Deep Learning-based Action Detection in Untrimmed Videos: A Survey

Elahe Vahdani and Yingli Tian*, *Fellow, IEEE*

Abstract—Understanding human behavior and activity facilitates advancement of numerous real-world applications, and is critical for video analysis. Despite the progress of action recognition algorithms in trimmed videos, the majority of real-world videos are lengthy and untrimmed with sparse segments of interest. The task of temporal activity detection in untrimmed videos aims to localize the temporal boundary of actions and classify the action categories. Temporal activity detection task has been investigated in full and limited supervision settings depending on the availability of action annotations. This paper provides an extensive overview of deep learning-based algorithms to tackle temporal action detection in untrimmed videos with different supervision levels including fully-supervised, weakly-supervised, unsupervised, self-supervised, and semi-supervised. In addition, this paper reviews advances in spatio-temporal action detection where actions are localized in both temporal and spatial dimensions. Action detection in online setting is also reviewed where the goal is to detect actions in each frame without considering any future context in a live video stream. Moreover, the commonly used action detection benchmark datasets and evaluation metrics are described, and the performance of the state-of-the-art methods are compared. Finally, real-world applications of temporal action detection in untrimmed videos and a set of future directions are discussed.

Index Terms—Action Understanding, Temporal Action Detection, Untrimmed Videos, Deep Learning, Full and Limited Supervision.

1 Introduction

This paper provides a comprehensive overview of automatic action detection in videos. Temporal action detection aims to detect the start and end of action instances in long untrimmed videos and predict the action categories. Spatio-temporal action detection further localizes actions in both temporal and spatial domains. Online action detection requires detecting actions in each frame without considering the future context. Action detection is crucial for many video analysis applications such as sport analysis, autonomous driving, anomaly detection in surveillance, understanding instructional videos, etc. Learning with limited supervision is a scheme where annotations of actions are unavailable or only partially available during the training phase. Because annotating long untrimmed videos is very time-consuming, designing action detection methods with limited supervision has been very popular. This survey reviews action detection task in temporal and spatio-temporal domains, offline and online setting, and with full and limited supervision.

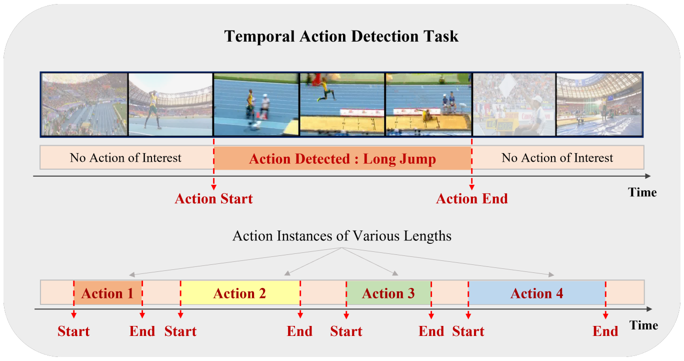

Fig. 1. Temporal action detection aims to localize action instances in time and recognize their categories. The first row demonstrates an example of action “long jump” detected in an untrimmed video from THUMOS14 dataset [1]. The second row is an example of an untrimmed video including several action instances of interest with various lengths.

1.1 Motivation

Social networks and digital cameras have led to substantial video and media content produced by individuals each day. Hence, video understanding continues to be one of the essential research subjects in computer vision. While deep learning has accomplished remarkable performance in many computer vision tasks, video understanding is still far from ideal. Action understanding, as a vital element of video analysis, facilitates the advancement of numerous real-world applications. Collaborative robots need to recognize how the human partner completes the job to cope with the variations in the task [2]. Sport analysis systems must comprehend game actions to report commentaries of live activities [3]. Autonomous driving cars demand an understanding of operations performed by surrounding cars and pedestrians [4].

In this paper, we define *trimmed videos* as pre-segmented video clips that each contains only one action instance. In other words, the *context* of the action, i.e., moments before

- *E. Vahdani is with the Department of Computer Science, The Graduate Center, The City University of New York, NY, 10016. E-mail: evahdani@gradcenter.cuny.edu*
- *Y. Tian is with the Department of Electrical Engineering, The City College, and the Department of Computer Science, the Graduate Center, the City University of New York, NY, 10031. E-mail:ytian@ccny.cuny.edu*
**Corresponding author*

This material is based upon work supported by the National Science Foundation under award number IIS-2041307.

or after the action are not included in the video. Therefore, action detection in trimmed videos only need to classify the action categories without the need to detect starting and ending timestamps. Recognizing actions in trimmed videos has many applications in video surveillance, robotics, medical diagnosis [5], and has achieved excellent performance in recent years [6], [7], [8].

However, the majority of videos *in the wild*, i.e., recorded in unconstrained environments, are naturally untrimmed. *Untrimmed videos* are lengthy unsegmented videos that may include several action instances, the moments before or after each action, and the transition from one action to another. The action instances in one video can belong to several action classes and have different duration.

Temporal activity detection in untrimmed videos aims to localize the action instances in time and recognize their categories. This task is considerably more complicated than action recognition which merely seeks to classify the categories of trimmed video clips. Fig. 1 shows an example of temporal activity detection in an untrimmed video recorded in a stadium. The first row demonstrates the detection of action ”long jump” in temporal domain where the start and end time of the action are localized. The goal is to only detect *the actions of interest*, i.e., actions that belong to a predefined set of action classes. The temporal intervals of other activities that do not belong to this set of actions are called *temporal background*. For example, the segments right before or right after action ”long jump” may belong to other diverse activities such as crowd cheering in the stadium. In some cases, the frames right before or right after an action are visually very similar to the start or end of the action which makes the localization of action intervals very challenging. Another challenge (as shown in the second row of Fig. 1) is that action instances may occur at any time of the video and have various duration, lasting from less than a second to several minutes [9].

Temporal action detection mainly targets activities of high-level semantics and videos with a sparse set of actions (e.g., actions only cover 30% of the frames in [10]). However, in some cases, the goal is to predict action labels at every frame of the video. In such cases, the task is referred to as *temporal action segmentation* which targets the fine-grained actions and videos with dense occurrence of actions (93% of the frames in [11]). One can convert between a given segmentation and a set of detected instances in the temporal domain by simply adding or removing temporal background segments [12]. Temporal action detection similar to object detection belongs to the family of detection problems. Both of these problems aim to localize the instances of interest, i.e., action intervals in temporal domain versus object bounding boxes in spatial domain, Fig. 2 (a and c). When targeting fine-grained actions, temporal action detection (segmentation) is similar to semantic segmentation as both aim to classify every single instance, i.e., frames in temporal domain versus pixels in spatial domain, Fig. 2 (b and d). As a result, many techniques for temporal action detection and segmentation are inspired by the advancements in object detection and semantic segmentation [13], [14], [15].

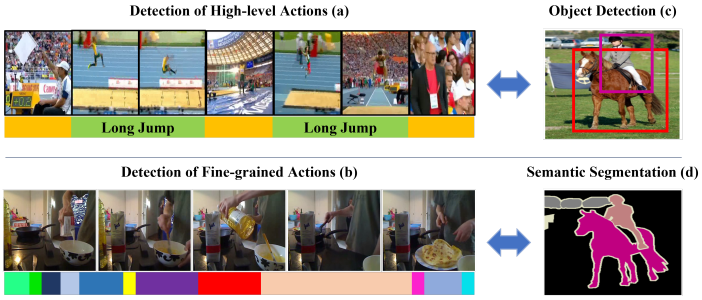
Detection of High-level Actions (a)
Object Detection (c)
Long Jump
Long Jump
Detection of Fine-grained Actions (b)
Semantic Segmentation (d)

Fig. 2. Task Relations: (a) Temporal detection of action “Long Jump” on THUMOS14 [1]. (b) Temporal detection (segmentation) of fine-grained actions shown by different colors in a “making pancakes” video on Breakfast [11]. (c) and (d) Results from [16] on PASCAL [17].

Spatio-temporal action detection further localizes actions in both temporal and spatial domains. In the “long jump” example (Fig. 1), this task requires predicting the temporal boundaries of the action, detecting the bounding box of the person performing the action in each frame, and recognizing the action category. Action detection can be studied in offline or online settings. In offline setting, the goal is to localize action instances and predict their categories in recorded videos using the full content of the video. *Online action detection* requires the detection of actions in each frame upon arrival without considering the future context in a live video stream.

Action detection has drawn much attention in recent years and has broad applications in video analysis tasks. As surveillance cameras are increasingly deployed in many places, the demand for anomaly detection has also surged. Anomalous events such as robbery or accidents occur less frequently compared with normal activities and it can be very time-consuming to detect such events by humans. Therefore, automatic detection of suspicious events has a great advantage. By growing popularity of social media many people follow online tutorials and instructional videos to learn how to perform a task such as “changing the car tire” properly for the first time. The instructional videos are usually untrimmed and include several steps of the main task, e.g., “jack up the car” and “put on the tire” for changing the tire. Automatic segmentation of these videos to the main action steps can facilitate and optimize the learning process. Another application is in sport video analysis to localize the salient actions and highlights of a game and analyze the strategies of specific teams. Furthermore, action detection has a critical role in self-driving cars to analyze the behavior of pedestrians, cyclists, and other surrounding vehicles to make safe autonomous decisions.

1.2 Taxonomy

To the best of our knowledge, this is the first comprehensive survey describing deep-learning based algorithms for activity detection in untrimmed videos with different supervision levels. Temporal action detection methods with full and limited supervision are discussed in Sections 2.2 and 2.3, respectively. Spatio-temporal action detection and online action detection are briefly reviewed in Sections 2.4 and 2.5, respectively. Section 3 summarizes action detection benchmark datasets, evaluation metrics, and performance comparison among the state-of-the-art methods. Finally, Section 4 discusses the real-world applications of action detection and potential future research directions. A brief introduction of the tasks is provided below.

Temporal Action Detection (TAD): This task aims to detect temporal boundaries and labels of action instances in untrimmed videos. Depending on annotation availability in training set, temporal action detection can be studied in the following settings (see Table 1).

- **Fully-supervised TAD:** Temporal boundaries and labels of action instances are available for training.
- **TAD with point-level supervision:** A single frame within the temporal window of each action instance is annotated for all videos.
- **Weakly-supervised TAD:** Only video-level labels of action instances are available in most cases.
- **Semi-supervised TAD:** The data is split to a small set S_S and a large set S_L. The videos in S_S are fully annotated while the videos in S_L are either not annotated or only annotated with video-level labels.
- **Unsupervised TAD:** No annotations of the action instances are available.
- **Self-supervised TAD:** A pretext task is defined to extract features from large databases in an unsupervised setting by leveraging its structure. The pre-trained models are then used to improve the performance for temporal action detection (downstream task) which can be supervised, unsupervised, or semi-supervised.
- **TAD with limited supervision:** This setting includes weakly-supervised, unsupervised, self-supervised, and semi-supervised settings.

TABLE 1
Main categories of temporal action detection task with different supervision levels in training set. "✓" indicates "available"; "✗" is for "unavailable", and $*$ is "partially available".

Supervision Level	Action Temporal Boundaries	Action Labels
Fully-supervised	✓	✓
Weakly-supervised	✗	✓
Unsupervised	✗	✗
Semi-supervised	$*$	$*$
Self-supervised	✓$*$✗	✓$*$✗

Spatio-temporal Action Detection: This task aims to localize action instances in both spatial and temporal domains, and recognize the action labels.

Online Temporal Action Detection: Given an incoming stream of video frames, this task aims to classify actions at each frame without seeing the future, by processing the data up to the current time.

2 Action Detection Methods

We begin this section by introducing important technical terms in Section 2.1. Temporal action detection methods with full and limited supervision are described in Sections 2.2 and 2.3, respectively. Spatio-temporal action detection is reviewed in Section 2.4, and online action detection is briefly discussed in Section 2.5.

2.1 Term Definition

To facilitate reading subsequent sections, we define some of the common terms here.

Definition 1. Temporal action detection. This task aims to find the temporal boundaries and categories of actions in untrimmed videos. For a given video, annotation Ψ_g includes a set of action instances as the following

$$\Psi_g = \{\varphi_n = (t_{s,n}, t_{e,n}, l_n)\}_{n=1}^{N}, \tag{1}$$

where N is the number of action instances, and φ_n is the n-th action instance. The start time, end time, and label of φ_n are denoted by $t_{s,n}$, $t_{e,n}$, and l_n, respectively. Label l_n belongs to set $\{1, \cdots, C\}$, where C is the number of action categories in the whole dataset. Annotation Ψ_g can be fully, partially, or not available for the videos in training set.

Definition 2. Temporal proposals. Temporal proposal P_n is a temporal region of an input video that is likely to contain an action of interest. P_n is identified with a starting time $t_{s,n}$, an ending time $t_{e,n}$, a predicted action label l_n, and a confidence score c_n. Confidence score is the predicted probability that the interval contains an action. Proposal P_n can be formulated as $P_n = (t_{s,n}, t_{e,n}, l_n, c_n)$.

Definition 3. Temporal IoU (tIoU). This is the ratio of temporal intersection over union between two temporal intervals. It is often measured between a predicted proposal (interval I_p) and its closest ground-truth action instance (interval I_g), formulated as $tIoU(I_p, I_g) = \frac{I_p \cap I_g}{I_p \cup I_g}$.

Definition 4. Temporal proposal labeling. For a given action class c, the predicted proposals with label c are matched to ground truth actions with label c using tIoU. Each proposal is matched with the annotated action with maximum tIoU. Proposals with tIoU above a given threshold are declared as *true positives*. To penalize multiple detections of the same action, at most one proposal (with the highest confidence score) is assigned to each annotated action instance and the remaining proposals are declared as *false positives*. Ground truth action instances with no matching proposals are declared as *false negatives*.

Definition 5. Precision and recall for proposal generation. Precision is the ratio of true positive proposals to the total number of predicted proposals. Precision must be high to avoid producing exhaustively many irrelevant proposals. Recall is the ratio of true positive proposals to the total number of ground-truth action instances. Recall must be high to avoid missing ground-truth instances.

Definition 6. Actionness score. Actionness score a_t is the occurrence probability of any action of interest at time t .

Definition 7. Startness and endness scores. Startness score (endness score) at any time is the probability that any action of interest starts (ends) at that time.

Definition 8. Action completeness score. The maximum tIoU between a candidate proposal and ground truth action instances is action completeness score of that proposal.

Definition 9. Action classification. Temporal proposals are fed to action classifiers to produce a probability distribution over all action classes where the maximum probability is the action classification score. Some methods use their own action classifier while others utilize classifiers from earlier work such as SCNN [18], UNet [19], and Cuhk [20] for a fair comparison. More details are provided in the Appendix.

Definition 10. Video Feature Encoding. Untrimmed videos are often lengthy and can be as long as several minutes. Thus, it is difficult to directly input the entire video to a visual encoder for feature extraction due to the limits of computational resources. A common strategy for video representation is to partition the video into equally sized temporal intervals called *snippets*, and then apply a pre-trained visual encoder over each snippet. Then, each video can be represented with a sequence of visual features that are further processed for action detection. Common visual encoders are I3D [21], Two-stream [22], C3D [23], TSN [24], ResNet50 [25], R(2+1)D [26], and P3D [27]. The features extracted from pre-trained visual encoders are typically trained for trimmed action classification tasks and are not necessarily suitable for temporal localization. Recently, researchers have proposed *pretraining for localization* to learn video representations that are more transferable to action localization [28], [29], [30], [31], [32]. More details are provided in the Appendix 1.1.1.

2.2 Temporal Action Detection with Full Supervision

In fully-supervised action detection, the temporal boundaries and labels of action instances are provided for each video of training set (annotation Ψ_g in Eq. (1) is fully provided). During inference, the goal is to detect the temporal boundaries of action instances and predict their labels. A main step in action detection is to generate temporal proposals (def 2) with high precision and recall (def 5). Fully-supervised temporal proposal generation methods can be categorized to anchor-based (Section 2.2.1) and anchor-free (Section 2.2.1). Some methods combine the advantages of anchor-free and anchor-based proposal generation methods (Section 2.2.3). Section 2.2.4 reviews common loss functions that are used during training for proposal evaluation. Section 2.2.5 discusses modeling long-range dependencies of video segments in untrimmed videos to improve action localization.

2.2.1 Anchor-based Proposal Generation and Evaluation

Anchor-based methods, also known as top-down methods, generate temporal proposals by assigning dense and multi-scale intervals with pre-defined lengths to uniformly distributed temporal locations in the input video. Formally, given a video with T frames, $\frac{T}{\sigma}$ temporal positions, known as *anchors*, are uniformly sampled from every σ frames. The proposal lengths must have a wide range to align with action instances of various lengths in untrimmed videos [9]. Therefore, multi-scale temporal windows are centered around each anchor as initial temporal proposals (e.g., windows with 2^k frames for $4 \leq k \leq 9$ in [18]). To evaluate the quality of temporal proposals for action classification, action completeness, boundary regression (Section 2.2.4), fixed-size features must be extracted from multi-scale proposals. Earlier methods uniformly sample fixed number of frames from proposals for feature extraction [18] or concatenate the features of fixed-size intervals [33], [34]. Such strategies do not extract rich features to represent the temporal and semantic structure of the proposals. The following feature extraction strategies were later proposed to improve the quality of features: 1) RoI Pooling [35], 2) Multi-tower Network [13], 3) Temporal Feature Pyramid Network (TFPN) [36], and 4) U-shaped Temporal Feature Pyramid Network (UTFPN) [37]. The details of these methods are explained in Appendix.

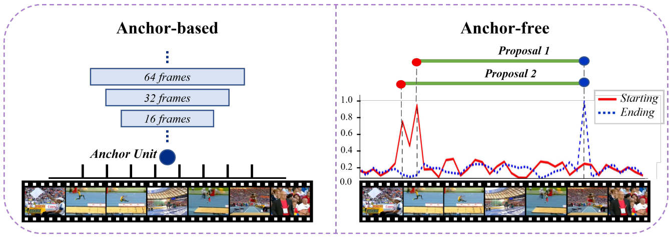

Fig. 3. Anchor-free vs anchor-based proposal generation.

2.2.2 Anchor-free Proposal Generation and Evaluation

Anchor-free methods employ a bottom-up grouping strategy for proposal generation, often based on predicting actionness, startness, and endness scores (def 6, 7) at each temporal position of the video. They are capable to generate proposals with precise boundaries and flexible duration as the proposal lengths are not predefined. Anchor-free temporal proposal generation was first introduced in TAG [38] to group continuous temporal regions with high actionness scores (def 6) using a classic watershed algorithm [39] on complemented actionness scores. The proposal features were extracted with temporal pooling which is too simple to represent the temporal context.

BSN [40] proposed to predict actionness, startness and endness signals (def 6, 7), and generate flexible proposals by matching the temporal positions that are high in startness and endness scores. The proposal features were constructed by concatenation of a fixed number of points, sampled from probability signals by linear interpolation. BSN Proposals are generated and evaluated separately which is inefficient and ignores the global context of the video. To mitigate this problem, BMN [41] captures the global context of the video by aggregating the features of all proposals and simultaneously evaluating them all. However, they ignore the global information for boundary prediction leading to inaccurate localization for actions with blurred boundaries. This issue was addressed in DBG [42] by employing global information to predict boundary probabilities. AFSD [43] predicts the distance to the temporal boundaries for each temporal location in the feature sequence, and proposed a novel boundary refinement strategy for precise temporal localization. BC-GNN [44] proposed to model the relations between the boundary and content of proposals by constructing a graph where boundaries and content of proposals are taken as nodes and edges, and their features are updated through graph operations. The updated edges and nodes are used to predict confidence scores of proposals.

2.2.3 Anchor-based and Anchor-free Combination

Anchor-based methods consider segments of various lengths as initial proposals but since the segment sizes are designed beforehand, they cannot accurately predict the temporal boundary of actions with various lengths. Anchor-free methods generate flexible proposals but usually exploit local context to extract action boundary information. Therefore, they are sensitive to noise, likely to produce incomplete proposals, and fail to yield robust detection results. Several methods proposed to balance the advantages and disadvantages between anchor-based and anchor-free approaches for proposal generation [45], [37], [46], [47]. Some methods such as MGG [37], PBRNet [46], and RapNet [47] proposed to generate coarse segment proposals with UTFPN (anchor-based), and simultaneously predict fine-level actionness scores (anchor-free) at each temporal position. Then, actionness information is used to adjust the segment boundary of proposals.

2.2.4 Common Loss Functions for Proposal Prediction

During the training, the predicted proposals are supervised with the following common loss functions.

Definition 11. Actionness loss. This is a binary cross-entropy loss that classifies the temporal proposals as action or background. Given N proposals, this loss is defined as:

$$L_{\text{act}} = -\frac{1}{N}\sum_{i=1}^{N} b_i \log(a_i) + (1-b_i)\log(1-a_i), \tag{2}$$

where a_i is the predicted actionness score (def 6), and $b_i \in \{0,1\}$ is ground-truth label for the i-th proposal. If the proposal is positive (def 4), then $b_i = 1$. Otherwise, $b_i = 0$.

Definition 12. Action completeness loss. Given N proposals, the completeness loss is defined as:

$$L_{\text{com}} = \frac{1}{N_{\text{pos}}}\sum_{i=1}^{N} d(c_i, g_i)\cdot[l_i > 0], \tag{3}$$

where c_i is the predicted action completeness score (def 8) for i-th proposal, and g_i is the ground-truth action completeness score. d is a distance metric which is often L_2 or smooth L_1 loss. l_i is the label of the i-th proposal and condition $[l_i > 0]$ implies that action completeness is only considered for positive proposals (def. 4). N_{pos} is the number of positive proposals during each mini-batch.

Definition 13. Action classification loss. This is the cross-entropy loss and the probability distribution is over all action classes and temporal background:

$$L_{\text{cls}} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_i^{l_i}), \tag{4}$$

where $l_i \in \{0,1,\cdots,C\}$ is the label of i-th proposal, and $p_i^{l_i}$ is the probability of class l_i.

Definition 14. Action regression loss. To adjust the temporal boundary of proposals, the start and end offset of proposals are supervised by regression loss:

$$L_{\text{reg}} = \frac{1}{N_{\text{pos}}}\sum_{i=1}^{N}|(o_{s,i} - o^{\star}_{s,i}) + (o_{e,i} - o^{\star}_{e,i})|\cdot[l_i > 0], \tag{5}$$

where term $o_{s,i}$ is the difference between the start coordinate of i-th proposal and the start coordinate of the closest ground truth action instance. The term $o^{\star}_{s,i}$ is the predicted offset. Similarly, $o_{e,i}$ and $o^{\star}_{e,i}$ are the ground-truth and predicted offset for end coordinate of the i-th proposal. The condition $[l_i > 0]$ implies that boundary adjustment is only considered for positive proposals (def. 4).

2.2.5 Modeling Long-range Dependencies

As mentioned earlier, untrimmed videos are often lengthy and must be partitioned into shorter clips for feature extraction. Processing these shorter clips independently can lead to loss of temporal or semantic dependencies between video segments. The following tools are employed to capture long-range dependencies in videos: 1) Recurrent Neural Networks (RNNs), 2) Graph Convolution Networks, 3) Transformers. We provide an overview of these methods here and more details are provided in Appendix.

Recurrent Neural Networks: In RNN-based methods such as [48], [49], [50], [51], [52], [53], [54], [55], the hidden state encodes the information from previous time steps which is useful to capture temporal dependencies. However, RNNs are not capable to encode long videos as the hidden vector gets saturated after some time steps.

Graph Convolution Networks: Graph models [56], [57], [44], [58], [59] are proposed to model the inter and intra dependencies of the proposals, exploit the relations between the boundary and action content of the proposals, and capture the relations between cross-scale snippets. In some methods [56], temporal dependencies are employed only for proposal refinement (not proposal generation).

Transformers: Transformer and attention mechanism are powerful tools to capture long-range dependencies in untrimmed videos [60], [61], [62]. Some methods proposed to directly generate action proposals by mapping a set of learnable embeddings (the latent representations of the action queries) to action instances [60], [61], [63]. Despite having advantage of modeling long-range dependencies, transformers have high parametric complexity and can lead to over-fitting on small datasets.

2.3 Temporal Action Detection with Limited Supervision

Fully-supervised action detection requires the annotation of temporal boundaries and action labels for all action instances in training videos, which is time-consuming and costly. To eliminate the need for exhaustive annotations in the training phase, researchers have explored the design of efficient models that require limited ground-truth annotations. We discuss weakly-supervised methods in Section 2.3.1, and other learning methods with limited supervision (unsupervised, semi-supervised, self-supervised, and special supervision) in Section 2.3.2.

2.3.1 Weakly-supervised Action Detection

Weakly-supervised learning scheme requires coarse-grained or noisy labels during the training phase. Following the work of [64], weakly-supervised action detection in common settings requires only the video-level labels of actions during training while the temporal boundaries of action instances are not needed. During testing both labels and temporal boundaries of actions are predicted. There are also other weak signals utilized for action detection such as frequency of action labels [65], and total number of events in each video [66]. A common strategy in weakly-supervised action detection is to use attention mechanism to focus on discriminative snippets and combine salient snippet-level features into a video-level feature. The attention scores are used to localize the action regions and eliminate irrelevant background frames. Attention signals are predicted with *class-specific attention* and *class-agnostic attention* methods which are discussed in this section.

2.3.1.1 Term Definition

To facilitate reading this section, we provide the definition of frequently used terminologies.

Definition 15. Temporal class activation maps (T-CAM). For a given video, T-CAM is a matrix denoted by A which represent the possibility of activities at each temporal position. Matrix A has n_c rows which is the total number of action classes, and T columns which is the number of temporal positions in the video. Value of cell $A[c,t]$ is the activation of class c at temporal position t. A is formulated as $A = WX \oplus b$ where $X \in \mathbb{R}^{d\times T}$ is a video-level feature matrix, and d is the feature dimension. Also, $W \in \mathbb{R}^{n_c \times d}$ and $b \in \mathbb{R}^{n_c}$, are learnable parameters and $\oplus$ is the addition with broadcasting operator.

Definition 16. Class-specific attention scores. In a given video, class-specific attention score is the occurrence probability of action class c at temporal position t, denoted by $a[c,t]$. Formally, $a[c,t]$ is computed by normalizing the activation of class c over temporal dimension:

$$a[c,t] = \frac{\exp(A[c,t])}{\sum_{t=1}^{T} \exp(A[c,t])}, \tag{6}$$

where A is the T-CAM (def. 15), and T is the number of temporal positions. Therefore, row a_c is the probability distribution of occurrence of class c over video length.

Definition 17. Class-agnostic attention score. In a given video, class-agnostic attention score, denoted by λ_t, is the occurrence probability of any action of interest at temporal position t, regardless of the action class. The attention vector for all temporal positions of the video is denoted by λ.

Definition 18. Attention-based aggregated features. The video-level foreground and background features are generated using temporal pooling of embedded features weighted by attention scores. Class-specific features are defined based on class-specific attention scores a_c (def. 16) for each class c while class-agnostic features are defined based on class-agnostic attention vector λ (def. 17). Aggregated foreground feature is most influenced by feature vectors with high attention that represent actions while background feature is impacted by features with low attention. T is the video length and X is the video feature matrix. These features are formulated as the following:

	Foreground:	Background:
Class-specific:	$f_c = Xa_c$	$b_c = \frac{1}{T-1}X(\mathbb{1} - a_c)$,
Class-agnostic:	$f = \frac{1}{T}X\lambda$	$b = \frac{1}{T}X(\mathbb{1} - \lambda)$.

2.3.1.2 Class-specific Attention for Action Localization

Class-specific attention module computes the attention weight $a[c,t]$ (def. 16) for all action classes c and all temporal positions t in each video. The attention scores attend to the portions of the video where an activity of a certain category occurs. Here, we review some of the strategies to learn class-specific attention with weak supervision.

Class-specific attention learning with MIL: In general scheme of MIL (multi-instance learning), training instances are arranged in sets, called bags, and a label is provided for the entire bag [67]. In the context of action detection, each video is treated as a bag of action instances. To compute the loss for each bag (video), each video should be represented using a single confidence score per category. The confidence score for each category is computed as the average of top k activation scores over the temporal dimension for that category. In a given video, suppose set $\{t_1^c, t_2^c, \cdots, t_k^c\}$ are k temporal positions with highest activation scores for class c. The video-level class-wise confidence score s^c for class c is defined as $s^c = \frac{1}{k}\sum_{l=1}^{k} A[c, t_l^c]$ where $A[c, t_l^c]$ is the activation (def 15) of class c at temporal position t_l^c. Probability distribution p^c is computed by applying softmax function on s^c scores over class dimension. MIL loss is a cross-entropy loss applied over all videos and all action classes. For video i and action class c, p_i^c is the class-wise probability score, and y_i^c is a normalized ground-truth binary label. The number of action classes and videos are denoted by n_c and n. MIL loss supervises class-wise probability scores and learns T-CAM (def. 15).

$$L_{MIL} = \frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{n_c} -y_i^c \log(p_i^c), \quad p^c = \frac{\exp\left(s^c\right)}{\sum_{c=1}^{n_c}\exp\left(s^c\right)}. \tag{7}$$

Class-specific attention learning with CASL: The CASL (co-activity similarity loss) was initially introduced in W-TALC [68] and is based on ranking hinge loss. The main idea is that for a pair of videos (with indices m and n) including the same action class c, the foreground features in both videos (f_c^m, f_c^n) should be more similar than the foreground feature in one video and the background feature in the other video (b_c^m or b_c^n) (def. 18). d is a metric (e.g., cosine similarity) and δ is a margin parameter. The average of L_c^{mn} is computed over all video pairs that include action class c. This loss trains class-specific attention scores a_c (f_c and b_c are defined based on a_c (def. 18)).

$$L_c^{mn} = \frac{1}{2}\{\max\left(0, d(f_c^m, f_c^n) - d(f_c^m, b_c^n) + \delta\right) \\ + \max\left(0, d(f_c^m, f_c^n) - d(b_c^m, f_c^n) + \delta\right)\}. \tag{8}$$

Class-specific attention learning with center loss: Center loss [69] learns the class-specific centers and penalizes the distance between the features and their class centers. 3C-Net [65] employed center loss to enhance the feature discriminability and reduce the intra-class variations. For each video i and each action class c, center loss computes the distance (L2 norm) between class-specific foreground feature f_c^i (def. 18) and cluster center feature z_c (updated during training), $\mathcal{L}_{center} = \frac{1}{N}\sum_i \sum_{c:y^i(c)=1} \left\|f_c^i - z_c\right\|_2^2$. Here, N is the number of videos, and condition $y^i(c) = 1$ checks if action class c occurs in video i.

2.3.1.3 Class-agnostic Attention for Action Localization

Class-agnostic attention module computes attention λ (def. 17) directly from raw data, by applying fully connected and ReLU layers over video features, followed by a sigmoid function to scale attention weights to $[0, 1]$.

Class-agnostic attention learning with cross-entropy: The video-level class-agnostic foreground and background features f and b (def. 18) are fed to a classification module, and supervised with a cross entropy loss, where w_c s are the weights of the classification module, C is the number of action classes, and y is the label of action that happens in the video. Label 0 represents the background class.

$$p_{fg}[c] = \frac{\exp\left(w_c \cdot f\right)}{\sum_{i=0}^{C} \exp\left(w_i \cdot f\right)}, \quad \mathcal{L}_{fg} = -\log(p_{fg}[y]). \tag{9}$$

Similarly, $\mathcal{L}_{bg}$ is defined for p_{bg} which is a softmax applied over multiplication of background feature b and the classification module. This loss trains attention vector λ through class-agnostic features f and b (def 18) [70].

Class-agnostic attention learning with clustering loss: Nguyen *et al* [71] separated foreground and background using clustering loss by penalizing the discriminative capacity of background features. Attention λ is trained by separating foreground and background features f and b (def. 18). Here, u, v are trainable parameters.

$$z_f = \frac{\exp(uf)}{\exp(uf) + \exp(vf)}\ ,\ z_b = \frac{\exp(vb)}{\exp(ub) + \exp(vb)}, \tag{10}$$

$$\mathcal{L}_{cluster} = -\log z_f - \log z_b. \tag{11}$$

Class-agnostic attention learning with prototypes: Prototypical network designed for classification task [72], represents each class as a prototype and matches each instance with a prototype with highest similarity. The semantically-related prototypes are pushed closer than unrelated prototypes. RPN [73] proposed a prototype learning scheme for action localization. For temporal position t and action class c, the similarity score $s_{t,c}$ between feature x_t and prototype p_c is computed and similarity vector s_t consists of $s_{t,c}$ for all classes, $s_{t,c} = -\left\|x_t - p_c\right\|_2^2$. The similarity vector s_t is fused with attention score λ_t into a video-level score $\hat{s}$, $\hat{s} = \sum_{t=1}^{T} \lambda_t s_t$. Score $\hat{s}$ is supervised by a classification loss with respect to the video-level labels to learn attention score λ_t.

Class-agnostic attention learning with CVAE: DGAM [74] separated actions from context frames by imposing different attentions on different features using CVAE [75]. The objective of DGAM is $\max_{\lambda\in[0,1]} \log p(y|X, \lambda) + \log p(X|\lambda)$ where X denotes the features, y is the video-level label, and λ is the attention signal. The first term encourages high discriminative capability of the foreground feature f and punishes any discriminative capability of the background feature b. The second term is approximated by a generative model which forces the feature representation X to be accurately reconstructed from the attention λ using CVAE. By maximizing this conditional probability, the frame-wise attention is optimized by imposing different attentions on different features to separate action from context.

2.3.1.4 Direct Action Proposal Generation

Some methods localize the actions by applying thresholds on attention scores [70], [68], [76], [19]. Thresholding treats the snippets independently, neglect their temporal relations, and is not robust to noises in class activation maps. AutoLoc [77] directly predicted the temporal boundary of action instances (inner boundaries), and obtained the outer boundaries by inflating the inner boundaries. They designed a novel loss to encourage high activations in the inner area and penalize high activations in the outer area because a complete action clip should look different from its neighbors. CleanNet [78] predicted a temporal contrast score by summing up actionness, starting and ending scores (def. 6, 7) for action proposals. The framework is trained by maximizing the average contrast score of the proposals, penalizing fragmented short proposals which promotes completeness and continuity in the proposals.

2.3.1.5 Action Completeness Modeling

Hide-and-seek [76] forced the model to see different parts of actions by randomly masking different video regions. However, random hiding does not always guarantee the discovery of new parts and disrupts the training process. Step-by-step [79] trained a series of classifiers iteratively to find complementary action parts, by erasing the predictions of predecessor classifiers from videos. The major drawback with this approach is the computational expense to train multiple classifiers. Similarly, Zeng *et al.* [80] proposed a strategy that selects the most discriminative action instances in each training iteration and hide them in the next iteration. CMCS [81] enforced multiple branches in parallel to discover complementary action parts where each branch generates a different class activation map (def. 15). They used diversity loss [82] to encourage the branches to produce activations on different action parts.

2.3.2 Action Detection with Different Levels of Supervision

In this section, we review action detection methods with point-level supervision, semi-supervision, self-supervision, omni-supervision, and in unsupervised setting.

Action Detection with Point-level Supervision: Point-level supervision is defined as annotating a single frame within the temporal window of each action instance in the input video [83], [84], [85]. Point-level supervision requires extra annotations compared to weakly supervised methods but still significantly reduces the labeling cost and provides rich information. SF-Net [83] argued that single-frame supervision provides strong signals about the background. They expanded each annotated single frame to its nearby frames to mine pseudo action frames and utilized the unannotated frames to mine pseudo background frames. PTAL [84] performed boundary regression based on keyframe prediction. Lee *et al.* [85] utilized action-background contrast to learn action completeness from dense pseudo-labels. BackTAL [86] introduced the background-click supervision by annotating a random frame from a series of consecutive background frames. They performed a supervised classification on annotated background frames to improve the quality of the class activation sequence (def. 15).

Semi-supervised Action Detection: In a semi-supervised setting, a small number of videos are fully annotated with the temporal boundary of actions and class labels, while many videos are either unlabeled or include only video-level labels. Many of semi-supervised methods employ consistency regularization by applying perturbations on video features and training the model to be robust to the perturbed inputs [87], [88], [89], [90]. Ji *et al.* [87] applied sequential perturbations (time warping and time masking [91]) on video features. The student model takes this perturbed sequence as the input, but the teacher model predicts directly on the original feature sequence. The student model is optimized with a supervised loss applied to labeled videos and a consistency loss for all videos. PM-MT [88] proposed a map warping to generate 2-D supervision for perturbed video features. KFC [90] proposed K-farthest crossover to construct perturbed features. SSTAP [89] introduced two temporal perturbations, i.e., temporal feature shift and temporal feature flip. All semi-supervised methods use 60% of annotated data for a fair comparison except TTC-Loc [92] (using 30% of labels on THUMOS14 and 2% on ActivityNet 1.3).

Self-supervised Action Detection: Self-supervised learning refers to training with pseudo labels where pseudo labels are automatically generated for a pre-defined pretext task without involving any human annotations. Actionbytes [93] adopted a self-supervised iterative approach for training boundary-aware models by decomposing trimmed videos into ActionBytes. SSTAP [89] designed two pretext tasks, i.e., masked feature reconstruction and clip-order prediction, to learn the relation of temporal clues and achieved the state-of-the-art results on ActivityNet-1.3 among semi-supervised methods. Gong *et al.* [94] proposed a self-supervised equivariant transform consistency constraint and attained the state-of-the-art results on ActivityNet-1.2 among weakly-supervised methods.

Unsupervised Action Detection: Unsupervised learning does not need any human-annotated labels during training. Gong *et al.* [95] used only the total count of unique actions that appear in the video set as a supervisory signal. They propose a two-step iterative clustering and localization procedure. The clustering step provides noisy pseudo-labels for the localization step, and the localization step provides temporal co-attention models to improve the clustering performance.

Omni-Supervised Action Detection: Shi *et al.* [96] proposed a multi-level supervision method for action detection. They incorporated state-of-the-art semi-supervised models into a fully-supervised action detection backbone [38]. They designed an unsupervised foreground attention module to recognize relatively complete actions without extra annotation cost. Moreover, they considered weakly-labeled data with only video-level labels and added a classification loss for the additional weakly-labeled data.

2.4 Spatio-temporal Action Detection

Spatio-temporal action detection aims to localize action instances in both space and time and recognize the action labels. In the fully-supervised setting of this task, the temporal boundary of action instances, the spatial bounding box of actions at the frame-level, and action labels are provided during training and must be detected during inference, shown in Fig. 4. For feature extraction, most methods combine a 3D-CNN (e.g., I3D [21]) with a region-based person detector (e.g., Faster R-CNN [97]).

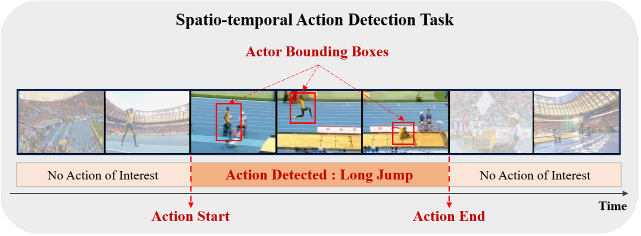

Fig. 4. Spatio-temporal activity detection: action ”long jump” is localized in time and space. Other than temporal interval of the action, bounding box of the person performing the action is detected in each frame.

Frame-level Action Detection: Advancements in object detection inspired frame-level action detection methods to recognize human action classes at the frame level. In the first stage, action proposals are produced by a region proposal algorithm or densely sampled anchors. In the second stage, the proposals are used for action classification and localization refinement. Hundreds of action proposals are extracted per video given low-level cues, such as super-voxels [98], [99] or dense trajectories [100], [101], and then proposals are classified to localize actions. After detecting the action regions in the frames, some methods [102], [103],

[104], [105], [106] use optical flow to capture motion cues. They employ linking algorithms to connect the frame-level bounding boxes into spatio-temporal action tubes. Another group [107], [108], [109] rely on an actionness measure, i.e., a pixel-wise probability of containing any action. They extract action tubes by thresholding the actionness scores [107] or by using a maximum set coverage [109]. The output is a rough localization of the action based on noisy pixel-level maps. The main disadvantage of these methods is that the temporal information is not fully exploited as the detection is performed on each frame independently.

Clip-level Action Detection: Effective temporal modeling is crucial as some actions are only identifiable when the temporal context is available. Kalogeiton *et al.* [110] proposed an action tubelet detector that takes as input a sequence of frames and outputs action categories and regressed tubelets, i.e., sequences of bounding boxes with associated scores. The tubelets are then linked to construct action tubes (sequence of action bounding boxes). Gu et al. [111] further demonstrate the importance of temporal information by using longer clips. They extend 2D region proposals to 3D by replicating them over time, but this approach fails if there is large spatial displacement over time. Moreover, using long cuboids directly as action proposals can generate extra noise for action classification. Yang *et al.* [112] perform action detection at clip level and then link them to build action tubes across the video. They employ a multi-step optimization process to refine the initial proposals progressively. Other methods [6], [113] exploited human proposals coming from pre-trained image detectors and replicated them in time to build spatio-temporal tubes.

Modeling Spatio-temporal Dependencies: To understand human actions, some methods model the relation between actors and contextual information such as other people and other objects. Some methods used the graph-structured networks [114], [115] and attention mechanism [113], [116], [117], [118] to aggregate the contextual information from other people and objects in the video. Wu *et al.* [113] provided long-term supportive information to model temporal dependencies and relate past and present information. Girdhar et al. [116] proposed a transformer-style architecture to aggregate features from the spatiotemporal context around the actors. Ji *et al.* proposed Action Genome [119] to model action-object interaction, by decomposing actions into spatio-temporal scene graphs. Ulutan et al. [117] proposed an attention mechanism to model the surrounding context of actors by combining the actor features and contextual features extracted from the scene. Tomei *et al.* [120] employed self-attention to encode people and object relationships in a graph structure, and use the spatio-temporal distance between proposals. Pan *et al.* [118] proposed a relational reasoning module to capture the relation between the two actors based on their respective relations with the context.

2.5 Online Action Detection

Given a video stream, online action detection aims to detect actions by processing the data up to the current time. Given the observed frames, a temporal modeling is required to aggregate discriminative information from the past and detect current actions. This problem was first introduced by De Geest *et al.* [121] and has been used in applications such as anomaly detection and autonomous driving. We briefly review action detection methods in online setting.

Temporal Modeling with RNNs: Several methods adopted RNNs to model temporal dependencies. Gao *et al.* [122] proposed an LSTM-based encoder-decoder network that takes past information and predicts the future representations, and recognizes actions as early as possible. Geest *et al.* [123] designed a two-stream feedback network using two LSTM streams. One stream focuses on interpreting the features, and the other captures the temporal structure and dependencies. Xu *et al.* [124] designed the Temporal Recurrent Network to model the temporal context by predicting future information. They utilized the future information with historical evidence under the constraint of an online setting to detect actions in the current time. Eun *et al.* [125] argued that RNN units operate without explicitly considering whether input information is relevant to the ongoing action. They proposed a novel recurrent unit that extends GRU (gated recurrent unit) and learns to determine if the input information is relevant to the current action. LAP-Net [126] is an RNN-based method where an adaptive sampling strategy was proposed to estimate current action progression and then decide what temporal ranges should be used to obtain the supplementary features.

Temporal Modeling with Transformer: OadTR [127] argued that RNNs suffer from nonparallelism and gradient vanishing and proposed a transformer to recognize current actions by simultaneously encoding historical information and predicting future context. The encoder captures the global interactions between historical observations while the decoder extracts auxiliary information by aggregating anticipated future clip representations. LSTR [128] presented an online temporal modeling to model activities at different temporal scales with a transformer. They divided the entire history into the long- and short-term memories. The encoder compresses the long-term memory into a latent representation of fixed length, and the decoder uses a short window of transient frames to perform self-attention and cross-attention operations.

3 Datasets and Evaluation

In this section, we provide a summary of benchmark datasets for action detection task in Table 2 (details in Appendix), describe the evaluation metrics and analyze the performance of the state-of-the-art methods.

3.1 Evaluation Metrics

We discuss the metrics designed to evaluate the performance of proposal generation, temporal action detection, spatio-temporal action detection, and online temporal action detection.

Temporal Action Proposal Generation. For this task, Average Recall (AR) with multiple IoU thresholds is

TABLE 2
The benchmark datasets for temporal, spatio-temporal, and online action detection.

Dataset	Activities Types	#Videos	#Action Categories	Avg Video Length (Sec)	#Action Instances (avg per video)	Multi-label (#labels per frame)
THUMOS [1]	Sports	413	20	212	15.5	No
MultiTHUMOS [52]	Sports	413	65	212	97	Yes
Breakfast [11]	Cooking	1,712	48	162	6	No
50Salads [129]	Cooking	50	17	384	20	No
MPII cooking 2 [130]	Cooking	273	59	356	51.6	No
Ava [111]	Movies	437	80	900	3,361.5	Yes
TVSeries [121]	TV series	27	30	2,133.3	231	Yes
ActivityNet [131]	Daily Activities	19,994	200	115	1.54	No
HACS Segment [132]	Daily Activities	50K	200	156	2.8	No
Charades [133]	Daily Activities	9,848	157	30	6.75	Yes
COIN [134]	Daily Activities	11,827	180	142	3.9	No
FineAction [135]	Daily Activities	17K	106	-	6	Yes

usually used as evaluation metrics. Most methods use IoU thresholds set $[0.5 : 0.05 : 0.95]$ in ActivityNet-1.3 [131] and $[0.5 : 0.05 : 1.0]$ in THUMOS14 [1]. To evaluate the relation between recall and proposals number, most methods evaluate AR with Average Number of proposals (AN) on both datasets, which is denoted as AR@AN. On ActivityNet-1.3, area under the AR vs. AN curve (AUC) is also used as metrics, where AN varies from 0 to 100.

Temporal Action Detection. For this task, mean Average Precision (mAP) is used as evaluation metric, where Average Precision (AP) is calculated on each action class. On ActivityNet-1.3 [131], mAP with IoU thresholds $\{0.5, 0.75, 0.95\}$ and average mAP with IoU thresholds set $[0.5 : 0.05 : 0.95]$ are often used. On THUMOS14 [1], mAP with IoU thresholds $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ is used.

Spatio-temporal Action Detection. Two metrics are frequently used for this task. First, *frame-AP* measures the area under the precision-recall curve of the detections for each frame. A detection is correct if the intersection-over-union with the ground truth at that frame is greater than a threshold and the action label is correctly predicted. Second, *video-AP* measures the area under the precision-recall curve of the action tubes predictions. A tube is correct if the mean per frame intersection-over-union with the ground truth across the frames of the video is greater than a threshold and the action label is correctly predicted.

Online Temporal Action Detection. There are two metrics for this task. Per-frame mean average precision (mAP) is the same as mAP for the offline setting. Calibrated average precision (cAP) was proposed [121] to correct the imbalance between positive and negative samples, $\text{cAP} = \sum_k \text{cPrec}(k) * \frac{I(k)}{P}$, where $\text{cPrec} = \frac{\text{TP}}{\text{TP+FP}/w}$, $I(k)$ is 1 if frame k is a true positive, P is the number of true positives, and w is the negative and positive ratio.

Temporal Action Detection Error Analysis. Alwassel *et al.* [136] proposed a diagnostic tool to analyze the performance of temporal action detectors on ActivityNet v1.3 dataset based on false-positive (FP) and false-negative (FN) errors (def 4). They classified FP errors into five main groups and analyzed the impact of error types on the average-mAP for several state-of-the-art methods. They concluded that *localization error* had the most impact among the studied methods. Localization error is a prediction with the correct label that fails to meet the tIoU threshold with the ground truth instance. They also defined six main action characteristics for the ground truth instances and analyzed the average FN rate across different algorithms for each characteristic. They found out that actions with ambiguous temporal boundaries, high temporal context, and short duration are harder to retrieve.

3.2 Performance Analysis

In this section, we analyze the performance of the state-of-the-art methods for temporal and spatio-temporal action detection, and online temporal action detection.

3.2.1 Spatio-temporal Action Detection

Table 3 provides the performance of offline spatio-temporal action detection methods on AVA 2.1 dataset. ACAR-Net [118] achieved the best performance by modeling the relation between two actors based on their interactions with the context. STAGE [120] achieved the second best performance by modeling spatio-temporal dependencies in a graph-based framework. SlowFast [6] achieved the third best performance by capturing spatial semantics in a slow pathway and learning temporal information in fast pathway. LFB [113], VATX [116], and ACAM [117] modeled spatio-temporal dependencies by using attention mechanism to aggregate the contextual information.

TABLE 3
Offline spatio-temporal action detection performance on AVA 2.1 validation set, measured by mAP (%) for IoU $= 0.5$. Some methods used their own person detector (marked with *).

Method	Flow	Visual Encoder	Pretrained	Person Detector	mAP
AVA [111]	✓	I3D [21]	Kinetics-400	FRCNN [97]	15.8
ACRN [114]	✓	S3D-G [137]	Kinetics-400	FRCNN [97]	17.4
STEP [112]	✗	I3D [21]	Kinetics-400	*	18.6
Better-b [138]	✗	I3D [21]	Kinetics-600	FRCNN [97]	21.9
SMAD [115]	✗	I3D [21]	Kinetics-400	Dave *et al.* [139]	22.2
RTPR [140]	✓	Res101 [25]	-	*	22.3
ACAM [117]	✗	I3D [21]	Kinetics-400	FRCNN [97]	24.4
VATX [116]	✗	I3D [21]	Kinetics-400	*	24.9
LFB [113]	✗	Res101 [25]	Kinetics-400	DTRN [141]	27.7
SlowFast [6]	✗	Res101 [25]	Kinetics-600	DTRN [141]	28.2
STAGE [120]	✗	Res101 [25]	Kinetics-600	DTRN [141]	29.8
ACAR [118]	✗	Res101 [25]	Kinetics-400	DTRN [141]	30.0

3.2.2 Online Temporal Action Detection

Table 4 provides the performance of online temporal action detection methods on THUMOS14 and TVSeries datasets. LSTR [128] and OadTR [127] are transformer-based models that achieved the state-of-the-art results and demonstrate the capability of transformers in temporal modeling. WOAD [142] is flexible to combine weak and strong supervision but only their results with full-supervision are reported in Table 4. TFN [143] employed a Non-local [144] block to capture long-range dependencies and attained stronger performance compared to RNN-based models [126], [145], [125].

TABLE 4
Online temporal action detection performance on THUMOS14 and TVSeries in terms of mAP and cAP, respectively. The methods are sorted based on their performance on THUMOS14.

Method	Visual Encoder	Pretrained-on	THUMOS14 mAP (%)	TVSeries cAP (%)
RED [122]	TS [22]	ActivityNet	45.3	79.2
TRN [124]	TS [22]	ActivityNet	47.2	83.7
IDN [125]	TS [22]	ActivityNet	50.0	84.7
FATS [145]	TS [22]	ActivityNet	51.6	81.7
LAP [126]	TS [22]	ActivityNet	53.3	85.3
TFN [143]	TS [22]	ActivityNet	55.7	85.0
OadTR [127]	TSN [24]	ActivityNet	58.3	85.4
LSTR [128]	TSN [24]	ActivityNet	65.3	88.1
FATS [145]	TSN [24]	Kinetics	59.0	84.6
IDN [125]	TSN [24]	Kinetics	60.3	86.1
PKD [146]	I3D [21]	Kinetics	64.5	86.4
OadTR [127]	TSN [24]	Kinetics	65.2	87.2
WOAD [142]	I3D [21]	Kinetics	67.1	82.2
LSTR [128]	TSN [24]	Kinetics	69.5	89.1

3.2.3 Temporal Action Detection

Table 5 provides the performance of offline temporal action detection methods with full and limited supervision on THUMOS14 and ActivityNet datasets. Table 6 summarizes the advantages and limitations of these methods.

Temporal Action Detection with Full Supervision: ActionFormer [147] outperformed all the state-of-the-art methods on THUMOS14 and ActivityNet 1.3 datasets, by combining the advantages of temporal feature pyramid network (TFPN) with transformer. After ActionFormer [147], the methods with superior performance on both datasets are PBRNet [148], VSGN [59], and TSP [28]. They all obtained average mAP more than 35% on ActivityNet 1.3 and mAP@50 more than 51% on THUMOS14. PBRNet [148] included a U-shaped temporal feature pyramid network (UTFPN) with several benefits such as receptive field alignment and enriching the features with high-level semantics as well as fine-grained details. VSGN [59] is graph-based model that exploits correlations between cross-scale snippets. TSP [28] proposed a temporally sensitive pretraining paradigm for clip features. The features trained with the proposed pretraining strategy combined with an anchor-free graph-based model (G-TAD [58]) achieved superior performance on ActivityNet 1.3. On average, methods with anchor-free components (AF or AB+AF) achieved better performance compared with anchor-based (AB) methods because they generated proposals with flexible boundaries. Transformer is a powerful tool to capture long-range dependencies in videos. Among transformer-based models, ATAG [62] obtained stronger performance on ActivityNet 1.3 but weaker performance on THUMOS14 which could be caused by high parametric complexity and over-fitting on such a small dataset. RTD-Net [61] reduced the number of parameters by designing a simpler encoder customized with a boundary-attentive architecture and achieved a better result on THUMOS14 but weaker performance on ActivityNet 1.3.

Temporal Action Detection with Limited Supervision: Among weakly-supervised methods, ASM-Loc [149] achieved the best performance on THUMOS14 by modeling temporal structures within and across action segments using self-attention, up-sampling action proposals with short duration, and proposal refinement. D2-Net [150] addressed noise in TCAMs (def. 15) by maximizing the MI between activations and labels within a video and across videos. ASM-Loc [149] and D2-Net [150] achieved comparable performance to some of the semi-supervised methods ([87], [92]). Several methods attempted to address action completeness with random masking [76], [79]), diversity loss [81], adversarial loss [151], EM process [152], or by sub-action modeling [153], [154], [155]. Among them, AUMN [155] achieved the best performance by modeling sub-actions with a memory network and self-attention. Several methods addressed action-Context separation by using a generative model [74], modeling context [156], or designing a three-branch attention module for action instances, context and background [157], among which ACM-Net [157] achieved the best result. Gong *et al.* [94] attained the state-of-the-art results on ActivityNet-1.2 among weakly-supervised methods by proposing a self-supervised equivariant transform consistency constraint, confirming the advantages of self-supervised learning. Methods with point-level supervision (PLS) required extra annotations compared to weakly supervised (WS) methods but still significantly reduced the labeling cost. Lee *et al.* [85] achieved a significantly stronger performance compared to all WS and PLS methods by modeling action completeness from dense pseudo-labels. Among semi-supervised methods, KFC [90] achieved the best performance by proposing K-farthest crossover to construct perturbed features. ACL [95] is an unsupervised method but still achieved a comparable performance with respect to earlier weakly-supervised methods [65].

4 Discussions

In this section, we describe the application of action detection and introduce future research directions.

4.1 Applications

Action detection has numerous real-world applications as most of the videos in practice are untrimmed. In this section, we describe several applications of this task.

Action Localization in Instructional videos: With the rising popularity of social media, people worldwide upload numerous instructional videos in diverse categories. Millions of people watch these tutorials to learn new tasks such as ”making pancakes” or ”changing a flat tire.” Based on the psychological studies, it has been shown that

TABLE 5

Performance of offline temporal action detection methods on testing set of THUMOS14 and validation set of ActivityNet (V is the version) measured by mAP (%) at tIoU thresholds. Temporal proposal generation is evaluated with AR@AN (%) metric. The methods are first sorted based on mAP@0.5 on THUMOS14 dataset, and then based on mAP-average on ActivityNet. The methods that used extra classifiers are marked with "*" (details in Appendix). Methods are grouped to FS (Fully-supervised), WS (Weakly-supervised), SS (Semi-supervised), SLS (Self-supervised), US (Unsupervised), MLS (Multi-level Supervision), and PLS (Point-level Supervision). Weakly-supervised methods with additional ground-truth annotations are marked with †. For fully-supervised methods, proposal generation is categorized to anchor-based (AB), anchor-free (AF), and both (AB+AF). All semi-supervised methods use 60% of annotated data for a fair comparison (except TTC-Loc [92]).

Group	Method	Visual Encoder	Model	THUMOS14						ActivityNet					V	Code
				mAP (%)					AR@AN	mAP (%)				AR@AN		
				0.3	0.4	0.5	0.6	0.7	200	0.5	0.75	0.95	Avg	100		
FS	SCNN [18]	C3D [23]	Multi-stage CNNs (AB)	36.3	28.7	19.0	-	-	20.0	-	-	-	-	-	1.3	Link
	*Sst [48]	C3D [23]	RNNs (AB)	-	-	23.0	-	-	-	-	-	-	-	-		Link
	CDC [14]	C3D [23]	Encoder-Decoder (AB)	40.1	29.4	23.3	13.1	7.9	-	45.3	26.0	0.2	23.8	-		Link
	SSAD [36]	TS [22]	TFPN (AB)	43	35	24.6	-	-	-	-	-	-	-	-		✗
	R-C3D [35]	C3D [23]	3D RoI Pooling (AB)	44.8	35.6	28.9	-	-	-	26.8	-	-	12.7	-		Link
	SS-TAD [49]	C3D [23]	RNNs (AB)	45.7	-	29.2	-	9.6	-	-	-	-	-	-		Link
	SSN [38]	TS [22]	Structured Pooling (AF)	51.9	41.0	29.8	-	-	48.9	39.12	23.48	5.49	23.98	-		Link
	*CTAP [45]	TS [22]	Confidence Estimator (AB+AF)	-	-	29.9	-	-	50.13	-	-	-	-	73.17		Link
	CBR [34]	TS [22]	Cascaded Regression (AB)	50.1	41.3	31.0	19.1	9.9	44.2	-	-	-	-	-		Link
	S3D [158]	C3D [23]	TFPN (AB)	47.9	41.2	32.6	23.3	14.3	-	-	-	-	-	-		Link
	DBS [15]	TS [22]	Spatial-temporal Dependences (AB)	50.6	43.1	34.3	24.4	14.7	-	43.2	25.8	6.1	26.1	-		✗
	*BSN [40]	TS [22]	Boundary Sensitive (AF)	53.5	45.0	36.9	28.4	20.0	53.2	52.50	33.53	8.85	33.72	74.16		Link
	*MGG [37]	TS [22]	UTFPN (AB+AF)	53.9	46.8	37.4	29.5	21.3	54.6	-	-	-	-	74.5		✗
	AGCN [57]	C3D [23]	Graphs + Attention (AB)	57.1	51.6	38.6	28.9	17.0	-	-	-	-	30.4	-		✗
	*BMN [41]	TS [22]	Boundary Matching (AF)	56.0	47.4	38.8	29.7	20.5	54.7	50.07	34.78	8.29	33.85	75.0		Link
	GTAN [159]	P3D [27]	Gaussian Kernel (AF)	57.8	47.2	38.8	-	-	54.3	52.61	34.14	8.91	34.31	74.8		✗
	*SRG [160]	TS [22]	Attention (AF)	54.5	46.9	39.1	31.4	22.2	56.7	46.53	29.98	4.83	29.72	74.6		✗
	*DBG [42]	TS [22]	Dense Boundary (AF)	57.8	49.4	39.8	30.2	21.7	54.5	-	-	-	-	76.6		Link
	*G-TAD [58]	TSN [24]	Graphs (AF)	54.5	47.6	40.2	30.8	23.4	-	50.36	34.60	9.02	34.09	-		Link
	*BC-GNN [44]	TS [22]	Graphs (AF)	57.1	49.1	40.4	31.2	23.1	56.3	50.56	34.75	9.37	34.26	76.7		✗
	*BSN++ [161]	TS [22]	UTFPN + Attention (AF)	59.9	49.5	41.3	31.9	22.8	57.6	51.27	35.70	8.33	34.88	76.5		✗
	TAL-Net [13]	I3D [21]	Multi-Tower Network (AB)	53.2	48.5	42.8	33.8	20.8	-	38.23	18.30	0	20.22	-		✗
	*BU [162]	I3D [21]	Consistency Loss (AF)	53.9	50.7	45.4	38.0	28.5	55.7	43.47	33.91	9.21	30.12	75.2		Link
	A2Net [163]	I3D [21]	TFPN (AB+AF)	58.6	54.1	45.5	32.5	17.2	-	43.55	28.69	3.7	27.75	-		Link
	*ATAG [62]	TS [22]	Graphs + Transformer (AF)	62.0	53.1	47.3	38.0	28.0	59.4	50.92	35.35	9.71	34.68	76.7		✗
	*Lianli [164]	TS [22]	RNNs + Attention (AF)	66.4	58.4	48.8	36.7	25.5	56.4	47.01	30.52	8.21	30.88	74.4		Link
	PGCN [56]	I3D [21]	Graphs (AF)	63.6	57.8	49.1	-	-	-	48.26	33.16	3.27	31.11	-		Link
	TadTR [63]	I3D [21]	Transformer (AF)	62.4	57.4	49.2	37.8	26.3	-	49.08	32.58	8.49	32.27	-		Link
	AFNet [165]	C3D [23]	3D RoI Pooling (AB)	63.4	58.5	49.5	36.9	23.5	49.1	36.1	17.8	5.2	18.6	-		✗
	AGT [60]	I3D [21]	Graphs + Transformer (AF)	65.0	58.1	50.2	-	-	-	-	-	-	-	-		Link
	PBRNet [148]	I3D [21]	UTFPN (AB+AF)	58.5	54.6	51.3	41.8	29.5	-	53.96	34.97	8.98	35.01	-		Link
	*RTD-Net [61]	I3D [21]	Transformer (AF)	68.3	62.3	51.9	38.8	23.7	56.4	47.21	30.68	8.61	30.83	73.2		Link
	C-TCN [166]	I3D [21]	UTFPN (AB)	68.0	62.3	52.1	-	-	-	47.6	31.9	6.2	31.1	-		Link
	VSGN [59]	TSN [24]	UTFPN + Graphs (AB+AF)	66.7	60.4	52.4	41.0	30.4	-	52.38	36.01	8.37	35.07	-		✗
	*TSA-Net [167]	P3D [27]	Multi-Tower Network (AB)	65.6	61.4	53.0	42.4	28.8	58.3	-	-	-	-	-		✗
	MLTPN [168]	I3D [21]	UTFPN + Attention (AB)	66.0	62.6	53.3	37.0	21.2	-	44.86	28.96	4.30	28.27	-		✗
	TSP [28]	R(2+1)D [26]	Discriminative Pretraining (AF)	69.1	63.3	53.5	40.4	26.0	-	51.26	37.12	9.29	35.81	76.6		Link
	DaoTAD [169]	R50-I3D [25]	UTFPN (AB)	62.8	59.5	53.8	43.6	30.1	-	-	-	-	-	-		Link
	AFSD [43]	I3D [21]	TFPN (AB+AF)	67.3	62.4	55.5	43.7	31.1	-	52.4	35.3	6.5	34.4	-		Link
	SP-TAD [170]	I3D [21]	UTFPN + Attention (AF)	69.2	63.3	55.9	45.7	33.4	-	50.06	32.92	8.44	32.99	-		Link
	*Liu [171]	TS [22]	Temporal Aggregation (AF)	68.9	64.0	56.9	46.3	31.0	-	50.02	34.97	6.57	33.99	-		Link
	ActionFormer [147]	I3D [21]	TFPN + Transformer (AF)	75.5	72.5	65.6	56.6	42.7	-	53.5	36.2	8.2	35.6	-		Link
WS	Hide-Seek [76]	-	Completeness with Random Masking	19.5	12.7	6.8	-	-	-	-	-	-	-	-	1.2	Link
	UNet [19]	UNet [19]	Multi Instance Learning	28.2	21.1	13.7	-	-	-	7.4	3.2	0.7	3.6	-	1.2	Link
	Step-by-step [79]	TSN [24]	Completeness with Iterative Removal	31.1	22.5	15.9	-	-	-	27.3	14.7	2.9	15.6	-	1.2	✗
	STPN [70]	I3D [21]	Cross-entropy Loss	35.5	25.8	16.9	9.9	4.3	-	29.3	16.9	2.6	-	-	1.3	Link
	MAAN [172]	I3D [21]	Marginalized Average Aggregation	41.1	30.6	20.3	12	6.9	-	33.7	21.9	5.5	-	-	1.3	Link
	AutoLoc [77]	UNet [19]	Boundary Contrastive Loss	35.8	29	21.2	13.4	5.8	-	27.3	15.1	3.3	16.0	-	1.2	Link
	W-TALC [68]	I3D [21]	Co-activity Similarity Loss	40.1	31.1	22.8	-	7.6	-	37.0	-	-	18.0	-	1.2	Link
	STAR [173]	I3D [21]	Temporal Modeling with LSTM	48.7	34.7	23.0	-	-	-	31.1	18.8	4.7	-	-	1.3	✗
	† CMCS [81]	I3D [21]	Completeness with Diversity Loss	41.2	32.1	23.1	15	7	-	36.8	22.0	5.6	22.4	-	1.2	Link
	Cleannet [78]	UNet [19]	Temporal Contrast Modeling	37	30.9	23.9	13.9	7.1	-	37.1	20.3	5.0	21.6	-	1.2	✗
	TSM [174]	I3D [21]	Temporal Structure Mining	39.5	31.9	24.5	13.8	7.1	-	28.3	17.0	3.5	17.1	-	1.2	✗
	† 3C-Net [65]	I3D [21]	Feature Learning with Center Loss	40.9	32.3	24.6	-	7.7	-	37.2	-	-	21.7	-	1.2	Link
	Shen *et al* [153]	I3D [21]	Completeness with Sub-actions	44.0	34.4	25.5	15.2	7.2	-	36.9	23.1	3.4	22.8	-	1.2	✗
	AG [175]	I3D [21]	Temporal Modeling with Graphs	47.3	36.4	26.1	-	-	-	29.4	-	-	-	-	1.2	Link
	† BG [71]	I3D [21]	Background Modeling	46.6	37.5	26.8	17.6	9	-	36.4	19.2	2.9	-	-	1.3	✗
	BaSNet [176]	I3D [21]	Background Suppression	44.6	36.0	27.0	18.6	10.4	-	38.5	24.2	5.6	24.3	-	1.2	Link
	RPN [73]	I3D [21]	Action Prototype Learning	48.2	37.2	27.9	16.7	8.1	-	37.6	23.9	5.4	23.3	-	1.2	✗
	TSCN [177]	I3D [21]	Attention Refinement	47.8	37.7	28.7	19.4	10.2	-	37.6	23.7	5.7	23.6	-	1.2	✗
	DGAM [74]	I3D [21]	Action-Context Separation with CVAE	46.8	38.2	28.8	19.8	11.4	-	41.0	23.5	5.3	24.4	-	1.2	Link
	Actionbytes [93]	I3D [21]	Knowledge Transfer from Clips (SLS)	43.0	35.8	29.0	-	9.5	-	39.4	-	-	-	-	1.2	✗
	ECM [178]	I3D [21]	Score Consistency	46.5	38.2	29.1	19.5	10.9	-	41.0	24.9	6.5	25.5	-	1.2	✗
	Deep Metric [179]	I3D [21]	Co-activity Loss with Metric Learning	46.8	-	29.6	-	9.7	-	35.2	-	-	-	-	1.2	Link
	A2CL-PT [151]	I3D [21]	Completeness with Adversarial Loss	48.1	39.0	30.1	19.2	10.6	-	36.8	22.0	5.2	22.5	-	1.3	Link
	EM-MIL [152]	I3D [21]	Completeness with EM	45.5	36.8	30.5	22.7	16.4	-	37.4	-	-	20.3	-	1.2	Link
	ASL [180]	I3D [21]	Generalized Cross-entropy Loss	51.8	-	31.1	-	11.4	-	40.2	-	-	25.8	-	1.2	Link
	Huang *et al* [154]	I3D [21]	Completeness with Sub-actions	49.1	40.0	31.4	18.8	10.6	-	36.5	22.8	6.0	22.9	-	1.3	✗
	CoLA [181]	I3D [21]	Refinement with Contrastive Loss	51.5	41.9	32.2	22.0	13.1	-	42.7	25.7	5.8	26.1	-	1.2	Link
	Acsnet [156]	I3D [21]	Action-Context Separation	51.4	42.7	32.4	22.0	11.7	-	40.1	26.1	6.8	26.0	-	1.2	✗
	AUMN [155]	I3D [21]	Completeness with Sub-actions	54.9	44.4	33.3	20.5	9.0	-	38.3	23.5	5.2	23.5	-	1.3	✗
	Gong *et al.* [94]	I3D [21]	Temporal Transformation (SLS)	50.8	42.2	32.9	21.0	10.1	-	45.5	27.3	5.4	27.6	-	1.2	✗
	FAC-Net [182]	I3D [21]	Foreground-action Consistency	52.6	44.3	33.4	22.5	12.7	-	37.6	24.2	6.0	24.0	-	1.3	Link
	Lee *et al.* [183]	I3D [21]	Uncertainty Modeling	52.3	43.4	33.7	22.9	12.1	-	41.2	25.6	6.0	25.9	-	1.2	Link
	ACM-Net [157]	I3D [21]	Action-Context Separation	55.0	44.6	34.6	21.8	10.8	-	40.1	24.2	6.2	24.6	-	1.3	Link
	UGCT [184]	I3D [21]	Uncertainty Modeling	55.5	46.5	35.9	23.8	11.4	-	39.1	22.4	5.8	23.8	-	1.3	✗
	D2-Net [150]	I3D [21]	Discriminative + Denoising Loss	52.3	43.4	36.0	-	-	-	42.3	25.5	5.8	26.0	-	1.2	Link
	ASM-Loc [149]	I3D [21]	Temporal Modeling with Self-attention	57.1	46.8	36.6	25.2	13.4	-	41.0	24.9	6.2	25.1	-	1.3	Link
PLS	Moltisanti *et al.* [185]	BN-I [186]	Sampling from Single Timestamps	15.9	12.5	9.0	-	-	-	-	-	-	-	-	-	✗
	SF-Net [83]	I3D [21]	Action-Background Mining	52.8	42.2	30.5	20.6	12.0	-	37.8	-	-	22.8	-	1.2	Link
	PTAL [84]	I3D [21]	Keypoint Detection + Mapper	58.2	47.1	35.9	23.0	12.8	-	-	-	-	-	-	-	✗
	BackTAL [86]	I3D [21]	Background-click Supervision	54.4	45.5	36.3	26.2	14.8	-	41.5	27.3	4.7	27.0	-	1.2	Link
	Lee *et al.* [85]	I3D [21]	Completeness with Action Contrast	64.6	56.5	45.3	34.5	21.8	-	44.0	26.0	5.9	26.8	-	1.2	Link
SS	TTC-Loc [92],	I3D [21]	Detection with Adaptive Thresholds	52.8	44.4	35.9	24.7	13.8	-	37.6	21.5	4.7	22.2	-	1.3	✗
	Ji *et al* [87]	TS [22]	Mean-Teacher Sequential Perturbations	53.4	45.2	37.2	29.5	20.5	53.3	-	-	-	-	75.1		✗
	PM-MT [88]	TS [22]	Mean-Teacher with Map Warping	-	-	-	-	-	54.2	-	-	-	-	75.5		✗
	*SSTAP [89]	TS [22]	Mean-Teacher + Pretext Tasks (SLS)	56.5	48.8	39.4	30.5	20.7	55.0	50.1	34.9	7.4	34.0	75.2		Link
	*KFC [90]	TSN [24]	K-farthest Crossover Perturbations	57.7	51.5	43.3	32.4	22.9	-	51.6	34.9	9.0	34.4	-		✗
US	ACL [95]	I3D [21]	Iterative Clustering and Localization	39.6	32.9	25.0	16.7	8.9	-	35.2	21.4	3.1	21.1	-	1.2	Link
MLS	Shi *et al.* [96]	TS [22]	Unlabeled and Weakly-labeled Data	45.6	36.4	26.2	15.5	7.10	-	19.47	12.54	1.88	12.27	-	1.2	Link

TABLE 6
Summary of temporal action detection methods with full-supervision (FS) and limited supervision (LS). $(+)$ and $(-)$ denote the advantages and disadvantages.

Group	Objective	Category	Methods	Advantages and Limitations
FS	Proposal Generation	Anchor-based	[13], [35], [165], [187] [57], [166], [168]	+ Efficiently generate multiple-scales proposals. - Proposals are not temporally flexible and precise.
		Anchor-free	[56], [63], [63], [164] [40], [41], [42], [170] [44], [58], [62], [161]	+ Generate proposals with flexible duration. - Proposal evaluation is not efficient in some cases. - Distorting the information of short actions due to down-scaling.
		Anchor-based +Anchor-free	[37], [43], [59], [148]	+ Combining advantages of anchor-based and anchor-free. - Not an end-to-end network in most cases.
	Proposal Feature Extraction and Evaluation	RoI Pooling	[35], [165]	+ Fast feature extraction from multi-scale proposals. - Proposal features may include insufficient or irrelevant information.
		Multi-tower Network	[13], [167]	+ Alignment of receptive field to proposal span to extract rich features from proposals. - Multiple networks for different anchor sizes (inefficient).
		TFPN	[36], [43], [158], [163]	+ Receptive field alignment for multiple anchor sizes in one network. - Lower layers are unaware of high-level information, top layers lack enough details.
		U-shaped TFPN	[37], [46], [166], [168] [47], [169], [170]	+ Receptive field alignment for multiple anchor sizes in one network. + Enriching features with both high-level semantics and fine-grained details. - No modeling of temporal dependencies.
		RNNs	[48], [49]	+ Modeling long-term dependencies in proposal feature extraction. - Hidden vector saturation for long sequences.
		Graphs	[44], [56], [57], [58] [59], [62]	+ Modeling temporal dependencies between proposals or video segments. - Temporal dependencies are used only for proposal refinement (not generation) in most cases.
		Transformer	[60], [61], [62], [63]	+ Modeling temporal structure for proposal generation and evaluation. - High parametric complexity.
LS	Localization with Class-specific Attention	Multi-Instance Learning (MIL)	[19], [68], [175], [176] [65], [93]	+ Learning temporal class activation maps. - Only supervising temporal positions with highest activation scores (top-k for predefined k).
		Co-activity Similarity Loss (CASL)	[68], [175], [179], [93]	+ Action-background separation, and action intra-class compactness are addressed . - Action-context confusion is not addressed. - Relation between different action categories, and action completeness are not modeled.
		Center Loss	[65]	+ Action intra-class compactness by pushing action features to class centers. + Separates different action classes with help of cross-entropy loss. - Center loss is sensitive to initialization of class centers.
	Localization with Class-agnostic Attention	Cross-entropy (CE) Loss	[70], [73], [71]	+ Learning class-agnostic attention. - Only sensitive to the most discriminative parts of the action, causing incomplete detection.
		Clustering Loss	[73], [71]	+ Separating foreground-background features. - Force all background features to belong to one class, even if they do not share common semantics.
		Prototype Learning	[73], [154]	+ Action intra-class compactness, and inter-class separateness. - Sensitive to initialization of prototypes.
		Conditional VAE	[74]	+ Separating actions from context frames with a generative model. - Not modeling temporal dependencies and relation between sub-actions.
	Direct Localization	Boundary Contrast Modeling	[77], [78]	+ Predicting boundaries of action instances instead of thresholding on attention signals. - Not modeling action completeness.
	Action Completeness Modeling	Masking or Prediction Removal	[76], [79]	+ Hiding video regions or erasing predictions to see different action parts. - Does not guarantee the discovery of new parts and action completeness.
		Diversity Loss	[81]	+ Enforcing the model to discover complementary pieces of an action. - Imprecise background modeling by violating MIL assumption (uniform negative distribution).
		Expectation-Maximization	[152]	+ Highlighting less discriminative segments and modeling background more accurately with EM process. - Not modeling the temporal structure of the video.

simplifying and segmenting the video into smaller steps (sub-actions) is a more effective way to learn a new task [188], [189]. Many datasets are designed to study action localization and action anticipation such as EPIC-Kitchen [190] and INRIA Instructional Videos Dataset [191]. All these tasks are directly related to action detection.

Anomaly Detection in Surveillance Videos: Surveillance cameras are increasingly deployed in public places, monitoring the areas of interest to ensure security. Anomalies are significant deviations of scene entities from normal behavior [192], [193] such as fighting, traffic accidents, burglary, and robbery. Compared to normal activities, anomalous events rarely occur. Therefore, intelligent computer vision algorithms are required to detect anomalous events automatically, to avoid the waste of time and labor [194], [195], [196], [197]. In many real-time applications, the system must detect anomalous events as soon as each video frame arrives, only based on history and the current data. To this end, online action detection algorithms are developed to accumulate historical observations and predicted future information to analyze the current events [198], [199], [200], [201], [124].

Action Spotting in Sports: Professional analysts utilize sports videos to investigate the strategies in a game, examine new players, and generate meaningful statistics. They watch many broadcasts to spot the highlights within a game, which is a time-consuming and costly process. Automated sports analytic methods can facilitate sports broadcasts understanding. Human activity localization in sports videos is studied in [202], [203], [204], [205], salient game actions are identified in [206], [207], automatic game highlights identification and summarization are performed in [208], [209], [210], [211], [212].Action spotting, which is the task of temporal localization of human-induced events, has been popular in soccer game broadcasts [3], [213] and some methods aimed to automatically detect goals, penalties, corner kicks, and card events [214].

Action Detection in Autonomous Driving: With the rapid development of vehicles in urban transportation, autonomous driving has attracted more attention in the last decades. The cameras assembled on the self-driving cars capture the real-time stream of videos that need to be processed with online algorithms. The car should be aware of the surrounding environment and detect and anticipate road users activities to adjust the speed and handle the situation. Therefore, spatio-temporal action localization algorithms need to be developed to guarantee the safety of self-driving cars [215], [216], [4].

4.2 Future work

Action localization with limited supervision has drawn much research attention by skipping exhaustive annotation of action instances in untrimmed videos. Subsequently, knowledge transfer from publicly available trimmed videos is a promising trend to make up for the coarse-grained video-level annotations in weakly-supervised settings [93], [217], [218], [219]. Domain-adaptation schemes must fulfill the domain gap between trimmed and untrimmed videos to transfer robust and reliable knowledge. The task of zero-shot temporal activity detection (ZSTAD) is introduced in [220] to generalize the applicability of action detection methods to newly emerging or rare events that are not included in the training set. The task of ZSTAD is highly challenging because each untrimmed video in the testing set possibly contains multiple novel action classes that must be localized and detected. It is worth mentioning that activity detection with few-shot learning has been recently explored in [93], [221], [222], [223], [224], [225]. The advancement of both zero-shot and few-shot action detection is anticipated in the near future.

5 Conclusion

Action detection schemes have expedited the progress in many real-world applications such as instructional video analysis, anomaly detection in surveillance videos, sports analysis, and autonomous driving. The advancement of learning methods with limited supervision has facilitated action detection by detachment from costly need to annotate the temporal boundary of actions in long videos. This survey has extensively studied recent deep learning methods for action detection from different aspects including fully-supervised schemes, methods with limited supervision, benchmark datasets, and applications. The performance analysis and future directions are summarized to inspire the design of new and efficient methods for action detection that serves the computer vision community.

References

[1] Y.-G. 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,” 2014.

[2] F. Rea, A. Vignolo, A. Sciutti, and N. Noceti, “Human motion understanding for selecting action timing in collaborative human-robot interaction,” *Front. Robot. AI*, vol. 6, p. 58, 2019.

[3] A. Cioppa, A. Deliege, S. Giancola, B. Ghanem, M. V. Droogenbroeck, R. Gade, and T. B. Moeslund, “A context-aware loss function for action spotting in soccer videos,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 126–13 136.

[4] A. Rasouli and J. K. Tsotsos, “Autonomous vehicles that interact with pedestrians: A survey of theory and practice,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 900–918, 2019.

[5] S. Herath, M. Harandi, and F. Porikli, “Going deeper into action recognition: A survey,” *Image and vision computing*, vol. 60, pp. 4–21, 2017.

[6] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “Slowfast networks for video recognition,” in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 6202–6211.

[7] D. Ghadiyaram, D. Tran, and D. Mahajan, “Large-scale weakly-supervised pre-training for video action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 046–12 055.

[8] H. Duan, Y. Zhao, Y. Xiong, W. Liu, and D. Lin, “Omni-sourced webly-supervised learning for video recognition,” *arXiv preprint arXiv:2003.13042*, 2020.

[9] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, “The thumos challenge on action recognition for videos “in the wild”,” *Computer Vision and Image Understanding*, vol. 155, pp. 1–23, 2017.

[10] A. Gorban, H. Idrees, Y.-G. Jiang, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar, “Thumos challenge: Action recognition with a large number of classes,” 2015.

[11] H. Kuehne, A. Arslan, and T. Serre, “The language of actions: Recovering the syntax and semantics of goal-directed human activities,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 780–787.

[12] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks for action segmentation and detection,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[13] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1130–1139.

[14] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5734–5743.

[15] Z. Gao, L. Wang, Q. Zhang, Z. Niu, N. Zheng, and G. Hua, “Video imprint segmentation for temporal action detection in untrimmed videos,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8328–8335.

[16] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[17] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge 2012 (voc2012) results (2012),” in *URL http://www. pascal-network. org/challenges/VOC/voc2011/workshop/index. html*, 2011.

[18] Z. Shou, D. Wang, and S.-F. Chang, “Temporal action localization in untrimmed videos via multi-stage cnns,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[19] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, “Untrimmednets for weakly supervised action recognition and detection,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 4325–4334.

[20] Y. Xiong, L. Wang, Z. Wang, B. Zhang, H. Song, W. Li, D. Lin, Y. Qiao, L. Van Gool, and X. Tang, “Cuhk & ethz & siat submission to activitynet challenge 2016,” *arXiv preprint arXiv:1608.00797*, 2016.

[21] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[22] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in neural information processing systems*, 2014, pp. 568–576.

[23] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[24] 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 *European conference on computer vision*. Springer, 2016, pp. 20–36.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[27] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[28] H. Alwassel, S. Giancola, and B. Ghanem, "Tsp: Temporally-sensitive pretraining of video encoders for localization tasks," *arXiv preprint arXiv:2011.11479*, 2020.

[29] M. Xu, J.-M. Pérez-Rúa, V. Escorcia, B. Martinez, X. Zhu, L. Zhang, B. Ghanem, and T. Xiang, "Boundary-sensitive pre-training for temporal localization in videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7220–7230.

[30] M. Xu, J. M. Perez Rua, X. Zhu, B. Ghanem, and B. Martinez, "Low-fidelity video encoder optimization for temporal action localization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9923–9935, 2021.

[31] M. Xu, E. Gundogdu, M. Lapin, B. Ghanem, M. Donoser, and L. Bazzani, "Contrastive language-action pre-training for temporal localization," *arXiv preprint arXiv:2204.12293*, 2022.

[32] C. Zhang, T. Yang, J. Weng, M. Cao, J. Wang, and Y. Zou, "Unsupervised pre-training for temporal action localization tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 031–14 041.

[33] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3628–3636.

[34] J. Gao, Z. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," *arXiv preprint arXiv:1705.01180*, 2017.

[35] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5783–5792.

[36] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 988–996.

[37] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S.-F. Chang, "Multi-granularity generator for temporal action proposal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3604–3613.

[38] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2914–2923.

[39] J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta informaticae*, vol. 41, no. 1, 2, pp. 187–228, 2000.

[40] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[41] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "Bmn: Boundary-matching network for temporal action proposal generation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3889–3898.

[42] C. Lin, J. Li, Y. Wang, Y. Tai, D. Luo, Z. Cui, C. Wang, J. Li, F. Huang, and R. Ji, "Fast learning of temporal action proposal via dense boundary generator." in *AAAI*, 2020, pp. 11 499–11 506.

[43] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Learning salient boundary feature for anchor-free temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3320–3329.

[44] Y. Bai, Y. Wang, Y. Tong, Y. Yang, Q. Liu, and J. Liu, "Boundary content graph neural network for temporal action proposal generation," *arXiv preprint arXiv:2008.01432*, 2020.

[45] J. Gao, K. Chen, and R. Nevatia, "Ctap: Complementary temporal action proposal generation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 68–83.

[46] Q. Liu and Z. Wang, "Progressive boundary refinement network for temporal action detection."

[47] J. Gao, Z. Shi, G. Wang, J. Li, Y. Yuan, S. Ge, and X. Zhou, "Accurate temporal action proposal generation with relation-aware pyramid network." in *AAAI*, 2020, pp. 10 810–10 817.

[48] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles, "Sst: Single-stream temporal action proposals," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2911–2920.

[49] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," 2019.

[50] J. Yuan, B. Ni, X. Yang, and A. A. Kassim, "Temporal action localization with pyramid of score distribution features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3093–3102.

[51] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2678–2687.

[52] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 375–389, 2018.

[53] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Daps: Deep action proposals for action understanding," in *European Conference on Computer Vision*. Springer, 2016, pp. 768–784.

[54] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, "A multi-stream bi-directional recurrent neural network for fine-grained action detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1961–1970.

[55] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1942–1950.

[56] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7094–7103.

[57] J. Li, X. Liu, Z. Zong, W. Zhao, M. Zhang, and J. Song, "Graph attention based proposal 3d convnets for action detection." in *AAAI*, 2020, pp. 4626–4633.

[58] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-tad: Sub-graph localization for temporal action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 156–10 165.

[59] C. Zhao, A. Thabet, and B. Ghanem, "Video self-stitching graph network for temporal action localization," *arXiv preprint arXiv:2011.14598*, 2020.

[60] M. Nawhal and G. Mori, "Activity graph transformer for temporal action localization," *arXiv preprint arXiv:2101.08540*, 2021.

[61] J. Tan, J. Tang, L. Wang, and G. Wu, "Relaxed transformer decoders for direct action proposal generation," *arXiv preprint arXiv:2102.01894*, 2021.

[62] S. Chang, P. Wang, F. Wang, H. Li, and J. Feng, "Augmented transformer with adaptive graph for temporal action proposal generation," *arXiv preprint arXiv:2103.16024*, 2021.

[63] X. Liu, Q. Wang, Y. Hu, X. Tang, S. Bai, and X. Bai, "End-to-end temporal action detection with transformer," *arXiv preprint arXiv:2106.10271*, 2021.

[64] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia, "Temporal localization of fine-grained actions in videos by domain transfer from web images," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 371–380.

[65] S. Narayan, H. Cholakkal, F. S. Khan, and L. Shao, "3c-net: Category count and center loss for weakly-supervised action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8679–8687.

[66] J. Schroeter, K. Sidorov, and D. Marshall, "Weakly-supervised temporal localization via occurrence count learning," *arXiv preprint arXiv:1905.07293*, 2019.

[67] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," *Pattern Recognition*, vol. 77, pp. 329–353, 2018.

[68] S. Paul, S. Roy, and A. K. Roy-Chowdhury, "W-talc: Weakly-supervised temporal activity localization and classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 563–579.

[69] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[70] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6752–6761.

[71] P. X. Nguyen, D. Ramanan, and C. C. Fowlkes, "Weakly-supervised action localization with background modeling," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5502–5511.

[72] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, 2017, pp. 4077–4087.

[73] L. Huang, Y. Huang, W. Ouyang, L. Wang *et al.*, "Relational prototypical network for weakly supervised temporal action localization," 2020.

[74] B. Shi, Q. Dai, Y. Mu, and J. Wang, "Weakly-supervised action localization by generative attention modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1009–1019.

[75] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in neural information processing systems*, 2015, pp. 3483–3491.

[76] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *2017 IEEE international conference on computer vision (ICCV)*. IEEE, 2017, pp. 3544–3553.

[77] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang, "Autoloc: Weakly-supervised temporal action localization in untrimmed videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 154–171.

[78] Z. Liu, L. Wang, Q. Zhang, Z. Gao, Z. Niu, N. Zheng, and G. Hua, "Weakly supervised temporal action localization through contrast based evaluation networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3899–3908.

[79] J.-X. Zhong, N. Li, W. Kong, T. Zhang, T. H. Li, and G. Li, "Step-by-step erasion, one-by-one collection: a weakly supervised temporal action detector," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 35–44.

[80] R. Zeng, C. Gan, P. Chen, W. Huang, Q. Wu, and M. Tan, "Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5797–5808, 2019.

[81] D. Liu, T. Jiang, and Y. Wang, "Completeness modeling and context separation for weakly supervised temporal action localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1298–1307.

[82] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[83] F. Ma, L. Zhu, Y. Yang, S. Zha, G. Kundu, M. Feiszli, and Z. Shou, "Sf-net: Single-frame supervision for temporal action localization," in *European conference on computer vision*. Springer, 2020, pp. 420–437.

[84] C. Ju, P. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Point-level temporal action localization: Bridging fully-supervised proposals to weakly-supervised losses," *arXiv preprint arXiv:2012.08236*, 2020.

[85] P. Lee and H. Byun, "Learning action completeness from points for weakly-supervised temporal action localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 648–13 657.

[86] L. Yang, J. Han, T. Zhao, T. Lin, D. Zhang, and J. Chen, "Background-click supervision for temporal action localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[87] J. Ji, K. Cao, and J. C. Niebles, "Learning temporal action proposals with fewer labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7073–7082.

[88] W. Wang, T. Lin, D. He, F. Li, S. Wen, L. Wang, and J. Liu, "Semi-supervised temporal action proposal generation via exploiting 2-d proposal map," *IEEE Transactions on Multimedia*, 2021.

[89] X. Wang, S. Zhang, Z. Qing, Y. Shao, C. Gao, and N. Sang, "Self-supervised learning for semi-supervised temporal action proposal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1905–1914.

[90] X. Ding, N. Wang, X. Gao, J. Li, X. Wang, and T. Liu, "Kfc: An efficient framework for semi-supervised temporal action localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 6869–6878, 2021.

[91] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[92] X. Lin, Z. Shou, and S.-F. Chang, "Towards train-test consistency for semi-supervised temporal action localization," *arXiv preprint arXiv:1910.11285*, 2019.

[93] M. Jain, A. Ghodrati, and C. G. Snoek, "Actionbytes: Learning from trimmed videos to localize actions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1171–1180.

[94] G. Gong, L. Zheng, W. Jiang, and Y. Mu, "Self-supervised video action localization with adversarial temporal transforms."

[95] G. Gong, X. Wang, Y. Mu, and Q. Tian, "Learning temporal co-attention models for unsupervised video action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9819–9828.

[96] B. Shi, Q. Dai, J. Hoffman, K. Saenko, T. Darrell, and H. Xu, "Temporal action detection with multi-level supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8022–8032.

[97] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[98] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek, "Action localization with tubelets from motion," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 740–747.

[99] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid, "Spatio-temporal object detection proposals," in *European conference on computer vision*. Springer, 2014, pp. 737–752.

[100] W. Chen and J. J. Corso, "Action detection by implicit intentional motion clustering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3298–3306.

[101] J. C. Van Gemert, M. Jain, E. Gati, C. G. Snoek *et al.*, "Apt: Action localization proposals from dense trajectories." in *BMVC*, vol. 2, 2015, p. 4.

[102] G. Gkioxari and J. Malik, "Finding action tubes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 759–768.

[103] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (t-cnn) for action detection in videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5822–5831.

[104] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," *arXiv preprint arXiv:1608.01529*, 2016.

[105] Z. Yang, J. Gao, and R. Nevatia, "Spatio-temporal action detection with cascade proposal and location anticipation," *arXiv preprint arXiv:1708.00042*, 2017.

[106] Y. Ye, X. Yang, and Y. Tian, "Discovering spatio-temporal action tubes," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 515–524, 2019.

[107] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek, "Videolstm convolves, attends and flows for action recognition," *Computer Vision and Image Understanding*, vol. 166, pp. 41–50, 2018.

[108] L. Wang, Y. Qiao, X. Tang, and L. Van Gool, "Actionness estimation using hybrid fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2708–2717.

[109] G. Yu and J. Yuan, "Fast action proposals for human action detection and search," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1302–1311.

[110] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Action tubelet detector for spatio-temporal action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4405–4413.

[111] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6047–6056.

[112] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L. S. Davis, and J. Kautz, "Step: Spatio-temporal progressive learning for video action detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 264–272.

[113] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 284–293.

[114] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid, "Actor-centric relation network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 318–334.

[115] Y. Zhang, P. Tokmakov, M. Hebert, and C. Schmid, "A structured model for action detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9975–9984.

[116] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 244–253.

[117] O. Ulutan, S. Rallapalli, M. Srivatsa, C. Torres, and B. Manjunath, "Actor conditioned attention maps for video action detection," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 527–536.

[118] J. Pan, S. Chen, M. Z. Shou, Y. Liu, J. Shao, and H. Li, "Actor-context-actor relation network for spatio-temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 464–474.

[119] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, "Action genome: Actions as compositions of spatio-temporal scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 236–10 247.

[120] M. Tomei, L. Baraldi, S. Calderara, S. Bronzin, and R. Cucchiara, "Video action detection by learning graph-based spatio-temporal interactions," *Computer Vision and Image Understanding*, vol. 206, p. 103187, 2021.

[121] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, "Online action detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 269–284.

[122] J. Gao, Z. Yang, and R. Nevatia, "Red: Reinforced encoder-decoder networks for action anticipation," *arXiv preprint arXiv:1707.04818*, 2017.

[123] R. De Geest and T. Tuytelaars, "Modeling temporal structure with lstm for online action detection," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1549–1557.

[124] M. Xu, M. Gao, Y.-T. Chen, L. S. Davis, and D. J. Crandall, "Temporal recurrent networks for online action detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5532–5541.

[125] H. Eun, J. Moon, J. Park, C. Jung, and C. Kim, "Learning to discriminate information for online action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 809–818.

[126] S. Qu, G. Chen, D. Xu, J. Dong, F. Lu, and A. Knoll, "Lap-net: Adaptive features sampling via learning action progression for online action detection," *arXiv preprint arXiv:2011.07915*, 2020.

[127] X. Wang, S. Zhang, Z. Qing, Y. Shao, Z. Zuo, C. Gao, and N. Sang, "Oadtr: Online action detection with transformers," *arXiv preprint arXiv:2106.11149*, 2021.

[128] M. Xu, Y. Xiong, H. Chen, X. Li, W. Xia, Z. Tu, and S. Soatto, "Long short-term transformer for online action detection," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[129] S. Stein and S. J. McKenna, "Combining embedded accelerometers with computer vision for recognizing food preparation activities," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013, pp. 729–738.

[130] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele, "Recognizing fine-grained and composite activities using hand-centric features and script data," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 346–373, 2016.

[131] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2015, pp. 961–970.

[132] H. Zhao, A. Torralba, L. Torresani, and Z. Yan, "Hacs: Human action clips and segments dataset for recognition and temporal localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8668–8678.

[133] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *European Conference on Computer Vision*. Springer, 2016, pp. 510–526.

[134] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J. Zhou, "Coin: A large-scale dataset for comprehensive instructional video analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1207–1216.

[135] Y. Liu, L. Wang, X. Ma, Y. Wang, and Y. Qiao, "Fineaction: A fined video dataset for temporal action localization," *arXiv e-prints*, pp. arXiv–2105, 2021.

[136] H. Alwassel, F. Caba Heilbron, V. Escorcia, and B. Ghanem, "Diagnosing error in temporal action detectors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 256–272.

[137] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning for video understanding," *arXiv preprint arXiv:1712.04851*, vol. 1, no. 2, p. 5, 2017.

[138] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "A better baseline for ava," *arXiv preprint arXiv:1807.10066*, 2018.

[139] A. Dave, P. Tokmakov, and D. Ramanan, "Towards segmenting anything that moves," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[140] D. Li, Z. Qiu, Q. Dai, T. Yao, and T. Mei, "Recurrent tubelet proposal and recognition networks for action detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 303–318.

[141] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[142] M. Gao, Y. Zhou, R. Xu, R. Socher, and C. Xiong, "Woad: Weakly supervised online action detection in untrimmed videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1915–1923.

[143] H. Eun, J. Moon, J. Park, C. Jung, and C. Kim, "Temporal filtering networks for online action detection," *Pattern Recognition*, vol. 111, p. 107695, 2021.

[144] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[145] Y. H. Kim, S. Nam, and S. J. Kim, "Temporally smooth online action detection using cycle-consistent future anticipation," *Pattern Recognition*, vol. 116, p. 107954, 2021.

[146] P. Zhao, L. Xie, Y. Zhang, Y. Wang, and Q. Tian, "Privileged knowledge distillation for online action detection," *arXiv preprint arXiv:2011.09158*, 2020.

[147] C. Zhang, J. Wu, and Y. Li, "Actionformer: Localizing moments of actions with transformers," *arXiv preprint arXiv:2202.07925*, 2022.

[148] Q. Liu and Z. Wang, "Progressive boundary refinement network for temporal action detection." in *AAAI*, 2020, pp. 11 612–11 619.

[149] B. He, X. Yang, L. Kang, Z. Cheng, X. Zhou, and A. Shrivastava, "Asm-loc: Action-aware segment modeling for weakly-supervised temporal action localization," *arXiv preprint arXiv:2203.15187*, 2022.

[150] S. Narayan, H. Cholakkal, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "D2-net: Weakly-supervised action localization via discriminative embeddings and denoised activations," *arXiv preprint arXiv:2012.06440*, 2020.

[151] K. Min and J. J. Corso, "Adversarial background-aware loss for weakly-supervised temporal activity localization," *arXiv preprint arXiv:2007.06643*, 2020.

[152] Z. Luo, D. Guillory, B. Shi, W. Ke, F. Wan, T. Darrell, and H. Xu, "Weakly-supervised action localization with expectation-maximization multi-instance learning," *arXiv preprint arXiv:2004.00163*, 2020.

[153] Z. Shen, F. Wang, and J. Dai, "Weakly supervised temporal action localization by multi-stage fusion network," *IEEE Access*, vol. 8, pp. 17 287–17 298, 2020.

[154] L. Huang, Y. Huang, W. Ouyang, and L. Wang, "Modeling subactions for weakly supervised temporal action localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 5154–5167, 2021.

[155] W. Luo, T. Zhang, W. Yang, J. Liu, T. Mei, F. Wu, and Y. Zhang, "Action unit memory network for weakly supervised temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9969–9979.

[156] Z. Liu, L. Wang, Q. Zhang, W. Tang, J. Yuan, N. Zheng, and G. Hua, "Acsnet: Action-context separation network for weakly supervised temporal action localization," *arXiv preprint arXiv:2103.15088*, 2021.

[157] S. Qu, G. Chen, Z. Li, L. Zhang, F. Lu, and A. Knoll, "Acm-net: Action context modeling network for weakly-supervised temporal action localization," *arXiv preprint arXiv:2104.02967*, 2021.

[158] D. Zhang, X. Dai, X. Wang, and Y.-F. Wang, "S3d: single shot multi-span detector via fully 3d convolutional networks," *arXiv preprint arXiv:1807.08069*, 2018.

[159] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 344–353.

[160] H. Eun, S. Lee, J. Moon, J. Park, C. Jung, and C. Kim, "Srg: Snippet relatedness-based temporal action proposal generator," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[161] H. Su, "Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[162] P. Zhao, L. Xie, C. Ju, Y. Zhang, Y. Wang, and Q. Tian, "Bottom-up temporal action localization with mutual regularization."

[163] L. Yang, H. Peng, D. Zhang, J. Fu, and J. Han, "Revisiting anchor mechanisms for temporal action localization," *IEEE Transactions on Image Processing*, vol. 29, pp. 8535–8548, 2020.

[164] L. Gao, T. Li, J. Song, Z. Zhao, and H. T. Shen, "Play and rewind: Context-aware video temporal action proposals," *Pattern Recognition*, p. 107477, 2020.

[165] G. Chen, C. Zhang, and Y. Zou, "Afnet: Temporal locality-aware network with dual structure for accurate and fast action detection," *IEEE Transactions on Multimedia*, 2020.

[166] X. Li, T. Lin, X. Liu, C. Gan, W. Zuo, C. Li, X. Long, D. He, F. Li, and S. Wen, "Deep concept-wise temporal convolutional networks for action localization," *arXiv preprint arXiv:1908.09442*, 2019.

[167] G. Gong, L. Zheng, and Y. Mu, "Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[168] X. Wang, C. Gao, S. Zhang, and N. Sang, "Multi-level temporal pyramid network for action detection," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2020, pp. 41–54.

[169] C. Wang, H. Cai, Y. Zou, and Y. Xiong, "Rgb stream is enough for temporal action detection," *arXiv preprint arXiv:2107.04362*, 2021.

[170] J. Wu, P. Sun, S. Chen, J. Yang, Z. Qi, L. Ma, and P. Luo, "Towards high-quality temporal action detection with sparse proposals," *arXiv preprint arXiv:2109.08847*, 2021.

[171] X. Liu, Y. Hu, S. Bai, F. Ding, X. Bai, and P. H. Torr, "Multi-shot temporal event localization: a benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 596–12 606.

[172] Y. Yuan, Y. Lyu, X. Shen, I. W. Tsang, and D.-Y. Yeung, "Marginalized average attentional network for weakly-supervised learning," *arXiv preprint arXiv:1905.08586*, 2019.

[173] Y. Xu, C. Zhang, Z. Cheng, J. Xie, Y. Niu, S. Pu, and F. Wu, "Segregated temporal assembly recurrent networks for weakly supervised multiple action detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9070–9078.

[174] T. Yu, Z. Ren, Y. Li, E. Yan, N. Xu, and J. Yuan, "Temporal structure mining for weakly supervised action detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5522–5531.

[175] M. Rashid, H. Kjellstrom, and Y. J. Lee, "Action graphs: Weakly-supervised action localization with graph convolution networks," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 615–624.

[176] P. Lee, Y. Uh, and H. Byun, "Background suppression network for weakly-supervised temporal action localization." in *AAAI*, 2020, pp. 11 320–11 327.

[177] Y. Zhai, L. Wang, W. Tang, Q. Zhang, J. Yuan, and G. Hua, "Two-stream consensus network for weakly-supervised temporal action localization," in *European conference on computer vision*. Springer, 2020, pp. 37–54.

[178] L. Yang, D. Zhang, T. Zhao, and J. Han, "Equivalent classification mapping for weakly supervised temporal action localization," *arXiv preprint arXiv:2008.07728*, 2020.

[179] A. Islam and R. Radke, "Weakly supervised temporal action localization using deep metric learning," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 547–556.

[180] J. Ma, S. K. Gorti, M. Volkovs, and G. Yu, "Weakly supervised action selection learning in video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7587–7596.

[181] C. Zhang, M. Cao, D. Yang, J. Chen, and Y. Zou, "Cola: Weakly-supervised temporal action localization with snippet contrastive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 010–16 019.

[182] L. Huang, L. Wang, and H. Li, "Foreground-action consistency network for weakly supervised temporal action localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8002–8011.

[183] P. Lee, J. Wang, Y. Lu, and H. Byun, "Weakly-supervised temporal action localization by uncertainty modeling," *arXiv preprint arXiv:2006.07006*, 2020.

[184] W. Yang, T. Zhang, X. Yu, T. Qi, Y. Zhang, and F. Wu, "Uncertainty guided collaborative training for weakly supervised temporal action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 53–63.

[185] D. Moltisanti, S. Fidler, and D. Damen, "Action recognition from single timestamp supervision in untrimmed videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9915–9924.

[186] S. Ioffe and C. S. B. Normalization, "Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2014.

[187] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen, "Temporal context network for activity localization in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5793–5802.

[188] Y. Tang, J. Lu, and J. Zhou, "Comprehensive instructional video analysis: The coin dataset and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[189] R. J. Nadolski, P. A. Kirschner, and J. J. Van Merriënboer, "Optimizing the number of steps in learning tasks for complex skills," *British Journal of Educational Psychology*, vol. 75, no. 2, pp. 223–237, 2005.

[190] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.

[191] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien, "Unsupervised learning from narrated instruction videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4575–4583.

[192] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection: A survey," *ACM Computing Surveys*, vol. 14, p. 15, 2007.

[193] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[194] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 18–32, 2013.

[195] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware activity recognition and anomaly detection in video," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 91–101, 2012.

[196] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6479–6488.

[197] C. He, J. Shao, and J. Sun, "An anomaly-introduced learning method for abnormal event detection," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29 573–29 588, 2018.

[198] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.

[199] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1992–2004, 2017.

[200] M. Sabokrou, M. Pourreza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, "Avid: Adversarial visual irregularity detection," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 488–505.

[201] K. Liu and H. Ma, "Exploring background-bias for anomaly detection in surveillance videos," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1490–1499.

[202] V. Bettadapura, C. Pantofaru, and I. Essa, "Leveraging contextual cues for generating basketball highlights," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 908–917.

[203] F. C. Heilbron, W. Barrios, V. Escorcia, and B. Ghanem, "Scc: Semantic context cascade for efficient action detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3175–3184.

[204] P. Felsen, P. Agrawal, and J. Malik, "What will happen next? forecasting player moves in sports videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3342–3351.

[205] R. Kapela, K. McGuinness, A. Swietlicka, and N. E. O'Connor, "Real-time event detection in field sport videos," in *Computer vision in Sports*. Springer, 2014, pp. 293–316.

[206] A. Cioppa, A. Deliege, and M. Van Droogenbroeck, "A bottom-up approach based on semantics for the interpretation of the main camera stream in soccer games," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1765–1774.

[207] T. Tsunoda, Y. Komori, M. Matsugu, and T. Harada, "Football action recognition using hierarchical lstm," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 99–107.

[208] Z. Cai, H. Neher, K. Vats, D. A. Clausi, and J. Zelek, "Temporal hockey action recognition via pose and optical flows," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[209] M. Sanabria, F. Precioso, and T. Menguy, "A deep architecture for multimodal summarization of soccer games," in *Proceedings Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports*, 2019, pp. 16–24.

[210] P. Shukla, H. Sadana, A. Bansal, D. Verma, C. Elmadjian, B. Raman, and M. Turk, "Automatic cricket highlight generation using event-driven and excitement-based features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1800–1808.

[211] G. Tsagkatakis, M. Jaber, and P. Tsakalides, "Goal!! event detection in sports video," *Electronic Imaging*, vol. 2017, no. 16, pp. 15–20, 2017.

[212] F. Turchini, L. Seidenari, L. Galteri, A. Ferracani, G. Becchi, and A. Del Bimbo, "Flexible automatic football filming and summarization," in *Proceedings Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports*, 2019, pp. 108–114.

[213] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, "Soccernet: A scalable dataset for action spotting in soccer videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1711–1721.

[214] C.-L. Huang, H.-C. Shih, and C.-Y. Chao, "Semantic analysis of soccer video using dynamic bayesian network," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 749–760, 2006.

[215] V. Fontana, G. Singh, S. Akrigg, M. Di Maio, S. Saha, and F. Cuzzolin, "Action detection from a robot-car perspective," *arXiv preprint arXiv:1807.11332*, 2018.

[216] Y. Yao, X. Wang, M. Xu, Z. Pu, E. Atkins, and D. Crandall, "When, where, and what? a new dataset for anomaly detection in driving videos," *arXiv preprint arXiv:2004.03044*, 2020.

[217] X.-Y. Zhang, C. Li, H. Shi, X. Zhu, P. Li, and J. Dong, "Adapnet: Adaptability decomposing encoder-decoder network for weakly supervised action recognition and localization," *IEEE transactions on neural networks and learning systems*, 2020.

[218] D. Cao, L. Xu, and H. Chen, "Action recognition in untrimmed videos with composite self-attention two-stream framework," in *Asian Conference on Pattern Recognition*. Springer, 2019, pp. 27–40.

[219] H. Shi, X. Zhang, and C. Li, "Weakly-supervised action recognition and localization via knowledge transfer," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2019, pp. 205–216.

[220] L. Zhang, X. Chang, J. Liu, M. Luo, S. Wang, Z. Ge, and A. Hauptmann, "Zstad: Zero-shot temporal activity detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 879–888.

[221] H. Xu, B. Kang, X. Sun, J. Feng, K. Saenko, and T. Darrell, "Similarity r-c3d for few-shot temporal activity detection," *arXiv preprint arXiv:1812.10000*, 2018.

[222] H. Xu, X. Sun, E. Tzeng, A. Das, K. Saenko, and T. Darrell, "Revisiting few-shot activity detection with class similarity control," *arXiv preprint arXiv:2004.00137*, 2020.

[223] H. Yang, X. He, and F. Porikli, "One-shot action localization by learning sequence matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1450–1459.

[224] Y. Huang, Q. Dai, and Y. Lu, "Decoupling localization and classification in single shot temporal action detection," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 1288–1293.

[225] D. Zhang, X. Dai, and Y.-F. Wang, "Metal: Minimum effort temporal activity localization in untrimmed videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3882–3892.

[226] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[227] R. Wang and D. Tao, "Uts at activitynet 2016," *AcitivityNet Large Scale Activity Recognition Challenge*, vol. 8, p. 2016, 2016.

[228] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[229] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[230] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[231] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[232] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom sequence models for efficient action detection," in *CVPR 2011*. IEEE, 2011, pp. 3201–3208.

[233] ——, "Temporal localization of actions with actoms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2782–2795, 2013.

[234] I. Laptev and P. Pérez, "Retrieving actions in movies," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[235] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic annotation of human actions in video," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1491–1498.

Elahe Vahdani received a B.S. degree in Mathematics from Sharif University of Technology and an MPhil in Computer Science from The Graduate Center of The City University of New York (CUNY) in 2020. She is currently a Ph.D. candidate at The Graduate Center of CUNY. Her research focuses on computer vision and machine learning for human action recognition and detection in videos.

Yingli Tian (M'99–SM'01–F'18) received the B.S. and M.S. degrees from Tianjin University, China, in 1987 and 1990, and the Ph.D. degree from Chinese University of Hong Kong, Hong Kong, in 1996. After holding a faculty position at National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, she joined Carnegie Mellon University in 1998, where she was a postdoctoral fellow at the Robotics Institute. She then worked as a research staff member in IBM T. J. Watson Research Center from 2001 to 2008. She is one of the inventors of the IBM Smart Surveillance Solutions. She is currently a CUNY Distinguished Professor in the Department of Electrical Engineering at the City College and the Department of Computer Science at the Graduate Center, the City University of New York. Her research focuses on a wide range of computer vision problems from scene understanding, medical imaging analysis, human behavior analysis, to facial expression recognition and assistive technology. She is a fellow of IEEE and IAPR.

Deep Learning-based Action Detection in Untrimmed Videos: A Survey Appendix

Elahe Vahdani and Yingli Tian*, *Fellow, IEEE*

Abstract—Understanding human behavior and activity facilitates advancement of numerous real-world applications, and is critical for video analysis. Despite the progress of action recognition algorithms in trimmed videos, the majority of real-world videos are lengthy and untrimmed with sparse segments of interest. The task of temporal activity detection in untrimmed videos aims to localize the temporal boundary of actions and classify the action categories. Temporal activity detection task has been investigated in full and limited supervision settings depending on the availability of action annotations. This paper provides an extensive overview of deep learning-based algorithms to tackle temporal action detection in untrimmed videos with different supervision levels including fully-supervised, weakly-supervised, unsupervised, self-supervised, and semi-supervised. In addition, this paper reviews advances in spatio-temporal action detection where actions are localized in both temporal and spatial dimensions. Action detection in online setting is also reviewed where the goal is to detect actions in each frame without considering any future context in a live video stream. Moreover, the commonly used action detection benchmark datasets and evaluation metrics are described, and the performance of the state-of-the-art methods are compared. Finally, real-world applications of temporal action detection in untrimmed videos and a set of future directions are discussed.

1 Methods

1.1 Visual Encoder

Untrimmed videos are often lengthy and can be as long as several minutes, and thus it is difficult to directly input the entire video to a visual encoder for feature extraction due to the limits of computational resources. For instance, popular video feature extractors such as 3D-CNNs can only operate on short clips spanning about 4 seconds. A common strategy for video representation is to partition the video into equally sized temporal intervals called *snippets*, and then apply a pre-trained visual encoder over each snippet. Therefore, each video can be represented with a sequence of visual features that are further processed for action detection. Formally, given input video X with l frames, a sequence S of snippets with regular duration σ is generated. Then, each snippet is fed to a pre-trained visual encoder such as two-stream [22], C3D [23], or I3D [21] for feature extraction. In two-stream network [22], snippet s_n which is centered at t_n-th frame of the video, has an RGB frame x_{t_n}, and a stacked optical flow o_{t_n} derived around the center frame. The RGB frame is fed to spatial network ResNet [25], and the optical flow is fed to temporal network BN-Inception [226]. The extracted spatial and temporal features are concatenated to represent the visual feature f_n for snippet s_n. Similarly, in I3D [21], a stack of RGB and optical flow frames from each snippet are fed to I3D network, extracting spatial and temporal feature vectors which are then concatenated. In C3D [23], the frames of each snippet s_n are directly fed to a 3D-CNN architecture to capture spatio-temporal information.

1.1.1 Pretraining for Localization

Many methods employ features extracted from pre-trained visual encoders that are trained for trimmed action classification task. Due to the inherent discrepancy between video-level classification and clip-level localization, these features are not necessarily suitable for temporal localization. Recently, researchers have proposed pretraining for localization to learn video representations that are more transferable to action localization. Alwassel *et al.* proposed a Temporally-Sensitive Pretraining (TSP) strategy [28], and demonstrated that using features pretrained with TSP significantly improves the performance on video action localization. They trained an encoder to explicitly discriminate between foreground and background clips in untrimmed videos. Xu *et al.* proposed a boundary-sensitive pretext (BSP) task [29] to model pre-training for temporal localization. They synthesized temporal boundaries in existing video action classification datasets, and classified the boundary types in a self-supervised manner. This strategy resulted in learning video representations that are more suitable for temporal localization by capturing temporal boundary information which is required for this task. Zhang *et al.* also proposed a self-supervised pretext task to pre-train feature encoders for temporal action localization [32] in an unsupervised setting. They randomly selected temporal regions from one video and pasted them onto different temporal positions of the

- E. Vahdani is with the Department of Computer Science, The Graduate Center, The City University of New York, NY, 10016. E-mail: evahdani@gradcenter.cuny.edu
- Y. Tian is with the Department of Electrical Engineering, The City College, and the Department of Computer Science, the Graduate Center, the City University of New York, NY, 10031. E-mail:ytian@ccny.cuny.edu
*Corresponding author

This material is based upon work supported by the National Science Foundation under award number IIS-2041307.

other two videos. The pretext task is to align the features of the pasted action regions from two synthetic videos and maximized the agreement between them. Xu *et al.* proposed a low-fidelity video encoder optimization method and reduced the mini-batch composition in terms of temporal, spatial or spatio-temporal resolution [30]. Through this approach, the gradients flow backwards through the video encoder conditioned on a temporal action localization loss, resolving the task discrepancy (between localization and classification) and providing more useful feature representations. Xu *et al.* proposed a novel post-pre-training approach [31] in which the video encoder is not frozen during post-pre-training and is trained end-to-end. They also introduced a contrastive loss to capture visio-linguistic relations between activities, background clips and language during training, which leads to learning video features that are proper for temporal action localization and video language grounding.

1.2 Action Classification

For a fair comparison, researchers utilize classifiers from earlier work SCNN-classifier [18], UntrimmedNet [18], [20], etc. They uniformly sample a constant number of frames from the video segment and feed it to ConvNets such as C3D [23], two stream CNNs [22] or temporal segment networks [24]. In some cases, the recognition scores of sampled frames are aggregated with the Top-k pooling or weighted sum to yield the final prediction. The following methods use classifier of UNet [19] On THUMOS14 and classifier of Cuhk [20] on ActivityNet 1.3: BMN [41], SRG [160], G-TAD [58], BC-GNN [44], BSN++ [161], BU [162], ATAG [62], Lianli [164], Liu [171], TadTR [63]. On THUMOS14, the classifier of UNet [19] is used in BSN [40], MGG [37], DBG [42], TSA-Net [167], classifier of SCNN [18] is used in Sst [48], CTAP [45], and classifier of PGCN [56] is used in RTD-Net [61]. On ActivityNet 1.3, the classifier of Uts [227] is used in BSN [40], and classifier of UNet [19] is used in RTD-Net [61].

1.3 Temporal Action Detection with Full Supervision

In this section, we first review some of the common strategies for proposal feature extraction in anchor-based temporal action detection. Second, we briefly describe the methods that model long-range dependencies in untrimmed videos for fully-supervised action detection.

1.3.1 Anchor-based Proposal Generation and Evaluation

In anchor-based action detection methods, fixed-size features must be extracted from multi-scale proposals to evaluate the quality of temporal proposals. We provide an overview of some of the popular feature extraction strategies in this section.

RoI Pooling: R-C3D [35] extended the idea of 2D RoI pooling for object detection [97] to 3D RoI pooling to extract fixed size features from multi-scale proposals. The limitation of this approach is that the multi-scale proposals at each location share the same receptive field, which may be too small or too large for some temporal scales. Therefore, the extracted feature may not contain sufficient information or include too much irrelevant information.

Multi-tower Network: TAL-Net [13] proposed a multi-tower network, compose of several temporal convNets, each one responsible for a certain temporal scale. In this design, the receptive field of each anchor segment is aligned with its temporal span using dilated temporal convolutions. The disadvantage of this model is that is built upon multiple networks and is not computationally fast.

Temporal Feature Pyramid Network (TFPN): In TFPN, the predictions are yielded from multiple resolution feature maps. This idea was first introduced in SSD [228] for object detection, and then extended to temporal domain for action detection in SSAD [36] and S^{3}D [158]. They proposed an end-to-end network where the lower-level feature maps with higher resolution and smaller receptive field are responsible to detect short action instances while the top layers with lower resolution and larger receptive field, detect long action instances. The limitation of this approach is that lower layers in the pyramid are unaware of high-level semantic information, and top layers lack enough details, so they all fail to localize the actions accurately.

U-shaped Temporal Feature Pyramid Network (UTFPN): UTFPN was designed to connect high-level and low-level features in TFPN. It was first proposed for object detection in [229], [230], [231]. Later, it was extended to temporal domain in MGG [37] and used in [46], [47], [166], [168]. UTFPN combines high-level features with corresponding low-level features with lateral connections. The limitation of UTFPN is lack of long-range dependencies modeling in videos.

1.3.2 Modeling Long-range Dependencies

Recurrent Neural Networks (RNNs), Graph Convolution Networks, and Transformers have been employed to capture temporal and semantic dependencies between video segments in untrimmed videos. We provide a high-level overview of these methods in this section.

Recurrent Neural Networks: RNNs are used for sequence modeling and are capable of capturing long-term dependencies in videos. Sst [48] and SS-TAD [49] used RNNs for action detection. They partition the video into equal-length segments and feed each segment to a visual encoder for feature extraction. At time t, visual feature f_t and the hidden state of the previous time step (h_{t-1}) are fed to a GRU-based architecture to produce hidden state h_t. This hidden state is then used to evaluate multi-scale proposals at time t. PSDF [50] captured the motion information over multiple resolutions and utilized RNNs to improve inter-frame consistency. Yeung *et al.* learn decision policies for an RNN-based agent [51], and later proposed an LSTM model to process multiple input frames with temporal attention mechanism [52]. LSTMs are also used in other frameworks such as [53], [54], [55] to evaluate temporal proposals. The advantage of RNNs is that the hidden state encodes the information from previous time steps which is useful to capture temporal dependencies. However, RNNs are not capable to encode long videos as the hidden vector gets saturated after some time steps.

Graph Convolution Networks: A full action often consists of several sub-actions that may independently be detected in several overlapping proposals. Based on this observation, PGCN [56] captured proposal-proposal relations by applying graph-convolution networks (GCNs). They constructed a graph where the nodes are the proposals and the edges weights model the relation between the proposals. Through graph convolutions feature of each proposal gets updated by aggregating the information from other proposals. Fig. 1 shows an example of modeling proposal-proposal relations with graphs. AGCN [57] proposed an attention based GCN to model the inter and intra dependencies of the proposals. Intra attention learns the long-range dependencies among pixels inside each action proposal and inter attention learns the adaptive dependencies among the proposals to adjust the boundaries. BC-GNN [44] proposed a graph neural network to model the relations between the boundary and action content of temporal proposals. G-TAD [58] captures the relations between different snippets of input video in a graph where the nodes are temporal segments of the video and the edges model the temporal and semantic context of the snippets. VSGN [59] proposed a cross-scale graph pyramid network which aggregates features from cross scales.

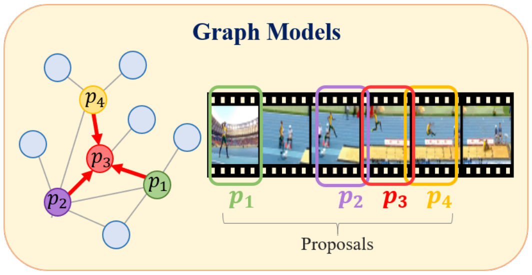

Fig. 1. Modeling proposal-proposal relations with graph networks. Proposal p_3 is influenced by proposals p_1, p_2, and p_4.

Transformers: Transformer and attention mechanism are powerful tools to capture long-range dependencies between video segments in untrimmed videos. ActionFormer [147] combined a multi-scale feature representation with local self-attention. They used a light-weighted decoder for action classification and regression. AGT [60] proposed an encoder decoder transformer to capture non-linear temporal structure by reasoning over videos as nonsequential entities. The encoder generates a context graph where the nodes are initially video level features and the interactions among nodes are modeled as learnable edge weights. The decoder learns the interactions between context graph and latent representations of the action queries. RTD-Net [61] proposed a relaxed transformer to directly generate action proposals. The transformer encoder models long-range temporal context and captures inter-proposal relationships from a global view to precisely localize action instances. They also argued that the snippet features in a video change at a very slow speed and direct employment of self-attention in transformers can lead to over-smoothing. They customized the encoder with a boundary-attentive architecture to enhance the discrimination capability of action boundary. ATAG [62] also designed an augmented transformer to mine long-range temporal context for noisy action instance localization. Despite having advantage of modeling long-range dependencies in sequential data, transformers have high parametric complexity and can cause over-fitting on small datasets.

2 Datasets

We provide a summary of benchmark datasets for action detection task, also shown in Table 1. Gaidon *et al.* [232], [233] introduced the problem of temporally localizing the actions in untrimmed videos, focusing on limited actions such as “drinking and smoking” [234] and “open door and sitdown” [235]. Later, researchers worked on building the following datasets that include large number of untrimmed videos with multiple action categories and complex background information. Some of these datasets target activities of high-level semantics (such as sports) while others include fine-grained activities (such as cooking). The details are summarized in Table 1.

- THUMOS14 [1] is the most widely used dataset for temporal action localization. There are 220 and 213 videos for training and testing with temporal annotations in 20 classes. Action instances are rather sparsely distributed through the videos and about 70% of all frames are labeled as background. The number of action instances per video on average is 15.5 (and 1.1 for distinct action instances). Also, maximum number of distinct actions per video is 3.

- MultiTHUMOS [52] has the same set of videos as in THUMOS14 [1], but it extends the latter from 20 action classes with 0.3 labels per frame to 65 classes with 1.5 labels per frame. Also, the average number of distinct action classes in a video is 10.5 (compared to 1.1 in THUMOS14), making it a more challenging multi-label dataset. Also, maximum number of distinct actions per video is 25.

- ActivityNet [131] has two versions, v1.2 and v1.3. The former contains $9,682$ videos in 100 classes, while the latter, which is a superset of v1.2 and was used in the ActivityNet Challenge 2016, contains $19,994$ videos in 200 classes. In each version, the dataset is divided into three disjoint subsets, training, validation, and testing, by 2:1:1.

- HACS [132] includes $504K$ untrimmed videos retrieved from YouTube where each one is strictly shorter than 4 minutes. HACS clips consists of $1.5M$ annotated clips of 2-second duration and HACS Segments contains $139K$ action segments densely annotated in $50K$ untrimmed videos spanning 200 action categories.

- CHARADES [133] consists of $9,848$ videos recorded by Amazon Mechanical Turk users based on provided scripts. This dataset contains videos with multiple actions and involves daily life activities from 157 classes of 267 people from three continents. Over 15% of the videos have more than one person.

- Breakfast [11] includes 1712 videos for breakfast preparation activities performed by 52 subjects. The videos were recorded in 18 different kitchens and belong to 10 different types of breakfast activities (such as fried egg or

TABLE 1
The benchmark datasets for temporal, spatio-temporal, and online action detection.

Dataset	Activities Types	#Videos	#Action Categories	Avg Video Length (Sec)	#Action Instances (avg per video)	Multi-label (#labels per frame)
THUMOS [1]	Sports	413	20	212	15.5	No
MultiTHUMOS [52]	Sports	413	65	212	97	Yes
Breakfast [11]	Cooking	1712	48	162	6	No
50Salads [129]	Cooking	50	17	384	20	No
MPII cooking 2 [130]	Cooking	273	59	356	51.6	No
Ava [111]	Movies	437	80	900	3361.5	Yes
TVSeries [121]	TV series	27	30	2133.3	231	Yes
ActivityNet [131]	Daily Activities	19,994	200	115	1.54	No
HACS Segment [132]	Daily Activities	50K	200	156	2.8	No
Charades [133]	Daily Activities	9,848	157	30	6.75	Yes
COIN [134]	Daily Activities	11,827	180	142	3.9	No
FineAction [135]	Daily Activities	17K	106	-	6	Yes

coffee) which consist of 48 different fine-grained actions. Each video contains 6 action instances on average and only 7% of the frames are background.

• 50Salads [129] contains 50 videos for salad preparation activities performed by 25 subjects and with 17 distinct action classes. On average, each video contains 20 action instances and is 6.4 minutes long.

• MPII Cooking 2 [130] consists of 273 videos with about 2.8 million frames. There are 59 action classes and about 29% of the frames are background. The dataset provides a fixed split into a train and test set, separating 220 videos for training.

• COIN dataset [134], [188] contains 180 tasks and $11,827$ videos and $46,354$ annotated segments. The videos are collected from YouTube in 12 domains (e.g., vehicles, gadgets, etc.) related to daily activities.

• AVA [111] is designed for spatio-temporal action detection and consists of 437 videos where each video is a 15 minute segment taken from a movie. Each person appearing in a test video must be detected in each frame and the multi-label actions of the detected person must be predicted correctly. The action label space contains 80 atomic action classes but often the results are reported on the most frequent 60 classes.

• TVSeries [121] contains 27 episodes of 6 popular TV series, totaling 16 hours of video. The dataset is annotated with 30 realistic, everyday actions (e.g., open door). 6,231 action instances. There are multiple actors, and everyone does an action his or her way. Different actions can occur at the same time, being performed by the same or multiple actors. Also, the way the action is recorded can be very different. The viewpoint is not fixed and part of the action can be occluded.

• FineAction [135] includes 103K instances of 106 action categories, annotated in 17K untrimmed videos. The action categories are collected from the existing benchmarks for video collection and annotation such as ActivityNet [131] and Kinetics [21], and contain a wide range from sports to daily activities. Several finegrained actions can happen simultaneously and 11.5% of temporal segments have multiple action labels with overlaps.