



Cascaded Pyramid Mining Network for Weakly Supervised Temporal Action Localization

Haisheng Su[✉], Xu Zhao[✉], and Tianwei Lin

Department of Automation, Shanghai Jiao Tong University, Shanghai, China
{suhaiseng, zhaoxu, wzmsltw}@sjtu.edu.cn

Abstract. Weakly supervised temporal action localization, which aims at temporally locating action instances in untrimmed videos using only video-level class labels during training, is an important yet challenging problem in video analysis. Many current methods adopt the “*localization by classification*” framework: first do video classification, then locate temporal area contributing to the results most. However, this framework fails to locate the entire action instances and gives little consideration to the local context. In this paper, we present a novel architecture called **Cascaded Pyramid Mining Network (CPMN)** to address these issues using two effective modules. First, to discover the entire temporal interval of specific action, we design a two-stage cascaded module with proposed **Online Adversarial Erasing (OAE)** mechanism, where new and complementary regions are mined through feeding the erased feature maps of discovered regions back to the system. Second, to exploit hierarchical contextual information in videos and reduce missing detections, we design a pyramid module which produces a scale-invariant attention map through combining the feature maps from different levels. Final, we aggregate the results of two modules to perform action localization via locating high score areas in temporal **Class Activation Sequence (CAS)**. Extensive experiments conducted on THUMOS14 and ActivityNet-1.3 datasets demonstrate the effectiveness of our method.

Keywords: Temporal action localization · Weak supervision · Online Adversarial Erasing · Scale invariance · Class Activation Sequence

1 Introduction

Due to the rapid development of computer vision along with the increasing amount of videos, many breakthroughs have been observed on video content analysis in recent years. Videos from realistic scenarios are often complex, which may contain multiple action instances of different categories with varied lengths.

This research is supported by the funding from NSFC programs (61673269, 61273285, U1764264).

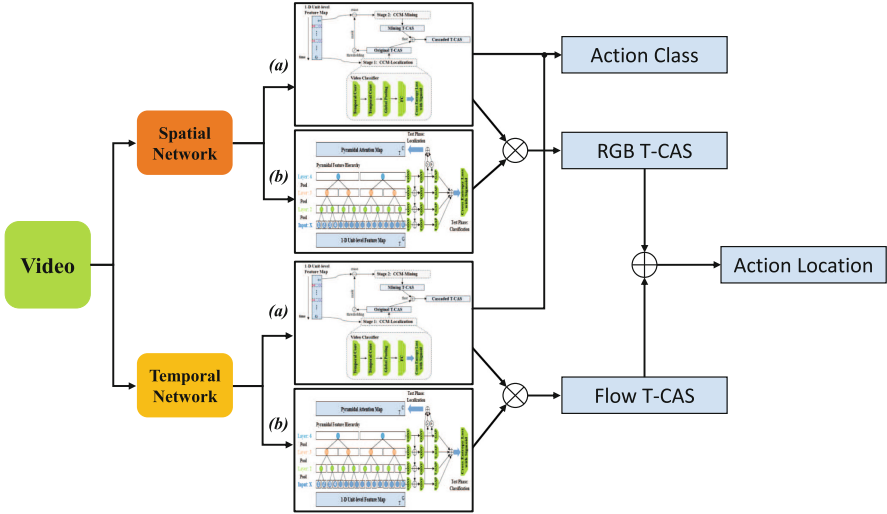


Fig. 1. Overview of our approach. Two-stream network is used to encode visual features for our algorithm to perform action classification and temporal action localization concurrently. The (a) Cascaded Classification Module (CCM) with Online Adversarial Erasing (OAE) method and the (b) Pyramid Attention Module (PAM) are proposed to compute attention-based cascaded Temporal Class Activation Sequence (T-CAS) from the two streams separately, which can be employed to locate the entire regions of specific actions in temporal domain with high accuracy.

This problem leads to a challenging task: temporal action localization, which requires to not only handle the category classification of untrimmed videos but also determine the temporal boundaries of action instances. Nevertheless, it implies the huge amounts of temporal annotations for training an action localization model, which are more labor-intensive to obtain than video-level class labels.

Contrary to the fully supervised counterparts, Weakly Supervised Temporal Action Localization (WSTAL) task learns TAL using only video-level class labels, which can be regarded as a temporal version of Weakly Supervised Object Detection (WSOD) in image. A popular series of models in WSOD generate Class Activation Maps (CAMs) [42] to highlight the discriminative object regions contributing to the classification results most. Inspired by [42], recently many WSTAL works generate the Class Activation Sequence (CAS) to locate the action instances in temporal domain. However, many drawbacks have been observed in this “*localization by classification*” mechanism: (1) the CAS fails to generate dense detections of target actions, causing many missing detections; (2) the classification network usually leverages features of discriminative rather than entire regions for recognition, failing to handle the action instances with varied lengths; (3) some true negative regions are falsely activated, which is mainly due

to the action classifier realizes the recognition task based on a global knowledge of the video, resulting in inevitably neglecting the local details.

To address these issues and generate high quality detections, we propose the Cascaded Pyramid Mining Network (CPMN), which adopts two effective modules to mine entire regions of target actions and remove the false positive regions respectively. Specifically, CPMN generates detections in three steps. **First**, CPMN adapts two classifiers with different input feature maps to discover discriminative regions separately, and the input feature maps of the second classifier are erased with the guidance of the CAS from the first one. **Second**, CPMN combines the discriminative regions discovered by the two classifiers to form the entire detections. **Final**, taking full advantage of hierarchical contextual representations, CPMN generates a scale-invariant attention map to correct the false positive regions and reduce the missing detections. These pyramidal feature representations offer “*local to global*” context information for better evaluation. The overview of our algorithm is illustrated in Fig. 1.

To sum up, the main contributions of our work are three-fold:

- (1) We propose a new architecture (CPMN) for weakly supervised temporal action localization in untrimmed videos, where entire temporal regions of action instances are located with less missing detections.
- (2) We introduce an Online Adversarial Erasing (OAE) method to discover entire regions of target actions using two cascaded classifiers with different input feature representations, and explicitly handle the action instances with varied lengths by exploiting hierarchical contextual information.
- (3) Extensive experiments demonstrate that our method achieves the state-of-the-art performance on both THUMOS14 and ActivityNet-1.3 datasets.

2 Related Work

Action Recognition. Action recognition has been widely studied in recent years, which aims to identify one or multiple action categories of a video. Earlier works mainly focus on hand-crafted feature engineering, such as improved Dense Trajectory (iDT) [31, 32]. With the development of convolutional neural networks, many deep learning based methods [6, 25, 29, 34] have been applied to action recognition task and achieve convincing performance. Two-stream network [6, 25, 34] typically consists of two branches which learn the appearance and motion information using RGB image and optical flow respectively. C3D network [29] simultaneously captures appearance and motion features using a series of 3D convolutional layers. These action recognition models are usually adopted to extract frame or unit level visual representation in long and untrimmed videos.

Weakly Supervised Object Detection. Weakly supervised object detection aims to locate the objects using only image-level labels. Current works mainly include bottom-up [2, 3] and top-down [27, 36, 40, 42] mechanisms. Proposals are first generated in [2, 3] using selective search [30] or edge boxes [43], which are further classified and the classification results are merged to match the image

labels. Zhou et al. [42] and Zhang et al. [40] aim to find out the relationship between the neural responses of image regions and classification results, and then locate top activations area as detections. Singh et al. [27] propose to improve the localization map by randomly hiding some regions during training, so as to force the network to look for other discriminative parts. However, without effective guidance, this attempt is blind and inefficient. Recently, Wei et al. [36] employ Adversarial Erasing (AE) approach to discover more object regions in images by training classification network repeatedly, with discriminative regions erasing of different degrees, which is somewhat impractical and time-consuming. Our work differs from these methods in designing an Online Adversarial Erasing (OAE) approach which only needs to train a network for entire regions mining.

Weakly Supervised Temporal Action Localization. Action localization in temporal domain [5, 15, 17, 38] is similar to object detection in spatial domain [8, 9, 18, 20, 21], as well as the case under weak supervision. WSTAL aims to locate action instances in untrimmed videos including both temporal boundaries and action categories while relying on video-level class label only. Based on the idea proposed in [3], Wang et al. [33] formulate this task as a proposal-based classification problem, where temporal proposals are extracted with the priors of action shot. However, the use of softmax function across proposals blocks it from distinguishing multiple action instances. Singh et al. [27] hide temporal regions to force attention learning. However, it's not applicable owing to the complexity and varied lengths of videos. In our work, the Pyramid Attention Module (PAM) is proposed to hierarchically classify the videos from local to global, thus the pyramidal attention map is generated by combining feature maps from different levels, which can be scale-invariant to the action instances.

3 Our Approach

3.1 Problem Definition

We denote an untrimmed video as $X_v = \{x_t\}_{t=1}^{l_v}$, where l_v is the number of frames and x_t is the t -th frame in X_v . Each video X_v is annotated with a set of temporal action instances $\Phi_v = \{\phi_n = (t_n^s, t_n^e, \varphi_n)\}_{n=1}^{N_v}$, where N_v is the number of temporal action instances in X_v , and t_n^s, t_n^e, φ_n are starting time, ending time and category of instance ϕ_n respectively, where $\varphi_n \in \{1, \dots, C\}$ and C is the number of action categories. During training phase, only the video-level action label set $\psi_v = \{\varphi_n\}_{n=1}^{N_v}$ is given, and during test phase, Φ_v need to be predicted.

3.2 Video Features Encoding

To apply CPMN, first feature representations need to be extracted to describe visual content of the input video in our work. UntrimmedNet [33] is employed as visual encoder, since this kind of architecture using multiple two-stream networks has shown great performance and becomes a prevalent practice adopted in action recognition and temporal action localization tasks.

Given a video containing l_v frames, we use video unit as the basic processing unit in our framework for computational efficiency. Hence the video is divided into l_v/n_u consecutive video units without overlap, where n_u is the frame number of a unit. Then we compose a unit sequence $U = \{u_j\}_{j=1}^{l_u}$ from the video X_v , where l_u is the number of units. A video unit can be represented as $u_j = \{x_t\}_{t=f_s}^{f_s+n_u}$, where f_s is the starting frame and $f_s + n_u$ is the ending frame. Each unit is fed to the pre-trained visual encoder to extract representation. Concretely, the center RGB frame inside a unit is processed by spatial network and stacked optical flow derived around the center frame is processed by temporal network, then we concatenate output scores of UntrimmedNet [33] in fc-action layer to form the feature vector $f_{u_j} = \{f_{S,u_j}, f_{T,u_j}\}$, where f_{S,u_j} and f_{T,u_j} are output score vector of spatial and temporal network respectively with length G . Final, the unit-level feature sequence $F = \{f_{u_j}\}_{j=1}^{l_u}$ is used as input of CPMN.

3.3 Cascaded Pyramid Mining Network

To generate high quality detections with accurate temporal regions under weak supervision, we propose a multi-stage framework to achieve this goal. In CPMN, we first design a module to discover the entire action regions in a cascaded way. Then we introduce another module to combine the temporal feature maps from pyramidal feature hierarchy for prediction, making it possible to handle action instances with varied lengths. Final, we fuse the results from these two modules for action localization in temporal domain.

Network Architecture. The architecture consists of three sub-modules: cascaded classification module, pyramid attention module and temporal action localization module. *Cascaded classification module* is a two-stage model which includes two classifiers as shown in Fig. 2, aiming to mine different but complementary regions of target action in the video through a cascaded manner. *Pyramid attention module* is proposed to generate the class probability of each input unit feature, through classifying the input feature sequence with hierarchical resolutions separately. The architecture of this module is illustrated in Fig. 3. Final, *temporal action localization module* fuses the cascaded localization sequence and the pyramidal attention map to make it more accurate.

Cascaded Classification Module. The goal of this module is to locate the entire regions of target actions in the video, where two cascaded classifiers are needed. As shown in Fig. 2, the Cascaded Classification Module (CCM) contains two separate classifiers with the same structure. In each stage (i.e. the localization stage and the mining stage), the classifier handles the input unit-level features and adopts two 1-D temporal convolutional layers followed by a global average pooling layer to get the video-level representation, which is then passed through a fully connected (FC) layer and a Sigmoid layer for video classification. The two convolutional layers have the same configurations: kernel size 3, stride 1 and 512 filters with ReLU activation. Denote the last convolutional features and the averaged feature representation as $Z \in \mathbb{R}^{T \times K}$ and $\bar{Z} = \{\frac{\sum_t Z_t^{(k)}}{T}\}_{k=1}^K$ respectively, where T is the length of input feature sequence, K is the channel

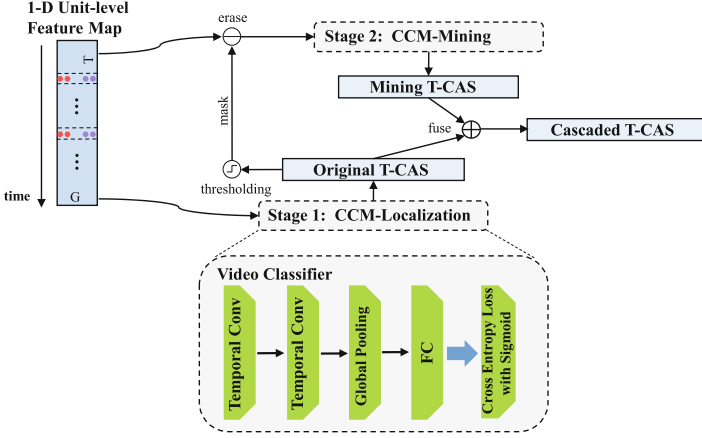


Fig. 2. Architecture of the cascaded classification module. The extracted unit-level feature representations are fed to two cascaded video classifiers for localization sequence inference individually. The two classifiers of the same structure share the input feature maps, and we erase the input features of discriminative regions highlighted by the first classifier, to drive the second classifier to discover more relevant regions of target actions in the video. Final, the two T-CASs are integrated for a better localization.

number and $Z_t(k)$ is the k -th feature map at time t . Then based on the idea in [42], we derive the 1-D Temporal Class Activation Sequence (T-CAS). We denote $w^c(k)$ as the k -th element of the weight matrix $\mathbf{W} \in \mathbb{R}^{K \times C}$ in the classification layer corresponding to class c . The input to the Sigmoid layer for class c is

$$s^c = \sum_{k=1}^K w^c(k) \bar{Z}(k) = \sum_{k=1}^K w^c(k) \sum_{t=1}^T Z_t(k) = \sum_{t=1}^T \sum_{k=1}^K w^c(k) Z_t(k), \quad (1)$$

$$M_t^c = \sum_{k=1}^K w^c(k) Z_t(k), \quad (2)$$

where M_t^c is denoted as T-CAS of class c , which indicates the activations of each unit feature contributing to a specific class of the video.

Then we conduct a threshold on the T-CAS obtained in the first stage to generate a mask which represents the discriminative regions discovered by the first classifier, and the mask is used to erase the input features of the second stage for classification. Such an online adversarial erasing operation allows the second classifier to leverage features from other regions for supporting the video-level labels. Final, we integrate the two T-CASs, $M^c(A)$ and $M^c(B)$, which are generated in the two stages separately, to form the cascaded localization sequence $M^c(Cas)$. Concretely, $M_t^c(Cas) = \max\{M_t^c(A), M_t^c(B)\}$, where $M_t^c(Cas)$ is the t -th element in the cascaded action localization sequence of class c .

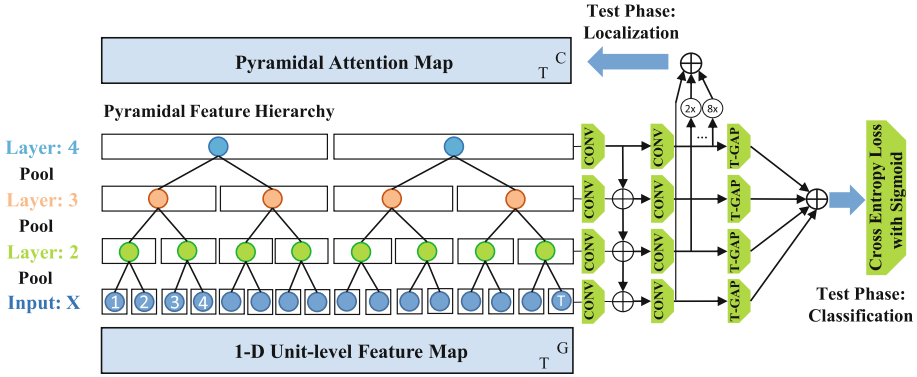


Fig. 3. Architecture of the pyramid attention module used for generating the multi-scale attention map. The pyramid attention module consists of several branches based on feature representations of hierarchical resolutions, for the purpose of classifying the video from local to global. And a global average pooling layer is employed to encode the video-level prediction of each branch and then we aggregate the multiple prediction results from different levels. Final, the pyramidal attention map is constructed through combining the multi-scale feature maps in the prediction layers during test phase.

Pyramid Attention Module. This module is designed to handle action instances with varied lengths. We achieve this goal in two steps. First, we semantically classify the feature representations with hierarchical temporal resolutions individually, aiming at processing the input unit-level feature sequence from local to global. Then we combine temporal feature maps in prediction layers from different levels to form the pyramidal attention map.

As shown in Fig. 3, we stack three 1-D max pooling layers on the input feature sequence, thus obtain the pyramidal feature hierarchy. After subsampling with a scaling step of 2 layer by layer, the feature sequence length of the l -th layer is $T_l = \frac{T}{2^{l-1}}$. Then for each level, we first conduct a temporal convolutional layer with kernel size 1, stride 1 and 512 filters to handle the feature sequence. Among these levels, we use lateral connections to exploit hierarchical context. Then we employ another convolutional layer to predict the classification scores of units associated with each feature map respectively. Each prediction layer consists of C filters with kernel size 1 and stride 1. And we continue to append a global average pooling layer on the label maps $\mathbf{K}_l \in \mathbb{R}^{T_l \times C}$ of each level separately to aggregate the video-level predictions. Final, we average among these prediction results to match the video-level class labels.

During test phase, we form the class heatmaps $\mathbf{H} = \{H^c\}_{c=1}^C$ by combining the output label maps in prediction layers from different levels. For example, with the coarser-resolution label map generated in l -th layer, we upsample the temporal resolution by a factor of 2^{l-1} through repeating the score vector in temporal dimension. Then every class heatmap is normalized to $[0, 1]$ as follows,

$$H^c = (H^c - \min(H^c)) / (\max(H^c) - \min(H^c)), \tag{3}$$

where H^c is the heatmap for class c , which indicates the class probability of each unit feature semantically.

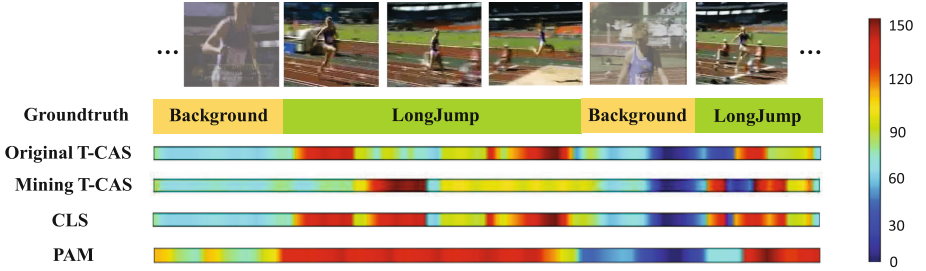


Fig. 4. Illustration of the comparison of ground-truth temporal intervals, original T-CAS, mining T-CAS, cascaded localization sequence (CLS) and pyramidal attention map (PAM) for the *LongJump* action class on THUMOS14.

Temporal Action Localization Module. The goal of this module is to fuse the cascaded action localization sequence and the pyramidal attention map obtained above for temporal action localization. As shown in Fig. 4, the CLS can identify more regions of target actions but some false positive regions are also activated by mistake. Since the class heatmap is generated using element-wise addition of label maps at multiple scales, the results are too smooth to indicate the accurate action boundaries. But they can provide important temporal priors to constrain the CLS. We let the class heatmap play the role of an attention map, which is used to correct the false positive regions and reduce missing detections. Then we fuse CLS and PAM to generate the attention-based cascaded T-CASs respectively as follows,

$$\Phi_{t,RGB}^c = \text{sigmoid}(M_{t,RGB}^c(Cas)) \cdot H_{t,RGB}^c, \quad (4)$$

$$\Phi_{t,Flow}^c = \text{sigmoid}(M_{t,Flow}^c(Cas)) \cdot H_{t,Flow}^c. \quad (5)$$

Then for each action class, we conduct a threshold on the $\Phi_{t,RGB}^c$ and $\Phi_{t,Flow}^c$ separately, and different from the method [42] which only retains the bounding box that covers the largest connected components, we keep all connected units that pass the predefined threshold θ_{TCAS} from each class and modality.

3.4 Training of CPMN

The training details of the cascaded classification module and the pyramid attention module in CPMN are introduced in this section.

Algorithm 1. Training procedure of cascaded classification module.

Input: Training data, $\Theta_{cas} = \{\Theta_{cas}(X_v)\}_{v=1}^D$; threshold, ζ ;
Output: Cascaded T-CAS, $M^c(Cas)$;
 1: **function** MAIN()
 2: **while** training is not convergent **do**
 3: $M^c(A), M^c(B) \leftarrow \text{CAS_TRAIN}(\Theta_{cas}(X_v), \zeta)$
 4: **end while**
 5: $M^c(Cas) \leftarrow \max(M_t^c(A), M_t^c(B))$
 6: **return** $M^c(Cas)$
 7: **end function**
 8: **function** CAS_TRAIN($\Theta_{cas}(X_v), \zeta$)
 9: Extract the feature sequence $F'(X_v)$
 10: Generate the original T-CAS $M^c(A) \leftarrow \text{infer}(F'(X_v), \psi_v)$
 11: Generate the mask $M^c(mask) \leftarrow M^c(A) > \zeta$
 12: Obtain the erased feature sequence $F'_{erase}(X_v) \leftarrow \text{erase}(F'(X_v), M^c(mask))$
 13: Generate the mining T-CAS $M^c(B) \leftarrow \text{infer}(F'_{erase}(X_v), \psi_v)$
 14: **return** $M^c(A), M^c(B)$
 15: **end function**

Cascaded Classification Module. Given an input video X_v , we form unit sequence U and extract corresponding feature sequence F with length l_u . Since an untrimmed video usually comes up with extremely irrelevant frames with action instances only occupying small parts, we totally test three sampling methods to simplify the feature sequence for computational cost reduction: (1) uniform sampling: units are extracted with a regular interval σ from U , thus the final unit sequence and feature sequence are $U' = \{u'_j\}_{j=1}^{l'_u}$ and F' separately, where $l'_u = \frac{l_u}{\sigma}$; (2) sparse sampling: we first divide the video into P segments $\{S_1, S_2, \dots, S_P\}$ with equal length, then during each training epoch, we randomly sample one unit from each segment to form the unit sequence of length P ; (3) shot-based sampling: considering the action structure, we sample the unit sequence U based on action shots, which are generated by shot boundary detector [28]. Evaluation results of these sampling methods are shown in Sect. 4.3.

After sampling, we construct the training data of each video as $\Theta_{cas}(X_v) = \{U'(X_v), F'(X_v), \psi_v\}$ and taking it as input, the first classifier leverages the most discriminative regions for classification while the second classifier handles the erased feature sequence for entire regions mining. We test different erasing thresholds ζ from 0.3 to 0.7 and the evaluation results are shown in Sect. 4.3. And we employ cross-entropy loss and l_2 regularization loss as final loss function to train the two multi-label classifiers separately:

$$L_{cas} = \sum_{v=1}^D \psi_v \cdot \log(y_v^{predict}) + \lambda \cdot L_2(\Xi_{cas}), \quad (6)$$

where $y_v^{predict}$ is the predicted class scores of the video and D is the number of training videos. λ balances the cross-entropy loss and l_2 loss, and Ξ_{cas} is the cascaded classification model. Algorithm 1 illustrates the training procedure.

Pyramid Attention Module. The pyramid attention module is trained to handle the action instances with arbitrary intervals. Considering the maximum length d_{max} of ground-truth action instances in dataset, we slide windows with length T_ω which can cover the d_{max} . The training data of video X_v is constructed as $\Theta_{mul}(X_v) = \{\Omega = \{U_\omega, F_\omega\}_{\omega=1}^{N_\omega}, \psi_v\}$, where N_ω is number of windows. Taking a sliding window with corresponding feature sequence F_ω as input, the pyramid attention module generates label maps with different lengths, then combine these maps in a bottom-up way and concatenate all windows results to form the video pyramidal attention maps. The multi-label cross-entropy loss function is also adopted to train the pyramid attention module.

3.5 Prediction and Post-processing

During prediction, we follow the same data preparation procedures of training phase to prepare the testing data, except for the following two changes: (1) in the cascaded classification module, we use uniform sampling method to sample the input feature sequence in order to increase the prediction speed and guarantee the stable results; (2) in the pyramid attention module, if the length of input video is shorter than T_ω , we will pad the input feature sequence to T_ω so that there is at least one window for the multi-tower network to predict. Then given a video X_v , we use CPMN to generate proposal set $\Gamma = \{\tau_n\}_{n=1}^{N_p}$ based on the thresholding attention-based cascaded T-CAS of top-2 predicted classes [33], where N_p is the number of candidate proposals and we set the threshold θ_{TCAS} as 20% of the max value of the derived T-CAS. For each proposal denoted by $[t_{start}, t_{end}]$, we first calculate the mean attention-based cascaded T-CAS among the temporal range of the proposal as p_{act} :

$$p_{act} = \sum_{t=t_{start}}^{t_{end}} \frac{M_{t,RGB}^c(Cas) \cdot H_{t,RGB}^c + M_{t,Flow}^c(Cas) \cdot H_{t,Flow}^c}{2 \cdot (t_{end} - t_{start} + 1)}, \quad (7)$$

then we fuse p_{act} and the class scores p_{class} with multiplication to get the confidence score p_{conf} :

$$p_{conf} = p_{act} \cdot p_{class}. \quad (8)$$

Since we keep all connected units that pass the predefined threshold θ_{TCAS} from each class and each modality as proposals, we may generate multiple predictions with different overlap. Then we conduct non-maximum suppression with predefined threshold θ_{NMS} in these prediction results to remove the redundant predictions of confidence score p_{conf} . Finally we get the prediction instances set $\Gamma' = \{\tau'_n\}_{n=1}^{N'_p}$, where N'_p is the number of final prediction instances.

4 Experiments

4.1 Dataset and Setup

Dataset. **ActivityNet-1.3** [4] is a large-scale video dataset for action recognition and temporal action localization tasks used in the ActivityNet Challenge 2017 and 2018, which consists of 10,024 videos for training, 4,926 for validation and 5,044 for testing, with 200 action classes annotated. Each video is annotated with average 1.5 temporal action instances. **THUMOS14** [14] dataset contains 1010 videos for validation and 1574 videos for testing with video-level labels of 101 action classes, while only a subset of 200 and 213 videos separately are temporally annotated among 20 classes. We train our model with the validation subset without using the temporal annotations. In this section, we compare our method with state-of-the-art methods on both ActivityNet-1.3 and THUMOS14.

Evaluation Metrics. Following the conventions, we use mean Average Precision (mAP) as evaluation metric, where Average Precision (AP) is calculated on each class separately. We report mAP values at different Intersection over Union (IoU) thresholds. On ActivityNet-1.3, mAP with IoU thresholds $\{0.5, 0.75, 0.95\}$ and average mAP with IoU thresholds set $\{0.5 : 0.05 : 0.95\}$ are used. On THUMOS14, mAP with IoU thresholds $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ are used.

Implementation Details. We use the UntrimmedNet [33] and TSN [34] for visual feature encoding of THUMOS14 and ActivityNet-1.3 separately, where ResNet network [10] is used as spatial network and BN-Inception network [12] is adopted as temporal network. The two visual encoders are both implemented using Caffe [13] and the TSN is pre-trained on ActivityNet-1.3. During feature extraction, the frame number of a unit n_u is set to 5 on THUMOS14 and is set to 16 on ActivityNet-1.3. In CPMN, the cascaded classification module and the pyramid attention module are both implemented using TensorFlow [1]. On both datasets, the two modules are both trained with batch size 16 and learning rate 0.001 for 30 epochs, then 0.0001 for another 120 epochs. The erasing threshold ζ used in CCM is 0.4 and the window size T_w used in the PAM is 64. Besides, the regularization term λ is 0.0025. For NMS, we set the threshold θ_{NMS} to 0.3 on THUMOS14 and 0.5 on ActivityNet-1.3 by empirical validation.

4.2 Comparison with State-of-the-Art Methods

We first evaluate the overall results of our proposed framework for action localization and compare our method with several state-of-the-art approaches including both fully and weakly supervised methods. Table 1 illustrates the localization performance on the THUMOS14 dataset. We can observe that our proposed algorithm achieves better performance than the two existing weakly supervised methods, and is even competitive to some fully supervised approaches. For example, when the IoU threshold α is 0.5, the mAP of our method is more than twice higher as the one of [27], which significantly convinces that our proposed online adversarial erasing strategy is more reasonable than randomly hiding. And the

substantial performance gaining over the previous works under different IoU thresholds confirms the effectiveness of our CPMN.

We also present our results on the validation set of the ActivityNet-1.3 dataset to further validate our localization performance. In ActivityNet-1.3, since the length of videos is not too long like THUMOS14, we directly resize the extracted feature sequence to length $l_w = 64$ by linear interpolation. And we choose attention-based cascaded T-CAS of top-1 class [35] as our detections. The evaluation results on ActivityNet-1.3 are shown in Table 2, from which we can see that our algorithm is generalized enough to this dataset, and significantly outperforms most fully supervised approaches with convincing performance. Meanwhile, we are the first to evaluate weakly supervised method on this dataset.

Table 1. Comparison of our method with other state-of-the-arts on THUMOS14 dataset for action localization, including both full supervision and weak supervision.

Supervision	Method	mAP@ <i>tIoU</i> (α)						
		0.1	0.2	0.3	0.4	0.5	0.6	0.7
Fully supervised	Oneata et al. [19]	36.6	33.6	27.0	20.8	14.4	-	-
	Richard et al. [22]	39.7	35.7	30.0	23.2	15.2	-	-
	Shou et al. [24]	47.7	43.5	36.3	28.7	19.0	10.3	5.3
	Yuan et al. [39]	51.0	45.2	36.5	27.8	17.8	-	-
	Lin et al. [15]	50.1	47.8	43.0	35.0	24.6	15.3	7.7
	Zhao et al. [41]	60.3	56.2	50.6	40.8	29.1	-	-
	Gao et al. [7]	60.1	56.7	50.1	41.3	31.0	19.1	9.9
Weakly supervised	Singh et al. [27]	36.4	27.8	19.5	12.7	6.8	-	-
	Wang et al. [33]	44.4	37.7	28.2	21.1	13.7	-	-
	CPMN	47.1	41.6	32.8	24.7	16.1	10.1	5.5

Table 2. Results on validation set of ActivityNet-1.3 in terms of mAP@*tIoU* and average mAP. Note that all compared methods are fully supervised.

Supervision	Method	0.5	0.75	0.95	Average
Fully supervised	Singh et al. [26]	34.5	-	-	-
	Heilbron et al. [11]	40.00	17.90	4.70	21.70
	Wang et al. [35]	42.28	3.76	0.05	14.85
	Shou et al. [23]	43.83	25.88	0.21	22.77
	Xiong et al. [37]	39.12	23.48	5.49	23.98
	Lin et al. [16]	48.99	32.91	7.87	32.26
Weakly supervised	CPMN	39.29	24.09	6.71	24.42

Figure 6 visualizes the localization performance of our proposed method on THUMOS14 dataset. As shown in Fig. 6(a), the video includes two different action classes, and our Attention-based Cascaded (AC) T-CAS of corresponding class can localize the entire regions of target actions individually. It confirms that the two cascaded classifiers are successful in mining different but complementary target regions. And in Fig. 6(b), there are many action instances densely distributed in the video, however, our method can effectively highlight the regions which may contain actions, which demonstrates that our modified T-CAS is able to generate dense detections. Besides, the Fig. 6(c) presents a video with similar appearance and little dynamic motions along the temporal dimension, which is a difficult scene even for humans to distinguish between adjacent frames, resulting in inevitably missing detections. Nevertheless, our algorithm is still robust enough to discover some discriminative regions even with some false positives. To conclude, the proposed cascaded action localization sequence together with the pyramidal attention map is intuitive to promote the overall localization performance.

4.3 Ablation Study

In this section, we evaluate CPMN with different implementation variations to study their effects and investigate the contribution of several components proposed in our network. All the experiments for our ablation study are conducted on THUMOS14 dataset.

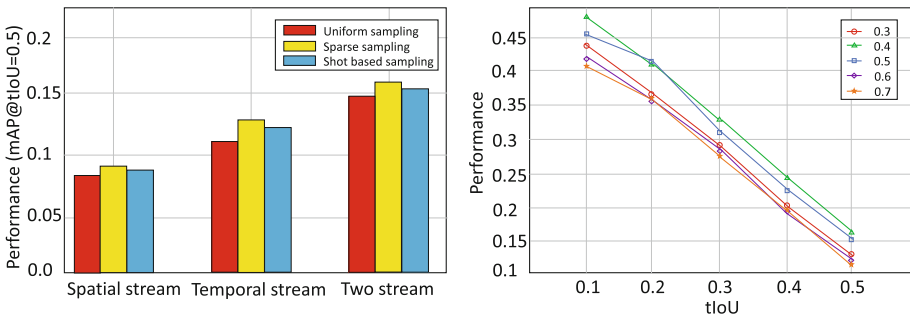


Fig. 5. Evaluation of the CPMN with different sampling methods (left) and erasing thresholds (right) used in the CCM on THUMOS14.

Sampling Strategy. The input untrimmed video usually exists substantial redundant frames which are useless for the cascaded model to leverage discriminative regions for recognition. In order to reduce computational cost, we sample the input feature sequence instead of using all units for video categorization. We first evaluate the Cascaded Classification Module (CCM) with different sampling methods, including uniform sampling, sparse sampling and shot-based sampling. The evaluation results are illustrated in Fig. 5 (left). We can observe

that the shot-based sampling method which takes action structure into consideration shows better performance than uniform sampling and sparse sampling which serves as a data augmentation step leads to the best performance. Therefore we finally adopt sparse sampling to sample the input feature sequence of the CCM during training phase.

Erasing Threshold. We continue to study the influence of different erasing thresholds ζ which we use to identify the discriminative regions highlighted by the first classifier of the CCM and create a mask for online adversarial erasing step. As shown in Fig. 5 (right), we test the thresholds from 0.3 to 0.7 and report the performance over different IoU. We observe that when the threshold $\zeta = 0.4$, the localization performance of the two-stage model is boosted and the

Table 3. Performance with respect to architecture choices. In first column, only original T-CAS without adversarial mining and attention map is used for action localization.

Original T-CAS	✓	✓	✓
Mining T-CAS		✓	✓
Pyramidal Attention Map (PAM)			✓
mAP ($\alpha = 0.5$)	11.4	14.5	16.1

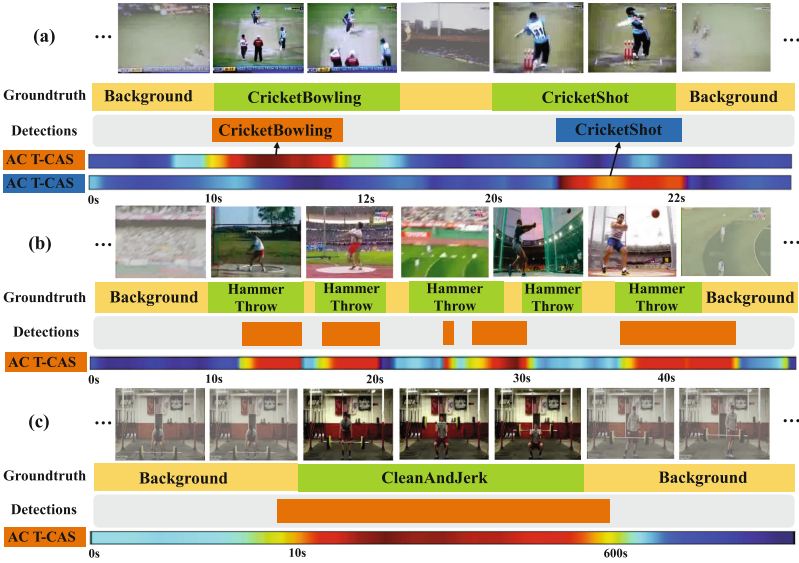


Fig. 6. Visualization of the action instances located by CPMN on THUMOS14. Figure (a) shows that entire regions of two actions are separately located in corresponding T-CAS. Figure (b) shows that dense predictions can be generated by our approach with less missing detections. Figure (c) shows that even the instances with small length and similar appearance, our model still can locate it with less false positives.

value larger or smaller than 0.4 would fail to encourage the second classifier to mine entire regions of target actions and may bring background noise.

Architecture of CPMN. As shown in Table 3, we investigate the contribution of each module proposed in our method. We choose the architecture sharing the same idea with CAM [42] for discriminative localization as our baseline model without online erasing step to mine entire regions and the pyramidal attention map. The comparison results reveal that the original T-CAS together with the mining T-CAS can promote the performance and with the help of pyramidal attention map, the localization performance can be further boosted. These observations suggest that these modules are all effective and indispensable in CPMN. Note that we also test the CCM with more classifiers. Specifically, we add the third classifier to handle the erased feature maps guided by the first two classifiers. However, we don't find any significant improvement.

5 Conclusion

In this paper, we propose the Cascaded Pyramid Mining Network (CPMN) for weakly supervised temporal action localization. Our method includes two main steps: cascaded adversarial mining and pyramidal attention map inference. Cascaded adversarial mining is realized by designing two cascaded classifiers to collaborate on locating entire regions of target actions with a novel online adversarial erasing step. And the pyramidal attention map tries explicitly handling the falsely activated regions and missing detections in the localization sequence, which is inferred upon the prediction results of the multi-tower network. Extensive experiments reveal that CPMN significantly outperforms other state-of-the-art approaches on both THUMOS14 and ActivityNet-1.3 datasets.

References

1. Abadi, M., Agarwal, A., Barham, P., et al.: Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint [arXiv:1603.04467](https://arxiv.org/abs/1603.04467) (2016)
2. Bai, P., Tang, X., Wang, X., Liu, W.: Multiple instance detection network with online instance classifier refinement. In: CVPR, pp. 4322–4328 (2017)
3. Bilen, H., Vedaldi, A.: Weakly supervised deep detection networks. In: CVPR, pp. 2846–2854 (2016)
4. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: ActivityNet: a large-scale video benchmark for human activity understanding. In: CVPR, pp. 961–970 (2015)
5. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster R-CNN architecture for temporal action localization. In: CVPR, pp. 1130–1139 (2018)
6. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: CVPR, pp. 1933–1941 (2016)
7. Gao, J., Yang, Z., Nevatia, R.: Cascaded boundary regression for temporal action detection. arXiv preprint [arXiv:1705.01180](https://arxiv.org/abs/1705.01180) (2017)

8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587 (2014)
9. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask-RCNN. [arXiv:1703.06870v2](https://arxiv.org/abs/1703.06870v2) (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
11. Heilbron, F.C., Barrios, W., Escorcia, V., Ghanem, B.: SCC: semantic context cascade for efficient action detection. In: CVPR, pp. 3175–3184 (2017)
12. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint [arXiv:1502.03167](https://arxiv.org/abs/1502.03167) (2015)
13. Jia, Y., et al.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
14. Jiang, Y., et al.: THUMOS challenge: action recognition with a large number of classes. In: Computer Vision-ECCV Workshop 2014 (2014)
15. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: Proceedings of the 2017 ACM on Multimedia Conference, pp. 988–996. ACM (2017)
16. Lin, T., Zhao, X., Shou, Z.: Temporal convolution based action proposal: submission to activitynet 2017. arXiv preprint [arXiv:1707.06750](https://arxiv.org/abs/1707.06750) (2017)
17. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: boundary sensitive network for temporal action proposal generation. arXiv preprint [arXiv:1806.02964](https://arxiv.org/abs/1806.02964) (2018)
18. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR, vol. 1, p. 4 (2017)
19. Oneata, D., Verbeek, J., Schmid, C.: The LEAR submission at THUMOS2014. THUMOS Action Recognition challenge (2014)
20. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR, pp. 779–788 (2016)
21. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. arXiv preprint [arXiv:1612.08242](https://arxiv.org/abs/1612.08242) (2016)
22. Richard, A., Gall, J.: Temporal action detection using a statistical language model. In: CVPR, pp. 3131–3140 (2016)
23. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: CVPR, pp. 1417–1426. IEEE (2017)
24. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage CNNs. In: CVPR, pp. 1049–1058 (2016)
25. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
26. Singh, G., Cuzzolin, F.: Untrimmed video classification for activity detection: submission to activitynet challenge. arXiv preprint [arXiv:1607.01979](https://arxiv.org/abs/1607.01979) (2016)
27. Singh, K.K., Lee, Y.J.: Hide-and-peek: forcing a network to be meticulous for weakly-supervised object and action localization. In: ICCV (2017)
28. Su, H., Zhao, X., Lin, T., Fei, H.: Weakly supervised temporal action detection with shot-based temporal pooling network. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) ICONIP 2018. LNCS, vol. 11304, pp. 426–436. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04212-7_37
29. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV, pp. 4489–4497 (2015)
30. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV **104**, 154–171 (2013)

31. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR, pp. 3169–3176. IEEE (2011)
32. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV, pp. 3551–3558 (2013)
33. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: UntrimmedNets for weakly supervised action recognition and detection. In: CVPR, vol. 2 (2017)
34. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
35. Wang, R., Tao, D.: UTS at activitynet 2016. ActivityNet large scale activity recognition challenge 2016, 8 (2016)
36. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: a simple classification to semantic segmentation approach. In: CVPR, vol. 1, p. 3 (2017)
37. Xiong, Y., Zhao, Y., Wang, L., Lin, D., Tang, X.: A pursuit of temporal accuracy in general activity detection. arXiv preprint [arXiv:1703.02716](https://arxiv.org/abs/1703.02716) (2017)
38. Xu, H., Das, A., Saenko, K.: R-C3D: region convolutional 3D network for temporal activity detection. In: ICCV, pp. 5794–5803 (2017)
39. Yuan, Z.H., Stroud, J.C., Lu, T., Deng, J.: Temporal action localization by structured maximal sums. In: CVPR, vol. 2, p. 7 (2017)
40. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. IJCV **126**, 1084–1102 (2018)
41. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: ICCV, vol. 2 (2017)
42. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR, pp. 2921–2929 (2016)
43. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 391–405. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_26