Action Duration Prediction for Segment-Level Alignment of Weakly-Labeled Videos

Reza Ghoddoosian Saif Sayed Vassilis Athitsos
Vision-Learning-Mining Lab, University of Texas at Arlington
{reza.ghoddoosian, saififtekar.sayed}@mavs.uta.edu, athitsos@uta.edu

Abstract

This paper focuses on weakly-supervised action alignment, where only the ordered sequence of video-level actions is available for training. We propose a novel Duration Network[1], which captures a short temporal window of the video and learns to predict the remaining duration of a given action at any point in time with a level of granularity based on the type of that action. Further, we introduce a Segment-Level Beam Search to obtain the best alignment, that maximizes our posterior probability. Segment-Level Beam Search efficiently aligns actions by considering only a selected set of frames that have more confident predictions. The experimental results show that our alignments for long videos are more robust than existing models. Moreover, the proposed method achieves state of the art results in certain cases on the popular Breakfast and Hollywood Extended datasets.

1. Introduction

Activity analysis covers a wide range of applications from monitoring systems to smart shopping and entertainment, and it is a topic that has been extensively studied in recent years. While good results have been obtained in recognizing actions in single-action RGB videos [5, 7, 10, 11, 37, 43], there are many real-life scenarios where we want to recognize a sequence of multiple actions, whose labels and start/end frames are unknown. Most work done in this area is fully supervised [15, 17, 21, 26, 28, 32, 35, 38, 40], requiring each frame in the training videos to be annotated. Given the need of deep learning algorithms for ever-larger training datasets, frame-level annotation can be expensive and unscalable. "Weak supervision" is an alternative, where each training video is only annotated with the ordered sequence of actions occurring in that video, with no start/end frame information for any action [3, 6, 8, 12, 18, 19, 29, 31].

This paper focuses on **weakly-supervised action alignment**, where it is assumed that the sequence of video-level action labels are provided as input for training and inference, and the output is the start and end time of each action.

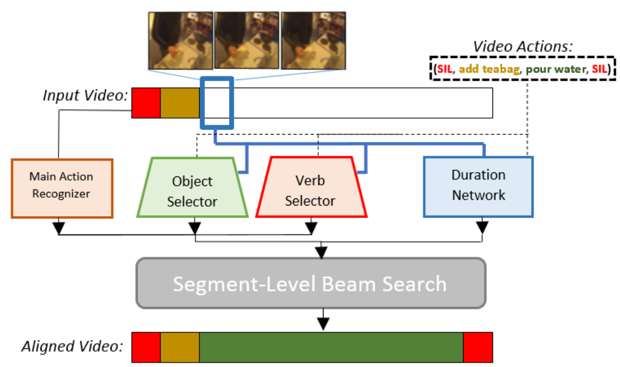

Figure 1. An overview of our proposed method. Based on the context of the temporal window, *pour water* is selected and its duration is predicted to align the given video-level actions

A key challenge in weakly supervised action alignment is correctly predicting the duration of actions. To achieve this goal, we propose a *Duration Network (DurNet)* that, unlike previous methods, takes video features into account. Video features contain valuable information that existing duration models ignore. As an example, video features can capture the pace (slow or fast) at which an action is performed. As another example, video features can capture the fact that an ongoing "frying" action is likely to continue for a longer time if the cook is currently away from the frying pan. Our duration model learns to estimate the remaining duration of an ongoing action based on the current visual observations. More specifically, the proposed DurNet mainly consists of a bi-directional Long Short-Term Memory (LSTM), which takes as inputs the set of frame features in a short temporal window at a given time, a hypothesized action class and its elapsed duration. The network outputs the probability of various durations (from a discretized set) for the remainder of that action.

We also introduce a Segment-Level Beam Search al-

[1]Code available at: https://github.com/rezaghoddoosian/DurNet

gorithm to efficiently maximize our factorized probability model for action alignment. This algorithm modifies the vanilla beam search to predict the most likely sequence of action segments without looping through all possible action-duration combination in all frames. Instead, it predicts the action and duration of segments by selecting a small subset of the frames that are significant enough to maximize the posterior probability. The time complexity of our Segment-Level Beam Search is linear to the number of action segments in the video, which is theoretically better than that of other Viterbi based alignment methods [19, 28, 31, 22]. In particular Richard *et al*. [31] considered visual and length models' frame-level outputs and their combinations over all the frames for action alignments. More recently [22] extended Richard *et al*.'s work [31] by incorporating all invalid action sequences in the loss function during training, but follows the same frame-level inference technique as in [31].

The main contributions of this paper can be summarized as follows: (1) We introduce a Duration Network for action alignment, that is explicitly designed to exploit information from video features and show its edge over the Poisson model used in previous work [31, 22]. (2) We propose a Segment-Level Beam Search that can efficiently align actions to frames without exhaustively evaluating each video frame as a possible start or end frame for an action (in contrast to [18, 19, 31, 22]). (3) In our experiments, we use two common benchmark datasets, the Breakfast [16] and Hollywood Extended [3], and we measure performance using three metrics from [8]. Depending on the metric and dataset, our method leads to results that are competitive or superior to the current state-of-the-art for action alignment.

2. Related Work

Weakly-Supervised Video Understanding. Existing methods for video activity understanding often differ in the exact version of the problem that they aim to solve. [9, 34] aim to associate informative and diverse sentences to different temporal windows for dense video captioning. [25, 39, 42] aim to do action *detection*, and are evaluated on videos that consist of typically a single unique action with a large portion of background frames.

Weakly-supervised action segmentation and alignment have been studied under different constraints at training time. Some works utilize natural language narrations of what is happening [2, 20, 24, 33, 44]. [30] use only unordered video-level action sets to infer video frames. Our work is closest to [3, 6, 8, 12, 18, 19, 29, 31, 22], where an ordered video-level sequence of actions is provided for training.

Our paper focuses on the task of weakly-supervised action alignment, where the video and an ordered sequence of action labels are provided as input, and frame-level annotations are the output.

Duration Modeling. One of the key innovations of our method is in weakly supervised modeling and prediction of action duration. Therefore, it is instructive to review how existing methods model duration. Some methods [6, 8, 12, 29] do not have an explicit duration model; the duration of an action is obtained as a by-product of the frame-by-frame action labels that the model outputs. [23, 1, 13] studied long term duration prediction. However they are fully supervised methods whose results are highly sensitive to ground-truth observations.

Most related to our duration model in action alignment are existing methods that model action duration as a Poisson function [31], or as a regularizer [3, 4, 19, 28] to penalize actions that last too long or too short. Specifically [31] and [22] integrated an action dependent Poisson model into their system which is characterized only by the average duration of each action based on current estimations. The key innovation of our method is that our duration model takes the video data into account. The video itself contains information that can be used to predict the remaining duration of the current action, and our method has the potential to improve prediction accuracy by taking this video information into account.

3. Method

In this section, we explain what probabilistic models our method consists of and how they are deployed for our Segment-Level Beam Search.

3.1. Problem Formulation

Our method takes two inputs. The first input is a video of T frames, represented by $\mathbf{x}_1^T$, which is the sequence of per-frame features. Feature extraction is a black box, our method is not concerned with how those features have been extracted from each frame. The second input is an ordered sequence $\boldsymbol{\tau} = (\tau_1, \tau_2, ..., \tau_M)$ of M action labels, that list the sequence of all actions taking place in the video.

A partitioning of the video into N consecutive segments is specified using a sequence $\mathbf{c}_1^N$ of action labels (c_n specifies the action label for the n-th segment) and a sequence $\mathbf{l}_1^N$ of corresponding segment lengths (l_n specifies the number of frames of the n-th segment). Given such a partition, we use notation π_n for the first frame of the n-th segment.

Given inputs $\mathbf{x}_1^T$ and $\boldsymbol{\tau}$, the goal of our method is to identify the most likely sequence $\bar{\mathbf{c}}_1^N$ of action labels c_n and corresponding sequence $\bar{\mathbf{l}}_1^N$ of durations l_n:

$$(\bar{\mathbf{c}}_1^N, \bar{\mathbf{l}}_1^N) = \underset{\mathbf{c}_1^N, \mathbf{l}_1^N}{\operatorname{argmax}}\, p(\mathbf{c}_1^N, \mathbf{l}_1^N | \mathbf{x}_1^T, \boldsymbol{\tau}) \tag{1}$$

We note that N (the number of segments identified by our method) can be different than M (the number of action labels in input $\boldsymbol{\tau}$). This happens because our method may

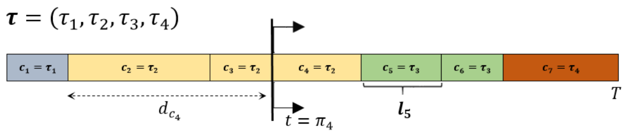

Figure 2. A sample segmented ($N = 7$) video given its video-level labels $\boldsymbol{\tau}$ ($M = 4$). One ground-truth action label τ can correspond to multiple consecutive segments

output the same action label for two or more consecutive segments, and all consecutive identical labels correspond to a single element of $\boldsymbol{\tau}$. We use Ω_n to denote the earliest segment number such that all segments from segment Ω_n up to and including segment n have the same action label. For example, in Fig. 2, $\Omega_4 = \Omega_3 = \Omega_2 = 2$.

Consider a frame π_n, that is the starting frame of the n-th segment. We assume that the remaining duration of an action at frame π_n depends on the type of action c_n, the elapsed duration $\mathbf{l}_{\Omega_n}^{n-1}$ of c_n up to frame π_n, and the visual features of a window of α frames starting at frame π_n. We denote this window as $\mathbf{w}_n = \mathbf{x}_{\pi_n}^{\pi_n+\alpha-1}$. Also, we decompose each action label c_n into a corresponding verb v_n and object o_n. For example the action "take cup" can be represented by the $(take, cup)$ pair, where $take$ and cup are the verb and object respectively. Working with "verbs" instead of "actions" lets us benefit from the shared information among "actions" with the same "verb". This specifically helps in analyzing any weakly-labeled video where the frame-level pseudo ground-truth is inaccurate. Based on the above, we rewrite $p(\mathbf{c}_1^N, \mathbf{l}_1^N | \mathbf{x}_1^T, \boldsymbol{\tau})$ as:

$$p(\mathbf{c}_1^N, \mathbf{l}_1^N | \mathbf{x}_1^T, \boldsymbol{\tau}) = \prod_{n=1}^{N} p(l_n | \mathbf{w}_n, \mathbf{l}_{\Omega_n}^{n-1}, c_n) \cdot p(c_n | \mathbf{x}_1^T, \boldsymbol{\tau}) \quad (2)$$

$$= \prod_{n=1}^{N} p(l_n | \mathbf{w}_n, \mathbf{l}_{\Omega_n}^{n-1}, v_n, o_n) \cdot p(c_n | \mathbf{x}_1^T, \boldsymbol{\tau}) \quad (3)$$

$$= \prod_{n=1}^{N} p(l_n | \mathbf{w}_n, \mathbf{l}_{\Omega_n}^{n-1}, v_n) \cdot p(c_n | \mathbf{x}_1^T, \boldsymbol{\tau}) \quad (4)$$

We should note that, in the above equations, in the boundary case where $\Omega_n = n$, we define $\mathbf{l}_{\Omega_n}^{n-1}$ to be 0. The Duration and Action Selector Network, described next, will be used to compute the probability terms in Eq. 4. Then, using our Segment-Level Beam Search, the most likely segment alignment will be identified.

3.2. Duration Network(DurNet)

Previous work [19, 31, 22] has tried to model the duration of actions. Richard *et al.* [31] have used a class-dependent Poisson distribution to model action duration, assuming that the duration of an action only depends on the type of that action. In contrast, we propose a richer duration

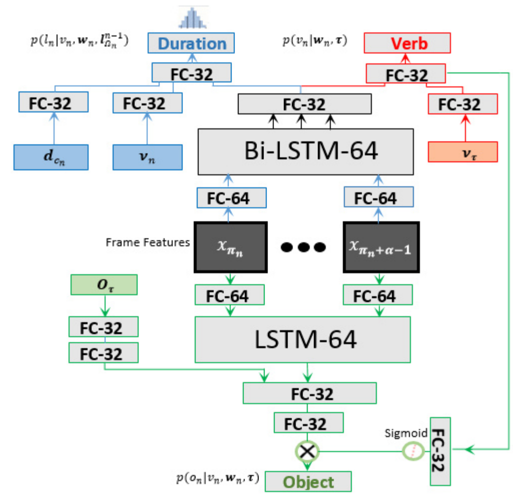

Figure 3. Architecture of the Duration, Object and Verb Selector Networks

model, where the length of an action segment depends not only on the type of that action, but also on the local visual features of the video, as well as on the length of the immediately preceding segments if they had the same action label as the current segment (Eq. 4).

The proposed model allows the estimate of the remaining length of an action to change based on video features. For example, our model can potentially predict a longer remaining duration for the action "squeeze orange" if the local visual cues correspond to a person just picking up the orange, compared to a person squeezing the orange.

In our method, the range of possible durations of a given action depends on the $verb$ of that action. For example, one second could be half of a short action associated with verb "take" and only one-hundredth of a longer action associated with verb "frying". We model this dependency by mapping time length to progress units for each verb. We denote by γ_v the median length of verb v across all training videos, and by L the number of time duration bins. We should note that the system cannot know the true value of γ_v, since frame-level annotations are not part of the ground truth. Instead, our system estimates γ_v based on pseudo-ground truth that is provided using an existing weakly supervised action alignment method, such as [8, 31]. Given this estimated γ_v, we discretize the elapsed and remaining time lengths into verb-dependent bins; i.e. the bin width b_v is calculated based on the type of each verb:

$$b_v = \frac{\gamma_v}{\lfloor \frac{L}{2} \rfloor + 1} \quad (5)$$

The above equation assures that the median length of a verb falls on or around the middle bin, which creates a more balanced distribution for learning.

In our method, $p(l_n | \mathbf{w}_n, \mathbf{l}_{\Omega_n}^{n-1}, v_n)$ is modeled by a Bi-

LSTM network preceded by a fully-connected layer and followed by fully connected layers and a softmax function σ as shown in Fig. 3. The input to this network, for any segment n at a given time π_n , is the one-hot vector representation of the verb $\mathbf{v}_n \in \mathbb{R}^V$ of a given action c_n and its discretized elapsed duration $\mathbf{d}_{c_n} \in \mathbb{R}^L$ as well as the local visual features $\mathbf{w}_n \in \mathbb{R}^{\Gamma \times F}$. Here, V is the total number of verbs, F is the input feature dimension, and Γ is the number of temporally sampled features over α frames starting with frame π_n. At the end, this network outputs the corresponding verb-dependent future progress probability corresponding to each bin. This probability is expressed as an L-dimensional vector $\mathbf{k}_{v_n}$, whose i-th dimension is the probability that the duration of action c_n falls in the i-th progress unit for verb v_n, given the inputs described above.

During training, we used a Gaussian to represent the progress probability labels as soft one-hot vectors. This representation considers the bins that are closer to the true bin more correct than the further ones. The resulting labels are used to compute the standard cross-entropy loss, as the DurNet loss function.

Finally, we translate this progress indicator back to time expressed as number of frames, according to verb-dependent steps s_v:

$$s_v = \lfloor \frac{\gamma_v}{L} \rfloor \tag{6}$$

$$l_{v,i} = (i+1) * s_v, i \in \{0, 1, ..., L-1\} \tag{7}$$

Thus, the i-th discretized duration $l_{v,i}$ for verb v corresponds to the i-th dimension of vector $\mathbf{k}_{v_n}$, and the value of $\mathbf{k}_{v_n}$ in the i-th dimension gives the probability of discretized duration $l_{v_n,i}$.

3.3. Action Selector Network

This network selects the label of the action occurring at any time in the video. Each action is decomposed as a $(verb, object)$ pair. The importance of objects and verbs in action recognition has been studied before [36, 44]. For example, the verb "take" in both "take bowl" and "take cup" is expected to visually look the same way. These two actions only differ in their corresponding objects. This approach has the advantage that not only the network can access more samples per class (verb/object), but also classification is done over fewer number of classes, because several actions share the same verb/object. This is specifically helpful in weakly-labeled data as the frame-level ground truth is not reliable. The probability of the selected action is obtained by the factorized equation below:

$$p(c_n|\mathbf{x}_1^T, \boldsymbol{\tau}) := \eta[p(o_n|v_n, \mathbf{w}_n, \boldsymbol{\tau})^\zeta p(v_n|\mathbf{w}_n, \boldsymbol{\tau})^\beta p(c_n|\mathbf{x}_1^T)^\lambda] \tag{8}$$

$\eta[\,]$ is a normalization function that assures :

$$\sum_{c_n \in \boldsymbol{\tau}} [p(c_n|\mathbf{x}_1^T, \boldsymbol{\tau})] = 1 \tag{9}$$

The Action Selector Network consists of three components: i) The verb selector network. ii) The object selector network. iii) The main action recognizer (Fig. 1). The influence of each network is adjusted by the ζ, β and λ hyper parameters.

i) The Verb Selector Network(VSNet): It focuses only on the local temporal features during the given time frame $[\pi_n, \pi_n + \alpha - 1]$ to select the correct verb v_n for segment n. The video-level verb labels $\mathbf{v}_{\boldsymbol{\tau}} \in \{0,1\}^V$ are also given as input to the network, where for every $i \in \{0, 1, ..., V-1\}$, $v_{\boldsymbol{\tau} i} = 1$ if $v_{\boldsymbol{\tau} i}$ is present in the video-level verbs, otherwise $v_{\boldsymbol{\tau} i} = 0$.

ii) The Object Selector Network(OSNet): Similar to the VSNet, using the local temporal features, this module selects the correct segment object o_n from the set of video-level objects $\mathbf{o}_{\boldsymbol{\tau}} \in \{0,1\}^O$, where O is the number of available objects in the dataset. Selecting the target object is also influenced by the type of the verb for a given action according to Eq. 8. In order to model this dependency, latent information from the VSNet flows into the OSNet (Fig. 3).

iii) The Main Action Recognizer(MAR): Unlike the other two components, this module produces frame-level probability distribution for the main actions. This network is more discriminative than the other two and particularly helpful in videos with repetitive verbs and objects. Note that the MAR module can be replaced by any baseline neural network architecture like CNNs or RNNs.

Finally, as shown in Eq. 8, the probability of a segment action is defined by fusing the output of the three above-mentioned networks. In the special case of $\zeta, \beta = 1$ and $\lambda = 0$, the definition of Eq. 8 would be truly probabilistic, and there would be no need for the normalization function η. The contribution of each network is quantitatively shown in Sec. 4.2.3. It is noteworthy to mention that our method is equally applicable without the verb-object decomposition assumption. In case there is no specific object associated with actions, our formulation still stands by setting $\zeta = 0$ and working with the actions as our set of verbs.

3.4. Segment-Level Beam Search

We introduce a beam search algorithm with beam size B to find the most likely sequence of segments, as specified by a sequence of labels $\mathbf{c}_1^N$ and a sequence of lengths $\mathbf{l}_1^N$. By combining Eq. 1 with Eq. 4 we obtain:

$$(\bar{\mathbf{c}}_1^N, \bar{\mathbf{l}}_1^N) = \underset{\mathbf{c}_1^N, \mathbf{l}_1^N}{\operatorname{argmax}} \{ \prod_{n=1}^{N} p(l_n|\mathbf{w}_n, \mathbf{l}_{\Omega_n}^{n-1}, v_n) \cdot p(c_n|\mathbf{x}_1^T, \boldsymbol{\tau}) \} \tag{10}$$

In frame-level beam search, different sequences of action classes are considered at every single frame until the end of the video. In contrast, our Segment-Level Beam Search

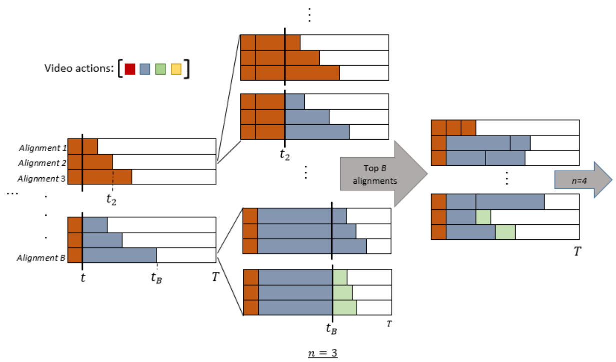

Figure 4. Our proposed Segment-Level Beam Search of beam size B during the estimation of the third segment ($n = 3$). For each alignment, different possibilities of next action and its predicted duration are evaluated. At each point in our method, all B hypothesized alignments consist of the same number of segments

allows the algorithm to consider such sequences only at the beginning of every segment. This technique is inspired by the fact that actions do not change rapidly from one frame to another.

We introduce the notation $A_i(c, l, t_i)$ to represent the probability of segment-level alignment i until frame t_i for each video, where c and l are the action class and length of the last segment. We also define $\max^B\{a_1, a_2, ..., a_n\}$ as the set of B greatest a_i, and calculate $A_i(c, l, t_i)$ of alignment i recursively for every action c_n and length l_n of segment n. Then, the B most probable alignments with n segments are selected over all combinations of c_n and l_n. Algorithm 1 summarizes the procedure for our proposed Segment-Level Beam Search with the following constraints:

- $c_1 = \tau_1, c_N = \tau_M$
- $t_i \leq T, \forall i \in \{1, 2, ..., B\}$

$\phi(c_{n-1})$ refers to the set of possible actions for segment n. $\phi(c_{n-1})$ is either a repetition of the action c_{n-1} of the previous segment or the start of the next action in $\boldsymbol{\tau}$. The final segment labels $\mathbf{c}_1^N$ and $\mathbf{l}_1^N$ are derived by keeping track of the maximizing arguments c_n and l_n in the maximization steps.

Note that $p(c_n|\mathbf{x}_1^T, \boldsymbol{\tau})$ in Algorithm 1 is factorized according to Eq. 8, and every $c_n \in \phi(c_{n-1})$ is broken down to its corresponding (v_n, o_n) pair. This factorization approach encourages segments that cover the whole duration of an action to avoid the penalty each time a new segment is added. This results in faster alignments with a smaller number of unreasonably short segments.

Time complexity of our Segment-Level Beam Search, for each video, depends on the beam size B, number of segments N and number of length bins L. As B and L are constant values, the time complexity for the algorithm above would be $O(N)$, and only limited to the number of segments per video. Based on our experiments, for the current public action alignment datasets, $N_{max} \approx 70$ is two orders of magnitude less than $T_{max} \approx 9700$. This makes the proposed beam search more efficient than the Viterbi algorithms used in [31, 22] and [19], which have the complexity of $O(T^2)$ and $O(T)$ respectively.

Algorithm 1 Segment-Level Beam Search

Input: Video features $\mathbf{x}_1^T$ and video-level labels $\boldsymbol{\tau}$, beam size B

Output: Action label and length sequences c_1^N and l_1^N.

$n \leftarrow 1,$ ▷ first segment

for $l_1 \in \{l_{v_1,0}, ..., l_{v_1,L-1}\}$ **do**:

$A_1(\tau_1, l_1, l_1) = p(l_1|\mathbf{x}_1^\alpha, 0, v_1) \cdot p(\tau_1|\mathbf{x}_1^T, \boldsymbol{\tau})$

$\mathbb{A}(n) = \max_{l_1}^B\{A_1(\tau_1, l_1, l_1)\},$ ▷ set of candidate alignments

while $t_i < T, \forall i \in \{1, 2, ..., B\}$ **do**:

$n \leftarrow n+1,$

for $i \leftarrow 1$ to B **do**:

for all $c_n \in \phi(c_{n-1}); l_n \in \{l_{v_n,0} ... l_{v_n,L-1}\}$ **do**:

$A_i(c_n, l_n, t_i + l_n) =$ ▷ $A_i(c_{n-1}, l_{n-1}, t_i) \in \mathbb{A}(n-1)$

$A_i(c_{n-1}, l_{n-1}, t_i) \cdot p(l_n|\mathbf{x}_{t_i}^{t_i+\alpha-1}, \mathbf{l}_{\Omega_n}^{n-1}, v_n) \cdot p(c_n|\mathbf{x}_1^T, \boldsymbol{\tau}),$

$\mathbb{A}(n) = \max_{c_n, l_n}^B\{A_i(c_n, l_n, t_i + l_n), \forall i \in \{1,2,...,B\}\}$

$A_{\text{final}}(c_N, l_N, T) = \max_{c_N, l_N}^1\{A_i(c_N, l_N, t_i), \forall i \in \{i|t_i = T\}\}$

4. Experiments

We show results on two popular weakly-supervised action alignment datasets based on three different metrics. We compare our method with several existing methods under different initialization schemes. Further, the contribution of each component of our model is quantitatively and qualitatively justified.

Datasets. 1) *The Breakfast Dataset (BD)* [16] consists of around 1.7k untrimmed instructional videos of few seconds to over ten minutes long. There are 48 action labels demonstrating 10 breakfast recipes with a mean of 4.9 instances per video. The overall duration of the dataset is 66.7h, and the evaluation metrics are conventionally calculated over four splits. 2) *The Hollywood Extended Dataset (HED)* [3] has 937 videos of 17 actions with an average of 2.5 non-background action instances per video. There are in total 0.8M frames of Hollywood movies and, following [3], we split the data into 10 splits for evaluation.

There are four main differences between these two datasets: i) Actions in the *BD* follow an expected scenario and context in each video. However, the relation between consecutive actions in the *HED* can be random. ii) Camera in the *BD* is fixed while there are scene cuts in the *HED*, making the duration prediction more challenging. iii) Background frames are over half of the total frames in the *HED*, while the percentage of them in the *BD* is about 10%, and iv) The inter-class duration variability in the *BD* is considerably higher than the *HED*.

Metrics. We use three metrics to evaluate performance: 1) *acc* is the frame-level accuracy averaged over all the videos. 2) *acc-bg* is the frame-level accuracy without the background frames. This is specifically useful for cases where the background frames are dominant as in the *HED*. 3) *IoU* defined as the intersection over union averaged across all videos. This metric is more robust to action label imbalance and is calculated over non-background segments.

Implementation. For a fair comparison, we obtained the pre-computed 64 dimensional features of previous work [6, 31, 22], computed using improved dense trajectories [41] and Fisher vectors [27], as described in [17]. A single layer bi-directional LSTM with 64 hidden units is shared between the DurNet and VSNet, and a single layer LSTM with 64 hidden units for the OSNet. We followed the same frame sampling as [22], [8] or [31], depending on the method we use for initialization. We use the cross-entropy loss function for all networks, using Adam optimization [14], learning rate of 10^{-5} and batch size of 64. L in the DurNet was set to 7 and 4 for the *BD* and *HED* respectively. In our experiments on the *BD*, we used an $alpha$ of 60 frames and ζ, β, and λ were adjusted to 1, 30, and 5 respectively for our selector network. Beam size in our beam search was set to 150 and other hyperparameters were picked after grid search optimization (refer to supplementary material).

Training Strategy. During training, alignment results of a baseline weakly-supervised method, *e.g.* CDFL [22], NNViterbi [31] or TCFPN [8], on the training data is used as the initial pseudo-ground truth. We also adopt the pre-trained frame-level action classifier (visual model) of the baseline (CDFL, NNViterbi or TCFPN) as our main action selector component. The initial pseudo-ground truth is used to train our duration and action selector networks. Then, new alignments are generated through the proposed Segment-Level Beam Search algorithm on the training videos. We call these new alignments the "new pseudo-ground truth". The adopted visual model is finally retrained based on our "new pseudo-ground truth", and used alongside our other components to align the test videos.

4.1. Comparison to State-of-the-Art Methods

Comparison Settings. In addition to evaluating existing methods, we also evaluate some combinations of existing methods, as follows: 1,2) *Ours/ [31] pg* and *Ours/ [22] pg*: Ours initialized with NNviterbi [31] and CDFL [22] pseudo-ground truth respectively, and a single layer GRU as the MAR. 3) *Ours/ [8] pg*: Ours initialized with the training results of [8] as our pseudo-ground truth, and the TCFPN [8] network as the MAR. 4) *[8]/ [31] pg*: The ISBA+TCFPN method [8] initialized with NNViterbi [31] pseudo-ground truth. 5) *[31]/ [8] pg*: The NNViterbi method [31] initialized with [8] pseudo-ground truth.

Table 1. Weakly-supervised action alignment results of existing methods on two main datasets. (* from [8], † best results obtained after running the author's source code multiple times,** after slight changes to the original source code for the specific task.)

	Breakfast (%)			Hollywood Extended (%)		
Models	acc	acc-bg	IoU	acc	acc-bg	IoU
HTK [18]*	43.9	-	26.6	49.4	-	29.1
ECTC [12]*	∼35	-	-	-	-	-
D^3TW [6]	57.0	-	-	59.4	-	-
TCFPN [8]†	51.7	48.2	33.0	57.6	46.1	28.2
[8]/ [31] pg**	56.4	53.4	36.2	-	-	-
NNViterbi [31]†	63.5	63.0	**47.5**	59.6	53.2	32.4
[31]/ [8] pg**	63.4	62.8	47.3	-	-	-
CDFL [22]	63.0	61.4	45.8	**65.0**†	63.7†	**40.2**†
Ours/ [8] pg	55.7	56.1	36.3	50.1	64.1	31.4
Ours/ [31] pg	63.7	65.0	42.5	56.0	64.3	34.3
Ours/ [22] pg	**64.1**	**65.5**	43.0	59.1	**65.4**	35.6

Action Alignment Results. Table 1 shows results for weakly-supervised action alignment. Our method produces better or competitive results for most cases on both datasets. Initialized with CDFL, our method achieves state-of-the-art in two of the three metrics for the Breakfast dataset and in one metric on the Hollywood. We compare our method with CDFL [22], NNViterbi [31] and TCFPN [8] more extensively, because they are the best open source methods that follow a similar pseudo-ground approach for training. Also for better comparison, in Table 1 we present the results of training NNViterbi on the pseudo ground-truth from TCFPN and vice versa.

In direct head-to-head comparisons with CDFL, NNViterbi and TCFPN, the proposed method often outperforms the respective competitor, and in some cases the head-to-head performance improvement by our method is quite significant. Our method improves action alignment results of TCFPN [8] and NNViterbi [31] in 5 (Table 2a) and 4 (Table 2b) out of 6 metrics respectively. In addition, we outperform CDFL in frame-level accuracy with and without background on the Breakfast dataset, and when tested on the Hollywood dataset, CDFL accuracy without background is improved while the inference complexity is decreased to $O(N)$ from CDFL's $O(T^2)$ (Table 2c).

In Table 2, our Segment-Level Beam Search achieves consistent improved results in frame accuracy for both datasets when the background frames are excluded. Considering *acc-bg* is essential especially for the Hollywood dataset as on average around 60% of the video frames are background, so *acc* values can be misleading.

There are two plausible explanations on why the performance of our method for non-background actions is not equally repeated for the background segments. First, there

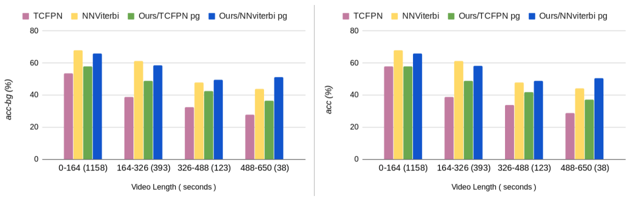

Figure 5. Weakly-supervised action alignment accuracy for videos of different lengths. Unlike the other two baselines, ours is more robust to longer videos. We obtained the results on four equal intervals considering the shortest and longest videos. The number of videos for each interval is mentioned in parentheses

Table 2. Head-to-head action alignment comparisons of the proposed model with the baselines († as specified in Table 1).

(a)

	Breakfast (%)			Hollywood Extended (%)		
Models	acc	acc-bg	IoU	acc	acc-bg	IoU
TCFPN [8]†	51.7	48.2	33.0	**57.6**	46.1	28.2
Ours/ [8] pg	**55.7**	**56.1**	**36.3**	50.1	**64.1**	**31.4**

(b)

	Breakfast (%)			Hollywood Extended (%)		
Models	acc	acc-bg	IoU	acc	acc-bg	IoU
NNViterbi [31]†	63.5	63.0	**47.5**	**59.6**	53.2	32.4
Ours/ [31] pg	**63.7**	**65.0**	42.5	56.0	**64.3**	**34.3**

(c)

	Breakfast (%)			Hollywood Extended (%)		
Models	acc	acc-bg	IoU	acc	acc-bg	IoU
CDFL [22]	63.0	61.4	**45.8**	**65.0**†	63.7†	**40.2**†
Ours/ [22] pg	**64.1**	**65.5**	43.0	59.1	**65.4**	35.6

is a lack of defined structure in what background can be, which makes it harder to learn. Second, there are cases where background depicts scenes where a person is still or no movement is happening. It is a tough task for even humans to predict how long that motionless scene would last, so the DurNet can easily make confident wrong predictions resulting in inaccurate alignments of background segments.

Fig. 5 shows how alignment results vary with video length on the Breakfast Dataset. The performance of our method compared to NNViterbi and TCFPN improves as video length increases. In longer videos, the DurNet can maintain the same action longer depending on the context, while in [31] any duration longer than the action average length gets penalized.

4.2. Analysis and Ablation Study

All analysis and ablation study is done using the TCFPN [8] pseudo ground-truth initialization. We also ran our ablation study experiments on the Breakfast dataset mainly, because it consists of videos with many actions and high duration variance, so the impact of learning duration can be measured more effectively.

Table 3. Comparison between our Duration Network and statistical Poisson length model on the breakfast dataset.

	Alignment	
Models	acc	acc-bg
Ours+Poisson	54.56%	54.95%
Ours+Duration Net	**55.70%**	**56.10%**

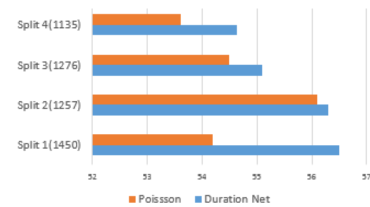

Figure 6. Split-wise frame accuracy on the Breakfast dataset. The number of training videos for each split is indicated in parentheses.

4.2.1 DurNet vs. Poisson Duration Model.

We compare our Duration Network with the Poisson length model used in [31, 22]. To compare the two models, we replaced the DurNet in our Segment-Level Beam Search with the Poisson model in [31, 22], while keeping all other parts of our method unchanged.

Table 3 quantitatively shows the advantage of using the context of the video, as it has improved the alignment accuracy by more than 1%. One reason for the small improvement, however, could be the imbalanced training set size across the four folds. Unlike the statistical Poisson approach, the performance of DurNet, as in other Neural Networks, depends on the training set size. As Figure 6 shows, the bigger the training data size, the better the performance of the DurNet.

4.2.2 Duration Step Size Granularity.

As explained in Section 3.2, the predicted durations are discretized into a fixed number L of bins, using different step sizes s_v for different verbs. In order to analyze the advantage of this duration modeling, we compare the weakly-supervised alignment results obtained when we replace this approach with fixed step size for all classes, as well as with different alternatives of adaptive steps (Table 4); *i.e.*, the predicted duration range of each action can depend on the maximum, mean or median length of that action calculated across all training videos. A fixed step and a step size dependent on maximum duration, both produce poor results. Step sizes dependent on mean and median durations of actions produce comparable results.

4.2.3 Analysis of the Action Selector Components.

We evaluate the effect of the OSNet, VSNet and MAR separately. Selecting verbs without objects fails in videos where

Table 4. The result of fixed step duration modeling with different alternatives of adaptive steps for weakly-supervised alignment.

	Alignment on Breakfast (%)		
Models	acc	acc-bg	IoU
Fixed steps($s_v = 5$ seconds)	49.9	49.6	32.3
Max-based adaptive steps	48.9	47.6	29.7
Mean-based adaptive steps	54.9	55.4	35.8
Median-based adaptive steps	**55.7**	**56.1**	**36.3**

Table 5. Contribution of each action selector component. Having all three components gives the best results.

	Alignment on Breakfast (%)		
Models	acc	acc-bg	IoU
Special case ($\zeta, \beta = 1, \lambda = 0$)	53.9	54.4	35.4
Action selector w/o main action	55.5	56.1	36.0
Action selector w/o object	54.8	54.6	35.9
Action selector w/o verb	50.9	50.8	32.8
All components	**55.7**	**56.1**	**36.3**

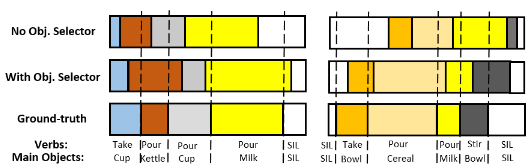

Figure 7. Two sample aligned videos, that consist of action labels with the same verb. The object selector component improves the results by aligning the segments with respect to the correct object.

two actions with the same verb happen consecutively in a video, *e.g.* pour cereal and pour milk (Fig. 7). Likewise, excluding the VSNet is problematic when two consecutive actions share the same object. Our experiments show that the VSNet and the MAR have the biggest and smallest contributions respectively(Table 5). We also include the results of the special case where we do not use hyperparameters in Eq. 8. As we see, a weighted combination of all three components performs best.

4.2.4 Qualitative Segment-Level Alignment Results.

One of the benefits of our Beam Search is predicting the class and length of segments without looping through all possible action-length combinations in all frames. Specifically, by predicting the duration of a segment in advance, only a limited set of more significant frames is processed. This leads to faster alignments with competitive accuracy compared to the frame-level Viterbi in [31, 22] (Table 2).

We demonstrate some success and failure cases of our segment predictions in Fig.8. It shows how a half minute video can be segmented in a small number of steps. Only a limited window of frames at the start of each step decides the class and length of the corresponding segment. Green and red arrows indicate valid and wrong step duration respectively. Similarly, the correctness of the action selector prediction is shown by the color of the square.

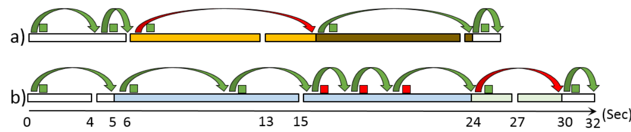

Figure 8. Separate segments denote the ground-truth and the color coded ones indicate the predicted segments. White segments are background. No-background actions are *add teabag* and *pour water* in video (a), and *pour cereal* and *pour milk* in video (b).

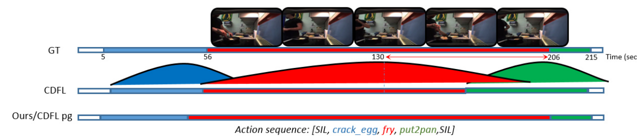

Figure 9. Alignment comparison between CDFL and our method for the test video "frying egg". Visual features in DurNet allow the predictions to adapt in duration. The color-coded curves represent the Poisson probability functions characterized by the expected duration of actions in CDFL.

Finally, Fig. 9 depicts a case where using visual features for length prediction outperforms the Poisson model in [22]. In this example "frying" is done slower than usual due to the subject turning away from the stove and the flipping of the egg. This makes the peak of the Poisson function temporally far from where "frying" actually ends resulting in the premature end of the action as longer predictions have very low probabilities and discouraged by the Poisson model. However, our DurNet takes the visual features into account and adapts to longer than expected action durations.

5. Conclusion

We have proposed our *Duration Network*, that predicts the remaining duration of an action taking the video frame-based features into account. We also proposed a Segment-Level Beam Search that finds the best alignment given the inputs from the DurNet and action selector module. Our beam search efficiently aligns actions by considering only a selected set of frames with more confident predictions. Our experimental results show that our method can be used to produce efficient action alignment results that are also competitive to state of the art.

Acknowledgement

This work is partially supported by National Science Foundation grant IIS-1565328. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors, and do not necessarily reflect the views of the National Science Foundation.

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5343–5352, 2018.

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016.

[3] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014.

[4] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *Proceedings of the IEEE international conference on computer vision*, pages 4462–4470, 2015.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[6] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3546–3555, 2019.

[7] César Roberto De Souza, Adrien Gaidon, Eleonora Vig, and Antonio Manuel López. Sympathy for the details: Dense trajectories and hybrid classification architectures for action recognition. In *European Conference on Computer Vision*, pages 697–716. Springer, 2016.

[8] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6508–6516, 2018.

[9] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. In *Advances in Neural Information Processing Systems*, pages 3059–3069, 2018.

[10] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Temporal residual networks for dynamic scene recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4728–4737, 2017.

[11] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 971–980, 2017.

[12] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *European Conference on Computer Vision*, pages 137–153. Springer, 2016.

[13] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9925–9934, 2019.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Hilde Kuehne, Ali Arslan, and Thomas 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*, pages 780–787, 2014.

[16] Hilde Kuehne, Ali Arslan, and Thomas 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*, pages 780–787, 2014.

[17] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[18] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017.

[19] Hilde Kuehne, Alexander Richard, and Juergen Gall. A hybrid rnn-hmm approach for weakly supervised temporal action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[20] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. 2008.

[21] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[22] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6243–6251, 2019.

[23] Tahmida Mahmud, Mahmudul Hasan, and Amit K Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5773–5782, 2017.

[24] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *arXiv preprint arXiv:1503.01558*, 2015.

[25] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761, 2018.

[26] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. The lear submission at thumos 2014. 2014.

[27] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007*

IEEE conference on computer vision and pattern recognition, pages 1–8. IEEE, 2007.

[28] Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3131–3140, 2016.

[29] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 754–763, 2017.

[30] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5996, 2018.

[31] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7386–7395, 2018.

[32] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1194–1201. IEEE, 2012.

[33] Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4480–4488, 2015.

[34] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. Weakly supervised dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1916–1924, 2017.

[35] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 585–594, 2017.

[36] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 585–594, 2017.

[37] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[38] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming 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*, pages 1961–1970, 2016.

[39] Krishna Kumar Singh and Yong Jae 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)*, pages 3544–3553. IEEE, 2017.

[40] Nam N Vo and Aaron F Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2641–2648, 2014.

[41] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[42] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4325–4334, 2017.

[43] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[44] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019.