Weakly Supervised Temporal Action Localization through Contrast based Evaluation Networks

Ziyi Liu[1] Le Wang[1*] Qilin Zhang[2] Zhanning Gao[3] Zhenxing Niu[4] Nanning Zheng[1] Gang Hua[5]
[1]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]HERE Technologies [3]DAMO Academy, Alibaba Group
[4]Machine Intelligence Israel Lab, Alibaba Group [5]Wormpex AI Research

Abstract

Weakly-supervised temporal action localization (WS-TAL) is a promising but challenging task with only video-level action categorical labels available during training. Without requiring temporal action boundary annotations in training data, WS-TAL could possibly exploit automatically retrieved video tags as video-level labels. However, such coarse video-level supervision inevitably incurs confusions, especially in untrimmed videos containing multiple action instances. To address this challenge, we propose the Contrast-based Localization EvaluAtioN Network (CleanNet) with our new action proposal evaluator, which provides pseudo-supervision by leveraging the temporal contrast in snippet-level action classification predictions. Essentially, the new action proposal evaluator enforces an additional temporal contrast constraint so that high-evaluation-score action proposals are more likely to coincide with true action instances. Moreover, the new action localization module is an integral part of CleanNet which enables end-to-end training. This is in contrast to many existing WS-TAL methods where action localization is merely a post-processing step. Experiments on THUMOS14 and ActivityNet datasets validate the efficacy of CleanNet against existing state-of-the-art WS-TAL algorithms.

1. Introduction

Temporal Action Localization (TAL) involves the localization of temporal starts and ends of specific categories of actions. Thanks to its numerous potential applications such as action retrieval, surveillance, and summary [1, 6, 18, 29], TAL has drawn increasing attention from the research community recently.

However, it could still be time-consuming and prohibitively expensive to manually label the temporal ranges of all action instances in untrimmed videos for a large-scale dataset. A more cost-effective alternative setting could be the weakly supervised temporal action localization (WS-TAL) that only relies on video-level categorical labels for training. The advantage of WS-TAL is in its training data collection, video-level labels are much easier to collect than temporal action boundaries. It might even be possible to automatically retrieve corresponding hashtags from video sharing website as labels. However, to be more focused and less ambitious, we limit the scope of our investigation to manually annotated video-level labels.

Currently, many existing WS-TAL methods [21, 23, 32, 37] localize actions by directly thresholding the classification score of each snippet. Therefore, those snippets are independently treated and their temporal relations are neglected. However, true action boundaries often depend heavily on the temporal contrast among those snippets, such as temporal discontinuities and sudden changes.

We propose a Contrast-based Localization EvaluAtioN Network (CleanNet) for WS-TAL, which leverages the temporal contrast cue among action classification predictions of snippets for action proposal evaluation. As illustrated in Figure 1, CleanNet consists of a feature embedding, an action classification, and an action localization module.

Given an untrimmed video input, snippet-level features are first extracted by the feature embedding. Subsequently, action classification produces Snippet-level Classification Predictions (SCP) and Snippet-level Attention Predictions (SAP), which are fused to get a video-level prediction by weighted summation multiplication. With the obtained video-level prediction and the video-level categorical label, the classification loss is calculated and the action classification is trained by minimizing it.

Meanwhile, after acquiring SCP and SAP, action localization proceeds to compute the "**contrast score**" for each action proposal provided by the action proposal generator. Then, only action proposals with higher contrast scores are kept and action localization is trained by maximizing the average contrast score of these survival proposals. During testing, after scoring all action proposals, duplicated action

*Corresponding author.

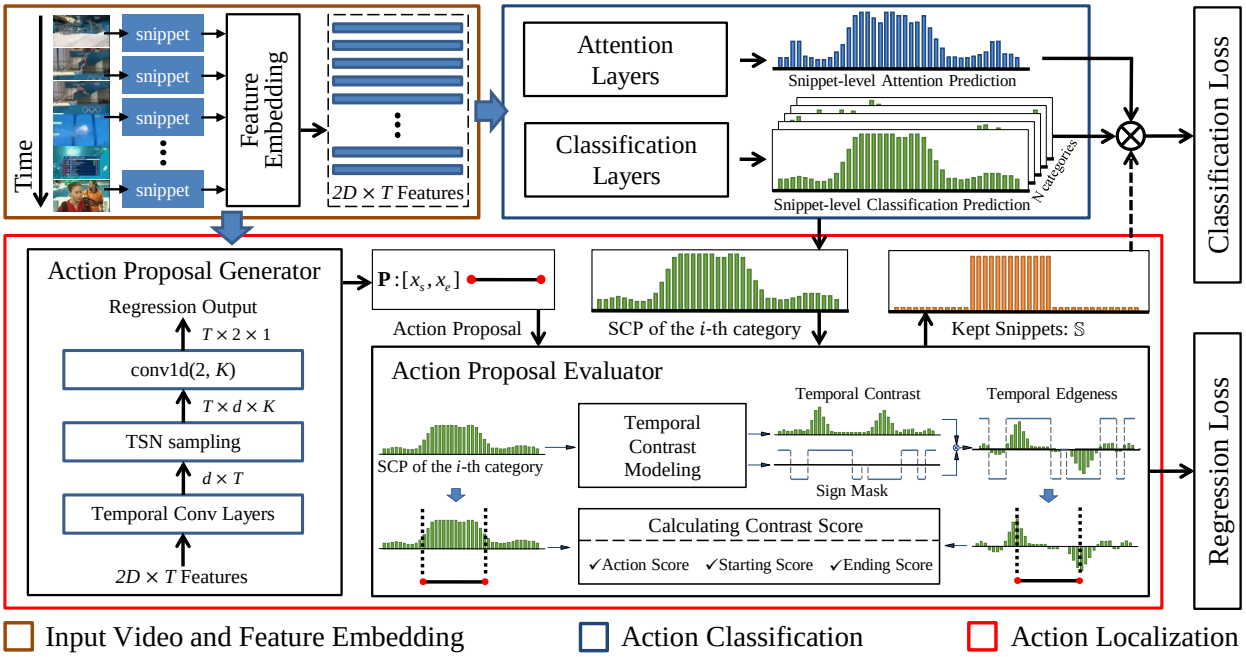


Figure 1: The proposed CleanNet consists of three parts, *i.e.*, a feature embedding, an action classification, and an action localization module, as denoted by brown, blue and red rectangles, respectively. Training inputs: untrimmed videos with video-level categorical labels. Prediction outputs: action instance category labels and temporal starts and ends.

proposals are removed by performing Non-Maximum Suppression (NMS). Finally, a set of predicted action instances are obtained, with both category labels and temporal boundaries.

Specifically, the action proposal evaluator calculates an action score and two edge scores (starting and ending scores) for each action proposal, representing the likelihood of the action proposal containing a specific action, and the consistency of the action proposal starts/ends with specific action edges, respectively. By combining the action, starting and ending scores, the new action proposal evaluator provides a comprehensive "contrast score" which measures both the content and the completeness of action proposals.

Moreover, there is a mutualism between action classification and action localization in CleanNet. Action classification provides SCP to the action proposal evaluator as the basis of contrast scores of action proposals; while the action localization offers localization-based filter of irrelevant frames, as illustrated by a dashed arrow in Figure 1, where irrelevant snippets are discarded in classification loss calculation.

In summary, the key contributions of this paper include (1) a new action proposal evaluator that quantifies the temporal contrast among SCP to facilitate WS-TAL; (2) an improved action proposal generator with matching receptive field with anchor size; (3) an end-to-end trainable CleanNet for WS-TAL, where action classification and localization are mutually beneficial; (4) the state-of-the-art WS-TAL performance on two benchmarks and even compares favorably with some fully-supervised TAL methods.

2. Related Work

We briefly review related work in action recognition, TAL with full supervision, and TAL with weak supervision.

2.1. Action Recognition

Prior to the prevalence of deep neural networks, action recognition is dominated by hand-crafted features-based methods [7, 19, 26, 36]. Recently, Convolutional Neural Networks (CNNs) have emerged as the state-of-the-art visual feature extractor and numerous CNNs-based action recognition methods are proposed. Two-stream networks [10, 30] incorporate optical flow in addition to images in a two-stream architecture, and recognition results are obtained by fusing both streams. 3D ConvNets [16, 34, 35] take video clip as input to acquire spatial and temporal correlations among video frames. TSN [38] captures the long-range temporal structure with sparse sampling. I3D [3] combines two-stream networks with 3D convolutions to further boost the recognition accuracy.

2.2. TAL with Full Supervision

Different from the task action recognition which only requires video-level categorical predictions, TAL requires finer-grained predictions with both categorical labels for

each of the action instances and their corresponding temporal boundaries. The fully-supervised TAL methods need both types of annotations during training.

Thanks to the advancements of deep learning-based object detection methods, such as R-CNN [14] and its variations [13, 25], many methods follow a similar structure of "generating and classifying action proposals" to perform TAL [2, 4, 5, 8, 12, 29, 39, 41]. Some works [2, 8, 29] generate action proposals using sliding windows or predefined temporal durations. Zhao *et al*. [41] adopts a watershed algorithm upon snippet-level "actioness" probabilities to generate action proposals with flexible durations. Some other works [4, 5, 12, 39] exploit the Faster R-CNN architecture [25] for TAL. Xu *et al*. [39] closely follows Faster R-CNN in multiple design settings. Some of these works [4, 5, 12] further adjust the Faster R-CNN architecture to resolve the receptive field issues and make better use of the contextual information. These architectural adjustments are reportedly responsible for the improved performances in the TAL task.

2.3. TAL with Weak Supervision

The idea of performing TAL using only video-level categorical annotations was first introduced in [33]. Hide-and-Seek [32] randomly hides regions to encourage the model to focus on both the most discriminative parts and other relevant parts of the target. UntrimmedNet [37] uses a soft selection module to locate target temporal action segments, which is similar to temporal attention weights, and the final localization is achieved by thresholding these segments after the scoring. STPN [21] proposes a sparse loss function to facilitate the selection of segments. W-TALC [23] proposes a co-activity loss and combine it with a multiple instance learning loss to train a weakly-supervised network. The localization parts of these methods are all based on thresholding on the final SCP.

The recent AutoLoc [28] directly predicts the temporal boundaries of each action instance by benefiting from its "outer-inner-contrastive loss". The proposed CleanNet is distinctive from [28] in the following three aspects. First of all, our action proposal evaluator exploits temporal contrast and treats the starting/ending boundaries separately to achieve better robustness to noise. Second, the action classification and action localization in CleanNet are interdependent and mutually beneficial, while the counterparts in [28] are independent. Moreover, our action proposal generator is specially designed to address the receptive field issue in the temporal dimension. All these three differences contribute to the superiority of CleanNet, as discussed in Section 4.2.

3. Proposed CleanNet

In this section, we introduce the proposed CleanNet. As illustrated in Figure 1, all three major components in CleanNet, *i.e*., the feature embedding, action classification, and action localization are described in detail as follows.

3.1. Snippet-Level Feature Embedding

The inputs to the feature embedding (the brown rectangle in Figure 1) are untrimmed videos, and the outputs are the corresponding features. The feature embedding mainly follows that in UntrimmedNet [37]. After dividing each video into non-overlapping snippets of the same length (15 frames), temporal features are extracted snippet-after-snippet, which are referred to as snippet-level features $\mathbf{F}$.

The backbone of the feature embedding is the TSN [38] with the Inception network architecture and Batch Normalization [15]. The pre-trained spatial stream (RGB input) and the temporal stream (optical flow input) are trained individually. The obtained D-dimensional ($D = 1024$) outputs after the `global_pool` layers from both streams are concatenated as one snippet-level feature. Specifically, for an input video with T snippets ($15T$ video frames), the output $\mathbf{F}$ is with $2D$ channels by T snippets. The feature of the t-th snippet is denoted as $\mathbf{F}(t) \in \mathbb{R}^{2D\times 1}$.

3.2. Action Classification

With $\mathbf{F} \in \mathbb{R}^{2D\times T}$, the action classification (blue rectangle in Figure 1) computes both the snippet-level classification prediction (SCP) and the snippet-level attention prediction (SAP) with two groups of fully connected layers, respectively. SCP and SAP are denoted as $\mathbf{\Psi} \in \mathbb{R}^{N\times T}$ and $\boldsymbol{\varphi} \in \mathbb{R}^{1\times T}$, where N and T are the numbers of action categories and snippets, respectively. Since our action classification has the same structure as the one in UntrimmedNet [37], a direct practice to obtain $\mathbf{\Psi}$ and $\boldsymbol{\varphi}$ is averaging the outputs of UntrimmedNet from both streams. To make this fusion step trainable, we design our action classification module as follow.

$$\mathbf{\Psi}(t) = (\mathbf{\Psi}^r(t) + \mathbf{\Psi}^f(t))/2, \tag{1}$$

$$\begin{bmatrix} \mathbf{\Psi}^r(t) \\ \mathbf{\Psi}^f(t) \end{bmatrix} = \mathbf{W}^c \cdot \mathbf{F}(t) + \mathbf{b}^c, \tag{2}$$

$$\boldsymbol{\varphi}(t) = \mathbf{w}^a \cdot \mathbf{F}(t) + b^a, \tag{3}$$

where $t = 1, \ldots, T$ is the snippet index. $\mathbf{\Psi}^r(t) \in \mathbb{R}^{N\times 1}$ and $\mathbf{\Psi}^f(t) \in \mathbb{R}^{N\times 1}$ are classification predictions of the t-th snippet from the spatial stream and temporal stream, respectively. $\mathbf{W}^c \in \mathbb{R}^{2N\times 2D}$ and $\mathbf{b}^c \in \mathbb{R}^{2N\times 1}$ are the parameters of classification layer. $\mathbf{w}^a \in \mathbb{R}^{1\times 2D}$ and b^a are the parameters of attention layer. They are initialized as

$$\mathbf{W}^c = \begin{bmatrix} \mathbf{W}^{c_r} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}^{c_f} \end{bmatrix}, \ \mathbf{b}^c = \begin{bmatrix} \mathbf{b}^{c_r} \\ \mathbf{b}^{c_f} \end{bmatrix}, \tag{4}$$

$$\mathbf{w}^a = \frac{1}{2}\begin{bmatrix} \mathbf{w}^{a_r} & \mathbf{w}^{a_f} \end{bmatrix}, b^a = \frac{b^{a_r} + b^{a_f}}{2}, \tag{5}$$

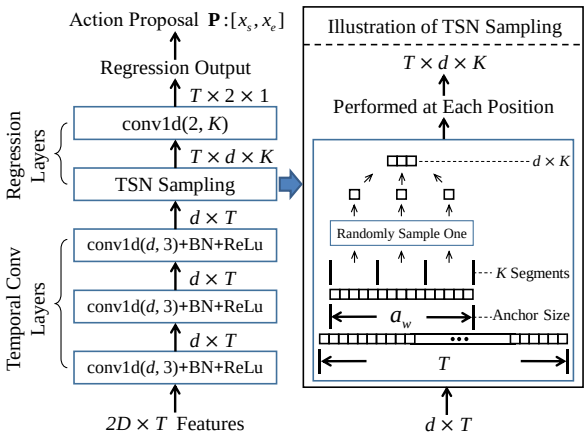


Figure 2: The structure of the action proposal generator. Input snippet-level features are fed into three stacked temporal convolutional layers before a *TSN sampling* layer, which matches its receptive field with the anchor size.

where $\mathbf{W}^{c_r} \in \mathbb{R}^{N \times D}$, $\mathbf{W}^{c_f} \in \mathbb{R}^{N \times D}$, $\mathbf{w}^{a_r} \in \mathbb{R}^{1 \times D}$ and $\mathbf{w}^{a_f} \in \mathbb{R}^{1 \times D}$ stand for the weights of the classification and attention layers with RGB input and optical flow input, respectively. $\mathbf{b}^{c_r} \in \mathbb{R}^{N \times 1}$, $\mathbf{b}^{c_f} \in \mathbb{R}^{N \times 1}$, b^{a_r} and b^{a_f} are the corresponding bias parameters. They are initialized by loading the pre-trained UntrimmedNet models[1]. By this initialization, our action classification achieves equivalent fusion output of averaging both streams from pre-trained UntrimmedNet and remains trainable for further finetuning. Finally, for each video with T snippets, we obtain its SCP ($\mathbf{\Psi} \in \mathbb{R}^{N \times T}$) and SAP ($\boldsymbol{\varphi} \in \mathbb{R}^{1 \times T}$).

3.3. Action Localization

The main contribution of this paper is reflected in the special design of the action localization (the red rectangle in Figure 1), which is composed of an action proposal generator and an action proposal evaluator.

3.3.1 Action Proposal Generator

The goal of the action proposal generator is to generate action proposals that can precisely cover the temporal range of action instances, which is obtained by temporal boundary regression. Inspired by existing anchor-based 2D bounding box regression techniques [24, 25], we utilize similar settings in this 1D temporal regression. Specifically, for an anchor with temporal duration (size) a_w and temporal location τ, its boundary regression value is a two-element vector, with r_c relevant to the regressed center and r_w relevant to the regressed duration. Let $\mathbf{P}$ denote the regressed anchor, the centroid x_c of $\mathbf{P}$ is obtained as $x_c = a_w \cdot r_c + \tau$, the temporal duration x_w of $\mathbf{P}$ is $x_w = a_w \cdot \exp(r_w)$, and the starting and ending boundaries of $\mathbf{P}$ can be calculated as $x_s = x_c - x_w/2$ and $x_e = x_c + x_w/2$, respectively. For notational simplicity, we choose $[x_s, x_e]$ to parameterize $\mathbf{P}$.

However, such direct adaptation of spatial bounding box regression algorithm is insufficient due to potential receptive field issues. More specifically, the spatial regression results in [25] are obtained from a 1×1 convolution layer upon the output of `pool5` in VGG16 [31], achieving a receptive field of 212, which is large enough given the input image resolution of 224×224. If such strategy is directly applied in 1D temporal regression, the receptive field of snippet-level features ($\mathbf{F} \in \mathbb{R}^{2D \times T}$) along the temporal dimension is merely 1, since they are extracted snippet-after-snippet. Thus, it is unrealistic to expect reasonable regression outputs when the receptive field is much smaller than the anchor size.

A direct remedy might be stacking multiple temporal convolutional layers upon snippet-level features $\mathbf{F}$, but the gain of the receptive field is still limited. To match the receptive field with corresponding anchor size, we exploit a sparse temporal sampling strategy inspired by TSN [38]. In detail, we divide each anchor into K segments and randomly sample a temporal location per segment, and then obtain a fixed size (K) representation regardless of the anchor size. We term this strategy as *TSN sampling*, as illustrated in Figure 2. Subsequently, the sampled features are fed into another convolutional layer to obtain the regression values.

3.3.2 Action Proposal Evaluator

To supervise the action proposal generator, an action proposal evaluator is necessary. In the fully supervised TAL setting where manually labeled temporal boundaries are available, action proposals can be readily evaluated by comparing with ground truth, with a metric such as Intersection-over-Union (IoU). However, in the WS-TAL setting where explicit temporal boundary annotations are unknown, the design of the action proposal evaluator is nontrivial.

In the CleanNet, we proposed a new action proposal evaluator to provide pseudo-supervision based on SCP values of the entire video. The intuition of exploiting all SCP values is to reward action proposals with both correct contents and complete action instances with less fragmentation. With extended SCP values beyond the starts and ends of an action proposal, the new action proposal evaluator penalizes fragmented short action proposals and promotes completeness and continuity.

The workflow of the action proposal evaluator is illustrated in Figure 3. To locate action instances of the i-th category ($i = 1, \ldots, N$) in a video, the inputs to the evaluator are all temporal SCP values corresponding to the i-th action category, *i.e.*, $\boldsymbol{\psi}_i \in \mathbb{R}^{1 \times T}$ (the i-th row of $\mathbf{\Psi}$, illustrated as a green histogram, provided by the action classification module) and an action proposal $\mathbf{P}$ (illustrated as the bolded

[1] https://github.com/wanglimin/UntrimmedNet

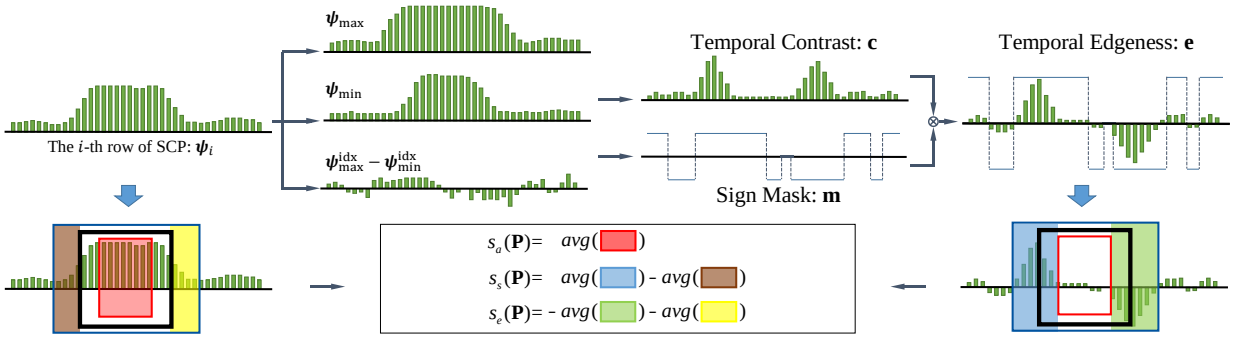


Figure 3: The work flow of the action proposal evaluator in CleanNet. To locate action instances of the i-th category in a video, the inputs to the evaluator are $\boldsymbol{\psi}_i \in \mathbb{R}^{1\times T}$ (illustrated as the green histogram) and an action proposal $\mathbf{P}$ (denoted as the black bounding boxes imposed on the green histogram). The output is the contrast score $s(\mathbf{P})$ of $\mathbf{P}$, according to Eq. (12).

black bounding boxes imposed on the histograms on bottom corners, provided by the action proposal generator). To simplify the subscripts of subsequent $\boldsymbol{\psi}$ variants, we temporarily replace $\boldsymbol{\psi}_i$ with $\boldsymbol{\psi}$ in this Section 3.3.2.

To account for the temporal contrast information, we propose the temporal contrast vector $\mathbf{c} \in \mathbb{R}^{1\times T}$ as

$$\mathbf{m} = (\boldsymbol{\psi}_{\text{max}} - \boldsymbol{\psi}_{\text{min}}) \odot [\text{abs}(\boldsymbol{\psi}^{\text{idx}}_{\text{max}} - \boldsymbol{\psi}^{\text{idx}}_{\text{min}})]^{-1}, \quad (6)$$

where $\odot$ indicates element-wise multiplication, abs$(\cdot)$ and $[\cdot]^{-1}$ represent element-wise absolute value and reciprocal function, respectively. $\boldsymbol{\psi}_{\text{max}} \in \mathbb{R}^{1\times T}$ is derived by sliding a max pooling window [2] upon $\boldsymbol{\psi}$, and $\boldsymbol{\psi}^{\text{idx}}_{\text{max}} \in \mathbb{R}^{1\times T}$ is the corresponding index vector of local maximums. Similarly, $\boldsymbol{\psi}_{\text{min}} \in \mathbb{R}^{1\times T}$ and $\boldsymbol{\psi}^{\text{idx}}_{\text{min}} \in \mathbb{R}^{1\times T}$ are the min pooling values and indexes, respectively. Intuitively, temporal contrast $\mathbf{c}$ represents the likelihood of each snippet being the boundary of an action instance. To distinguish the starts and ends of action instances (*i.e.*, the rising and falling edges in $\boldsymbol{\psi}$), a sign mask $\mathbf{m} \in \mathbb{R}^{1\times T}$ is defined as

$$\mathbf{m}(t) = \begin{cases} 1 & \text{if } \boldsymbol{\psi}^{\text{idx}}_{\text{min}}(t) \le t < \boldsymbol{\psi}^{\text{idx}}_{\text{max}}(t), \\ -1 & \text{if } \boldsymbol{\psi}^{\text{idx}}_{\text{min}}(t) > t \ge \boldsymbol{\psi}^{\text{idx}}_{\text{max}}(t), \\ 0 & \text{otherwise, } t = 1, \ldots, T. \end{cases} \quad (7)$$

Subsequently, the temporal edgeness $\mathbf{e} \in \mathbb{R}^{1\times T}$ is calculated by $\mathbf{e}{=}\mathbf{m}\odot\mathbf{c}$, illustrated as the histogram on the top-right in Figure 3. Positive and negative values indicate the starting and ending boundaries of action instances, respectively.

For an action proposal $\mathbf{P}{:}[x_s, x_e]$, we compute its inflated and deflated regions $\mathbf{P}_{\text{inf}}{:}[x^{\text{inf}}_s, x^{\text{inf}}_e]$, $\mathbf{P}_{\text{def}}{:}[x^{\text{def}}_s, x^{\text{def}}_e]$ as

$$\begin{aligned} x^{\text{inf}}_s &= x_s - x_w/4, \quad & x^{\text{inf}}_e &= x_e + x_w/4, \\ x^{\text{def}}_s &= x_s + x_w/4, \quad & x^{\text{def}}_e &= x_e - x_w/4, \end{aligned} \quad (8)$$

which are illustrated as the blue and red bounding boxes imposed on the histograms on bottom corners in Figure 3, respectively. Definitions of x_c and x_w are included in Section 3.3.1.

With $\boldsymbol{\psi}$, $\mathbf{e}$, $\mathbf{P}$, $\mathbf{P}_{\text{inf}}$ and $\mathbf{P}_{\text{def}}$, three scores are calculated, *i.e.*, the action score $s_a(\mathbf{P})$ represents the likelihood of $\mathbf{P}$ containing a specific action instance, the starting score $s_s(\mathbf{P})$ reflects the likelihood of $\mathbf{P}$'s start stage coinciding with the beginning of an action instance, and the ending score $s_e(\mathbf{P})$ indicates the likelihood of $\mathbf{P}$'s end stage coinciding with the ending of an action instance. They are

$$s_a(\mathbf{P}) = \text{avg}(\boldsymbol{\psi}(x^{\text{def}}_s : x^{\text{def}}_e)), \quad (9)$$

$$s_s(\mathbf{P}) = \text{avg}(\mathbf{e}(x^{\text{inf}}_s : x^{\text{def}}_s)) - \text{avg}(\boldsymbol{\psi}(x^{\text{inf}}_s : x_s)), \quad (10)$$

$$s_e(\mathbf{P}) = -\text{avg}(\mathbf{e}(x^{\text{def}}_e : x^{\text{inf}}_e)) - \text{avg}(\boldsymbol{\psi}(x_e : x^{\text{inf}}_e)), \quad (11)$$

where avg$(\cdot)$ denotes arithmetic average. The final contrast score $s(\mathbf{P})$ is a weighted summation,

$$s(\mathbf{P}) = s_a(\mathbf{P}) + \frac{1}{2}\left(s_s(\mathbf{P}) + s_e(\mathbf{P})\right). \quad (12)$$

By summing up action scores and edge scores, the contrast score penalizes fragmented short action proposals and promotes completeness and continuity in action proposals. Ablation experimental results in Section 4.2 validate the contributions of each term in Eq. (12).

3.4. Training CleanNet

Having introduced the architecture of CleanNet, this section will discuss how to train the model. As shown in Figure 1, there are two losses, regression loss and classification loss, which are responsible for the two outputs of CleanNet, *i.e.*, localization and classification, respectively. For the training of action localization, we first select "*positive*" action proposals according to their contrast scores assigned by the action proposal evaluator. Specifically, when locating an action of the i-th category, if the i-th category prediction of the t-th snippet $\boldsymbol{\psi}_i(t)$ or its attention prediction $\boldsymbol{\varphi}(t)$ is lower than corresponding pre-defined thresholds, all anchors centered at this snippet will be discarded. Then, the

[2] The max pooling kernel size is 7. To ensure the output $\boldsymbol{\psi}_{\text{max}}$ is identical in size with the input $\boldsymbol{\psi}$, stride and padding are 1 and 3, respectively.

remained anchors are regressed to be action proposals. One proposal $\mathbf{P}$ will be selected to be "positive" if its contrast score $s(\mathbf{P})$ is higher than 0.5. The set of all selected "positive" proposals is denoted as $\mathbb{P}$. With $\mathbb{P}$, the regression loss L_{reg} is defined as

$$L_{\text{reg}} = \frac{1}{\|\mathbb{P}\|} \sum_{\mathbf{P} \in \mathbb{P}} \max(m - s(\mathbf{P}), 0), \tag{13}$$

where m is a margin parameter to ensure L_{reg} be lager than 0 and $\|\cdot\|$ denotes the cardinality (number of elements).

The solely training of the action classification is the same as UntrimmedNet [37], which is achieved by minimizing cross-entropy loss between the video-level category label $\mathbf{y}$ and video-level category prediction $\mathbf{x} = \sum_{t=1}^{T} \boldsymbol{\varphi}(t)\boldsymbol{\Psi}(t)$. Intuitively, $\mathbf{x}$ is the weighted summation of all snippet-level predictions in the video, regardless of a snippet is background or not. In the case of videos with multiple labels, $\mathbf{y}$ will be normalized with ℓ_1-norm before training.

But the drawback of this training scheme is evident. All snippets are engaged in training regardless they are background or not, which will introduce noise to the training procedure of action classification. Here we propose a simple yet effective way to further finetune the action classification together with the action localization (C_5 in Table 1). First, we find all snippets covered by action proposals in $\mathbb{P}$ and define this snippet set as $\mathbb{S}$. Intuitively, $\mathbb{S}$ contains all positive snippets that covered by any positive proposal. Then, all snippets not contained by $\mathbb{S}$ are eliminated during the training of action classification. As we assume snippets not covered by any positive proposals as irrelevant, and thus they should be neglected during training. By this way, less noise will be introduced during training. The analysis of the performance contribution of this joint training will be discussed in Section 4.2.

4. Experiments

In this section, we evaluate the TAL performance of the proposed CleanNet, and carry out detailed ablation studies to explore the performance contribution of each component in CleanNet. Meanwhile, we compare our method with existing WS-TAL methods and recent fully-supervised TAL methods on two standard benchmarks.

4.1. Experimental Setting

Evaluation Datasets: THUMOS14 [17] dataset contains 413 untrimmed videos of 20 actions in the temporal action localization task, where 200 untrimmed videos from validation set and 213 untrimmed videos from test set. Each video contains at least one action. The validation and test sets are leveraged to train and evaluate our CleanNet, respectively. ActivityNet v1.2 [9] covers 100 activity classes. The training set includes $4,819$ videos and the validation set includes $2,383$ videos[3], which are used in our training and evaluation, respectively.

Evaluation metric: We evaluate the TAL performances using mean average precision (mAP) values at different levels of IoU thresholds. Both THUMOS14 and ActivityNet v1.2 benchmarks provide standard evaluation implementations, which are directly exploited in our experiments for fair comparison.

Implementation details: We implement our CleanNet using PyTorch [22] on one NVIDIA GeForce GTX TITAN X-p GPU. We adopt stochastic gradient descent (SGD) solver for optimization, with the initial learning rate of 0.0001 and divided by 10 after every 200 batches (one batch contains one whole untrimmed video). Following [28], the anchor sizes are set as 1, 2, 4, 8, 16, 32 snippets for THUMOS14 and 16, 32, 64, 128, 256, 512 snippets for ActivityNet v1.2, respectively. During testing, NMS with IoU threshold 0.4 is used to remove duplicated action proposals. For videos with multiple labels, we perform action localization to all actions with a classification score higher than 0.1.

4.2. Ablation Study

We present multiple ablation studies to explore the performance contribution of each component in CleanNet. We first divide CleanNet into five components as listed in Table 1. Then ablated variants with different combination of these five components are evaluated on THUMOS14, together with the baseline method UntrimmedNet [37], as presented in Table 2.

Using Proposal Evaluator without Training Generator: Note that our action proposal evaluator can assign contrast scores to arbitrary action proposals no matter they are generated from the regressor or not. Thus, without training the action proposal generator, our CleanNet can still function well. In this way, all "proposals without regression" (*i.e.*, anchors) are directly produced by sampling and scored by the proposal evaluator. The rest steps remain the same. This ablated variant is denoted as "Plain-Model" in Table 2, since there is no trainable parameter for action localization. With such settings, the action localization degenerates as a post-processing procedure and achieves a fair comparison with the thresholding component in UntrimmedNet [37]. Our method offers substantial improvements over UntrimmedNet [37] as the mAP is boosted from 15.4% to 21.6% at IoU threshold 0.5. This ablation study validates the efficacy of the contrast scores provided by action proposal evaluator, which is responsible for the major improvement of our TAL performance.

With training of action proposal generator enabled ("CleanNet-Simple" in Table 2), the generated action proposals are more flexible in centroid locations and durations,

[3] In our experiments, there were $4,471$ and $2,211$ videos accessible from YouTube in the training and validation set, respectively.

Table 1: Five main components of CleanNet divided for detailed ablation studies.

Notation	Explanation
C_1	Training the action proposal generator.
C_2	Using s_a to evaluate proposals.
C_3	Using s_s and s_e to evaluate proposals.
C_4	Using TSN sampling strategy.
C_5	Joint finetuning of action classification.

Table 2: TAL performance comparison of our method's variants with different combination of components on THUMOS14 test set, at IoU threshold 0.5.

Method	C_1	C_2	C_3	C_4	C_5	mAP(%)
UntrimmedNet [37]	Baseline					15.4
Plain-Model		✓	✓			21.6
Actioness-Only	✓	✓		✓		1.2
Edgeness-Only	✓		✓	✓		11.4
CleanNet-Simple	✓	✓	✓			22.9
CleanNet-T	✓	✓	✓	✓		23.4
CleanNet-J	✓	✓	✓		✓	23.6
CleanNet	✓	✓	✓	✓	✓	**23.9**

allowing them a better chance to overlap with the ground truth action instances, which leads to further mAP improvements over Plain-Model.

Variants of Proposal Scores: As alternatives to the contrast score $s(\mathbf{P})$ defined in Eq. (12), two ablated versions are studied, termed "Actioness-Only" and "Edgeness-Only" in Table 2. The Actioness-Only replaces Eq. (12) with action score (C_2) only, *i.e.*, $s(\mathbf{P}) = s_a(\mathbf{P})$; while the Edgeness-Only replaces Eq. (12) with starting and ending scores (C_3) only, *i.e.*, $s(\mathbf{P}) = s_s(\mathbf{P}) + s_e(\mathbf{P})$.

As shown in Table 2, without $s_s(\mathbf{P})$ and $s_e(\mathbf{P})$, Actioness-Only suffers from such dramatic performance degradations that it is significantly worse than UntrimmedNet [37]. In addition, the performance of Edgeness-Only is marginally better than Actioness-Only, but the degradation is still evident. This is because without regard to content (the action score), Edgeness-Only is likely to be more susceptible to fluctuations of SCP (*e.g.*, due to noises). Comparing these two variants with others (both C_2 and C_3 are enabled), we provide performance advantage attribution to each term in Eq. (12), confirming $s_a(\mathbf{P})$, $s_s(\mathbf{P})$, and $s_e(\mathbf{P})$ are all indispensable components of the contrast score $s(\mathbf{P})$.

TSN Sampling and Joint Training: With only components C_1, C_2 and C_3, the ablated version "CleanNet-Simple" in Table 2 has already achieved state-of-the-art performance, as presented in Table 3. Besides, enabling the TSN sampling ("CleanNet-T") or the joint finetuning of action classification ("CleanNet-J") can lead to further improvements over CleanNet-Simple. Comparison of CleanNet-T, CleanNet-J and CleanNet shows that, the contributions of C_4 and C_5 are compatible. Finally, with all five components, CleanNet achieves the best action localization performance among all variants.

Table 3: TAL performance comparison on THUMOS14 test set. Fully-supervised methods have access to both video-level category labels and temporal annotations during training; while the weakly-supervised methods only have video-level category labels. Methods sharing the same network backbone are indicated with the symbol *.

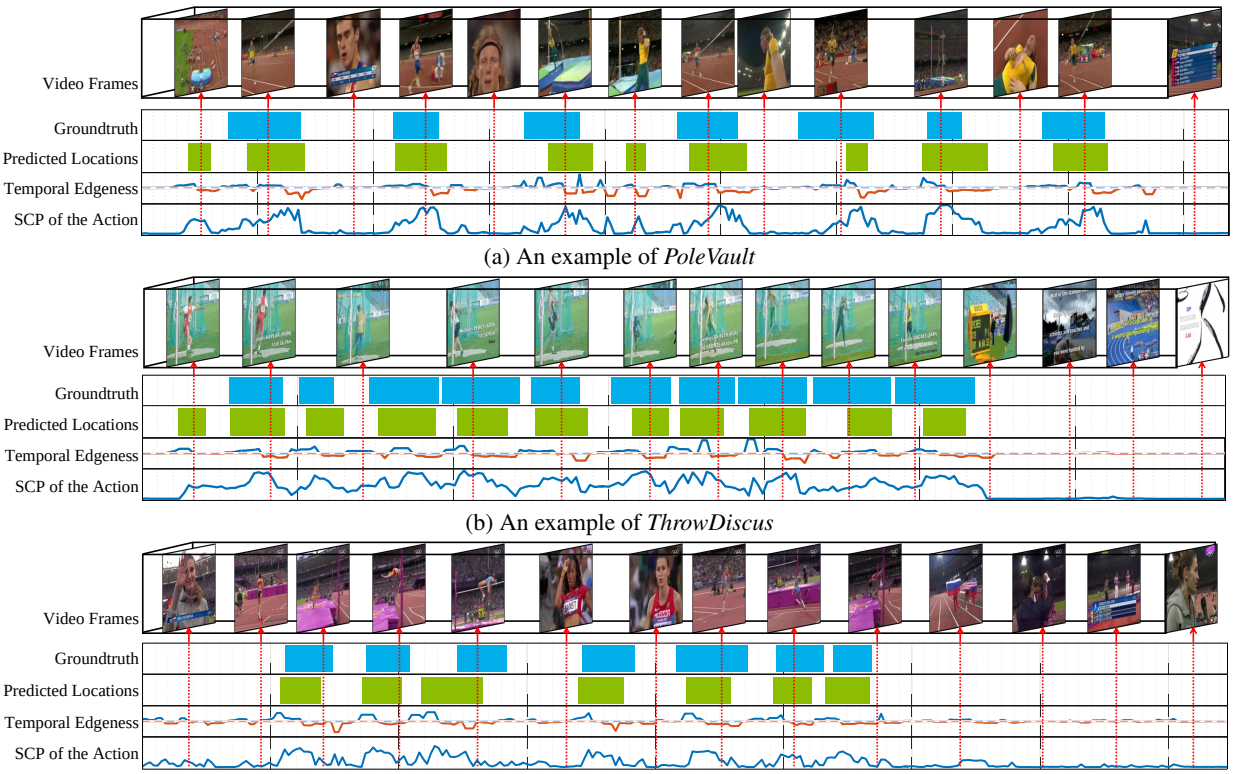
	Method	mAP(%)@IoU				
		0.3	0.4	0.5	0.6	0.7
Fully-supervised	Yuan *et al.* [40]	36.5	27.8	17.8	-	-
	S-CNN [29]	36.3	28.7	19.0	10.3	5.3
	SST [2]	37.8	-	23.0	-	-
	CDC [27]	40.1	29.4	23.3	13.1	7.9
	Dai *et al.* [5]	-	33.3	25.6	15.9	9.0
	R-C3D [39]	44.7	35.6	28.9	-	-
	Gao *et al.* [11]	50.1	41.3	31.0	19.1	9.9
	SSN* [41]	51.9	41.0	29.8	19.6	10.7
	Chao *et al.* [4]	53.2	48.5	42.8	33.8	20.8
	BSN [20]	53.5	45.0	36.9	28.4	20.0
Weakly-supervised	Hide-and-Seek [32]	19.5	12.7	6.8	-	-
	UntrimmedNet* [37]	29.8	22.8	15.4	8.3	4.2
	STPN* [21]	31.1	23.5	16.2	9.8	5.1
	W-TALC* [23]	32.0	26.0	18.8	10.9	6.2
	AutoLoc* [28]	35.8	29.0	21.2	13.4	5.8
	CleanNet-Simple*	36.3	29.6	22.9	13.8	5.3
	CleanNet*	**37.0**	**30.9**	**23.9**	**13.9**	**7.1**

4.3. Performance Comparison

As summarized in Table 3, our CleanNet (shown on last row) outperforms all the compared WS-TAL methods on THUMOS14 test set. The performance advantage of CleanNet is especially evident if compared against thresholding-based methods, *e.g.*, Hide-and-Seek [32], UntrimmedNet [37], STPN [21], and W-TALC [23], which implies the superiority of action proposal generation and evaluation scheme over thresholding. Moreover, CleanNet-Simple can be regarded as a direct comparison to AutoLoc [28], since it differs from AutoLoc only in action proposal evaluation. Thanks to all the distinct designs (see Section 2.3 for details) of CleanNet, it outperforms AutoLoc with all IoU threshold settings. Surprisingly, CleanNet even achieves comparable performances with some fully-supervised methods (*e.g.*, S-CNN [29], SST [2], and CDC [27]). Some qualitative examples are presented in Figure 4.

As the comparison results on ActivityNet v1.2 in Table 4 shown[4], CleanNet outperforms all other weakly-supervised methods on average mAP for IoU thresholds 0.5:0.05:0.95. Note that ActivityNet v1.2 validation set has only an average of 1.5 action instances and 34.6% background per video, while THUMOS14 has an average of 15.4 action instances and 71.4% background per video. Under such a low noise ratio, it is not surprising that the thresholding method W-TALC [23] can achieve good performances when IoU threshold is lower. As the ascending of the IoU threshold,

[4]The mAPs of UntrimmedNet [37] are obtained using the trained models and source codes released by the authors. The mAPs of W-TALC [23] are acquired from the authors.



(a) An example of *PoleVault*

(b) An example of *ThrowDiscus*

(c) An example of *HighJump*

Figure 4: Qualitative TAL examples of the proposed CleanNet on THUMOS14 test set. The ground truth action instances and predicted ones are illustrated with blue and green bars, respectively. Both the corresponding temporal edgeness ($\mathbf{e}$) and snippet-level classification prediction of the specific action (ψ_i) are included. Specifically, to illustrate $\mathbf{e}$, a two-tone color scheme is used, with blue and orange colors representing positive and negative values, respectively.

Table 4: TAL mAP (%) under different IoU thresholds on ActivityNet v1.2 validation set. All methods are trained with weak supervision (video-level labels only). Methods sharing the same network backbone are indicated with the symbol *.

Supervision	IoU threshold	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	Avg
Weakly-supervised	UntrimmedNet* [37]	7.4	6.1	5.2	4.5	3.9	3.2	2.5	1.8	1.2	0.7	3.6
	W-TALC [23]	37.0	**33.5**	**30.4**	25.7	14.6	12.7	10.0	7.0	4.2	1.5	18.0
	AutoLoc* [28]	27.3	24.9	22.5	19.9	17.5	15.1	13.0	10.0	6.8	3.3	16.0
	CleanNet*	**37.1**	33.4	29.9	**26.7**	**23.4**	**20.3**	**17.2**	**13.9**	**9.2**	**5.0**	**21.6**

W-TALC [23] dramatically deteriorates compared with AutoLoc [28] and CleanNet. When the IoU threshold is larger than 0.65, CleanNet significantly outperforms all other methods. This verifies that CleanNet can generate action proposals with large overlaps of ground truth temporal action instances.

To summarize, our CleanNet achieves state-of-the-art WS-TAL performance on both THUMOS14 and ActivityNet v1.2 datasets. Moreover, extensive experiments in the ablation study provide some insights into the performance contribution of each component in CleanNet.

5. Conclusion

We propose CleanNet for WS-TAL, which leverages the temporal contrast among snippet-level action classification predictions to locate the temporal action boundaries. The new action proposal evaluator provides contrast scores as pseudo-supervision to replace manually labeled temporal boundaries. The proposed CleanNet outperforms existing WS-TAL methods on both the THUMOS14 and ActivityNet v1.2 datasets. It can even outperform some recent fully-supervised TAL methods.

6. Acknowledgment

This work was supported partly by National Key R&D Program of China Grant 2017YFA0700800, NSFC Grants 6162930, 61773312, and 61976171, China Postdoctoral Science Foundation Grant 2019M653642, and Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001.

References

[1] M. Asadiaghbolaghi, A. Clapes, M. Bellantonio, H. J. Escalante, V. Poncelopez, X. Baro, I. Guyon, S. Kasaei, and S. Escalera. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *FG*, pages 476–483, 2017. 1

[2] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, pages 6373–6382, 2017. 3, 7

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 4724–4733, 2017. 2

[4] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, pages 1130–1139, 2018. 3, 7

[5] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen. Temporal context network for activity localization in videos. In *ICCV*, pages 5727–5736, 2017. 3, 7

[6] O. Dan, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, pages 1817–1824, 2013. 1

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005. 2

[8] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, pages 768–784, 2016. 3

[9] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 6

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016. 2

[11] J. Gao, Z. Yang, and R. Nevatia. Cascaded boundary regression for temporal action detection. In *BMVC*, 2017. 7

[12] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, pages 3628–3636, 2017. 3

[13] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 3

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 3

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 3

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. 2

[17] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 6

[18] S. M. Kang and R. P. Wildes. Review of action recognition and detection methods. *arXiv preprint arXiv:1610.06906*, 2016. 1

[19] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005. 2

[20] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018. 7

[21] P. Nguyen, T. Liu, G. Prasad, and B. Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, pages 6752–6761, 2018. 1, 3, 7

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6

[23] S. Paul, S. Roy, and A. K. Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, pages 588–607, 2018. 1, 3, 7, 8

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 4

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 3, 4

[26] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, pages 1234–1241, 2012. 2

[27] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, pages 1417–1426, 2017. 7

[28] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, pages 154–171, 2018. 3, 6, 7, 8

[29] Z. Shou, D. Wang, and S. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016. 1, 3, 7

[30] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 2

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[32] K. K. Singh and Y. J. Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, pages 3524–3533, 2017. 1, 3, 7

[33] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *ACM MM*, pages 371–380, 2015. 3

[34] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *CVPR*, pages 4597–4605, 2015. 2

[35] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *CVPR*, pages 4489–4497, 2015. 2

[36] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013. 2

[37] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, pages 4325–4334, 2017. 1, 3, 6, 7, 8

[38] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016. 2, 3, 4

[39] H. Xu, A. Das, and K. Saenko. R-c3d: region convolutional 3d network for temporal activity detection. In *ICCV*, pages 5794–5803, 2017. 3, 7

[40] Z.-H. Yuan, J. C. Stroud, T. Lu, and J. Deng. Temporal action localization by structured maximal sums. In *CVPR*, pages 3684–3692, 2017. 7

[41] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2933–2942, 2017. 3, 7