What do I Annotate Next? An Empirical Study of Active Learning for Action Localization

Fabian Caba Heilbron[1]*, Joon-Young Lee[2], Hailin Jin[2], and Bernard Ghanem[1]

[1] King Abdullah University of Science and Technology (KAUST), Saudi Arabia
[2] Adobe Research, San Jose, CA, USA
{fabian.caba,bernard.ghanem}@kaust.edu.sa
{jolee,hljin}@adobe.com
https://cabaf.github.io/what-to-annotate-next

Abstract. Despite tremendous progress achieved in temporal action localization, state-of-the-art methods still struggle to train accurate models when annotated data is scarce. In this paper, we introduce a novel active learning framework for temporal localization that aims to mitigate this data dependency issue. We equip our framework with active selection functions that can *reuse knowledge* from previously annotated datasets. We study the performance of two state-of-the-art active selection functions as well as two widely used active learning baselines. To validate the effectiveness of each one of these selection functions, we conduct simulated experiments on ActivityNet. We find that using previously acquired knowledge as a bootstrapping source is crucial for active learners aiming to localize actions. When equipped with the right selection function, our proposed framework exhibits significantly better performance than standard active learning strategies, such as uncertainty sampling. Finally, we employ our framework to augment the newly compiled Kinetics action dataset with ground-truth temporal annotations. As a result, we collect *Kinetics-Localization*, a novel large-scale dataset for temporal action localization, which contains more than 15K YouTube videos.

Keywords: Video Understanding · Temporal Action Localization · Active Learning · Video Annotation

1 Introduction

Video data arguably dominates the largest portion of internet content. With more than 74% of total internet traffic being video [15], a need that arises is to automatically understand and index such massive amounts of data. The computer vision community has embraced this problem, and during the last decade, several approaches for video analysis have been proposed [8,26,31,39,41,48,52,58,76]. One of the most challenging tasks in this field, which has recently gained much attention, is to understand and temporally localize human actions in untrimmed videos. Such a task, which is widely known as temporal action localization, aims to produce temporal bounds in a video, during which human actions occur.

*Work done during internship at Adobe Research

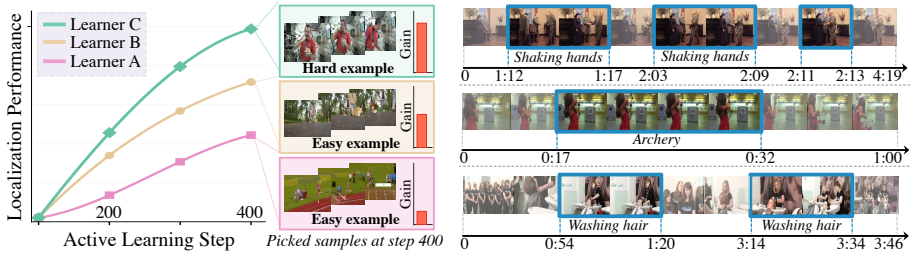

Fig. 1: **Active Learning for Action Localization.** We compare three different active learners for temporal action localization. We plot the localization performance (mAP) of each learner at different active learning steps. Each learner's aim is to use the least number of training samples as possible, which are obtained sequentially by annotating samples from an unlabeled set. The proposed method resembles Learner C, which minimizes the number of active learning steps to reach a target performance. Using our active learner, we construct Kinetics-Localization, a novel and large-scale dataset for temporal action localization.

Datasets such as Thumos14 [35], ActivityNet [8], and Charades [58] have enabled the development of innovative approaches addressing the temporal action localization problem [50,56,71,75,77]. These approaches have been successful in increasing localization performance while maintaining a low computational footprint [5,71]. For instance, current state-of-the-art approaches [44,77] have improved more than three times the first reported performance on datasets like Thumos14 and ActivityNet. However, despite those great achievements, a crucial limitation persists, namely the dependence of these models on large-scale annotated data for training. This limitation often prevents the deployment of action localization methods at scale, due to the large costs associated with video labeling (*e.g.* Charades authors [58] spent $1 per video).

Additionally, given that datasets for temporal action localization are relatively small, it is unclear whether existing methods will reach performances like the ones obtained in other vision tasks such as object detection [54]. To overcome some of these issues, Wang *et al.* [68] propose a new model that uses video-level annotations combined with an attention mechanism to pinpoint actions temporal bounds. Although it does not require temporal ground-truth, their performance is significantly lower than that achieved by fully-supervised approaches, thus, restricting its applications that do not require accurate detection.

In this paper, we propose an active learning method that aims to ease the large-scale data dependence of current temporal localization methods. As in every active learning setting [55], our goal is to develop a learner that selects samples (videos in this case) from unlabeled sets to be annotated by an oracle. As compared to traditional active learners [27,42] where heuristics such as uncertainty sampling are used to perform the sample selection, we explore novel selection functions [25,40] that reuse knowledge from a previously existing dataset. For instance, we study a learnable selection function that learns a

mapping from a model-sample state pair to an expected improvement in performance. In doing so, such function learns to score the unlabelled samples based on the expected performance gain they are likely to produce if they are annotated and used to update the current version of the localization model being trained.

Figure 1 depicts the learning process of three different action localization strategies. To evaluate each learner, we measure the performance improvements, which are assessed on a labeled set, at different training dataset sizes (or learning stages). We associate traditional action localization approaches [5,71,77] to Learner A (passive learning), which randomly picks samples to be annotated for future training iterations. Learner A exhibits passive behavior in making smart selections of samples to augment its training set. Learner B is an active learner that uses uncertainty sampling [42] to select the samples (the learner chooses instances whose labels are most uncertain). Learner C is a learning-based active learner. Because it incorporates historical knowledge from previous dataset selections, Learner C enables a better learning process. In this paper, we introduce an active learning framework that minimizes the number of active learning steps required to reach the desired performance.

Contributions. The core idea of the paper is to develop an active learning framework for temporal action localization. Specifically, the contributions of this paper are threefold. **(1)** We introduce a new active learner for action localization (see Section 3). To develop our approach, we thoughtfully study different sampling functions, including those that can exploit previously labeled data to learn or bootstrap a selection function that chooses unlabelled samples with the aim of improving the localization model the most. **(2)** We conduct extensive experiments in Section 4 demonstrating the capabilities of the proposed framework. When compared to traditional learning (random sampling), our approach learns to detect actions significantly quicker. Additionally, we show that our active learner can be employed in batch-mode, and is robust to noisy ground-truth annotations. **(3)** We employ our active learner to construct a novel dataset for temporal action localization (see Section 5). Using videos from the Kinetics [39] dataset, we apply our learner to request temporal annotations from Amazon Mechanical Turk workers. We name this collected dataset *Kinetics-Localization* and it comprises more than $15K$ YouTube videos.

2 Related Work

This section briefly discusses the most relevant work to ours, namely those related to active learning and temporal action localization.

Active Learning tackles the problem of selecting samples from unlabeled sets to be annotated by an oracle. In the last decade, several active learning strategies have been proposed [27,42,63] and applied to several research fields, including speech recognition [32], natural language processing [62], chemistry [18], just to name a few. We refer the reader to the survey of Settles [55] for an extensive review of active learning methods. Active learning has also been used in traditional

computer vision tasks, such as image classification [4,22,25,36,37,53] and object detection [64], or to construct large-scale image and video datasets [16,66,72]. Very recently, active learning approaches have emerged in more contemporary vision tasks, including human pose estimation [46] and visual question answering [45]. Most of the active learning approaches in computer vision have used the simple but effective uncertainty sampling query strategy [42,43], where unlabelled samples are selected based on the entropy of their scores generated by the current discriminative model (least confidence and margin based score selections are other popular query strategies). However, the main limitation of this strategy is its inability to handle complex scenarios where factors such as label noise, outliers, or shift in data distribution arise in the active learning setting [40]. Inspired by very recent ideas in active learning [1,25,40,70,74], our proposed active learning framework learns (or bootstrap) a function that selects samples for annotation based on knowledge extracted from a previous dataset. One variant of our approach estimates the effect of labeling a particular instance on the performance of the current discriminative model. As such, this learnable function is able to overcome the shortcomings of heuristic active learners, such as uncertainty sampling (see Section 4).

Temporal Action Localization. Many techniques have been developed over the years to recognize [11,12,49,59,67], and localize human activities, either in images [28,47,73] or videos [29,34,69]. Our work focuses on the temporal action localization problem in video, whose goal is to provide starting and ending times of an action occurring within an untrimmed video. Researchers have explored innovative ideas to efficiently and accurately address this problem. Earlier methods rely on applying action classifiers in a sliding window fashion [19,23,50]. To unburden the computational requirements of sliding windows, a new line of work studies the use of action proposals to quickly scan a video in an attempt to reduce the search space [6,7,10,20,24,56]. More recently, end-to-end approaches have boosted the performance of stage-wise methods, demonstrating the importance of jointly optimizing classifiers and feature extractors [13,71,75,77].

Despite the large body of work on action localization, most methods focus on either improving performance [77] or boosting speed [5], while very few investigate the use of active learning to mitigate the data dependency problem. To the best of our knowledge, only the work of Bandla and Grauman [2] has incorporated active learning to train an action detection model. However, their method relies on hand-crafted active selection functions such as uncertainty sampling [42], which works well in controlled scenarios where statistical properties of the dataset can be inferred. However, it fails when more complex shifts in data distribution are present. In contrast and inspired by recent works [25,40], our approach avoids predefined heuristics and instead *learns* or *bootstraps* the active selection function from existing data. We will show that learning such a function not only improves the learning process of an action localization model on a given dataset, but it is also adaptable for use when annotating new data.

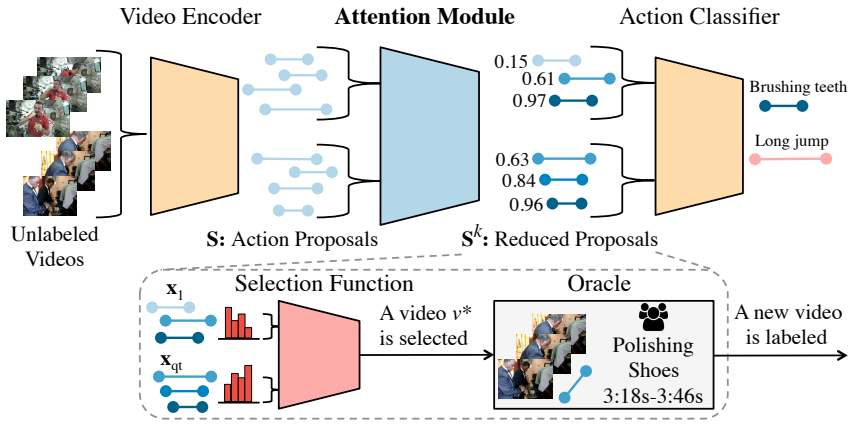

Fig. 2: **Active Learner for Temporal Action Localization.** Firstly, we train an action localization model with a labeled set of videos. Then, using the trained model, we generate video predictions in an unlabeled set and select one of the videos that is expected to improve the learner the most. Finally, an oracle temporally annotates the selected video and then added into the labeled set.

3 Active Learner for Temporal Action Localization

We propose an active learning framework for temporal action localization. Our goal is to train accurate detection models using a reduced amount of labeled data. At every learning step t, a set of labeled samples $\mathcal{L}_t$ is first used to train a model f_t. Then, from an unlabeled pool $\mathcal{U}_t$, a video instance v^* is chosen by a selection function g. Afterwards, an oracle provides temporal ground-truth for the selected instance, and the labeled set $\mathcal{L}_t$ is augmented with this new annotation. This process repeats until the desired performance is reached or the set $\mathcal{U}_t$ is empty. As emphasized in previous work [37,46], the key challenge in active learning is to design the proper selection function, which seeks to minimize the number of times an oracle is queried to reach a target performance. Accordingly, we empower our proposed framework with state-of-the-art selection functions that exploit previously labeled datasets as bootstrapping.

This section provides a complete walk-through of our approach (see Figure 2). We describe our model for temporal action localization, elaborate on our proposed active selection function, and explain in detail the oracle's task.

3.1 Localization Model Training Step

Much progress has been made in designing accurate action detection models [5,24,71,77]. So ideally, any of these detectors can be used here. These detectors can be grouped into two categories, namely, stage-wise and end-to-end models. Models trained end-to-end have shown superior detection rates. However, such methods cannot decompose the localization problem into simpler tasks. We argue

that decomposing the action localization task is key, specially for active learning methods that use previous knowledge to bootstrap the selection function learning process. As such, we opt for designing a stage-wise action localization model.

Our model takes as input a video v described by a set of n temporal segments, denoted by $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_n\}$ where $\mathbf{s_i} = [t^{start}, t^{end}]$ is a 2D vector containing starting and ending times of a segment. In this paper, these temporal segments are action proposals generated by DAPs [20]. Our localization model's goal is to select k temporal segments $\mathbf{S}^k$ from the initial set $\mathbf{S}$ and produce a vector of confidence scores $\mathbf{z}_c \in \mathbb{R}^k$ for each action class c in the dataset. In short, our model maps an input video described by a large set of candidate segments into a small set of temporal predictions: $f_t(v, \mathbf{S}) \rightarrow \left\{\mathbf{S}^k, \{\mathbf{z}_c\}_{c\in\mathcal{C}}\right\}$ where $\mathcal{C}$ is the set of action classes.

To that end, we organize our model into three modules: a *video encoder* whose goal is to describe temporal segments $\mathbf{S}$ in terms of a feature vector $\mathbf{o}$, an *attention module* which picks k segments based on a binary action classifier h_t, and an *action classifier* $\phi(\mathbf{S}^k)$ which generates the confidence scores $\mathbf{z}_c$ for each class in $\mathcal{C}$. Below, we provide design details for each component.

Video Encoder. Given a set of temporal segments $\mathbf{S}$, our aim is to encode each individual segment $\mathbf{s}_i$ with a compact representation. We first extract frame-level features using a CNN and then aggregate these representations into a single feature vector $\mathbf{o_i}$. In our experiments, we train an Inception V3 network [61] using the Kinetics dataset [39] and extract features from the *pool3* layer (a feature vector with 2048 dimensions). To reduce the temporal receptive field, we opt for average pooling, which beyond its simplicity has demonstrated competitive performance in various tasks [38,60]. Thus, our video encoder generates a matrix of visual observations, $\mathbf{O} = [\mathbf{o}_1 \cdots \mathbf{o}_n] \in \mathbb{R}^{2048\times n}$.

Attention Module. This module receives a visual observation matrix $\mathbf{O}$ to pick k temporal segments $\mathbf{S}^k$ which are most likely to contain an action. We adopt a linear Support Vector Machine (SVM) [17,21] to learn a binary classifier that discriminates between actions and background. We employ Platt scaling [51] to obtain probabilistic scores from the SVM outputs. Finally, to select the output segments, we perform hard attention pooling and pick the *top-k* segments with high confidence scores. We set $k = 10$ in our experiments. Accordingly, our attention module h_t outputs a small number of segments $\mathbf{S}^k$, which are encoded with their corresponding visual representations in $\mathbf{O}$.

Action Classifier. Taking as input a reduced set of temporal segments $\mathbf{S}^k$, the action classifier aims to generate a set of confidence scores $\mathbf{z}_c$ for each action category in $\mathcal{C}$. Consciously, we build a model composed of a fully-connected layer and a soft-max classifier. Thus, our action classifier ϕ generates the final detection results $\left\{\mathbf{S}^k, \{\mathbf{z}_c\}_{c\in\mathcal{C}}\right\}$.

Training. We define the labeled set at learning step t of size p_t as $\mathcal{L}_t = \left\{ (v_1^{train}, \mathbf{y}_1), (v_2^{train}, \mathbf{y}_2), \cdots (v_{p_t}^{train}, \mathbf{y}_{p_t})\right\}$, where $\mathbf{Y} = [\mathbf{y}_1|\cdots|\mathbf{y}_{p_t}] \in \mathbb{R}^{2\times p_t}$

contains the temporal annotations of all action instances. We also define the set of temporal segments of size m as $\mathbf{S}_i^{(t)} = \{\mathbf{s}_1^{train}, \cdots, \mathbf{s}_m^{train}\}$, where $i \in \{1, 2, ..., p_t\}$ describes each video. We train our attention and action classifier modules separately. To train the attention module, we define instances in $\mathbf{S}_i^{(t)}$ as positives if the temporal Intersection over Union (tIoU) with any ground-truth instance is greater than 0.7. Similarly, for training the action classifier, we use temporal instances with tIoU greater than 0.7, but considering only the *top-k* segments chosen by our attention module.

3.2 Active Selection Step

Our aim is to design a selection function g that picks an instance v^* from the unlabeled set $\mathcal{U}_t$. Our primary challenge is to develop this function such that it selects the samples that are expected to improve the localization model the most. Additionally, we want the selection function to generalize to unseen action categories. Purposefully, instead of sampling directly from the f_t predictions, we cast the selection problem into a meta-learning task; *pick samples that improve attention module h_t the most*. Here, we focus the learner on the attention module as opposed to the action classifier, since the former deals with a more complex task (temporal boundary generation) and its output directly impacts the latter. Formally, our learnable selector g takes as input confidence scores produced by the action classifier h_t when applied to the unlabeled set $\mathcal{U}_t$: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{q_t}]$ where $\mathbf{X} \in \mathbb{R}^{l \times q_t}$ with l being the number of temporal segments and q_t the number of unlabeled videos. In this section, we introduce three different sampling functions, which are studied and diagnosed in Section 4.

Learning Active Learning (LAL). Here, we follow [40] and formulate the learning of the selection function as a regression problem, which predicts the improvement in performance of our attention module for all samples belonging to $\mathcal{U}_t$. We construct a feature matrix $\mathbf{F}$ from pairs of model state and sample description. We choose the model state to be the SVM weights defining h_t and the sample description to be the histogram of confidence scores in $\mathbf{X}$. The target vector used for regression is η, which corresponds to the improvement δ in localization performance (in practice mean Average Precision) after the model h_t is trained with each of the samples in a Set of previously labeled examples $\mathcal{K}_t$ individually. In our experiments, we refer to $\mathcal{K}_t$ as the Knowledge-Source Set. To generate a matrix $\mathbf{F}$ that explores enough pairs of model and sample states, we follow the Monte-Carlo procedure used in [40]. Once matrix $\mathbf{F}$ and targets η are constructed, we learn g using Support Vector Regression (SVR). Once trained, we can apply g to the unlabelled set to select the sample with the highest predicted performance improvement: $g(\mathcal{U}_t) \rightarrow v^*$.

Maximum Conflict Label Equality (MCLE). This method leverages knowledge from past existing datasets. We closely follow [25] and devise a method that uses zero-shot learning as warm initialization for active learning. We opt

for simplicity and implement a Video Search zero-shot learning approach, which uses top results from YouTube search as positive samples [14]. This approach's implementation is based on the code provided by [25].

Uncertainty Sampling (US). This baseline samples videos with the most uncertain predictions. Following standard uncertainty sampling approaches [42], we compute the entropy of the video predictions (*i.e.* the histogram of confidence scores in the columns of $\mathbf{X}$) and select the one with highest entropy value. This baseline is popularly used in computer vision applications such as image classification [53] or human pose estimation [46].

3.3 Annotation Step

The oracle's task is to annotate videos chosen by the selection function g. Specifically, the oracle is asked to provide temporal bounds of all instances of an intended action. Towards this goal, several researchers have proposed efficient strategies to collect such annotations [9,57]. Most of them have focused their approaches to exploit crowd-sourcing throughput and have used Amazon Mechanical Turk to annotate their large-scale video datasets. In this work, we experiment with two type of oracles: (i) simulated ones, which we emulate by using the ground-truth from existing and completely annotated datasets, and (ii) real human annotators, who are Amazon Mechanical Turk workers. We observe that the proposed framework is indiscriminately good in both cases.

4 Diagnostic Experiments

To evaluate our framework, we analyze its performance, including all its variants of selection functions, when oracles are simulated, *i.e.* we emulate an oracle's outcome by using the ground-truth from existing datasets that have already been completely annotated.

4.1 Experimental Settings

Dataset. We choose ActivityNet [8], the largest available dataset for temporal action localization, to conduct the diagnostic experiments in this section. Specifically, we use the training and validation sets of ActivityNet 1.3, which include 14950 videos from 200 activity classes.

Metrics. We use the mean Average Precision (mAP) metric to assess the performance of an action localization model. Following the standard evaluation of ActivityNet, we report mAP averaged in a range of tIoU thresholds, *i.e.* from 0.5 to 0.95 with an increment of 0.05. To quantify the merits of a sampling function, we are particularly interested in observing the rate of increase of mAP with increasing training set size (*i.e.* increasing percentage of the dataset used to train the localization model).

Setup. LAL and MCLE approaches (introduced in Section 3.2) leverage knowledge extracted from previous datasets to bootstrap the selection function learning process. To exploit each of these methods to their full extent, we extract two category-disjoint subsets from ActivityNet. The first subset, dubbed KNOWLEDGE-SOURCE, contains 2790 videos from 50 action categories. This subset is used to bootstrap the LAL and MCLE sampling functions. The second subset, dubbed ACTIVITYNET-SELECTION, consists of 11160 videos with 150 action categories, which do not overlap with the ones in KNOWLEDGE-SOURCE. We mainly conduct the active learning experiments on ACTIVITYNET-SELECTION. Additionally, to measure the performance of the localization model, we define a TESTING SET, which contains 3724 unseen videos from the same 150 categories as ACTIVITYNET-SELECTION. The TESTING SET videos do not overlap with ACTIVITYNET-SELECTION nor KNOWLEDGE-SOURCE videos.

We use the following protocol in our diagnostic experiments. We bootstrap LAL and MCLE using the labeled data in KNOWLEDGE-SOURCE by following the method described in Section 3.2. Note that US does not need previous knowledge to operate. Once the selection function is available, we randomly select 10% from ACTIVITYNET-SELECTION as a training set to build an initial action localization model (refer to Section 3.1). Then, we evaluate the model's mAP performance on the TESTING SET, and we apply our active learner onto the remaining videos of ACTIVITYNET-SELECTION to select one or more of them, which will be annotated in the next step. Subsequently, we probe the oracle, which is simulated in this case by using the ground-truth directly provided by ACTIVITYNET-SELECTION, to obtain temporal annotations for the selected videos. Finally, we augment the training set with the newly annotated samples, which in turn are used to re-train the localization model. This sequential process repeats until we have used 100% of the videos in ACTIVITYNET-SELECTION for training.

4.2 Selection Function Ablation Study

Comparison under Controlled settings. Figure 3 (**Left**) compares mAP performance between the three selection functions introduced in Section 3.2 on the TESTING SET. We also report the performance of a Random Sampling baseline for reference. We report how the mAP of the localization model increases with the increase in training data, which is iteratively sampled according to the three active learning methods. These results help us investigate the effectiveness of each method in terms of how much improvement is obtained by adding a certain amount of training data. It is clear that LAL and MCLE significantly outperform US and the random sampling baseline. For example, to achieve 80% of the final mAP (*i.e.* when all of ACTIVITYNET-SELECTION is used for training), LAL and MCLE require only 35% and 38% of the training data to be labelled respectively, while Uncertainty and Random Selection need 42% and 65% respectively to achieve the same performance. We attribute the superiority of LAL and MCLE to the fact that both approaches reuse information from labeled classes in the KNOWLEDGE-SOURCE Set. Additionally, LAL directly exploits the current state of the localization model to make its selection at every

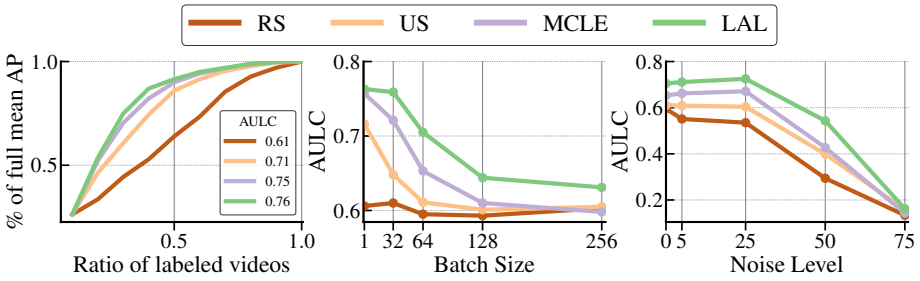

Fig. 3: **Selection Function Ablation Study. Left.** We show the % of full mAP (full training) achieved at different ratios of labeled videos. We report the Area Under the Learning Curve (AULC) for each sampling function. LAL and MCLE present steeper increases on mAP. **Center.** We report the AULC at different batch sizes. LAL is robust to large batch sizes. **Right.** We compute AULC against different level of noise from oracle annotations. All methods tolerate small levels of noise.

training step. As such, it has inherently broader knowledge about the dataset it is annotating as compared to the simple heuristics used by Uncertainty Selection.

Effect of Sampling Batch Size. Re-training a model whenever a single new sample is made available is prohibitively expensive. To alleviate this problem, researchers often consider active learning in batch-mode [3]. In batch-mode, our active learner selects groups of samples instead of just one. For LAL, we simply rank all the unlabelled samples and pick the top scoring ones based on LAL's predictions (*i.e.* the performance gain they are expected to cause when they are individually added to the training). For MCLE and Uncertainty Sampling, we select one unlabeled instance at a time until we completely fill the batch that will be annotated by the oracle. Figure 3 (**Center**) shows the Area Under the Learning Curve for different sampling batch sizes. The AULC value summarizes the performance of an active learner by computing the area under the *"percentage of full mAP vs ratio of labeled videos"* curve. For reference, we include the performance when using a single selection (*i.e.* batch size of 1). Uncertainty Sampling performance is poor after increasing the sampling batch size to 32. Interestingly, MCLE performance is strongly degraded at larger sampling batch sizes. The AULC score jumps from 0.75% down to 0.65% when the batch size is set to 64. On the other hand, we observe that LAL is relatively robust to larger sampling batch sizes. For instance, for a batch of size 64, the AULC drops only 0.05. We attribute the robustness of LAL to the fact that it estimates the selection score of each sample independently. Motivated by a trade-off between computational footprint and performance, we fix the selection batch size to 64 for the remaining experiments.

Effect of Noisy Annotations. Here, we analyze the performance of the selection functions when exposed to noisy oracles. To evaluate robustness against noisy annotations, we measure the performance of our active learner when different levels of noise are injected into the oracle responses. We quantify the noise in terms of how much an oracle response differs, in tIoU, from the original ground-truth. For example, at 5% noise level, the oracle returns temporal instances sampled from a Gaussian distribution with mean equal to 95% tIoU.

Similar to previous analysis, Figure 3 (**Right**) reports the AULC at different noise levels. We observe that all sampling functions tolerate high levels of noise and in some cases (LAL) their performance can even improve when small (5%) noise levels are added. We conjecture that this improvement is due to the fact that such small levels of noise can be seen as adversarial examples, which previous works have demonstrated to be beneficial for training [30].

5 Online Experiments: Collecting Kinetics-Localization

In this section, we perform live experiments, where we employ our active learner to build a new dataset. Instead of collecting the dataset from scratch, we exploit Kinetics [39] videos (and its video-level labels) and enrich them with temporally localized annotations for actions. We call our novel dataset Kinetics-Localization. First, we run our active learner to collect temporal annotations from Kinetics videos. Then, we present statistics of the collected data. Finally, we evaluate the performance of models trained with the collected data.

5.1 Active Annotation Pipeline

The Kinetics dataset [39] is one of the largest available datasets for action recognition. To construct the dataset, the authors used Amazon Mechanical Turk (AMT) to decide whether a 10 seconds clip contains a target action. To gather the pool of clips to be annotated, first a large set of videos are obtained by matching YouTube titles with action names. Then, a classifier, which is trained with images returned by Google Image Search, decides where the 10 seconds clip to be annotated is extracted from. As a result, Kinetics provides more than $300K$ videos among 400 different action labels. There is only one annotated action clip in each video. The scale of the dataset has enabled the development of novel neural network architectures for video [12]. Unfortunately, despite the tremendous effort needed to build Kinetics, the dataset is not designed for the task of temporal action localization. Thus, we commit our active learner to collect temporal annotations for a portion of Kinetics.

We employ our active learner to gather temporal annotations for Kinetics videos from 75 action classes. It needs to select samples that will be annotated online by real human oracles. Following standard procedure for temporal video annotation, we design a user interface that allows people to determine the temporal bounds of actions in videos [9,57,65]. We rely on Amazon Mechanical Turk workers (turkers) to annotate the videos. Snapshots of the user interface and details about the annotation process are available in the *supplementary material*.

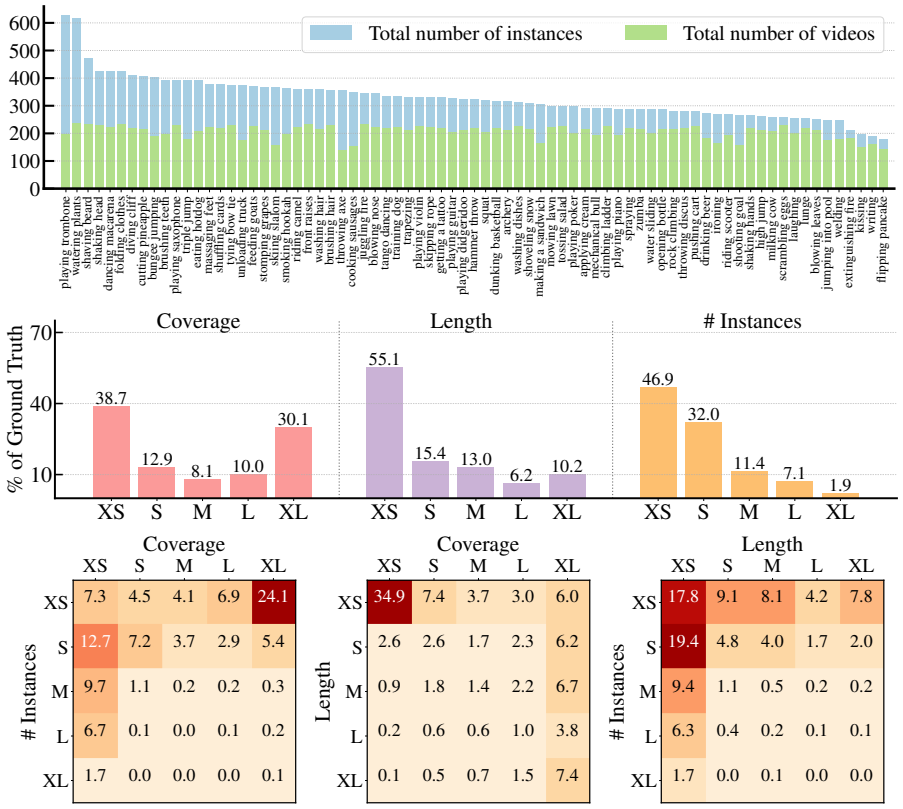

Fig. 4: **Kinetics-Localization at a Glance.** We introduce Kinetics-Localization, a novel dataset for temporal action localization. **Top:** Distribution of number of videos and instances per class. **Middle.** Kinetics-Localization attributes. We show the distribution of ground-truth instances for different attributes including Coverage, Length, and Number of Instances per video. **Bottom.** We analyze the distribution of ground-truth instances for pairwise interactions of attributes. Each bin reports the percentage of ground-truth that belongs to such bin.

5.2 Kinetics-Localization at a Glance

As a result of our annotation campaign, we effectively compile a temporal action localization dataset comprising 15000 videos from 75 different action categories, resulting in more than 30000 temporal annotations. Figure 4 summarizes Kinetics-Localization properties. Figure 4 **(Top)** shows the number of videos and instances per class in the current version of the dataset. The distribution of number of videos/instances is close to uniform. Also notice that the ratio of instances per video is 2.2.

Figure 4 **(Middle)** shows the ground-truth distribution for three different inherent attributes of the dataset. (i) Coverage, which we measure as the fraction between an instance's length and the duration of the video it belongs to. We group instance coverage into five groups: Extra Small (XS: $(0, 0.2]$); Small (S: $(0.2, 0.4]$); Medium (M: $(0.4, 0.6]$); Large (L: $(0.6, 0.8]$); Extra Large (XL: $(0.8, 1.0]$). (ii) Length, measured as the duration, in seconds, of an instance. We define five bins to plot the distribution of this attribute: Extra Small (XS: $(0, 30]$), Small (S: $(30, 60]$), Medium (M: $(60, 120]$), Large (L: $(120, 180]$), and Extra Large (XL: > 180). (iii) Number of instances in a video (# instances), which we cluster into five bins as well: Extra Small (XS: $[0, 1]$); Small (S: $(1, 4]$); Medium (M: $(4, 8]$); Large (L: $(8, 16]$); Extra Large (XL: > 16). In terms of coverage, extra small and extra large instances have a large portion of ground-truth instances assigned. Also note that more than half of the instances have at most small coverage (< 0.4). The dataset comprises 55.1% of instances that are relatively small. We hypothesize that such small instances will enable new challenges, as is the case in other fields such as face detection [33].

We also study the distribution between pairs of instance attributes (see Figure 4 **(Bottom)**). We observe three major trends from the ground-truth distribution: (i) as expected, instances with high coverage tend to have no neighbours (single instance per video); (ii) 34.9% of instances have extra small coverage and extra small length, which we argue may be the hardest type of sample for current detectors; (iii) In summary, we find that the dataset exhibits challenging types of ground-truth instances, which may span ranges of difficulty.

5.3 Kinetics-Localization Benchmark

We evaluate two different temporal action localization models: (i) our temporal localization model (Stage-Wise), which we introduced in Section 3.1; (ii) the Structured Segmented Network (SSN) introduced by Zhao *et al.* [77] (we refer to this approach as End-to-End). Although we could have employed other action detectors such as [5,71], we choose SSN because it registers state-of-the-art performance. We train each of the models either using Kinetics-Localization or the original Kinetics dataset. Table 1 summarizes the results. We use the provided 10 second clips to train the action localization models, and assume that all remaining content in the video is background information. Even though background might also contain some valid action instances, we argue there is no systematic way to add those for training without fully annotating them.

To properly quantify performance, we fully annotate a portion of the Kinetics validation subset with temporal annotations, which we refer from now on as Kinetics-Localization Validation Set. Table 1 shows the temporal localization performance of both approaches at different tIoU thresholds on the Kinetics-Localization Validation Set. We observe that the performance at lower tIoU thresholds (*e.g.* 0.1) for both models is close to the achieved performance of previous work on the trimmed classification task [12]. However, when the tIoU threshold is increased to 0.2, the performance drastically drops. For example, the mAP of the End-to-End SSN model (trained on the original Kinetics) decreases

	mAP (%) at tIoU									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg.
Kinetics-Localization										
Stage-Wise	72.1	59.2	52.8	48.7	45.1	31.0	26.4	17.7	3.9	21.3
End-to-End	72.8	61.3	54.9	52.3	49.6	32.7	28.2	19.5	5.2	22.8
Kinetics [39]										
Stage-Wise	43.2	34.7	22.8	15.1	13.7	11.0	8.9	5.7	2.9	8.2
End-to-End	59.4	40.1	28.3	20.8	15.0	11.8	9.4	5.2	1.2	8.3

Table 1: **Kinetics-Localization benchmark.** We report the mAP at different tIoU thresholds of the Stage-Wise and End-to-End models. We averaged mAP in a range of tIoU thresholds, *i.e.* from 0.5 to 0.95 with an increment of 0.05 (Avg. mAP). Notably, training with Kinetics-Localization dataset offers significant gains in performance as compared to using the original Kinetics dataset.

from 59.4% to 40.1%. Also, once typical tIoU thresholds for localization are used (0.5 to 0.9), both approaches perform poorly. We attribute this behavior to the fact that Kinetics does not include accurate temporal action bounds, thus, preventing the localization models to reason about temporal configurations of actions. When comparing the performance of the Stage-Wise approach to that of the same model trained with the newly collected Kinetics-Localization data, an improvement of 13.1% mAP is unlocked on the validation set. This validates the need for accurate temporal annotations to train localization models as well as the need for cost effective frameworks to collect these annotations (like the the active learner method we propose in this paper).

6 Conclusion

We introduced a novel active learning framework for temporal action localization. Towards this goal, we explored several state-of-the-art active selection functions and systematically analyzed their performance. We showed that our framework outperforms baseline approaches when the evaluation is conducted with simulated oracles. We also observed interesting properties of our framework when equipped with its LAL variant; (1) it exhibited good performance in batch-mode, and (2) is robust to noisy oracles. After validating the contributions of our active learner, we employed it to gather a novel dataset for temporal localization, which we called Kinetics-Localization. We presented statistics of the datasets as well as a novel established benchmark for temporal action localization. We hope that the collected Kinetics-Localization dataset helps to encourage the design of novel methods for action localization.

Acknowledgments. This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2017-3405.

References

1. Bachman, P., Sordoni, A., Trischler, A.: Learning Algorithms for Active Learning. arXiv preprint arXiv:1708.00088 (2017)
2. Bandla, S., Grauman, K.: Active learning of an action detector from untrimmed videos. In: ICCV (2013)
3. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: International Conference on Machine Learning (ICML) (2003)
4. Bruzzone, L., Marconcini, M.: Domain adaptation problems: A dasvm classification technique and a circular validation strategy. PAMI **32**(5), 770–787 (2010)
5. Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.C.: End-to-end, single-stream temporal action detection in untrimmed videos. In: BMVC (2017)
6. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C.: Sst: Single-stream temporal action proposals. In: CVPR (2017)
7. Caba Heilbron, F., Barrios, W., Escorcia, V., Ghanem, B.: Scc: Semantic context cascade for efficient action detection. In: CVPR (2017)
8. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR (2015)
9. Caba Heilbron, F., Niebles, J.C.: Collecting and annotating human activities in web videos. In: Proceedings of International Conference on Multimedia Retrieval (ICMR) (2014)
10. Caba Heilbron, F., Niebles, J.C., Ghanem, B.: Fast temporal activity proposals for efficitent detection of human actions in untrimmed videos. In: CVPR (2016)
11. Caba Heilbron, F., Thabet, A., Niebles, J.C., Ghanem, B.: Camera motion and surrounding scene appearance as context for action recognition. In: ACCV (2014)
12. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
13. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: CVPR (2018)
14. Chen, X., Shrivastava, A., Gupta, A.: Neil: Extracting visual knowledge from web data. In: ICCV (2013)
15. Cisco: The zettabyte era: Trends and analysis. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html (2017)
16. Collins, B., Deng, J., Li, K., Fei-Fei, L.: Towards scalable dataset construction: An active learning approach. ECCV (2008)
17. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
18. Cronin, L., Duros, V., Grizou, J., Xuan, W., Hosni, Z., Long, D.L., Miras, H.: Human vs robots in the discovery and crystallization of gigantic polyoxometalates. Angewandte Chemie (2017)
19. Duchenne, O., Laptev, I., Sivic, J., Bach, F., Ponce, J.: Automatic annotation of human actions in video. In: ICCV (2009)
20. Escorcia, V., Caba Heilbron, F., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: ECCV (2016)
21. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of machine learning research **9**(Aug), 1871–1874 (2008)

22. Freytag, A., Rodner, E., Denzler, J.: Selecting influential examples: Active learning with expected model output changes. In: ECCV (2014)
23. Gaidon, A., Harchaoui, Z., Schmid, C.: Actom sequence models for efficient action detection. In: CVPR (2011)
24. Gao, J., Yang, Z., Sun, C., Chen, K., Nevatia, R.: Turn tap: Temporal unit regression network for temporal action proposals. In: ICCV (2017)
25. Gavves, S., Mensink, T., Tommasi, T., Snoek, C., Tuytelaars, T.: Active transfer learning with zero-shot priors: Reusing past datasets for future tasks. In: ICCV (2015)
26. Giancola, S., Amine, M., Dghaily, T., Ghanem, B.: Soccernet: A scalable dataset for action spotting in soccer videos. In: CVPR Workshops (2018)
27. Gilad-Bachrach, R., Navot, A., Tishby, N.: Query by committee made real. In: NIPS (2006)
28. Gkioxari, G., Hariharan, B., Girshick, R., Malik, J.: R-cnns for pose estimation and action detection (2014)
29. Gkioxari, G., Malik, J.: Finding action tubes. In: CVPR (2015)
30. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. ICLR (2014)
31. Gu, C., Sun, C., Vijayanarasimhan, S., Pantofaru, C., Ross, D.A., Toderici, G., Li, Y., Ricco, S., Sukthankar, R., Schmid, C., et al.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: CVPR (2018)
32. Hakkani-Tür, D., Riccardi, G., Gorin, A.: Active learning for automatic speech recognition. In: Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on (2002)
33. Hu, P., Ramanan, D.: Finding tiny faces. In: CVPR (2017)
34. Jain, M., van Gemert, J., Jegou, H., Bouthemy, P., Snoek, C.G.: Action localization with tubelets from motion. In: CVPR (2014)
35. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/ (2014)
36. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: CVPR (2009)
37. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Active learning with gaussian processes for object categorization. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. pp. 1–8. IEEE (2007)
38. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
39. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
40. Konyushkova, K., Sznitman, R., Fua, P.: Learning active learning from data. In: Advances in Neural Information Processing Systems. pp. 4226–4236 (2017)
41. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: ICCV (2017)
42. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: ICML (1994)
43. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (1994)
44. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: ECCV (2018)

45. Lin, X., Parikh, D.: Active learning for visual question answering: An empirical study. arXiv preprint arXiv:1711.01732 (2017)
46. Liu, B., Ferrari, V.: Active learning for human pose estimation. In: ICCV (2017)
47. Maji, S., Bourdev, L., Malik, J.: Action recognition from a distributed representation of pose and appearance. In: CVPR (2011)
48. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: CVPR (2009)
49. Niebles, J.C., Chen, C.W., Fei-Fei, L.: Modeling temporal structure of decomposable motion segments for activity classification. In: ECCV (2010)
50. Oneata, D., Verbeek, J., Schmid, C.: Efficient action localization with approximately normalized fisher vectors. In: CVPR (2014)
51. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers **10**(3), 61–74 (1999)
52. Poppe, R.: A survey on vision-based human action recognition. Image and vision computing **28**(6), 976–990 (2010)
53. Qi, G.J., Hua, X.S., Rui, Y., Tang, J., Zhang, H.J.: Two-dimensional active learning for image classification. In: CVPR (2008)
54. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
55. Settles, B.: Active learning literature survey. University of Wisconsin, Madison **52**(55-66), 11 (2010)
56. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: CVPR (2016)
57. Sigurdsson, G.A., Russakovsky, O., Farhadi, A., Laptev, I., Gupta, A.: Much ado about time: Exhaustive annotation of temporal data. In: AAAI Conference on Human Computation and Crowdsourcing (HCOMP) (2016)
58. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: ECCV (2016)
59. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
60. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)
61. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
62. Thompson, C.A., Califf, M.E., Mooney, R.J.: Active learning for natural language parsing and information extraction
63. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: Proceedings of the ninth ACM international conference on Multimedia. pp. 107–118. ACM (2001)
64. Vijayanarasimhan, S., Grauman, K.: Large-scale live active learning: Training object detectors with crawled data and crowds. International Journal of Computer Vision **108**(1-2), 97–114 (2014)
65. Vondrick, C., Patterson, D., Ramanan, D.: Efficiently scaling up crowdsourced video annotation. International Journal of Computer Vision (2012)
66. Vondrick, C., Ramanan, D.: Video annotation and tracking with active learning. In: NIPS (2011)

67. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR (2011)
68. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: CVPR (2017)
69. Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Learning to track for spatio-temporal action localization. In: ICCV (2015)
70. Woodward, M., Finn, C.: Active one-shot learning. In: NIPS (2016)
71. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection (2017)
72. Yang, J., et al.: Automatically labeling video data using multi-class active learning. In: ICCV (2003)
73. Yao, B., Jiang, X., Khosla, A., Lin, A.L., Guibas, L., Fei-Fei, L.: Human action recognition by learning bases of action attributes and parts. In: ICCV (2011)
74. Yeung, S., Ramanathan, V., Russakovsky, O., Shen, L., Mori, G., Fei-Fei, L.: Learning to learn from noisy web videos. In: CVPR (2017)
75. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: CVPR (2016)
76. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm computing surveys (CSUR) **38**(4), 13 (2006)
77. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Lin, D., Tang, X.: Temporal action detection with structured segment networks. In: ICCV (2017)