

Adversarial Background-Aware Loss for Weakly-supervised Temporal Activity Localization

Kyle Min and Jason J. Corso

University of Michigan, Ann Arbor, MI 48109
`{kylemin,jjcorso}@umich.edu`

Abstract. Temporally localizing activities within untrimmed videos has been extensively studied in recent years. Despite recent advances, existing methods for weakly-supervised temporal activity localization struggle to recognize when an activity is not occurring. To address this issue, we propose a novel method named A2CL-PT. Two triplets of the feature space are considered in our approach: one triplet is used to learn discriminative features for each activity class, and the other one is used to distinguish the features where no activity occurs (i.e. background features) from activity-related features for each video. To further improve the performance, we build our network using two parallel branches which operate in an adversarial way: the first branch localizes the most salient activities of a video and the second one finds other supplementary activities from non-localized parts of the video. Extensive experiments performed on THUMOS14 and ActivityNet datasets demonstrate that our proposed method is effective. Specifically, the average mAP of IoU thresholds from 0.1 to 0.9 on the THUMOS14 dataset is significantly improved from 27.9% to 30.0%.

Keywords: A2CL-PT, temporal activity localization, adversarial learning, weakly-supervised learning, center loss with a pair of triplets

1 Introduction

The main goal of temporal activity localization is to find the start and end times of activities from untrimmed videos. Many of the previous approaches are fully supervised: they expect that ground-truth annotations for temporal boundaries of each activity are accessible during training [22, 20, 26, 32, 2, 11, 14]. However, collecting these frame-level activity annotations is time-consuming and difficult, leading to annotation noise. Hence, a weakly-supervised version has taken foot in the community: here, one assumes that only video-level ground-truth activity labels are available. These video-level activity annotations are much easier to collect and already exist across many datasets [8, 23, 6, 15, 31], thus weakly-supervised methods can be applied to a broader range of situations.

Current work in weakly-supervised temporal activity localization shares a common framework [12, 16, 17, 19, 9]. First, rather than using a raw video, they

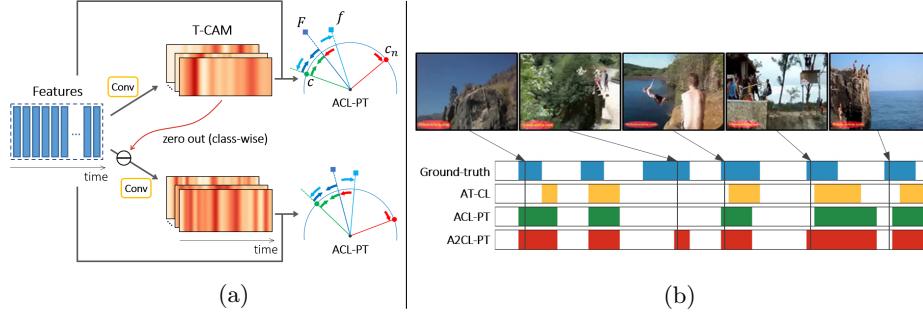


Fig. 1. (a): An illustration of the proposed A2CL-PT. F and f are aggregated video-level features where f is designed to be more attended to the background features. c is their corresponding center and c_n is the negative center. A triplet of (F, c, c_n) is used to learn discriminative features. We propose to exploit another triplet of (c, F, f) which distinguishes background features from the activity-related features. We call this method of two triplets ACL-PT. In addition, we design our network with two parallel branches so that the two separate sets of centers can be learned in an adversarial way. We call our final proposed method A2CL-PT. (b): Sample frames of a video containing *Diving* activity class from THUMOS14 dataset [5] and the corresponding results of activity localization. It is shown that our final method A2CL-PT performs the best.

use a sequence of features extracted by deep networks where the features are much smaller than the raw video in size. Second, they apply a fully-connected layer to embed the pre-extracted features to the task-specific feature space. Third, they project the embedded features to the label space by applying a 1-D convolutional layer to those features. The label space has the same dimension as the number of activities, so the final output becomes a sequence of vectors that represents the classification scores for each activity over time. Each sequence of vectors is typically referred to as CAS (Class Activation Sequence) [21] or T-CAM (Temporal Class Activation Map) [17]. Finally, activities are localized by thresholding this T-CAM. T-CAM is sometimes applied with the softmax function to generate class-wise attention. This top-down attention represents the probability mass function for each activity over time.

An important component in weakly-supervised temporal activity localization is the ability to automatically determine background portions of the video where no activity is occurring. For example, BaS-Net [9] suggests using an additional suppression objective to suppress the network activations on the background portions. Nguyen et al. [18] proposes a similar objective to model the background contents. However, we argue that existing methods are not able to sufficiently distinguish background information from activities of interest for each video even though such an ability is critical to strong temporal activity localization.

To this end, we propose a novel method for the task of weakly-supervised temporal activity localization, which we call Adversarial and Angular Center Loss with a Pair of Triplets (A2CL-PT). It is illustrated in Fig. 1(a). Our key innovation is that we explicitly enable our model to capture the background re-

gion of the video while using an adversarial approach to focus on completeness of the activity learning. Our method is built on two triplets of vectors of the feature space, and one of them is designed to distinguish background portions from the activity-related parts of a video. Our method is inspired by the angular triplet-center loss (ATCL) [10] originally designed for multi-view 3D shape retrieval. Let us first describe what ATCL is and then how we develop our novel method of A2CL-PT.

In ATCL [10], a *center* is defined as a parameter vector representing the center of a cluster of feature vectors for each class. During training, the centers are updated by reducing the angular distance between the embedded features and their corresponding class centers. This groups together features that correspond to the same class and distances features from the centers of other class clusters (i.e. negative centers), making the learned feature space more useful for discriminating between classes. It follows that each training sample is a triplet of a feature vector, its center, and a negative center where the feature serves as an anchor.

Inspired by ATCL, we first formulate a loss function to learn discriminative features. ATCL cannot be directly applied to our problem because it assumes that all the features are of the same size, whereas an untrimmed video can have any number of frames. Therefore, we use a different feature representation at the video-level. We aggregate the embedded features by multiplying the top-down attention described above at each time step. The resulting video-level feature representation has the same dimension as the embedded features, so we can build a triplet whose anchor is the video-level feature vector (it is (F, c, c_n) in Fig. 1(a)). This triplet ensures that the embedded features of the same activity are grouped together and that they have high attention values at time steps when the activity occurs.

More importantly, we argue that it is possible to exploit another triplet. Let us call the features at time steps when some activity occurs *activity features*, and the ones where no activity occurs *background features*. The main idea is that the background features should be distinguished from the activity features for each video. First, we generate a new class-wise attention from T-CAM. It has higher attention values for the background features when compared to the original top-down attention. If we aggregate the embedded features with this new attention, the resulting video-level feature will be more attended to the background features than the original video-level feature is. In a discriminative feature space, the original video-level feature vector should be closer to its center than the new video-level feature vector is. This property can be achieved by using the triplet of the two different video-level feature vectors and their corresponding center where the center behaves as an anchor (it is (c, F, f) in Fig. 1(a)). The proposed triplet is novel and will be shown to be effective. Since we make use of a pair of triplets on the same feature space, we call it Angular Center Loss with a Pair of Triplets (ACL-PT).

To further improve the localization performance, we design our network to have two parallel branches which find activities in an adversarial way, also illus-

trated in Fig. 1(a). Using a network with a single branch may be dominated by salient activity features that are too short to localize all the activities in time. We zero out the most salient activity features localized by the first branch for each activity so that the second (adversarial) branch can find other supplementary activities from the remaining parts of the video. Here, each branch has its own set of centers which group together the features for each activity and one 1-D convolutional layer that produces T-CAM. The two adversary T-CAMs are weighted to produce the final T-CAM that is used to localize activities. We want to note that our network produces the final T-CAM with a single forward pass so it is trained in an end-to-end manner. We call our final proposed method Adversarial and Angular Center Loss with a Pair of Triplets (A2CL-PT). It is shown in Fig. 1(b) that our final method performs the best.

There are three main contributions in this paper:

- We propose a novel method using a pair of triplets. One facilitates learning discriminative features. The other one ensures that the background features are distinguishable from the activity-related features for each video.
- We build an end-to-end two-branch network by adopting an adversarial approach to localize more complete activities. Each branch comes with its own set of centers so that embedded features of the same activity can be grouped together in an adversarial way by the two branches.
- We perform extensive experiments on THUMOS14 and ActivityNet datasets and demonstrate that our method outperforms all the previous state-of-the-art approaches.

2 Related Work

Center loss (CL) [25] is recently proposed to reduce the intra-class variations of feature representations. CL learns a center for each class and penalizes the Euclidean distance between the features and their corresponding centers. Triplet-center loss (TCL) [4] shows that using a triplet of each feature vector, its corresponding center, and a nearest negative center is effective in increasing the inter-class separability. TCL enforces that each feature vector is closer to its corresponding center than to the nearest negative center by a pre-defined margin. Angular triplet-center loss (ATCL) [10] further improves TCL by using the angular distance. In ATCL, it is much easier to design a better margin because it has a clear geometric interpretation and is limited from 0 to π .

BaS-Net [9] and Nguyen et al. [18] are the leading state-of-the-art methods for weakly-supervised temporal activity localization. They take similar approaches to utilize the background portions of a video. There are other recent works without explicit usage of background information. Liu et al. [12] utilizes multi-branch network where T-CAMs of these branches differ from each other. This property is enforced by the diversity loss: the sum of the simple cosine distances between every pair of the T-CAMs. 3C-Net applies an idea of CL, but the performance is limited because CL does not consider the inter-class separability.

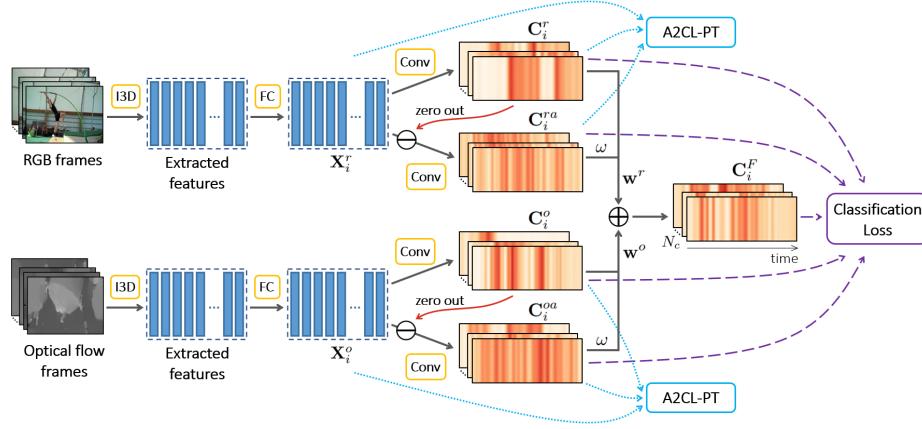


Fig. 2. An illustration of our overall architecture. It consists of two streams (RGB and optical flow), and each stream consists of two (first and adversarial) branches. Sequences of features are extracted from two input streams using pre-trained I3D networks [1]. We use two fully-connected layers with ReLU activation (FC) to compute the embedded features $\mathbf{X}_i^r, \mathbf{X}_i^o$. Next, T-CAMs $\mathbf{C}_i^r, \mathbf{C}_i^o$ are computed by applying 1-D convolutional layers (Conv). The most salient activity features localized by the first branch are zeroed out for each activity class, and the resulting features are applied with different 1-D convolutional layers (Conv) to produce $\mathbf{C}_i^{ra}, \mathbf{C}_i^{oa}$. Using the embedded features $\mathbf{X}_i^r, \mathbf{X}_i^o$ and T-CAMs $\mathbf{C}_i^r, \mathbf{C}_i^o, \mathbf{C}_i^{ra}, \mathbf{C}_i^{oa}$, we compute the term of A2CL-PT (Eq. 16). The final T-CAM \mathbf{C}_i^F is computed from the four T-CAMs and these T-CAMs are used to compute the loss function for classification (Eq. 19).

Using an end-to-end two-branch network that operates in an adversarial way is proposed in Adversarial Complementary Learning (ACoL) [30] for the task of weakly-supervised object localization. In ACoL, object localization maps from the first branch are used to erase the salient regions of the input feature maps for the second branch. The second branch then tries to find other complementary object areas from the remaining regions. To the best of our knowledge, we are the first to merge the idea of ACoL with center loss and to apply it to weakly-supervised temporal activity localization.

3 Method

The overview of our proposed method is illustrated in Fig. 2. The total loss function is represented as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{A2CL-PT}} + \mathcal{L}_{\text{CLS}} \quad (1)$$

where $\mathcal{L}_{\text{A2CL-PT}}$ and \mathcal{L}_{CLS} denote our proposed loss term and the classification loss, respectively. α is a hyperparameter to control the weight of A2CL-PT term. In this section, we describe each component of our method in detail.

3.1 Feature Embedding

Let us say that we have N training videos $\{v_i\}_{i=1}^N$. Each video v_i has its ground-truth annotation for video-level label $\mathbf{y}_i \in \mathbb{R}^{N_c}$ where N_c is the number of activity classes. $\mathbf{y}_i(j) = 1$ if the activity class j is present in the video and $\mathbf{y}_i(j) = 0$ otherwise. We follow previous works [19, 16] to extract the features for both RGB and optical flow streams. First, we divide v_i into non-overlapping 16-frame segments. We then apply I3D [1] pretrained on Kinetics dataset [6] to the segments. The intermediate D -dimensional ($D = 1024$) outputs after the global pooling layer are the pre-extracted features. For the task-specific feature embedding, we use two fully-connected layers with ReLU activation. As a result, sequences of the embedded features $\mathbf{X}_i^r, \mathbf{X}_i^o \in \mathbb{R}^{D \times l_i}$ are computed for RGB and optical flow stream where l_i denotes the temporal length of the features of the video v_i .

3.2 Angular Center Loss with a Pair of Triplets (ACL-PT)

For simplicity, we first look at the RGB stream. The embedded features \mathbf{X}_i^r are applied with a 1-D convolutional layer. The output is T-CAM $\mathbf{C}_i^r \in \mathbb{R}^{N_c \times l_i}$ which represents the classification scores of each activity class over time. We compute class-wise attention $\mathbf{A}_i^r \in \mathbb{R}^{N_c \times l_i}$ by applying the softmax function to T-CAM:

$$\mathbf{A}_i^r(j, t) = \frac{\exp(\mathbf{C}_i^r(j, t))}{\sum_{t'=1}^{l_i} \exp(\mathbf{C}_i^r(j, t'))} \quad (2)$$

where $j \in \{1, \dots, N_c\}$ denotes each activity class and t is for each time step. Since this top-down attention represents the probability mass function of each activity over time, we can use it to aggregate the embedded features \mathbf{X}_i^r :

$$\mathbf{F}_i^r(j) = \sum_{t=1}^{l_i} \mathbf{A}_i^r(j, t) \mathbf{X}_i^r(t) \quad (3)$$

where $\mathbf{F}_i^r(j) \in \mathbb{R}^D$ denotes a video-level feature representation for the activity class j . Now, we can formulate a loss function that is inspired by ATCL [10] on the video-level feature representations as follows:

$$\mathcal{L}_{\text{ATCL}}^r = \frac{1}{N} \sum_{i=1}^N \sum_{j: \mathbf{y}_i(j)=1} \max \left(0, \mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_j^r) - \mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_{n_{i,j}^r}^r) + m_1 \right) \quad (4)$$

where $\mathbf{c}_j^r \in \mathbb{R}^D$ is the center of activity class j , $n_{i,j}^r = \operatorname{argmin}_{k \neq j} \mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_k^r)$ is an index for the nearest negative center, and $m_1 \in [0, \pi]$ is an angular margin. It is based on the triplet of $(\mathbf{F}_i^r(j), \mathbf{c}_j^r, \mathbf{c}_{n_{i,j}^r}^r)$ that is illustrated in Fig. 1(a). Here, $\mathcal{D}(\cdot)$ represents the angular distance:

$$\mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_j^r) = \arccos \left(\frac{\mathbf{F}_i^r(j) \cdot \mathbf{c}_j^r}{\|\mathbf{F}_i^r(j)\|_2 \|\mathbf{c}_j^r\|_2} \right) \quad (5)$$

Optimizing the loss function of Eq. 4 ensures that the video-level features of the same activity class are grouped together and that the inter-class variations of those features are maximized at the same time. As a result, the embedded features are learned to be discriminative and T-CAM will have higher values for the activity-related features.

For the next step, we exploit another triplet. We first compute a new class-wise attention $\mathbf{a}_i^r \in \mathbb{R}^{N_c \times l_i}$ from T-CAM:

$$\mathbf{a}_i^r(j, t) = \frac{\exp(\beta \mathbf{C}_i^r(j, t))}{\sum_{t'=1}^{l_i} \exp(\beta \mathbf{C}_i^r(j, t'))} \quad (6)$$

where β is a scalar between 0 and 1. This new attention still represents the probability mass function of each activity over time, but it is supposed to have lower values for the activity features and higher values for the background features when compared to the original attention \mathbf{A}_i^r . Therefore, if we aggregate the embedded features \mathbf{X}_i^r using \mathbf{a}_i^r , the resulting new video-level feature \mathbf{f}_i^r should attend more strongly to the background features than \mathbf{F}_i^r is. This property can be enforced by introducing a different loss function based on the new triplet of $(\mathbf{c}_j^r, \mathbf{F}_i^r(j), \mathbf{f}_i^r(j))$ that is also illustrated in Fig. 1(a):

$$\mathcal{L}_{\text{NT}}^r = \frac{1}{N} \sum_{i=1}^N \sum_{j: \mathbf{y}_i(j)=1} \max(0, \mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_j^r) - \mathcal{D}(\mathbf{f}_i^r(j), \mathbf{c}_j^r) + m_2) \quad (7)$$

where the subscript NT refers to the new triplet and $m_2 \in [0, \pi]$ is an angular margin. Optimizing this loss function makes the background features more distinguishable from the activity features. Merging the two loss functions of Eq. 4 and Eq. 7 gives us a new loss based on a pair of triplets, which we call Angular Center Loss with a Pair of Triplets (ACL-PT):

$$\mathcal{L}_{\text{ACL-PT}}^r = \mathcal{L}_{\text{ATCL}}^r + \gamma \mathcal{L}_{\text{NT}}^r \quad (8)$$

where γ is a hyperparameter denoting the relative importance of the two losses.

Previous works on center loss [25, 4, 10] suggest using an averaged gradient (typically denoted as $\Delta \mathbf{c}_j^r$) to update the centers for better stability. Following this convention, the derivatives of each term of Eq. 8 with respect to the centers are averaged. For simplicity, we assume that the centers have unit length. Refer to the supplementary material for general case without such assumption. Let $\tilde{\mathcal{L}}_{\text{ATCL}_{i,j}}^r$ and $\tilde{\mathcal{L}}_{\text{NT}_{i,j}}^r$ be the loss terms inside the max operation of the i -th sample and of the j -th activity class as follows:

$$\tilde{\mathcal{L}}_{\text{ATCL}_{i,j}}^r = \mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_j^r) - \mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_{n_{i,j}^r}^r) + m_1 \quad (9)$$

$$\tilde{\mathcal{L}}_{\text{NT}_{i,j}}^r = \mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_j^r) - \mathcal{D}(\mathbf{f}_i^r(j), \mathbf{c}_j^r) + m_2 \quad (10)$$

Next, let $\mathbf{g}_{1_{i,j}}^r$ and $\mathbf{g}_{2_{i,j}}^r$ be the derivatives of Eq. 9 with respect to \mathbf{c}_j^r and $\mathbf{c}_{n_{i,j}^r}^r$, respectively; and let $\mathbf{h}_{i,j}^r$ be the derivative of Eq. 10 with respect to \mathbf{c}_j^r . For

example, $\mathbf{g}_{1_{i,j}}^r$ is given by:

$$\mathbf{g}_{1_{i,j}}^r = -\frac{\mathbf{F}_i^r(j)}{\sin(\mathcal{D}(\mathbf{F}_i^r(j), \mathbf{c}_j^r)) \|\mathbf{F}_i^r(j)\|_2} \quad (11)$$

Then, we can represent the averaged gradient considering the three terms:

$$\Delta \mathbf{c}_j^r = \Delta \mathbf{g}_{1_{i,j}}^r + \Delta \mathbf{g}_{2_{i,j}}^r + \Delta \mathbf{h}_{t_{i,j}}^r \quad (12)$$

For example, $\Delta \mathbf{g}_{1_{i,j}}^r$ is computed as follows:

$$\Delta \mathbf{g}_{1_{i,j}}^r = \frac{1}{N} \left(\frac{\sum_{i: \mathbf{y}_i(j)=1} \mathbf{g}_{1_{i,j}}^r \delta(\tilde{\mathcal{L}}_{\text{ATCL}_{i,j}}^r > 0)}{1 + \sum_{i: \mathbf{y}_i(j)=1} \delta(\tilde{\mathcal{L}}_{\text{ATCL}_{i,j}}^r > 0)} \right) \quad (13)$$

Here, $\delta(\text{condition}) = 1$ if the *condition* is true and $\delta(\text{condition}) = 0$ otherwise. Finally, the centers are updated using $\Delta \mathbf{c}_j^r$ for every iteration of the training process by a gradient descent algorithm. More details can be found in the supplementary material.

3.3 Adopting an adversarial approach (A2CL-PT)

We further improve the performance of the proposed ACL-PT by applying an adversarial approach inspired by ACoL [30]. For each stream, there are two parallel branches that operate in an adversarial way. The motivation is that a network with a single branch might be dominated by salient activity features that are not enough to localize all the activities in time. We zero out the most salient activity features localized by the first branch for activity class j of v_i as follows:

$$\mathbf{X}_{i,j}^{ra}(t) = \begin{cases} \mathbf{0}, & \text{if } \mathbf{C}_i^r(j, t) \in \text{top-}k_a \text{ elements of } \mathbf{C}_i^r(j) \\ \mathbf{X}_i^r(t), & \text{otherwise} \end{cases} \quad (14)$$

where $\mathbf{X}_{i,j}^{ra} \in \mathbb{R}^{D \times l_i}$ denotes the input features of activity class j for the second (adversarial) branch and k_a is set to $\lfloor \frac{l_i}{s_a} \rfloor$ for a hyperparameter s_a that controls the ratio of zeroed-out features. For each activity class j , a separate 1-D convolutional layer of the adversarial branch transforms $\mathbf{X}_{i,j}^{ra}$ to the classification scores of the activity class j over time. By iterating over all the activity classes, new T-CAM $\mathbf{C}_i^{ra} \in \mathbb{R}^{N_c \times l_i}$ is computed. We argue that \mathbf{C}_i^{ra} can be used to find other supplementary activities that are not localized by the first branch. By using the original features \mathbf{X}_i^r , new T-CAM \mathbf{C}_i^{ra} , and a separate set of centers $\{\mathbf{c}_j^{ra}\}_{j=1}^{N_c}$, we can compute the loss of ACL-PT for this adversarial branch $\mathcal{L}_{\text{ACL-PT}}^{ra}$ in a similar manner (Eq. 1-7). We call the sum of the losses of the two branches Adversarial and Angular Center Loss with a Pair of Triplets (A2CL-PT):

$$\mathcal{L}_{\text{A2CL-PT}}^r = \mathcal{L}_{\text{ACL-PT}}^r + \mathcal{L}_{\text{ACL-PT}}^{ra} \quad (15)$$

In addition, the losses for the optical flow stream $\mathcal{L}_{\text{ACL-PT}}^o$ and $\mathcal{L}_{\text{ACL-PT}}^{oa}$ are also computed in the same manner. As a result, the total A2CL-PT term is given by:

$$\mathcal{L}_{\text{A2CL-PT}} = \mathcal{L}_{\text{A2CL-PT}}^r + \mathcal{L}_{\text{A2CL-PT}}^o \quad (16)$$

3.4 Classification Loss

Following the previous works [19, 12, 16], we use the cross-entropy between the predicted pmf (probability mass function) and the ground-truth pmf of activities for classifying different activity classes in a video. We will first look at the RGB stream. For each video v_i , we compute the class-wise classification scores $\mathbf{s}_i^r \in \mathbb{R}^{N_c}$ by averaging *top-k* elements of \mathbf{C}_i^r per activity class where k is set to $\lceil \frac{l_i}{s} \rceil$ for a hyperparameter s . Then, the softmax function is applied to compute the predicted pmf of activities $\mathbf{p}_i^r \in \mathbb{R}^{N_c}$. The ground-truth pmf \mathbf{q}_i is obtained by l_1 -normalizing \mathbf{y}_i . Then, the classification loss for the RGB stream is:

$$\mathcal{L}_{\text{CLS}}^r = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{N_c} -\mathbf{q}_i(j) \log (\mathbf{p}_i^r(j)) \quad (17)$$

The classification loss for the optical flow stream $\mathcal{L}_{\text{CLS}}^o$ is computed in a similar manner. $\mathcal{L}_{\text{CLS}}^{ra}$ and $\mathcal{L}_{\text{CLS}}^{oa}$ of adversarial branches are also computed in the same way.

Finally, we compute the final T-CAM \mathbf{C}_i^F from the four T-CAMs (two from the RGB stream: $\mathbf{C}_i^r, \mathbf{C}_i^{ra}$, two from the optical flow stream: $\mathbf{C}_i^o, \mathbf{C}_i^{oa}$) as follows:

$$\mathbf{C}_i^F = \mathbf{w}^r \cdot (\mathbf{C}_i^r + \omega \mathbf{C}_i^{ra}) + \mathbf{w}^o \cdot (\mathbf{C}_i^o + \omega \mathbf{C}_i^{oa}) \quad (18)$$

where $\mathbf{w}^r, \mathbf{w}^o \in \mathbb{R}^{N_c}$ are class-specific weighting parameters that are learned during training and ω is a hyperparameter for the relative importance of T-CAMs from the adversarial branch. We can then compute the classification loss for the final T-CAM $\mathcal{L}_{\text{CLS}}^F$ in the same manner. The total classification loss is given by:

$$\mathcal{L}_{\text{CLS}} = \mathcal{L}_{\text{CLS}}^r + \mathcal{L}_{\text{CLS}}^{ra} + \mathcal{L}_{\text{CLS}}^o + \mathcal{L}_{\text{CLS}}^{oa} + \mathcal{L}_{\text{CLS}}^F \quad (19)$$

3.5 Classification and Localization

During the test time, we use the final T-CAM \mathbf{C}_i^F for the classification and localization of activities following the previous works [19, 16]. First, we compute the class-wise classification scores $\mathbf{s}_i^F \in \mathbb{R}^{N_c}$ and the predicted pmf of activities $\mathbf{p}_i^F \in \mathbb{R}^{N_c}$ as described in Section 3.4. We use \mathbf{p}_i^F for activity classification. For activity localization, we first find a set of possible activities that has positive classification scores, which is $\{j : \mathbf{s}_i^F(j) > 0\}$. For each activity in this set, we localize all the temporal segments that has positive T-CAM values for two or more successive time steps. Formally, a set of localized temporal segments for v_i is:

$$\{[s, e] : \forall t \in [s, e], \mathbf{C}_i^F(t) > 0 \text{ and } \mathbf{C}_i^F(s-1) < 0 \text{ and } \mathbf{C}_i^F(e+1) < 0\} \quad (20)$$

where $\mathbf{C}_i^F(0)$ and $\mathbf{C}_i^F(l_i+1)$ are defined to be any negative values and $e \geq s+2$. The localized segments for each activity are non-overlapping. We assign a confidence score for each localized segment, which is the sum of the maximum T-CAM value of the segment and the classification score of it.

4 Experiments

4.1 Datasets and Evaluation

We evaluate our method on two datasets: THUMOS14 [5] and ActivityNet1.3 [3]. For the THUMOS14 dataset, the validation videos are used for training without temporal boundary annotations and the test videos are used for evaluation following the convention in the literature. This dataset is known to be challenging because each video has a number of activity instances and the duration of the videos varies widely. For the ActivityNet1.3 dataset, we use the training set for training and the validation set for evaluation. Following the standard evaluation protocol, we report mean average precision (mAP) at different intersection over union (IoU) thresholds.

4.2 Implementation Details

First, we extract RGB frames from each video at 25 fps and generate optical flow frames by using the TV-L1 algorithm [29]. Each video is then divided into non-overlapping 16-frame segments. We apply I3D networks [1] pre-trained on Kinetics dataset [6] to the segments to obtain the intermediate 1024-dimensional features after the global pooling layer. We train our network in an end-to-end manner using a single GPU (TITAN Xp).

For the THUMOS14 dataset [5], we train our network using a batch size of 32. We use the Adam optimizer [7] with learning rate 10^{-4} and weight decay 0.0005. The centers are updated using the SGD algorithm with learning rate 0.1 for the RGB stream and 0.2 for the optical flow stream. The kernel size of the 1-D convolutional layers for the T-CAMs is set to 1. We set α in Eq. 1 to 1 and γ in Eq. 8 to 0.6. For β in Eq. 6, we randomly generate a number between 0.001 and 0.1 for each training sample. We set angular margins m_1 to 2 and m_2 to 1. s_a of Eq. 14 and s for the classification loss are set to 40 and 8, respectively. Finally, ω in Eq. 18 is set to 0.6. The whole training process of 40.5k iterations takes less than 14 hours.

For the ActivityNet1.3 dataset [3], it is shown from the previous works [19, 16] that post-processing of the final T-CAM is required. We use an additional 1-D convolutional layer (kernel size=13, dilation=2) to post-process the final T-CAM. The kernel size of the 1-D convolutional layers for T-CAMs is set to 3. In addition, we change the batch size to 24. The learning rate for centers are 0.05 and 0.1 for the RGB and optical flow streams, respectively. We set α to 2, γ to 0.2, and ω to 0.4. The remaining hyperparameters of β , m_1 , m_2 , s_a , and s are the same as above. We train the network for 175k iterations.

4.3 Comparisons with the State-of-the-art

We compare our final method A2CL-PT with other state-of-the-art approaches on the THUMOS14 dataset [5] in Table 1. Full supervision refers to training from frame-level activity annotations, whereas weak supervision indicates training

Table 1. Performance comparison of A2CL-PT with state-of-the-art methods on the THUMOS14 dataset [5]. A2CL-PT significantly outperforms all the other weakly-supervised methods. † indicates an additional usage of other ground-truth annotations or independently collected data. A2CL-PT also outperforms all weakly†-supervised methods that use additional data at higher IoUs (from 0.4 to 0.9). The column AVG is for the average mAP of IoU threshold from 0.1 to 0.9.

Supervision	Method	mAP(%)@IoU									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	AVG
Full	S-CNN [22]	47.7	43.5	36.3	28.7	19.0	10.3	5.3	-	-	-
	R-C3D [26]	54.5	51.5	44.8	35.6	28.9	-	-	-	-	-
	SSN [32]	66.0	59.4	51.9	41.0	29.8	-	-	-	-	-
	TAL-Net [2]	59.8	57.1	53.2	48.5	42.8	33.8	20.8	-	-	-
	BSN [11]	-	-	53.5	45.0	36.9	28.4	20.0	-	-	-
	GTAN [14]	69.1	63.7	57.8	47.2	38.8	-	-	-	-	-
Weak†	Liu et al. [12]	57.4	50.8	41.2	32.1	23.1	15.0	7.0	-	-	-
	3C-Net [16]	59.1	53.5	44.2	34.1	26.6	-	8.1	-	-	-
	Nguyen et al. [18]	64.2	59.5	49.1	38.4	27.5	17.3	8.6	3.2	0.5	29.8
	STAR [27]	68.8	60.0	48.7	34.7	23.0	-	-	-	-	-
Weak	UntrimmedNet [24]	44.4	37.7	28.2	21.1	13.7	-	-	-	-	-
	STPN [17]	52.0	44.7	35.5	25.8	16.9	9.9	4.3	1.2	0.1	21.2
	W-TALC [19]	55.2	49.6	40.1	31.1	22.8	-	7.6	-	-	-
	AutoLoc [21]	-	-	35.8	29.0	21.2	13.4	5.8	-	-	-
	CleanNet [13]	-	-	37.0	30.9	23.9	13.9	7.1	-	-	-
	MAAN [28]	59.8	50.8	41.1	30.6	20.3	12.0	6.9	2.6	0.2	24.9
	BaS-Net [9]	58.2	52.3	44.6	36.0	27.0	18.6	10.4	3.9	0.5	27.9
A2CL-PT (Ours)		61.2	56.1	48.1	39.0	30.1	19.2	10.6	4.8	1.0	30.0

only from video-level activity labels. For fair comparison, we use the symbol † to separate methods utilizing additional ground-truth annotations [16, 27] or independently collected data [12, 18]. The column AVG is for the average mAP of IoU thresholds from 0.1 to 0.9 with a step size of 0.1. Our method significantly outperforms other weakly-supervised methods across all metrics. Specifically, an absolute gain of 2.1% is achieved in terms of the average mAP when compared to the best previous method (BaS-Net [9]). We want to note that our method performs even better than the methods of weak† supervision at higher IoUs.

We also evaluate A2CL-PT on the ActivityNet1.3 dataset [3]. Following the standard evaluation protocol of the dataset, we report mAP at different IoU thresholds, which are from 0.05 to 0.95. As shown in Table 2, our method again achieves the best performance.

4.4 Ablation Study and Analysis

We perform an ablation study on the THUMOS14 dataset [5]. In Table 3, we analyze the two main contributions of this work, which are the usage of the newly-suggested triplet (Eq. 7) and the adoption of adversarial approach (Eq. 15). ATCL refers to the baseline that uses only the loss term of Eq. 4. We use the superscript + to indicate the addition of adversarial branch. As described in Section 3.2, ACL-PT additionally uses the new triplet on top of the baseline.

Table 2. Performance comparison on the ActivityNet1.3 dataset [3]. A2CL-PT again achieves the best performance. † indicates an additional usage of other ground-truth annotations or independently collected data. The column AVG is for the average mAP of IoU threshold from 0.5 to 0.95.

Supervision	Method	mAP(%)@IoU										
		0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	AVG
Weak†	Liu et al. [12]	34.0	-	-	-	-	20.9	-	-	-	5.7	21.2
	Nguyen et al. [18]	36.4	-	-	-	-	19.2	-	-	-	2.9	-
	STAR [27]	31.1	-	-	-	-	18.8	-	-	-	4.7	-
Weak	STPN [17]	29.3	-	-	-	-	16.9	-	-	-	2.6	-
	MAAN [28]	33.7	-	-	-	-	21.9	-	-	-	5.5	-
	BaS-Net [9]	34.5	-	-	-	-	22.5	-	-	-	4.9	22.2
	A2CL-PT (Ours)	36.8	33.6	30.8	27.8	24.9	22.0	18.1	14.9	10.2	5.2	22.5

Table 3. Performance comparison of different ablative settings on the THUMOS14 dataset [5]. The superscript + indicates that we add an adversarial branch to the baseline method. It demonstrates that both components are effective.

Method	New triplet	Adversarial	mAP(%)@IoU					
			0.3	0.4	0.5	0.6	0.7	AVG(0.1:0.9)
ATCL			44.7	34.8	25.7	15.8	8.3	27.4
ATCL ⁺		✓	43.7	35.1	26.3	15.7	8.3	27.2
ACL-PT	✓		46.6	37.2	28.9	18.2	10.0	29.2
A2CL-PT	✓	✓	48.1	39.0	30.1	19.2	10.6	30.0

We can observe that our final proposed method, A2CL-PT, performs the best. It implies that both components are necessary to achieve the best performance and each of them is effective. Interestingly, adding an adversarial branch does not bring any performance gain without our new triplet. We think that although using ACL-PT increases the localization performance by learning discriminative features, it also makes the network sensitive to salient activity-related features.

We analyze the impact of two main hyperparameters in Fig. 3. The first one is α that controls the weight of A2CL-PT term (Eq. 1), and the other one is ω that is for the relative importance of T-CAMs from adversarial branches (Eq. 18). We can observe from Fig. 3(a) that positive α always brings the performance gain. It indicates that A2CL-PT is effective. As seen in Fig. 3(b), the performance is increased by using an adversarial approach when ω is less or equal to 1. If ω is greater than 1, T-CAMs of adversarial branches will play a dominant role in activity localization. Therefore, the results tell us that the adversarial branches provide mostly supplementary information.

4.5 Qualitative Analysis

We perform a qualitative analysis to better understand our method. In Fig. 4, qualitative results of our A2CL-PT on four videos from the test set of the THUMOS14 dataset [5] are presented. (a), (b), (c), and (d) are examples of *JavelinThrow*, *HammerThrow*, *ThrowDiscus*, and *HighJump*, respectively. Detection denotes the localized activity segments. For additional comparison, we

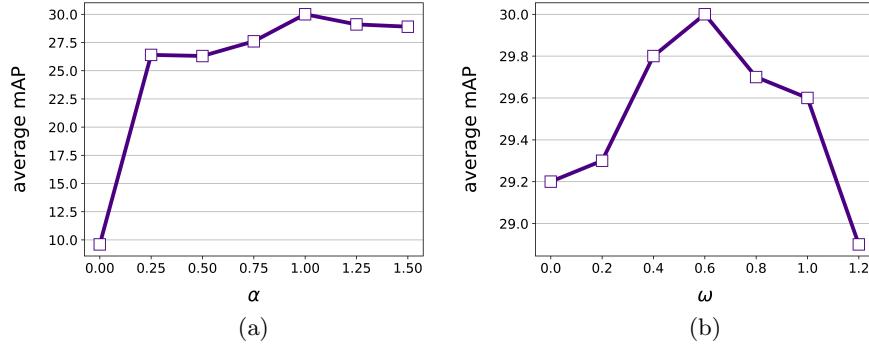


Fig. 3. We analyze the impact of two main hyperparameters α and ω . (a): Positive α always provides the performance gain, so it indicates that our method is effective. (b): If ω is too large, the performance is decreased substantially. It implies that T-CAMs of adversarial branches provide mostly supplementary information.

also show the results of BaS-Net [9], which is the leading state-of-the-art method. We use three different colors on the contours of sampled frames: blue, green, and red which denote true positive, false positive, and false negative, respectively. In (a), there are multiple instances of false positive. These false positives are challenging because the person in the video swings the javelin, which can be mistaken for a throw. Similar cases are observed in (b). One of the false positives includes the person drawing the line on the field, which looks similar to a *HammerThrow* activity. In (c), some false negative segments are observed. Interestingly, this is because the ground-truth annotations are wrong; that is, the *ThrowDiscus* activity is annotated but it does not actually occur in these cases. In (d), all the instances of the *HighJump* activity are successfully localized. Other than the unusual situations, our method performs well in general.

5 Conclusion

We have presented A2CL-PT as a novel method for weakly-supervised temporal activity localization. We suggest using two triplets of vectors of the feature space to learn discriminative features and to distinguish background portions from activity-related parts of a video. We also propose to adopt an adversarial approach to localize activities more thoroughly. We perform extensive experiments to show that our method is effective. A2CL-PT outperforms all the existing state-of-the-art methods on major datasets. Ablation study demonstrates that both contributions are significant. Finally, we qualitatively analyze the effectiveness of our method in detail.

Acknowledgement We thank Stephan Lemmer, Victoria Florence, Nathan Louis, and Christina Jung for their valuable feedback and comments. This research was, in part, supported by NIST grant 60NANB17D191.

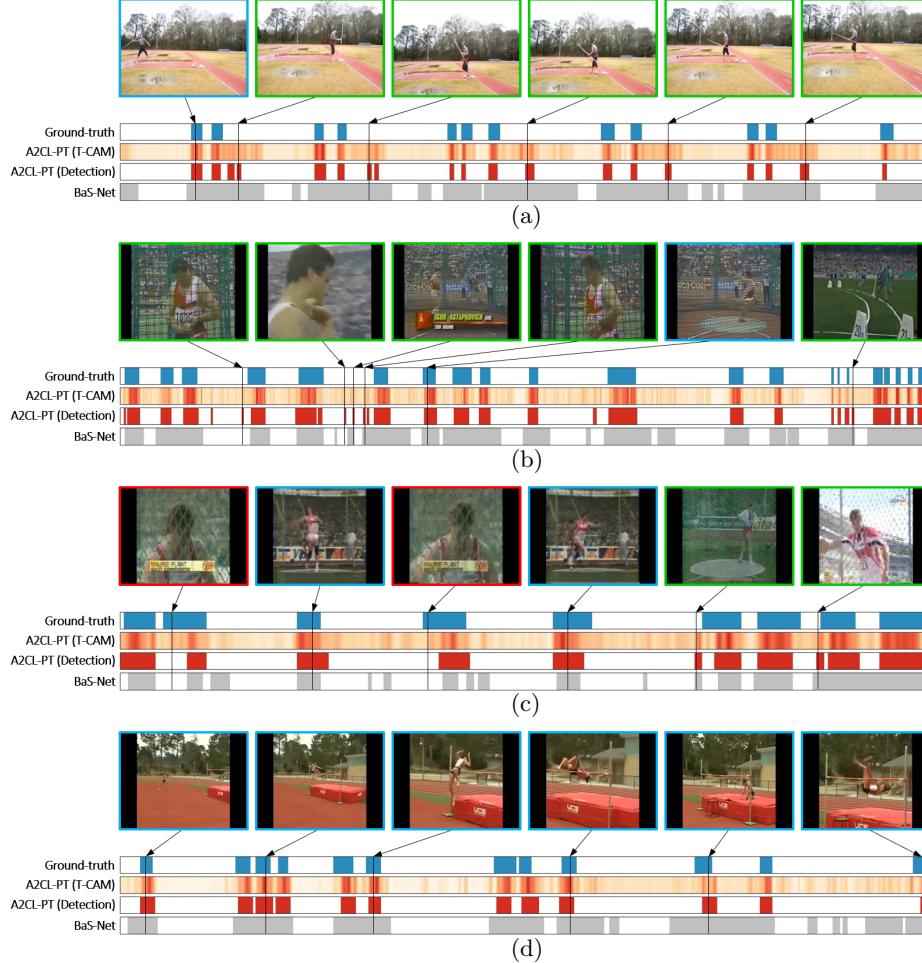


Fig. 4. Qualitative results on the THUMOS14 dataset [5]. Detection denotes the localized activity segments. The results of BaS-Net [9] are also included for additional comparison. Contours of the sampled frames have three different colors. We use blue, green, and red to indicate true positives, false positives, and false negatives, respectively. (a): An example of *JavelinThrow* activity class. The observed false positives are challenging. The person in the video swings the javelin on the frames of these false positives, which can be mistaken for a throw. (b): An example of *HammerThrow*. One of the false positives include the person who draws the line on the field. It is hard to distinguish the two activities. (c): An example of *ThrowDiscus*. Multiple false negatives are observed, which illustrates the situations where the ground-truth activity instances are wrongly annotated. (d): An example of *HighJump* without such unusual cases. It can be observed that our method performs well in general.

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
2. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1130–1139 (2018)
3. Fabian Caba Heilbron, Victor Escorcia, B.G., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 961–970 (2015)
4. He, X., Zhou, Y., Zhou, Z., Bai, S., Bai, X.: Triplet-center loss for multi-view 3d object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1945–1954 (2018)
5. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes. <http://crcv.ucf.edu/THUMOS14/> (2014)
6. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
8. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: 2011 International Conference on Computer Vision. pp. 2556–2563. IEEE (2011)
9. Lee, P., Uh, Y., Byun, H.: Background suppression network for weakly-supervised temporal action localization. In: AAAI (2020)
10. Li, Z., Xu, C., Leng, B.: Angular triplet-center loss for multi-view 3d shape retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8682–8689 (2019)
11. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
12. Liu, D., Jiang, T., Wang, Y.: Completeness modeling and context separation for weakly supervised temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1298–1307 (2019)
13. Liu, Z., Wang, L., Zhang, Q., Gao, Z., Niu, Z., Zheng, N., Hua, G.: Weakly supervised temporal action localization through contrast based evaluation networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3899–3908 (2019)
14. Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., Mei, T.: Gaussian temporal awareness networks for action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 344–353 (2019)
15. Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S.A., Yan, T., Brown, L., Fan, Q., Gutfruend, D., Vondrick, C., et al.: Moments in time dataset: one million videos for event understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–8 (2019). <https://doi.org/10.1109/TPAMI.2019.2901464>

16. Narayan, S., Cholakkal, H., Khan, F.S., Shao, L.: 3c-net: Category count and center loss for weakly-supervised action localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8679–8687 (2019)
17. Nguyen, P., Liu, T., Prasad, G., Han, B.: Weakly supervised action localization by sparse temporal pooling network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6752–6761 (2018)
18. Nguyen, P.X., Ramanan, D., Fowlkes, C.C.: Weakly-supervised action localization with background modeling. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5502–5511 (2019)
19. Paul, S., Roy, S., Roy-Chowdhury, A.K.: W-talc: Weakly-supervised temporal activity localization and classification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 563–579 (2018)
20. 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: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5734–5743 (2017)
21. Shou, Z., Gao, H., Zhang, L., Miyazawa, K., Chang, S.F.: Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 154–171 (2018)
22. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1049–1058 (2016)
23. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
24. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 4325–4334 (2017)
25. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European conference on computer vision. pp. 499–515. Springer (2016)
26. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. In: Proceedings of the IEEE international conference on computer vision. pp. 5783–5792 (2017)
27. Xu, Y., Zhang, C., Cheng, Z., Xie, J., Niu, Y., Pu, S., Wu, F.: Segregated temporal assembly recurrent networks for weakly supervised multiple action detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 9070–9078 (2019)
28. Yuan, Y., Lyu, Y., Shen, X., Tsang, I.W., Yeung, D.Y.: Marginalized average attentional network for weakly-supervised learning. In: International Conference on Learning Representations (ICLR) (2019)
29. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: Joint pattern recognition symposium. pp. 214–223. Springer (2007)
30. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.S.: Adversarial complementary learning for weakly supervised object localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1325–1334 (2018)
31. Zhao, H., Torralba, A., Torresani, L., Yan, Z.: Hacs: Human action clips and segments dataset for recognition and temporal localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8668–8678 (2019)
32. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2914–2923 (2017)