and Applications, vol. 18(1s), pp. 1–18 (2021)
24. O. Köpüklü, X. Wei, G. Rigoll, You only watch once: a unified CNN architecture for real-time spatiotemporal action localization. arXiv preprint (2020) arXiv:1911.06644
25. L. Wang, Y. Xiong, D. Lin, et al., UntrimmedNets for weakly supervised action recognition and detection. In: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4325–4334 (2017)
26. P. Nguyen, T. Liu, G. Prasad, B. Han, Weakly supervised action localization by sparse temporal pooling network. In: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6752–6761 (2018)
27. Z. Shou, H. Gao, L. Zhang, K. Miyazawa, S. F. Chang, AutoLoc: weakly-supervised temporal action localization in untrimmed videos. In: In Proceedings of the European Conference on Computer Vision (ECCV), pp. 154–171 (2018)
28. S. Paul, S. Roy, A. K. Roy-Chowdhury, WTALC: weakly-supervised temporal activity localization and classification. In: In Proceedings of the European Conference on Computer Vision (ECCV), pp. 563–579 (2018)
29. A. Islam, R. Radke, Weakly supervised temporal action localization using deep metric learning. In: IEEE Winter Conference on Applications of Computer Vision, pp. 547–556 (2020)
30. T. Yu, Z. Ren, E. Yan, Temporal structure mining for weakly supervised action detection. In: In Proceedings of the IEEE International Conference on Computer Vision, pp. 5522–5531 (2019)
31. B. Fernando, C. Tan, H. Bilen, Weakly supervised Gaussian networks for action detection. In: In The IEEE Winter Conference on Applications of Computer Vision, pp. 537–546 (2020)
32. B. Shi, Q. Dai, Y. Mu, Weakly-supervised action localization by generative attention modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1009–1019 (2020)
33. Z. Liu, L. Wang, Q. Zhang, Z. Gao, Z. Niu, N. Zheng, G. Hua, Weakly supervised temporal action localization through contrast based evaluation networks. In: In Proceedings of the IEEE International Conference on Computer Vision, pp. 3899–3908 (2019)
34. S. Narayan, H. Cholakkal, F. S. Khan, L. Shao, 3C-net: category count and center loss for weaklysupervised action localization. In: In Proceedings of the IEEE International Conference on Computer Vision, pp. 8679–8687 (2019)
35. P. Lee, Y. Uh, H. Byun, Background suppression network for weakly-supervised temporal action localization. In: The AAAI Conference on Artificial Intelligence (2020) aaai.v34i07.6793
36. P. Nguyen, T. Liu, G. Prasad, B. Han, Weakly supervised action localization by sparse temporal pooling network. In: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6752–6761 (2018)
37. N. Yu, L. Huang, Z. Wei, W. Zhang, B. Wang, Weakly supervised fine-grained recognition based on spatial-channel aware attention filters. In: Multimedia Tools and Applications (2021) https://doi.org/10.1007/s11042-020-10268-y
38. D. Liu, T. Jiang, Y. Wang, Completeness modeling and context separation for weakly supervised temporal action localization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1298–1307 (2019)
39. A. Islam, C. Long, R. Radke, A hybrid attention mechanism for weakly-supervised temporal action localization. AAAI Conf. Artif. Intell. **35**(2), 1637–1645 (2021)
40. S. Xie, R. Girshick, P. Dollar, Z. Tu, K. He, Aggregated residual transformations for deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5987–5995 (2017)
41. J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
42. H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition. In: International Conference on Computer Vision, pp. 2556–2563 (20110)
43. K. Soomro, A. R. Zamir, M. Shah, Ucf101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint (2012) arXiv:1212.0402v1
44. B. Babenko, M. H. Yang, S. Belongie, Visual tracking with online multiple instance learning. In: Conference on Computer Vision and Pattern Recognition, pp. 983–990 (2009)
45. D. Martin, G. Hager, F. Shahbaz, M. Felsberg, Learning spatially regularized correlation filters for visual tracking. In: Conference on Computer Vision and Pattern Recognition (2015). https://doi.org/10.1109/ICCV.2015.490
46. X. Yang, M. Y. Liu, F. Xiao, L. S. Davis, J. Kautz, STEP: spatiotemporal progressive learning for video action detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 264–272 (2019)
47. C. Gu, C. Sun, D. A. Ross, AVA: a video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6047–6056 (2018)
48. M. Xu, C. Zhao, D. S. Rojas, G-TAD: sub-graph localization for temporal action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10156–10165 (2020)
49. R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, C. Gan, Graph convolutional networks for temporal action localization. In: In The IEEE International Conference on Computer Vision, pp. 7094–7103 (2019)
50. B. Zhou, A. Khosla, L. A. Oliva, A. Torralba, Learning deep features for discriminative localization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)

Publisher's Note