Colar: Effective and Efficient Online Action Detection by Consulting Exemplars

Le Yang Junwei Han* Dingwen Zhang
School of Automation, Northwestern Polytechnical University, China
https://nwpu-brainlab.gitee.io/index_en

Abstract

Online action detection has attracted increasing research interests in recent years. Current works model historical dependencies and anticipate the future to perceive the action evolution within a video segment and improve the detection accuracy. However, the existing paradigm ignores category-level modeling and does not pay sufficient attention to efficiency. Considering a category, its representative frames exhibit various characteristics. Thus, the category-level modeling can provide complimentary guidance to the temporal dependencies modeling. This paper develops an effective exemplar-consultation mechanism that first measures the similarity between a frame and exemplary frames, and then aggregates exemplary features based on the similarity weights. This is also an efficient mechanism, as both similarity measurement and feature aggregation require limited computations. Based on the exemplar-consultation mechanism, the long-term dependencies can be captured by regarding historical frames as exemplars, while the category-level modeling can be achieved by regarding representative frames from a category as exemplars. Due to the complementarity from the category-level modeling, our method employs a lightweight architecture but achieves new high performance on three benchmarks. In addition, using a spatio-temporal network to tackle video frames, our method makes a good trade-off between effectiveness and efficiency. Code is available at https://github.com/VividLe/Online-Action-Detection.

1. Introduction

With the development of mobile communications, video has become a powerful medium to record life and transform information. As a result, video understanding technologies have aroused increasing research interests. Among these technologies, temporal action detection [36, 39, 62] can discover action instances from untrimmed videos and extract valuable information. Well-performed action detection algorithms can benefit smart surveillance [32], anomaly detection [4] *etc*. In recent years, along with action detection technologies becoming mature, a more challenging but more practical task, namely online action detection, has been proposed [8]. The online action detection algorithm tackles a streaming video, reports the occurrence of an action instance, and keeps alarming until the action ends [8]. In inference, the algorithm only employs historical frames that have been observed, but has no access to future frames.

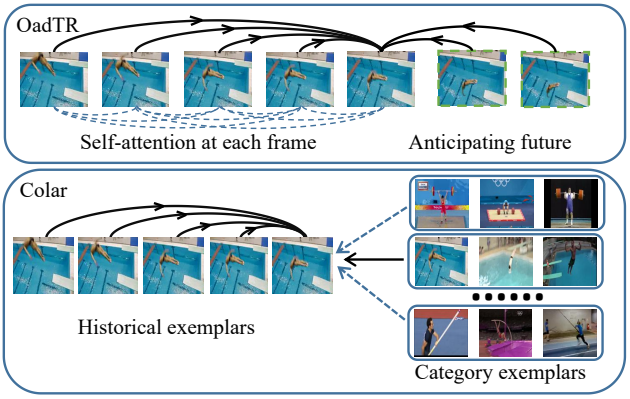


Figure 1. Comparison between existing state-of-the-art method OadTR [42] and our proposed Colar. Unlike OadTR, Colar consults historical exemplars to model long-term dependencies and consults category exemplars to capture category-level particularity, forming an effective and efficient method.

As an early exploration, Geest *et al*. [8] discovered the importance of modeling long-term dependencies. Later, Xu *et al*. [45] revealed the value of anticipating future status to enhance the long-term dependencies modeling. OadTR [42] recently utilized the multi-head self-attention module to jointly model historical dependencies and anticipate the future, which achieved promising online action detection results.

As an under-explored domain, there are three core challenges for online action detection: How to model long-term dependencies? How to associate a frame with representative frames from the same category? How to conduct detection efficiently? Existing works [8, 42, 45] primarily focus on the long-term dependencies modeling, but ignore the

*Corresponding author.

other two challenges. However, as shown in Figure 1 (a), both analyzing historical frames and anticipating future status only model relationships within a video segment, leaving the category-level modeling under-explored. Because an action category contains multiple instances and each instance exhibits special appearance and motion characteristic, the guidance of exemplary frames can make the online detection algorithm more robust to resist noises within a video segment. In addition, a practical online action detection algorithm should always consider the computational efficiency, including both the efficiency to perform online detection and the efficiency to extract video features.

This paper develops an exemplar-consultation mechanism to tackle above three challenges in a unified framework. The exemplar-consultation mechanism first jointly transforms a frame and its exemplary frames to the key space and value space. Then, it measures the similarity in the key space and employs the similarity to aggregate information in the value space. As both feature transformation and similarity measurement require limited computations, the proposed exemplar-consultation mechanism is efficient. Considering a video segment, we can effectively model long-term dependencies by using historical frames as exemplars based on the exemplar-consultation mechanism. As we only compare one frame with its historical frames, rather than performing self-attention on all frames, the computational burden is alleviated. Similarly, we can also regard representative frames of each category as exemplars and conduct category-level modeling based on the exemplar-consultation mechanism. Compared with a video segment, category exemplars can provide complementary guidances and make the algorithm more robust.

By **co**nsulting exemp**lar**s, we build a unified framework, namely Colar, to perform online action detection, as shown in Figure 2. Colar maintains the dynamic branch and the static branch in parallel, where the former models long-term dependencies within a video segment and the latter models category-level characteristics. In the dynamic branch, Colar consults previous frames and aggregates historical features. In the static branch, Colar first obtains category exemplars via clustering, then consults exemplars and aggregate category features. Finally, two classification scores are fused to detect actions. Moreover, we analyze the running time bottleneck of existing works and discover the expensive costs to extract flow features. Thus, we employ a spatio-temporal network to only dispose of video frames and perform end-to-end online action detection, which only takes 9.8 seconds to tackle a one-minute video. To sum up, this paper makes the following contributions:

- We make an early attempt to conduct category-level modeling for the online action detection task, which provides holistic guidance and makes the detection algorithm more robust.
- We propose the exemplar-consultation mechanism to compare similarities and aggregate information, which can efficiently model long-term dependencies and category particularities.
- Due to the effectiveness of the exemplar-consultation mechanism and the complimentary guidance from category-level modeling, our method employs a lightweight architecture. Still, it achieves superior performance and builds new state-of-the-art performance on three benchmarks.

2. Related work

Modeling temporal dependencies. Different from image-based task, *e.g*. detection [12–14, 48], localization [15, 55, 59] and segmentation [56], it is crucial to model temporal dependencies for online action detection. Existing works rely on recurrent networks, including both LSTM-based methods [9, 45, 50] and GRU-based methods [11]. Specifically, Geest *et al*. [9] proposed a two-stream LSTM [17] network. Similarly, TRN [45] employed LSTM blocks to model historical temporal dependencies. Recently, OadTR [42] drove the recurrent-network paradigm into a transformer-based paradigm and effectively captured the long-term relationship via self-attention. Although OadTR [42] effectively models long-term dependencies, the self-attention process for all frames leads to the computational burden problem. This work regards historical frames as exemplars and utilizes the exemplar-consultation mechanism to model long-term dependencies.

Anticipating future. Although online action detection algorithms cannot access future frames, anticipating future features can assist the decision of current frame. In RED [22], Gao *et al*. estimated features for future frames and calculated the classification loss and the feature regression loss to improve the anticipation quality, which is further developed by TRN [45] and OadTR [42]. In this paper, the static branch employs the exemplar-consultation mechanism to compare a frame with representative exemplars of each category and brings complementary information to the dynamic branch.

Offline action detection. The offline action detection algorithm aims to discover action instances from untrimmed videos [5, 29, 30], where all video frames can be utilized. Some algorithms [19,36,39,44] tackled video frames to perform localization. In addition, a majority of works [28, 62] first extracted video features from powerful backbone networks [3, 40, 60, 61], then performed action localization based on video features. From the view of anchor mechanism, the representative works include anchor-based methods [28, 62] and anchor-free methods [26, 27, 47]. Besides, multiple effective modules have been proposed, *e.g*. graph convolutional module [53, 54]. Moreover, action detection

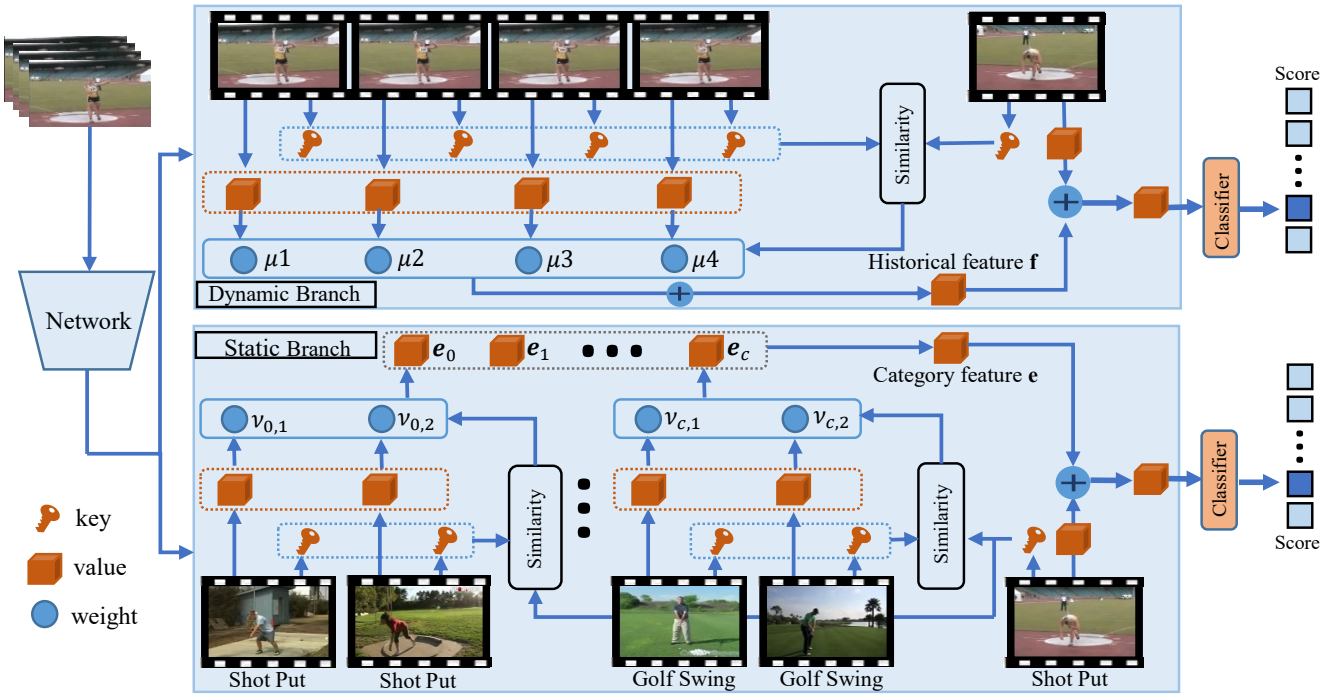


Figure 2. Framework of the proposed Colar method for online action detection. Given a video, the dynamic branch compares a frame with its historical exemplars and models temporal dependencies, while the static branch compares a frame with category exemplars and captures the category particularity.

under the weakly supervised setting [46,51,52,58] was also well explored.

The primary difference between online action detection and offline action detection algorithms lies in whether future frames can be accessed. In offline algorithms, Xu *et al*. [44] performed data augmentation via playing the video in reverse order, while Zhu [62] modeled the relationship among multiple proposals within a video. However, these procedures are unsuitable for the studied online action detection task.

Space-time memory network. Oh *et al*. [33] proposed the space-time memory network to efficiently connect a frame and its previous frames via space-time memory read. It has verified effective performance on modeling temporal information, and has been extended to multiple tasks, *e.g*. video object detection [6], video object segmentation [18, 31], tracking [25]. In contrast to the space-time memory network, our static branch employs the exemplar-consultation mechanism to model the intra-category relationship. Specifically, it first aggregates particular features from each category, and then combines multiple features to obtain the category feature.

3. Method

Given a video stream, the online action detection algorithm should report the occurrence once the action starts and keep alarming until the action ends. The learning process is guided by the frame-level classification label $\mathbf{y} = [y_0, y_1, ..., y_C]$, where $y_c \in \{0, 1\}$ indicates whether frame $\mathbf{f}_0$ belongs to the c^{th} category. As shown in Figure 2, we first employ a backbone network to extract video features. Then, we propose the dynamic branch to model long-term dependencies within a segment and propose the static branch to capture the holistic particularity for each category. Finally, two detection results are fused to perform the online action detection task.

3.1. Dynamic branch

As neighboring frames can provide rich contextual cues to determine the category label of the current frame, the core idea of the dynamic branch is to model local evolution by comparing a frame with its previous historical frames and dynamically aggregating the local features. The upper part of Figure 2 exhibits the detailed operations in the dynamic branch. Compared with the standard multi-head self-attention mechanism of OadTR [42], our proposed dynamic branch makes two reasonable designs, which sufficiently benefit the online action detection task. First, we use temporal convolution with kernel size 3 to model local cues among historical frames, which is complementary to the global modeling of self-attention. Second, we make two simplifications over OadTR, *i.e*. removing the class to-

ken and replacing multi-head self-attention with one-head attention on the current frame. The simplifications reduce learning difficulty and benefit the performance when training data is not rich enough.

Given a video feature sequence, we first transform a feature $\mathbf{f}_t$ to the key space and the value space, where the former is responsible for comparing similarity, and the latter can be used for feature aggregation.

$$\mathbf{f}_t^k = \Phi^k(\mathbf{f}_t), \;\; \mathbf{f}_t^v = \Phi^v(\mathbf{f}_t), \tag{1}$$

where Φ^k and Φ^v indicate two convolutional layers in the dynamic branch. Then, we measure the pair-wise affinity between $\mathbf{f}_0^k$ and other key features (*e.g.* $\mathbf{f}_t^k$) via calculating the cosine similarity:

$$\mu_t = cos(\mathbf{f}_0^k, \mathbf{f}_t^k) = \frac{\mathbf{f}_0^k \cdot \mathbf{f}_t^k}{||\mathbf{f}_0^k|| \cdot ||\mathbf{f}_t^k||}. \tag{2}$$

Given a series of affinity values $[\mu_{-T}, ..., \mu_{-1}, \mu_0]$, we perform softmax normalization and obtain the attention mask $[\hat{\mu}_{-T}, ..., \hat{\mu}_{-1}, \hat{\mu}_0]$. As each element μ_t indicates the similarity between the previous t^{th} frame and the current frame, we can aggregate value features among previous frames and obtain the historical feature $\mathbf{f}$:

$$\mathbf{f} = \sum_{t=-T}^{0} \hat{\mu}_t \cdot \mathbf{f}_t^v. \tag{3}$$

In the end, the dynamic branch jointly considers value feature $\mathbf{f}_0^v$ and the historical feature $\mathbf{f}$ (*e.g.* via summation) and conduct online action detection:

$$\mathbf{s}^{\mathrm{d}} = \Omega^{\mathrm{d}}(\mathbf{f}_0^v, \mathbf{f}|\Theta^{\mathrm{d}}), \tag{4}$$

where Ω^{d} is the classifier in the dynamic branch with parameter Θ^{d}, and $\mathbf{s}^{\mathrm{d}} \in \mathbb{R}^{C+1}$ is the classification score from the dynamic branch.

3.2. Static branch

Considering action instances from the same category, some instances with distinctive appearance characteristics and clear motion patterns can be selected as exemplars to represent this category. We employ the K-means clustering algorithm for each category, carry out clustering, and obtain M exemplary features. On this basis, the online action detection task can be formulated as comparing a frame with representative exemplars of each category. As a result, the static branch can provide complementary cues to the dynamic branch and makes the online detection algorithm robust to noises within the local video segment.

Before stepping to detailed operations in the static branch, it is necessary to analyze its efficiency. First, using another branch increases a certain computation. However, compared with OadTR [42], we not only simplify the attention computation but also remove decoder layers. Thus, our holistic computation is smaller than OadTR [42], and we require less memory as well (see experiments in Sec.4.2). In addition, even given a dataset with millions of samples and thousands of categories, the modern implementation [23] of the K-Means algorithm can still efficiently generate exemplars, as verified by DeepCluster [2].

As shown in the bottom part of Figure 2, the static branch operates with the category exemplars $\{\mathcal{E}_c = [\mathbf{e}_{c,1}, \mathbf{e}_{c,2}, ..., \mathbf{e}_{c,M}]\}_{c=0}^{C}$ to classify feature $\mathbf{f}_0$, where each category contains M representative exemplars. At first, we convert each exemplar $\mathbf{e}_{c,i}$ to the key space and the value space:

$$\mathbf{e}_{c,i}^k = \Psi^k(\mathbf{e}_{c,i}), \;\; \mathbf{e}_{c,i}^v = \Psi^v(\mathbf{e}_{c,i}), \tag{5}$$

and convert the frame feature $\mathbf{f}_0$ to the key space and value space as well:

$$\mathbf{e}_0^k = \Gamma^k(\mathbf{f}_0), \;\; \mathbf{e}_0^v = \Gamma^v(\mathbf{f}_0), \tag{6}$$

where Ψ^k, Ψ^v, Γ^k and Γ^v indicate convolutional layers. In the key space, we can measure the similarity between feature $\mathbf{f}_0$ and exemplar $\mathcal{E}_c$ from the c^{th} category:

$$\nu_{c,i} = cos(\mathbf{e}_0^k, \mathbf{e}_{c,i}^k) = \frac{\mathbf{e}_0^k \cdot \mathbf{e}_{c,i}^k}{\|\mathbf{e}_0^k\| \cdot \|\mathbf{e}_{c,i}^k\|}. \tag{7}$$

Based on the pair-wise similarity between $\mathbf{e}_0^k$ and all exemplars $[\mathbf{e}_{c,1}, \mathbf{e}_{c,2}, ..., \mathbf{e}_{c,M}]$ from the c^{th} category, we can first calculate the attention mask $[\hat{\nu}_{c,1}, ..., \hat{\nu}_{c,M}]$ via softmax normalization, and then aggregate all exemplars to represent the current frame from the perspective of the c^{th} category:

$$\mathbf{e}_c = \sum_{i=1}^{M} \hat{\nu}_{c,i} \cdot \mathbf{e}_{c,i}^v. \tag{8}$$

After comparing the current frame with representative exemplars of all categories, we obtain category-specific features $[\mathbf{e}_0, \mathbf{e}_1, ..., \mathbf{e}_C]$. Considering a feature from the c^{th} category, it would be similar to exemplars from the c^{th} category while be different from other exemplars. Thus, we use a convolutional layer to estimate the attention weight $\mathbf{a} \in \mathbb{R}^{C+1}$ and aggregate category feature $\mathbf{e}$:

$$\mathbf{e} = \sum_{c=0}^{C} a_c \cdot \mathbf{e}_c. \tag{9}$$

The exemplary feature $\mathbf{e}$ is generated from all exemplars and can reveal the category characteristics. In the end, the static branch employs both value feature $\mathbf{e}_0^v$ and category feature $\mathbf{e}$ to predict the classification score $\mathbf{s}^{\mathrm{s}}$:

$$\mathbf{s}^{\mathrm{s}} = \Omega^{\mathrm{s}}(\mathbf{e}_0^v, \mathbf{e}|\Theta^{\mathrm{s}}), \tag{10}$$

where Ω^{s} is the classifier with parameter Θ^{s}.

3.3. Efficient online action detection

Given a series of pre-extracted video features, the dynamic branch connects a frame with its historical neighbors and models local evolution, while the static branch compares a frame with representative exemplars and models category particularity. It is convenient to fuse the predictions of two branches and detect actions online. However, the feature extraction process, especially calculating optical flows, requires heavy computations, which prevents us from conducting online action detection in practical scenarios.

To alleviate the computational burdens, we can employ a spatio-temporal network to tackle video frames and provide representative features for the dynamic and the static branch. Considering the video recognition performance and calculation efficiency, we utilize the ResNet-I3D network [41], discard the last classification layer, and construct our feature extraction backbone. Given a video sequence with T frames, the output of the backbone network is $\mathbf{x} \in \mathbb{R}^{D \times T/8}$, where D indicates the feature dimension. In practice, as the benchmark datasets contain limited training videos, we find frozen the first three blocks can produce more accurate detection results.

3.4. Training and inference

Given a frame, the dynamic branch and the static branch predict its classification score $\mathbf{s}^{\mathrm{d}}$ and $\mathbf{s}^{\mathrm{s}}$, respectively. We calculate the cross-entropy loss to guide the learning process:

$$\mathcal{L}_{cls}^{\mathrm{d}} = -\sum_{c=0}^{C} \mathbf{y}_c \log(\hat{\mathbf{s}_c}^{\mathrm{d}}), \quad \mathcal{L}_{cls}^{\mathrm{s}} = -\sum_{c=0}^{C} \mathbf{y}_c \log(\hat{\mathbf{s}_c}^{\mathrm{s}}), \tag{11}$$

where $\hat{\mathbf{s}_c}^{\mathrm{d}}$ and $\hat{\mathbf{s}_c}^{\mathrm{s}}$ indicate scores after softmax normalization. Besides, as the dynamic and static branches tackle the same frame, two classification scores should be consistent. Thus, we introduce the consistency loss $\mathcal{L}_{cons}$ to enable mutual guidance among two branches:

$$\mathcal{L}_{cons} = \mathcal{L}_{KL}(\hat{\mathbf{s}}^{\mathrm{d}} \| \hat{\mathbf{s}}^{\mathrm{s}}) + \mathcal{L}_{KL}(\hat{\mathbf{s}}^{\mathrm{s}} \| \hat{\mathbf{s}}^{\mathrm{d}}), \tag{12}$$

where $\mathcal{L}_{KL}$ indicates the KL-divergence loss. As verified by Zhang *et al*. [57], the consistency loss can lead to a robust model with better generalization. To sum up, the training process is guided by the following loss:

$$\mathcal{L} = \mathcal{L}_{cls}^{\mathrm{d}} + \mathcal{L}_{cls}^{\mathrm{s}} + \lambda \mathcal{L}_{cons}, \tag{13}$$

where λ is a trade-off parameter.

In inference, the dynamic classification score $\mathbf{s}^{\mathrm{d}}$ and the static classification score $\mathbf{s}^{\mathrm{s}}$ are fused via a balance coefficient β to perform online action detection:

$$\mathbf{s} = \beta\, \hat{\mathbf{s}}^{\mathrm{s}} + (1-\beta)\, \hat{\mathbf{s}}^{\mathrm{d}}. \tag{14}$$

Table 1. Comparison experiments on THUMOS14 dataset, measured by mAP (%).

Setups	Method	mAP(%)
Offline	CNN [38]ICLR15	34.7
	CNN [37]NIPS14	36.2
	LRCN [10]CVPR15	39.3
	MultiLSTM [49]IJCV18	41.3
	CDC [35]CVPR17	44.4
Online (TSN-Anet)	RED [22]BMVC17	45.3
	TRN [45]ICCV19	47.2
	IDU [11]CVPR20	50.0
	OadTR [42]ICCV21	58.3
	Colar	**59.4**
Online (TSN-Kinetics)	IDU [11]CVPR20	60.3
	OadTR [42]ICCV21	65.2
	Colar	**66.9**
RGB end-to-end	Colar	58.6

4. Experiments

4.1. Setups

Dataset. We carry out experiments on three widely used benchmarks, THUMOS14 [21], TVSeries [8] and HDD [34]. THUMOS14 [21] includes sports videos from 20 action categories, where the validation set and test set contain 200 and 213 videos, respectively. On THUMOS14, challenges for online action detection include drastic intra-category varieties, motion blur, short action instances *etc*. We follow previous works [9, 11, 42, 45], train the model on the validation set, and evaluate performance on the test set.

TVSeries [8] collects about 16 hours of videos from 6 popular TV series. The dataset contains 30 daily actions, where the total instance number is 6231. The TVSeries dataset exhibits some challenging characteristics, *e.g*. temporal overlapping action instances, a large proportion of background frames and unconstrained perspectives.

HDD [34] contains 104 hours of human driving video, belonging to 11 action categories. The videos were collected from 137 driving sessions using an instrumented vehicle equipped with different sensors. Following existing works [9, 42, 45], we use 100 sessions for training and 37 sessions for testing.

Metric. We adopt mean average precision (mAP) and calibrated mean average precision (cmAP) to measure the performance of online action detection algorithms. As for mAP, we first collect classification scores for all frames and then calculate precision and recall based on sorted results. Afterward, we calculate interpolated average precision to obtain AP scores for a category and finally regard the mean

Table 2. Comparison experiments on TVSeries dataset, measured by mcAP(%).

Setups	Method	mcAP(%)
RGB	LRCN [10]CVPR15	64.1
	RED [22]BMVC17	71.2
	2S-FN [9]WACV18	72.4
	TRN [45]ICCV19	75.4
	IDU [11]CVPR20	76.6
Flow	FV-SVM [8]ECCV2016	74.3
	IDU [11]CVPR20	80.3
Online (TSN-Anet)	RED [22]BMVC17	79.2
	TRN [45]ICCV19	83.7
	IDU [11]CVPR20	84.7
	OadTR [42]ICCV21	85.4
	Colar	**86.0**
Online (TSN-Kinetics)	IDU [11]CVPR20	86.1
	OadTR [42]ICCV21	87.2
	Colar	**88.1**
RGB end-to-end	Colar	86.8

Table 3. Comparison experiments on HDD dataset, measured by mAP (%).

Setups	Method	mAP(%)
Sensors	CNN [8]ICLR15	22.7
	LSTM [34]CVPR18	23.8
	ED [22]BMVC17	27.4
	TRN [45]ICCV19	29.2
	OadTR [42]ICCV21	29.8
	Colar	**30.6**

Table 4. Comparison between our proposed Colar method and existing methods. The inference time (in second) is measured on a 1080Ti GPU when tackling the same one-minute video. Both "Colar*" and Colar† directly tackle video frames, where the former uses a fixed backbone and the latter is end-to-end trained.

Method	RGB Feature	Optical Flow	Flow Feature	Action Detection	Inference Time	mAP (%)
Given pre-extracted features, Colar is faster and more accurate.						
IDU [11]	2.3	39.8	4.4	52.8	99.3	60.3
OadTR [42]	2.3	39.8	4.4	4.7	51.2	65.2
Colar	2.3	39.8	4.4	4.2	**50.7**	**66.9**
OadTR-Flow	-	39.8	4.4	4.5	48.7	57.8
Colar-Flow	-	39.8	4.4	4.0	**48.2**	**59.6**
OadTR-RGB	2.3	-	-	4.5	6.8	51.2
Colar-RGB	2.3	-	-	4.0	**6.3**	**52.1**
Given frames, Colar provides a trade-off between speed and accuracy.						
Colar*	5.8	-	-	4.0	9.8	53.4
Colar†	5.8	-	-	4.0	9.8	**58.8**

value of AP scores among all categories as mAP. Considering the drastically imbalanced frame numbers of different categories, Geest *et al*. [8] proposed to calibrate the mAP score. In particular, we first calculate the ratio w between background frames and action frames and then calculate the calibrated precision as:

$$cPre(i) = \frac{w \cdot TP(i)}{w \cdot TP(i) + FP(i)}. \tag{15}$$

Afterward, the calibrated average precision cAP for a category can be calculated as:

$$cAP = \frac{\sum_i cPre(i) \cdot \mathbf{1}(i)}{\sum_i \mathbf{1}(i)}, \tag{16}$$

where $\mathbf{1}(\cdot)$ indicates whether the i^{th} frame belongs to the considered action category. Finally, cmAP can be obtained via calculating the mean value among all cAPs.

Implementation details. Following previous works [9, 11, 42, 45], we first conduct experiments with pre-extracted features. The feature extractor uses the two-stream network [43], whose spatial stream adopts ResNet-200 [16] and temporal stream adopts BN-Inception [20]. We report two experiments where the two-stream network [40, 43] is trained on the ActivityNet v1.3 dataset [1] or the Kinetics-400 [3] dataset to verify the generalization of the proposed Colar method. As for end-to-end online action detection, our backbone network is based on the ResNet50-I3D architecture [41], where the last average pooling layer and classification layer are removed. The ResNet50-I3D network is pretrained on Kinetics-400 [3] dataset, and we use the weight file provided by MMAction2 [7]. In training, we freeze the first three blocks of the backbone network. Video frames are extracted with a frame rate of 25*fps*, where the spatial size is set as 224×224. We use the Adam [24] algorithm to optimize the whole network and set the batchsize as 16. The initial learning rate is 3×10^{-4} and decays every five epochs.

4.2. Comparison experiments

Quantitative comparisons. We make a comparison with current state-of-the-art methods [9, 11, 42, 45] and consistently build new high performance on THUMOS14 [21], TVSeries [8], and HDD [34] benchmarks. As shown in Table 1, based on TSN-ActivityNet features, our Colar brings an mAP gain of 1.1% over OadTR [42], and the improvements would be an mAP of 1.7% if the comparison is based on TSN-Kinetics features. The consistent improvements

Table 5. Detailed online action detection performances under different action portions, measured by mcAP (%) on TVSeries dataset.

Setups	Method	Portion of actions									
		0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1
RGB	CNN [8]ICLR15	61.0	61.0	61.2	61.1	61.2	61.2	61.3	61.5	61.4	61.5
	LSTM [8]ICLR15	63.3	64.5	64.5	64.3	65.0	64.7	64.4	64.4	64.4	64.3
Flow	FV-SVM [8]ECCV2016	67.0	68.4	69.9	71.3	73.0	74.0	75.0	75.4	76.5	76.8
Online (TSN-Anet)	TRN [45]ICCV19	78.8	79.6	80.4	81.0	81.6	81.9	82.3	82.7	82.9	83.3
	IDU [11]CVPR20	**80.6**	81.1	81.9	82.3	82.6	82.8	82.6	82.9	83.0	83.9
	OadTR [42]ICCV21	79.5	83.9	86.4	85.4	86.4	87.9	87.3	87.3	85.9	84.6
	Colar	80.2	**84.4**	**87.1**	**85.8**	**86.9**	**88.5**	**88.1**	**87.7**	**86.6**	**85.1**
Online (TSN-Kinetics)	IDU [11]CVPR20	81.7	81.9	83.1	82.9	83.2	83.2	83.2	83.0	83.3	86.6
	OadTR [42]ICCV21	81.2	84.9	87.4	87.7	88.2	89.9	88.9	88.8	87.6	86.7
	Colar	**82.3**	**85.7**	**88.6**	**88.7**	**88.8**	**91.2**	**89.6**	**89.9**	**88.6**	**87.3**
RGB end-to-end	Colar	80.8	84.4	87.2	87.5	87.8	89.4	88.4	88.5	87.3	86.4

over current state-of-the-art methods verify the efficacy of our proposed exemplar-consultation mechanism. In addition, the proposed Colar can directly tackle video frames and perform online action detections, which achieves 58.6% mAP. In addition to THUMOS14, experiments on TVSeries [8] and HDD [34] benchmarks also verify the superiority of our method, as shown in Table 2 and Table 3.

Effectiveness and efficiency. Table 4 analyzes the performance and running time under different setups. When using pre-extracted features, Colar performs superior to existing methods [11, 42]. It is worth noting that extracting RGB and flow features takes 46.5 seconds, of which calculating optical flow costs the majority of the time. When only flow features or RGB features are available, the feature extraction cost is reduced, but both OadTR [34] and our Colar observe performance drop. In particular, it costs 44.2s to extract flow features, where OadTR [34] and Colar observe 7.4% and 7.3% performance drops, respectively. The cost to extract RGB features is small, but the online detection performance decreases a lot.

Given the ResNet50-I3D network [41], we first extract features from video frames and then train the proposed Colar method, which gets 53.4%. In contrast, the proposed end-to-end learning paradigm achieves 58.8%. To sum up, our proposed Colar method achieves a good balance between effectiveness and efficiency. Given pre-extracted features, Colar makes accurate detection results. Given only video frames, Colar costs 9.8 seconds to tackle a one-minute video and achieves comparable performance. In addition, we measure the memory cost under identical setups, where Colar requires 2235M memory and OadTR [42] requires 4375M memory.

Performance under different action portions. Table 5 elaborately studies the online action detection performance when different action portions are observed. The proposed Colar achieves promising accuracy when using the TSN-ActivityNet feature, the TSN-Kinetics feature, and only video frames. In particular, considering the most severe cases that only the first 10% portion of actions are observed, the previous state-of-the-art method OadTR [42] shows inferior performance to IDU [11]. However, the proposed Colar consistently exceeds OadTR, due to that the static branch effectively connects a frame with representative exemplars of each category and provides complimentary guidance.

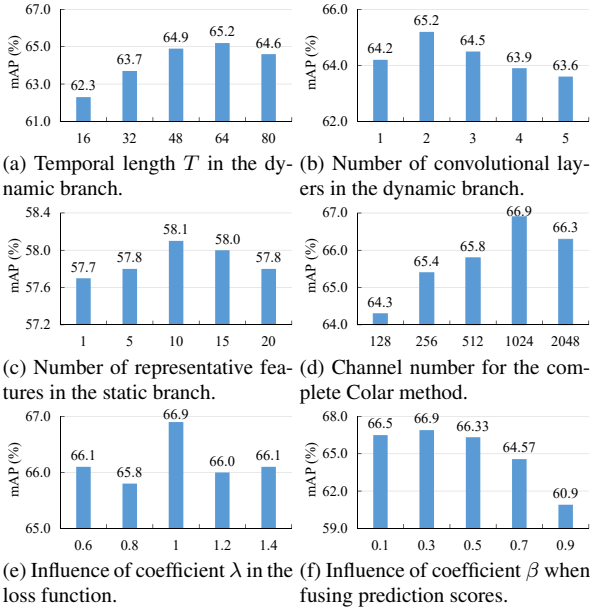
Table 6. Ablation studies about the efficacy of each component, measured by mAP(%) on three benchmarks.

Dynamic	Static	$\mathcal{L}_{cons}$	THUMOS14	TVSeries	HDD
✓			65.2	86.3	29.5
	✓		58.1	83.5	26.4
✓	✓		65.8	86.9	29.9
✓	✓	✓	66.9	88.1	30.6

4.3. Ablation experiments

Efficacy of each component. The proposed Colar method consists of the dynamic branch and the static branch, as well as a consistency loss $\mathcal{L}_{cons}$ to enable mutual guidance between two branches. Table 6 studies the efficacy of each component on all three benchmark datasets. Firstly, the dynamic branch performs superior to the static branch, demonstrating the necessity of carefully modeling temporal dependencies. Besides, without the consistency loss, directly fusing prediction scores of two branches (*e.g.* using Eq. (14)) only observes limited improvements, while $\mathcal{L}_{cons}$ can further improve the detection performance.

Ablations about the dynamic branch. As shown in



(a) Temporal length T in the dynamic branch. (b) Number of convolutional layers in the dynamic branch.

(c) Number of representative features in the static branch. (d) Channel number for the complete Colar method.

(e) Influence of coefficient λ in the loss function. (f) Influence of coefficient β when fusing prediction scores.

Figure 3. Ablation studies about hyper-parameters in the proposed Colar method, measured by mAP (%) on THUMOS14 dataset.

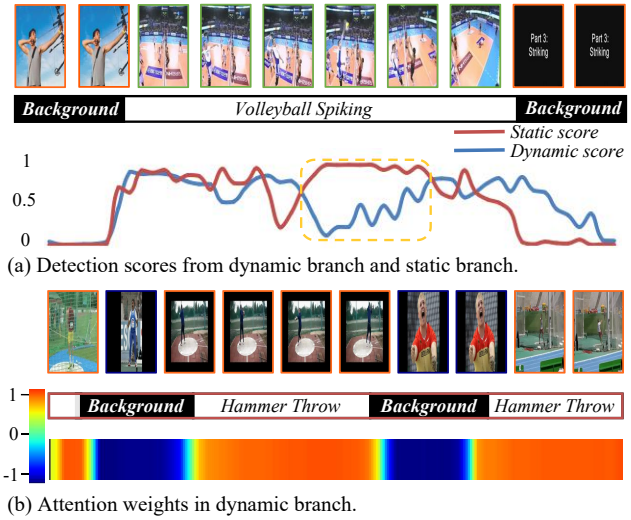
Figure 3 (a), we first study the influence of temporal scope T in modeling temporal dependencies and find 64 is a proper choice for the dynamic branch. The too-short temporal scope is insufficient to perceive evolvement within a video segment, while too long temporal scope would bring noises. In addition, we vary the number of convolutional layers and choose two layers, as shown in Figure 3 (b).

Ablations about the static branch. Based on K-Means clustering, the number of exemplars is an influential parameter for the static branch. As shown in Figure 3 (c), the ability of limited exemplars is insufficient and overwhelming exemplars would damage the performance as well.

Ablations about the complete method. Given the complete method, Figure 3 (d) studies the performance under different feature channels and verifies 1024 is a proper choice. Figure 3 (e) studies the coefficient λ for the consistency loss in the training phase, while Figure 3 (f) verifies the influence of coefficient β in the inference phase. We find $\lambda = 1$ and $\beta = 0.3$ are proper choices.

4.4. Qualitative analysis

Figure 4 qualitatively analyzes the proposed Colar method. Figure 4 (a) exhibits the static and dynamic scores within a video segment. Because the *Volleyball Spiking* instance shows dramatic viewpoint changes, the dynamic branch predicts low confident scores for some unique frames (shown in the yellow dotted box). In contrast, the static branch consults representative exemplars from the *Volleyball Spiking* category and consistently predicts high scores for these unique frames. Figure 4 (b) presents a video



(a) Detection scores from dynamic branch and static branch.

(b) Attention weights in dynamic branch.

Figure 4. Qualitative analysis of the proposed Colar method.

segment containing multiple action instances, exhibits the similarity between the current frame (the last one) and its historical frames. The similarity weights clearly highlight historical action frames and suppress background frames, which contributes to aggregating temporal features.

5. Conclusion

This paper proposes Colar, based on the exemplar-consultation mechanism, to conduct category-level modeling for each frame and capture long-term dependencies within a video segment. Colar compares a frame with exemplar frames, aggregates exemplar features, and carries out online action detection. In the dynamic branch, Colar regards historical frames as exemplars and models long-term dependency with a lightweight network structure. In the static branch, Colar employs representative exemplars of each category and captures the category particularity. The prominent efficacy of Colar would inspire future works to pay attention to category-level modeling. In addition, as Colar has made a good trade-off between effectiveness and efficiency, it is a promising direction to conduct online action detection directly from streaming video data, which can benefit practical usage.

Limitations. Because Colar is only verified on the benchmark datasets, it may observe performance drop in practical scene due to new challenges, *e.g.* long-tail distribution, open-set action categories. Besides, the unintended usage of Colar for surveillance may violate individual privacy.

Acknowledgments. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province (No.2019B010110001) and the National Natural Science Foundation of China under Grant U21B2048, 62036011, and the Open Research Projects of Zhejiang Lab (No.2019KD0AD01/010).

References

[1] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 6

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 4

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2, 6

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009. 1

[5] Peihao Chen, Chuang Gan, Guangyao Shen, Wenbing Huang, Runhao Zeng, and Mingkui Tan. Relation attention for temporal action localization. *IEEE TMM*, 2019. 2

[6] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *CVPR*, pages 10337–10346, 2020. 3

[7] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020. 6

[8] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *ECCV*, pages 269–284. Springer, 2016. 1, 5, 6, 7

[9] Roeland De Geest and Tinne Tuytelaars. Modeling temporal structure with lstm for online action detection. In *WACV*, pages 1549–1557. IEEE, 2018. 2, 5, 6

[10] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015. 5, 6

[11] Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. Learning to discriminate information for online action detection. In *CVPR*, pages 809–818, 2020. 2, 5, 6, 7

[12] Xiaoxu Feng, Junwei Han, Xiwen Yao, and Gong Cheng. Progressive contextual instance refinement for weakly supervised object detection in remote sensing images. *IEEE TGRS*, 58(11):8002–8012, 2020. 2

[13] Xiaoxu Feng, Junwei Han, Xiwen Yao, and Gong Cheng. Tcanet: Triple context-aware network for weakly supervised object detection in remote sensing images. *IEEE TGRS*, 2020. 2

[14] Xiaoxu Feng, Xiwen Yao, Gong Cheng, Jungong Han, and Junwei Han. Saenet: Self-supervised adversarial and equivariant network for weakly supervised object detection in remote sensing images. *IEEE TGRS*, 2021. 2

[15] Guangyu Guo, Junwei Han, Fang Wan, and Dingwen Zhang. Strengthen learning tolerance for weakly supervised object localization. In *CVPR*, pages 7403–7412, 2021. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[18] Peiliang Huang, Junwei Han, Nian Liu, Jun Ren, and Dingwen Zhang. Scribble-supervised video object segmentation. *IEEE/CAA Journal of Automatica Sinica*, 9(2):339–353, 2021. 3

[19] Peiliang Huang, Junwei Han, Dingwen Zhang, and Mingliang Xu. Clrnet: Component-level refinement network for deep face parsing. *IEEE TNNLS*, 2021. 2

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456. PMLR, 2015. 6

[21] Y.-G. Jiang, J. Liu, A. Roshan 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. 5, 6

[22] Zhenheng Yang Jiyang Gao and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In *BMVC*, pages 92.1–92.11, 2017. 2, 5, 6

[23] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE TBD*, 7(3):535–547, 2019. 4

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[25] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, pages 6479–6488, 2020. 3

[26] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *CVPR*, pages 3320–3329, 2021. 2

[27] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3889–3898, 2019. 2

[28] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *ACM MM*, pages 988–996, 2017. 2

[29] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip HS Torr. Multi-shot temporal event localization: a benchmark. In *CVPR*, pages 12596–12606, 2021. 2

[30] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *arXiv preprint arXiv:2106.10271*, 2021. 2

[31] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *ECCV*, pages 661–679. Springer, 2020. 3

[32] Ala Mhalla, Thierry Chateau, Sami Gazzah, and Najoua Essoukri Ben Amara. An embedded computer-vision system for multi-object detection in traffic surveillance. *IEEE T-ITS*, 20(11):4006–4018, 2018. 1

[33] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9226–9235, 2019. 3

[34] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for

learning driver behavior and causal reasoning. In *CVPR*, pages 7699–7707, 2018. 5, 6, 7

[35] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, pages 5734–5743, 2017. 5

[36] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016. 1, 2

[37] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *NeurIPS*, 2014. 5

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014. 5

[39] Chenhao Wang, Hongxiang Cai, Yuxin Zou, and Yichao Xiong. Rgb stream is enough for temporal action detection. *arXiv preprint arXiv:2107.04362*, 2021. 1, 2

[40] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016. 2, 6

[41] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 5, 6, 7

[42] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. In *ICCV*, 2021. 1, 2, 3, 4, 5, 6, 7

[43] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016. 6

[44] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, pages 5783–5792, 2017. 2, 3

[45] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S Davis, and David J Crandall. Temporal recurrent networks for online action detection. In *ICCV*, pages 5532–5541, 2019. 1, 2, 5, 6, 7

[46] Le Yang, Junwei Han, Tao Zhao, Tianwei Lin, Dingwen Zhang, and Jianxin Chen. Background-click supervision for temporal action localization. *IEEE TPAMI*, 2021. 3

[47] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE TIP*, 29:8535–8548, 2020. 2

[48] Xiwen Yao, Xiaoxu Feng, Junwei Han, Gong Cheng, and Lei Guo. Automatic weakly supervised object detection from high spatial resolution remote sensing images via dynamic curriculum learning. *IEEE TGRS*, 59(1):675–685, 2020. 2

[49] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *IJCV*, 126(2-4):375–389, 2018. 5

[50] Yuan Yuan, Xiaodan Liang, Xiaolong Wang, Dit-Yan Yeung, and Abhinav Gupta. Temporal dynamic graph lstm for action-driven video object detection. In *ICCV*, pages 1801–1810, 2017. 2

[51] Yuan Yuan, Yueming Lyu, Xi Shen, Ivor W Tsang, and Dit-Yan Yeung. Marginalized average attentional network for weakly-supervised learning. In *ICLR*, 2019. 3

[52] Runhao Zeng, Chuang Gan, Peihao Chen, Wenbing Huang, Qingyao Wu, and Mingkui Tan. Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization. *IEEE TIP*, 28(12):5797–5808, 2019. 3

[53] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, pages 7094–7103, 2019. 2

[54] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional module for temporal action localization in videos. *IEEE TPAMI*, 2021. 2

[55] Dingwen Zhang, Junwei Han, Gong Cheng, and Ming-Hsuan Yang. Weakly supervised object localization and detection: a survey. *IEEE TPAMI*, 2021. 2

[56] Dingwen Zhang, Wenyuan Zeng, Guangyu Guo, Chaowei Fang, Lechao Cheng, and Junwei Han. Weakly supervised semantic segmentation via alternative self-dual teaching. *arXiv preprint arXiv:2112.09459*, 2021. 2

[57] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, pages 4320–4328, 2018. 5

[58] Tao Zhao, Junwei Han, Le Yang, Binglu Wang, and Dingwen Zhang. Soda: Weakly supervised temporal action localization based on astute background response and self-distillation learning. *IJCV*, 129(8):2474–2498, 2021. 3

[59] Linchao Zhu, Hehe Fan, Yawei Luo, Mingliang Xu, and Yi Yang. Few-shot common-object reasoning using common-centric localization network. *IEEE TIP*, 30:4253–4262, 2021. 2

[60] Linchao Zhu, Hehe Fan, Yawei Luo, Mingliang Xu, and Yi Yang. Temporal cross-layer correlation mining for action recognition. *IEEE TMM*, 2021. 2

[61] Linchao Zhu and Yi Yang. Label independent memory for semi-supervised few-shot video classification. *IEEE TPAMI*, 44(1):273–285, 2020. 2

[62] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. Enriching local and global contexts for temporal action localization. In *ICCV*, pages 13516–13525, 2021. 1, 2, 3