Set-Constrained Viterbi for Set-Supervised Action Segmentation

Jun Li
Oregon State University
liju2@oregonstate.edu

Sinisa Todorovic
Oregon State University
sinisa@oregonstate.edu

Abstract

This paper is about weakly supervised action segmentation, where the ground truth specifies only a set of actions present in a training video, but not their true temporal ordering. Prior work typically uses a classifier that independently labels video frames for generating the pseudo ground truth, and multiple instance learning for training the classifier. We extend this framework by specifying an HMM, which accounts for co-occurrences of action classes and their temporal lengths, and by explicitly training the HMM on a Viterbi-based loss. Our first contribution is the formulation of a new set-constrained Viterbi algorithm (SCV). Given a video, the SCV generates the MAP action segmentation that satisfies the ground truth. This prediction is used as a framewise pseudo ground truth in our HMM training. Our second contribution in training is a new regularization of feature affinities between training videos that share the same action classes. Evaluation on action segmentation and alignment on the Breakfast, MPII Cooking2, Hollywood Extended datasets demonstrates our significant performance improvement for the two tasks over prior work.

1. Introduction

This paper addresses action segmentation by labeling video frames with action classes under set-level weak supervision in training. Set-supervised training means that the ground truth specifies only a set of actions present in a training video. Their temporal ordering and the number of their occurrences remain unknown. This is an important problem arising from the proliferation of big video datasets where providing detailed annotations of a temporal ordering of actions is prohibitively expensive. One example application is action segmentation of videos that have been retrieved from a dataset based on word captions [6, 20], where the captions do not describe temporal relationships of actions.

There is scant work on this problem. Related work [16] combines a frame-wise classifier with a Hidden Markov Model (HMM). The classifier assigns action scores to all frames, and the HMM models a grammar and temporal lengths of actions. However, their HMM and classifier are not jointly learned, and performance is significantly below that of counterpart approaches with access to ground-truth temporal ordering of actions in training [13].

In this paper, we also adopt an HMM model that is grounded on a fully-connected two-layer neural network which extracts frame features and scores label assignments to frames. Our HMM models temporal extents and co-occurrence of actions. We jointly train the HMM and neural network on a maximum posterior probability (MAP) action assignment to frames that satisfy the set-level ground truth of a training video. We expect that the inferred MAP action sequence is more optimal for joint training than the pseudo ground truth generated independently for each frame as in [16]. We cast this MAP inference as a set-constrained structured prediction of action labels such that every label from the ground-truth action set appears at least once in the MAP prediction. This problem has been shown to be NP-hard [1]. Therefore, our main novelty is an efficient approximation of this NP-hard problem for our set-supervised training.

Fig. 1 illustrates our two contributions. First, we propose an efficient Set Constrained Viterbi (SCV) algorithm for the MAP inference on training videos. Our SCV consists of two steps. In the first step, we run the Viterbi algorithm for inferring a MAP action assignment to frames, where the predicted actions are restricted to appear in the ground-truth set. When our initial prediction is missing some actions from the ground truth, we proceed to our second step that sequentially flips the predicted labels of lowest scoring video segments into the missing labels, so as to minimally decrease the assignment's posterior probability. This is done until the predicted sequence contains all actions from the ground truth. While this second step could alternatively use a more principled algorithm for enforcing the ground-truth constraint (e.g., Markov Chain Monte Carlo sampling), such an alternative would be significantly more complex and hence poorly justified.

Our second contribution is that we specify a new regularization of our set-supervised learning. This regularization is aimed at our two-layer neural network for extracting frame features. The regularization enforces that frame features are

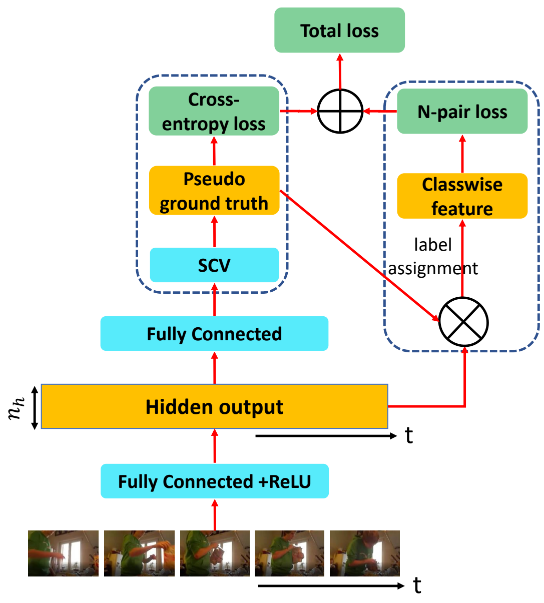


Figure 1. Our two contributions for set-supervised training include: (1) Set Constrained Viterbi (SCV) algorithm for estimating the MAP pseudo ground truth, and (2) Regularization for maximizing a margin between shared action classes and non-shared action classes in training videos. The pseudo ground truth is used for computing the cross-entropy loss and n-pair loss.

closer for frames belonging to the same actions than other frames of different actions. Our key novelty is a new n-pair loss that deals with sequences of actions, unlike the standard n-pair loss used for distance-metric learning of images or videos with a single class label. For pairs of training videos with generally different ground-truth sets of actions, our n-pair loss minimizes and maximizes distances of frame features over shared and non-shared actions, respectively.

For evaluation, we address action segmentation and action alignment on the Breakfast, MPII Cooking2, Hollywood Extended datasets. As in [16], action alignment means that for a test video we know the set of actions present, and the task is to accurately label frames with these actions. Note that this definition increases difficulty over the definition of action alignment used in prior work [15, 17], where for a test video we would also know the temporal ordering of actions present. Our experiments demonstrate that we significantly outperform existing work for the two tasks.

The rest of the paper is organized as follows. Sec. 2 reviews related work, Sec. 3 formalizes our problem, Sec. 4 specifies our SCV and regularization, Sec. 5 describes our inference in testing, and Sec. 6 presents our experiments.

2. Related Work

This section reviews closely related work on weakly supervised: (a) Action segmentation and localization; (b) Distance metric learning over sequences of labels. A review of both fully supervised and unsupervised action segmentation is beyond our scope, for obvious reasons.

Learning with the Temporal Ordering Constraint. A number of related approaches assume access to the temporal ordering of actions in training [2, 7, 9, 10, 12, 15, 5, 17, 4]. For example, Huang *et al*. [7] propose an extended connectionist temporal classification for taking into account temporal coherence of features across consecutive frames. Bojanowski *et al*. [2] formulate a convex relaxation of discriminative clustering based on a conditional gradient (Frank-Wolfe) algorithm for action alignment. Other approaches [15, 17] use a statistical language model for action segmentation. All these methods in training generate a framewise pseudo ground truth which satisfies the provided ground-truth action ordering. As our training is inherently less constrained, for generating a pseudo ground truth, we face a much larger search space of label sequences that satisfy our ground truth. This informs our design choices to adopt simpler models than theirs (e.g., our two-layer network vs. a more complex recurrent network in [17]) so as to facilitate our set-supervised learning.

Set-supervised Learning. Action localization under unordered set-level supervision has recently made significant progress [24, 22, 26, 21, 14, 8, 23]. All of these methods, however, address videos with few action classes and a lot of background frames. In contrast, we consider a more challenging setting, where videos show significantly more action classes. As shown in [16], these methods are particularly designed to detect sparse actions in relatively long background intervals, and thus are not suitable for our setting of densely labeled videos. For example, Shou *et al.* [21] use the outer-inner-contrastive loss for detecting action boundaries, and Paul *et al.* [14] minimize an action affinity loss for multi-instance learning (MIL) [27]. The affinity loss in [14] is related to our regularization. However, they enforce only similarity between video features of the same action class, whereas our more general n-pair loss additionally maximizes a distance between features of frames belonging to non-shared action classes.

The most related approach [16] uses MIL [27] to train a frame-wise classifier. Instead, we formulate the new set-constrained Viterbi algorithm to predict MAP action sequences for the HMM training, as well as regularize our training with distance-metric learning on frame features.

3. Our Problem Setup and Model

Problem Formulation. A training video of length T is represented by a sequence of frame features, $\boldsymbol{x} = [x_1, ..., x_t, ..., x_T]$, and annotated with an unordered set of action classes $C = \{c_1, \cdots, c_m, \cdots, c_M\}$, where C is a subset of a large set of all actions, $C \subseteq \mathcal{C}$. T and M may vary across the training set. There may be multiple instances of the same action in a training video. For a given video, our goal is to find an optimal action segmentation, $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$, where $\hat{\boldsymbol{c}} = [\hat{c}_1, ..., \hat{c}_n, ..., \hat{c}_{\hat{N}}]$ denotes the predicted temporal ordering of action labels of length $\hat{N}$, $\hat{c}_n \in \mathcal{C}$, and $\hat{\boldsymbol{l}} = [\hat{l}_1, \cdots, \hat{l}_{\hat{N}}]$ are their corresponding temporal extents.

Our Model. We use an HMM to estimate the MAP $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$ for a given video $\boldsymbol{x}$ as

$$(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}}) = \arg\max_{N,\boldsymbol{c},\boldsymbol{l}} \ p(\boldsymbol{c}, \boldsymbol{l}|\boldsymbol{x}) = \arg\max_{N,\boldsymbol{c},\boldsymbol{l}} \ p(\boldsymbol{c})p(\boldsymbol{l}|\boldsymbol{c})p(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{l}),$$
$$= \arg\max_{N,\boldsymbol{c},\boldsymbol{l}} \Big[\prod_{n=1}^{N-1} p(c_{n+1}|c_n)\Big]\Big[\prod_{n=1}^{N} p(l_n|c_n)\Big] \cdot \Big[\prod_{t=1}^{T} p(x_t|c_{n(t)})\Big]. \tag{1}$$

A similar HMM formulation is used in [16], but with a notable difference in the definition of $p(\boldsymbol{c})$. In [16], $p(\boldsymbol{c})$ is equal to a positive constant for all legal action grammars, where they declare $\boldsymbol{c}$ as a legal grammar if all actions in $\boldsymbol{c}$ regardless of their ordering are seen in at least one ground truth of the training dataset. In contrast, we specify $p(\boldsymbol{c})$ in (1) as a product of transition probabilities between actions along the sequence $\boldsymbol{c}$, $\prod_{n=1}^{N-1} p(c_{n+1}|c_n)$. Another difference from [16] is that we consider both *static* and *dynamic* formulations of the HMM, and [16] uses only the former.

In this paper, we specify and evaluate two distinct versions of the HMM given by (1) – namely, the static and dynamic HMM. The static probabilities in (1) are precomputed directly from the available set-supervised training ground truth. The dynamic probabilities in (1) are iteratively updated based on the previous MAP assignments $\{(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})\}$ over all training videos.

Static HMM. The static transition probability $p^s(c_{n+1}|c_n)$ in (1) is defined as

$$p^s(c_{n+1}|c_n) = \#(c_{n+1}, c_n) \ / \ \#(c_n), \tag{2}$$

where $\#(\cdot)$ denotes the number of action or action-pair occurrences in the training ground truth.

The static likelihood of action lengths $p^s(l|c)$ in (1) is modeled as a class-specific Poisson distribution:

$$p^s(l|c) = \frac{(\lambda_c^s)^l}{l!} e^{-\lambda_c^s}, \tag{3}$$

where λ_c^s is the static, expected temporal length of $c \in \mathcal{C}$. For estimating λ_c^s, we ensure that, for all training videos indexed by v, the accumulated mean length of all classes from the ground truth C_v is close to the video's length T_v. This is formalized as minimizing the following quadratic objective:

$$\text{minimize} \quad \sum_v \Big(\sum_{c \in C_v} \lambda_c^s - T_v\Big)^2, \quad \text{s.t.} \quad \lambda_c^s > l_{\min}, \tag{4}$$

where $l_{\min}$ is the minimum allowed action length which ensures a reasonable action duration.

The static likelihood $p^s(x|c)$ in (1) is estimated as:

$$p^s(x|c) \propto \frac{p(c|x)}{p^s(c)}, \qquad p^s(c) = \frac{\sum_v T_v \cdot 1(c \in C_v)}{\sum_v T_v}. \tag{5}$$

where $p(c|x)$ is the softmax score of our neural network, and $1(\cdot)$ is the indicator function. The static class prior per frame $p^s(c)$ in (5) is a percentage of the total footage of training videos having c in their ground truth.

Dynamic HMM. The dynamic transition probability $p^d(c_{n+1}|c_n)$ in (1) is defined as

$$p^d(c_{n+1} = \hat{c}_{n+1}|c_n = \hat{c}_n) = \#(\hat{c}_{n+1}, \hat{c}_n) \ / \ \#(\hat{c}_n), \tag{6}$$

where $\#(\hat{c}_{n+1}, \hat{c}_n)$ and $\#(\hat{c}_n)$ are the numbers of predicted consecutive pairs and single actions, respectively, in the previous MAP assignments $\{(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})\}$ over all training videos.

The dynamic likelihood of action lengths in (1) is also the Poisson distribution, $p^d(l|c) = \frac{(\lambda_c^d)^l}{l!} e^{-\lambda_c^d}$, where for the previous MAP assignments $\{(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})\}$ over all training videos, we estimate the expected temporal length of $c \in \mathcal{C}$ as

$$\lambda_c^d = \Big[\sum_v \sum_{n=1}^{\hat{N}} \hat{l}_{v,n} \cdot 1(c = \hat{c}_{v,n})\Big] \Big/ \sum_v \sum_{n=1}^{\hat{N}} 1(c = \hat{c}_{v,n}). \tag{7}$$

Finally, $p^d(x|c)$ in (1) is defined as

$$p^d(x|c) \propto \frac{p(c|x)}{p^d(c)}, \quad p^d(c) = \frac{\sum_v \sum_{n=1}^{\hat{N}} \hat{l}_{v,n} \cdot 1(c = \hat{c}_{v,n})}{\sum_v T_v}. \tag{8}$$

where $p^d(c)$ is a percentage of training frames predicted to belong to class c in the previous MAP assignments $\{(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})\}$ over all training videos.

Two-layer Neural Network. Our HMM is grounded on the frame features $\boldsymbol{x}$ via a two-layer fully connected neural network. The first fully connected layer uses ReLU to extract hidden features $\boldsymbol{h}$ as

$$\boldsymbol{h} = \max(0, \boldsymbol{W}^1 \boldsymbol{x} \oplus \boldsymbol{b}^1), \tag{9}$$

where $\boldsymbol{x} \in \mathbb{R}^{d \times T}$ denotes d-dimensional unsupervised features of T frames, $\boldsymbol{W}^1 \in \mathbb{R}^{n_h \times d}$ for $n_h = 256$ hidden units, $\boldsymbol{b}^1 \in \mathbb{R}^{n_h \times 1}$, $\boldsymbol{h} \in \mathbb{R}^{n_h \times T}$, and $\oplus$ indicates that the bias $\boldsymbol{b}^1$ is added to every column in the matrix $\boldsymbol{W}^1 \boldsymbol{x}$.

The second fully connected layer computes the matrix of unnormalized scores for each action class in $\mathcal{C}$ as

$$\boldsymbol{f} = \boldsymbol{W}^2 \boldsymbol{h} \oplus \boldsymbol{b}^2, \tag{10}$$

where $\boldsymbol{W}^2 \in \mathbb{R}^{|\mathcal{C}| \times n_h}$, $\boldsymbol{b}^2 \in \mathbb{R}^{|\mathcal{C}| \times 1}$, and $\boldsymbol{f} \in \mathbb{R}^{|\mathcal{C}| \times T}$. Thus, an element $\boldsymbol{f}[c, t]$ of matrix $\boldsymbol{f}$ represents the network's unnormalized score for class c at video frame t.

We use both $\boldsymbol{h}$ and $\boldsymbol{f}$ for estimating loss in our training.

4. Our Set-Supervised Training

For our training, we generate the MAP labels $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$ for every training video by using our new Set-Constrained Viterbi algorithm (SCV). $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$ is taken as the pseudo ground truth and used to estimate the cross-entropy loss and n-pair regularization loss for updating parameters of the HMM and two-layer neural network. In the following, we first formulate our SCV, and then the two loss functions.

4.1. Set Constrained Viterbi

Inference in training is formulated as the NP-hard all-color shortest path problem [1]. Given a training video and its ground truth C, our goal is to predict the MAP $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$, such that every ground-truth action $c \in C$ occurs at least once in $\hat{\boldsymbol{c}}$. As shown in Fig. 2, our efficient solution to this NP-hard problem consists of two steps.

In the first step, we ignore the all-color constraint, and conduct the HMM inference by using the vanilla Viterbi algorithm for predicting an initial MAP $(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{l}})$, as specified in (1), where actions in $\tilde{\boldsymbol{c}}$ are constrained to come from C. From (1) and (10), our first step is formalized as

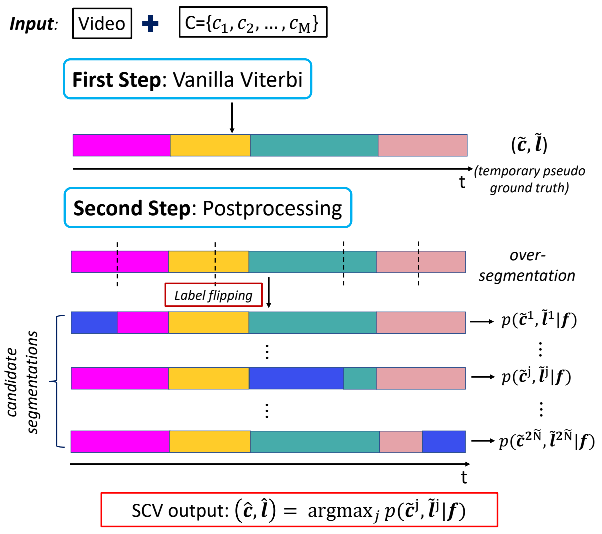
$$(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{l}}) = \underset{\substack{N,\boldsymbol{c},\boldsymbol{l} \\ \boldsymbol{c} \in C^N}}{\arg\max} \Big[\prod_{n=1}^{N-1} p(c_{n+1}|c_n)\Big] \Big[\prod_{n=1}^{N} p(l_n|c_n)\Big] \cdot \Big[\prod_{t=1}^{T} \frac{p(\boldsymbol{f}[c_{n(t)}, t])}{p(c_{n(t)})}\Big], \tag{11}$$

where $p(\boldsymbol{f}[c_{n(t)}, t])$ denotes the softmax score for class $c_n \in C$ at frame t. For estimating the likelihood of action lengths $p(l_n|c_n)$ in (11), we use the Poisson distribution, parameterized by either the static expected length λ_c^s as in (4) or dynamic λ_c^d as in (7). For the transition probabilities in (11), $p(c_{n+1}|c_n)$, we use either the static definition given by (2) or the dynamic definition given by (6).

The vanilla Viterbi begins by computing: (i) Softmax scores $p(\boldsymbol{f}[c_{n(t)}, t])$ with the two-layer neural network for $t = 1, \ldots, T$, and (ii) Per-frame class priors $p(c_{n(t)})$ for all $c \in \mathcal{C}$ as in (5) for the static HMM or in (8) for the dynamic HMM. Then, the algorithm uses a recursion to efficiently compute $(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{l}})$. Let $\boldsymbol{\ell}[c, t]$ denote the maximum log-posterior of all action sequences $\{\boldsymbol{c}\}$ ending with action $c \in C$ at video frame t, $t = 1, \ldots, T$. From (11), $\boldsymbol{\ell}[c, t]$ can be recursively estimated as

$$\boldsymbol{\ell}[c, t] = \max_{\substack{t' < t \\ c' \neq c \\ c' \in C}} \Big[\boldsymbol{\ell}[c', t'] + \log p(t - t'|c) + \log p(c|c') + \sum_{k=t'+1}^{t} \log \frac{p(\boldsymbol{f}[c, k])}{p(c)}\Big], \tag{12}$$

where $\boldsymbol{\ell}[c, 0] = 0$ and $p(t-t'|c)$ is the Poisson likelihood that action c has length $(t-t')$. During the recursion, we discard action sequences $\boldsymbol{c}$, whose accumulated mean length exceeds the video length T, *i.e.* $\sum_{c \in \boldsymbol{c}} \lambda_c > T$.



Input:
Video
C={c_1, c_2, …, c_M}
First Step: Vanilla Viterbi
$(\tilde{c}, \tilde{l})$
(temporary pseudo ground truth)
t
Second Step: Postprocessing
over-segmentation
Label flipping
candidate segmentations
$p(\tilde{c}^1, \tilde{l}^1|f)$
$p(\tilde{c}^j, \tilde{l}^j|f)$
$p(\tilde{c}^{2\tilde{N}}, \tilde{l}^{2\tilde{N}}|f)$
t
SCV output: $(\hat{c}, \hat{l}) = \text{argmax}_j\, p(\tilde{c}^j, \tilde{l}^j|f)$

Figure 2. Our SCV consists of two steps shown top-to-bottom. The first step is the vanilla Viterbi algorithm that infers an initial MAP action sequence $(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{l}})$. $\tilde{\boldsymbol{c}}$ may be missing some classes (e.g., "blue" label) from the ground truth C. The second step sequentially flips labels of video oversegments to the missing labels, such that the MAP score is minimally reduced, until the final prediction $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$ includes all actions from C.

After estimating the entire matrix of maximum log-posteriors, $\boldsymbol{\ell}[c, t]$, the Viterbi algorithm back-traces the optimal path in the matrix, starting from the maximum element of the last video frame $\max_{c \in C} \boldsymbol{\ell}[c, T]$, resulting in $(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{l}})$.

The second step is aimed at satisfying the ground-truth constraint by adding labels from C that have been missed in $(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{l}})$. If $\tilde{\boldsymbol{c}}$ consists of all classes from the ground truth C, we stop our inference. Otherwise, we proceed by sequentially searching for the missing classes. We first oversegment the video in an unsupervised manner, as explained below. Then, among many candidate oversegments, the best oversegment is selected for replacing its predicted class $\tilde{c}$ with the best missed class $c' \in C$ and $c' \notin \tilde{\boldsymbol{c}}$, such that the resulting posterior $p(\tilde{\boldsymbol{c}}', \tilde{\boldsymbol{l}}'|\boldsymbol{f})$ is minimally decreased relative to $p(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{l}}|\boldsymbol{f})$. Note that, due to the splitting of video segments into oversegments, our class flipping of oversegments does not remove the predicted classes that belong to C. The class flipping is sequentially conducted until all classes from C appear in the solution. The final solution is $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$.

For efficiency, our unsupervised video oversegmentation is a simple splitting of every predicted video segment in $\tilde{\boldsymbol{l}}$ into two parts. Specifically, for every nth predicted video segment of length $\tilde{l}_n$ starting at time $t_n < T$, we estimate the cosine similarities of consecutive frames as

$\{\boldsymbol{h}[t_n + k]^\top \boldsymbol{h}[t_n + k + 1] : k = 0, \dots, \tilde{l}_n - 2\}$, where $\boldsymbol{h}$ is specified in (9). Then, we select the $\hat{k}$th frame with the minimum similarity to its neighbor to split the interval into two oversegments $[t_n, t_n + \hat{k}]$ and $[t_n + \hat{k} + 1, t_n + \tilde{l}_n - 1]$.

4.2. The Cross-Entropy Loss

After the SCV infers the pseudo ground truth of a given training video, $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{l}})$, we compute the incurred cross-entropy loss as

$$\mathcal{L}_{\text{CE}} = -\sum_{t=1}^{T} \log p(\hat{c}_{n(t)}|x_t), \;\; \hat{c}_{n(t)} \in \hat{\boldsymbol{c}} \tag{13}$$

where $p(\hat{c}_{n(t)}|x_t) = p(\boldsymbol{f}[\hat{c}_{n(t)}, t])$ is the softmax score of the two-layer neural network for the nth predicted class $\hat{c}_{n(t)} \in \hat{\boldsymbol{c}}$ at frame t.

4.3. Regularization with the N-Pair Loss

Our training is regularized with the n-pair loss aimed at minimizing a distance between pairs of training videos sharing action classes in their respective set-level ground truths. We expect that the proposed regularization would help constrain our set-supervised problem in training.

For every pair of training videos v and v', we consider an intersection of their respective ground truths: $C_{vv'} = C_v \cap C_{v'}$. When $C_{vv'} \neq \emptyset$, for every shared class $c \in C_{vv'}$, our goal is two-fold:

1. Minimize a distance, $d_{vv'}^{cc}$, between features of v and v' that are relevant for predicting the shared class $c \in C_{vv'}$, and simultaneously.
2. Maximize distances $d_{vv'}^{ac}$ and $d_{vv'}^{cb}$ between features of the shared class $c \in C_{vv'}$ and features of non-shared classes $a \in C_v \setminus C_{vv'}$ and $b \in C_{v'} \setminus C_{vv'}$.

This is formalized as minimizing the following n-pair loss in terms of three distances illustrated in Fig. 3:

$$\begin{aligned}\mathcal{L}_{\text{NP}} =& \frac{1}{|C_{vv'}|} \sum_{c \in C_{vv'}} \log\Big[1 + \sum_{a \in C_v \setminus C_{vv'}} \exp(d_{vv'}^{cc} - d_{vv'}^{ac}) \\ &+ \sum_{b \in C_{v'} \setminus C_{vv'}} \exp(d_{vv'}^{cc} - d_{vv'}^{cb})\Big].\end{aligned} \tag{14}$$

The distances in (14) are expressed in terms of the first-layer features $\boldsymbol{h} \in \mathbb{R}^{n_h \times T}$, given by (9). Specifically, features relevant for a particular class $c \in \mathcal{C}$, denoted as $\overline{\boldsymbol{h}}^c$, are computed by averaging $\boldsymbol{h}$ over all video frames labeled with c in the MAP assignment. We consider both hard and soft class assignments to frames, and define:

$$\overline{\boldsymbol{h}}_{\text{hard}}^c = \frac{1}{\sum_{t=1}^T 1(c = \hat{c}_t)} \sum_{t=1}^{T} \boldsymbol{h}[t] \cdot 1(c = \hat{c}_t), \tag{15}$$

$$\overline{\boldsymbol{h}}_{\text{soft}}^c = \sum_{t=1}^{T} \boldsymbol{h}[t] \cdot p(\boldsymbol{f}[c, t]), \tag{16}$$

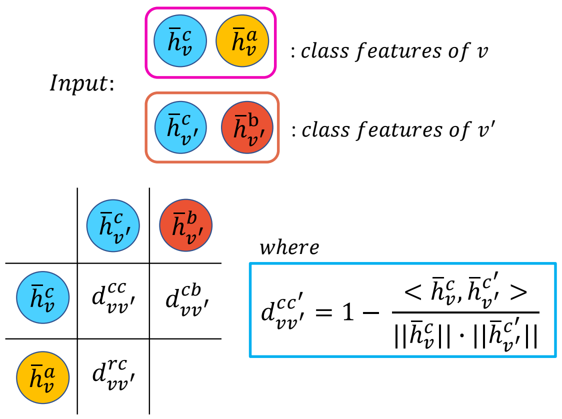


Figure 3. An example of estimating three cosine distances that are used for computing the n-pair loss. Given video v with only two classes $\{c, a\}$ and another video v' with also two classes $\{c, b\}$, we first compute their average class features: $\{\bar{h}_v^c, \bar{h}_v^a\}$ and $\{\bar{h}_{v'}^c, \bar{h}_{v'}^b\}$. Then, we estimate the cosine distance $d_{vv'}^{cc}$ of $\bar{h}_v^c$ and $\bar{h}_{v'}^c$ for the shared class c, and the cosine distances $d_{vv'}^{ac}$ and $d_{vv'}^{cb}$ of the video features for c and the non-shared classes a and b.

where $\hat{c}_t$ is the label predicted for tth frame by the SCV, and $p(\boldsymbol{f}[c, t])$ is the softmax score of $\boldsymbol{f}[c, t]$ specified in (10). Finally, we estimate the feature distances in (14) as

$$d_{vv'}^{cc'} = 1 - \frac{<\overline{\boldsymbol{h}}_v^c, \overline{\boldsymbol{h}}_{v'}^{c'}>}{\|\overline{\boldsymbol{h}}_v^c\| \|\overline{\boldsymbol{h}}_{v'}^{c'}\|}. \tag{17}$$

where we omit the subscript "hard" or "soft", for simplicity.

Our total loss is a weighted average of $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{NP}}$:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{CE}} + (1 - \lambda) \mathcal{L}_{\text{NP}}, \tag{18}$$

where we experimentally found that $\lambda = 0.5$ is optimal.

4.4. Complexity of Our Training

Complexity of the SCV is $O(T^2|\mathcal{C}|^2)$, where T is the video length, and $|\mathcal{C}|$ is the number of all action classes. The complexity mainly comes from the vanilla Viterbi algorithm in the first step of the SCV, whose complexity is $O(T^2|\mathcal{C}|^2)$. Complexity of the second step of the SCV is $O(T|\mathcal{C}|)$.

Complexity of the n-pair loss is $O(T|\mathcal{C}| + |\mathcal{C}|^2)$, where $O(T|\mathcal{C}|)$ accounts for computation of the average feature $\overline{\boldsymbol{h}}_v^c$, and $O(|\mathcal{C}|^2)$ accounts for estimation of the distances between features of shared and non-shared classes.

Therefore, our total training complexity is $O(T^2|\mathcal{C}|^2)$, whereas the training complexity of [16] is $O(T|\mathcal{C}|)$.

5. Inference on a Test Video

For inference on a test video, we follow the well-motivated Monte Carlo sampling of [16], which is not suitable for our training due to its prohibitive time complexity.

As in [16], we make the assumption that the set of actions appearing in a test video has been seen in our set-supervised training, i.e., there is at least one training video with the same ground-truth set of actions as the test video. Our sampling has two differences as explained below.

Given a test video with length T, we begin by uniformly sampling action sets C from all ground-truth sets of training videos. For every sampled C, we sequentially generate an action sequence $\boldsymbol{c}$ by uniformly sampling one action c from C at a time, allowing non-consecutive repetitions of actions along $\boldsymbol{c}$. The sampling stops when $\sum_{c\in\boldsymbol{c}}\lambda_c > T$. The first difference from [16] is that we discard the obtained $\boldsymbol{c}$ if it does not include all actions from C, and proceed to randomly generating a new $\boldsymbol{c}$. Thus, for every sampled C, we generate a number of legal $\{\boldsymbol{c}\}$ that include all actions from C and satisfy $\sum_{c\in\boldsymbol{c}}\lambda_c \leq T$. The uniform sampling of sets and sequential generation of their respective legal action sequences is repeated until the total number of legal action sequences is $K = \{\boldsymbol{c}\} = 1000$.

For every generated legal $\boldsymbol{c}$, as in [16], we use the standard dynamic programming to infer temporal extents $\boldsymbol{l}$ of actions from $\boldsymbol{c}$. The second difference is the evaluation of the new HMM's posterior as in (1), $p(\boldsymbol{c},\boldsymbol{l}|\boldsymbol{f}) = \Big[\prod_{n=1}^{N-1} p(c_{n+1}|c_n)\Big]\Big[\prod_{n=1}^{N} p(l_n|c_n)\Big]\Big[\prod_{t=1}^{T} \frac{p(\boldsymbol{f}[c_{n(t)},t])}{p(c_{n(t)})}\Big]$. Out of K candidate sequences, as the final solution, we select $(\boldsymbol{c}^*,\boldsymbol{l}^*)$ that gives the maximum posterior $p(\boldsymbol{c}^*,\boldsymbol{l}^*|\boldsymbol{f})$.

As in [16], complexity of our inference is $O(T^2|\mathcal{C}|K)$, where T is the video length, $|\mathcal{C}|$ is the number of action classes, K is the number of candidate action sequences from the Monte Carlo sampling.

6. Experiments

6.1. Setup

Datasets. As in prior work [16], evaluation is done on the tasks of action segmentation and action alignment on three datasets, including Breakfast [11], MPII Cooking 2 [19], and Hollywood Extended (Ext) [2].

Breakfast consists of 1,712 videos with 3.6 million frames showing a total of 48 human actions for preparing breakfast. Each video has on average 6.9 action instances. For Breakfast, we report mean of frame accuracy (Mof) over the same 4-fold cross validation as in [11].

MPII Cooking 2 consists of 273 videos with 2.8 million frames showing 67 human cooking actions. We use the standard split of training and testing videos provided by [18]. For evaluation on MPII Cooking 2, as in [18], we use the midpoint hit criterion, *i.e.* the midpoint of a correct detection segment should be within the ground-truth segment.

Hollywood Ext consists of 937 video clips with 800,000 frames showing 16 different classes. Each video has on average 2.5 action instances. We perform the same 10-fold cross validation as in [2]. For evaluation on Hollywood Ext, we use the Jaccard index, *i.e.* intersection over detection (IoD), defined as IoD $= |GT \cap D|/|D|$, where GT and D denote the ground truth and detected action segments with the largest overlap.

Features. For fair comparison, we use the same unsupervised video features as in [16]. For all three datasets, features are the framewise Fisher vectors of improved dense trajectories [25]. The Fisher vectors for each frame are extracted over a sliding window of 20 frames. They are first projected to a 64-dimensional space by PCA, and then normalized along each dimension.

Training. We train our model on a total number of 50,000 iterations. In each iteration, we randomly select two training videos that share at least one common action. We set the initial learning rate as 0.01 and reduce it to 0.001 at the 10,000th iteration. Parameters of the dynamic HMM – namely, dynamic transition probabilities, dynamic mean action length, and dynamic class prior per frame – are initialized as for the static model, and then updated after each iteration as in (6), (7), (8), respectively. In practice, it takes around 12 hours for training with one TITAN XP GPU.

Ablations. We consider several variants of our approach for testing how its components affect performance:

- SCV = Our full approach with: the n-pair loss for regularization, the dynamic HMM whose parameters are given by (6), (7), (8), and hard class assignments to frames given by (15) for computing feature distances.
- SCVnoreg = SCV, but without any regularization.
- SCVbasereg = SCV, but with a simplified baseline regularization loss that minimizes only feature distances of shared classes by setting $d^{ac}_{vv'} = d^{cb}_{vv'} = 0$ in (14).
- SCVsoft = SCV, but instead of hard we use soft class assignments to frames given by (16).
- SCVstatic = Our full approach as SCV, but instead of the dynamic HMM we use the static HMM whose parameters are given by (2), (4), (5).

6.2. Action Segmentation

This section presents our evaluation on action segmentation. Tab. 1 shows our results on the three datasets. As can be seen, our SCV outperforms the state of the art by 6.9% on Breakfast, 3.9% on Cooking 2, and 8.4% on Hollywood Ext, respectively. Tab. 1 also shows results of the prior work with stronger supervision in training, where ground truth additionally specifies the temporal ordering of actions, called transcript. Surprisingly, despite weaker supervision in training, our SCV outperforms some recent transcript-supervised approaches, e.g., ECTC [7] on Breakfast, and HMM+RNN [15] and TCFPN [5] on Hollywood Ext. The superior performance of SCV over the state of the art justifies our increase in training complexity.

Fig. 4 illustrates our action segmentation on an example test video *P03_cam01_P03_tea* from Breakfast. As can be

Model	Breakfast (*Mof*)	Cooking2 (*midpoint hit*)	Holl.Ext (*IoD*)
(Set-supervised)			
Action Set [16]	23.3	10.6	9.3
SCV	**30.2**	**14.5**	**17.7**
(Transcript-supervised)			
OCDC [2]	8.9	-	-
HTK [12]	25.9	20.0	8.6
CTC [7]	21.8	-	-
ECTC [7]	27.7	-	-
HMM+RNN [15]	33.3	-	11.9
TCFPN [5]	38.4	-	18.3
NN-Viterbi [17]	43.0	-	-
D3TW [3]	45.7	-	-
CDFL [13]	50.2	-	25.8

Table 1. Weakly supervised action segmentation. Our SCV uses set-level supervision in training, while [2, 12, 7, 15, 5, 17] use stronger transcript-level supervision in training. The dash means that the prior work did not report the corresponding result.

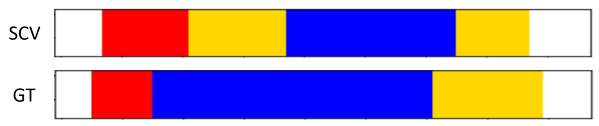


Figure 4. Action segmentation on an example test video *P03_cam01_P03_tea* from Breakfast. Top row: SCV result. Bottom row: ground truth with the color-coded action sequence {take_cup, add_teabag, pour_water}. The background frames are marked white. In general, SCV accurately detects true actions present in videos, but may miss their true ordering and locations.

seen, SCV can detect true actions present in videos, but may miss their true ordering and true locations.

Ablation — Grammar. As explained in Sec. 5, we use Monte Carlo sampling to generate legal action sequences, called Monte Carlo grammar. In Table. 2, we test this component of our approach relative to the following baselines on Breakfast. The upper-bound baselines use a higher level of supervision — transcripts of temporal orderings of actions made available for their inference. Thus, instead of generating the Monte Carlo grammar, the upper-bound baselines explicitly constrain their inference to search the solution only within the transcript grammar consisting of all ground-truth transcripts seen in their training. The lower-bound baselines do not consider legal action sequences in inference, but only predict actions from the provided set of all actions. Tab. 2 shows that SCV outperforms the state of the art for all types of grammars used in inference. From Tab. 2, our Monte Carlo estimation of legal action sequences is reasonable, since the upper-bound SCV with the transcript grammar has an improvement of only 4.5% on Breakfast.

(Grammar) Method	Breakfast (*Mof*) train	Breakfast (*Mof*) test
(No Grammar)		
Action Set [16]	14.7	9.9
SCV	16.6	12.0
(Monte Carlo Grammar)		
Action Set [16]	28.2	23.3
SCV	34.5	30.2
(Transcript Grammar)		
Action Set [16]	36.7	26.9
SCV	40.5	34.7

Table 2. Evaluation of our Monte Carlo sampling for generating legal action sequences (Monte Carlo Grammar) in inference relative to the upper-bound case that uses ground-truth transcripts of temporal orderings of actions (Transcript Grammar) in inference, and the lower-bound case that only predicts actions from the set of all actions in inference. Our SCV is superior for all grammars.

Model parameters	Breakfast (*Mof*)	Cooking 2 (*midpoint hit*)	Holl.Ext (*IoD*)
SCV+ground truth	33.4	15.3	19.2
SCVstatic	28.7	13.7	16.1
SCV	30.2	14.5	17.7

Table 3. Evaluation of SCV when using ground-truth, static, and dynamic model parameters. SCV with ground-truth model parameters an upper-bound baseline.

Model	Breakfast (*Mof*)	Cooking 2 (*midpoint hit*)	Holl.Ext (*IoD*)
SCVnoreg	26.5	12.3	12.6
SCVbasereg	27.8	13.1	14.8
SCVsoft	29.3	14.2	16.1
SCV	**30.2**	**14.5**	**17.7**

Table 4. Evaluation of SCV with different regularizations. SCV gives the best results.

Ablation — Static vs. Dynamic Tab. 3 shows that our SCV with the dynamic HMM outperforms the variant with the static HMM – SCVstatic. The table also shows the upper-bound performance of SCV when the mean action length λ_c and class priors $p(c)$ are estimated directly from framewise ground-truth video labels. From Tab. 3, our dynamic estimation of model parameters in (6), (7), (8) is very accurate, since SCV with the framewise ground truth model parameters has an improvement of only 3.2% on Breakfast.

Ablation — Regularization. Tab. 4 shows our evaluation of SCV when using different types of regularization, including no regularization. As can be seen, SCV with the n-pair loss and the hard class assignments in

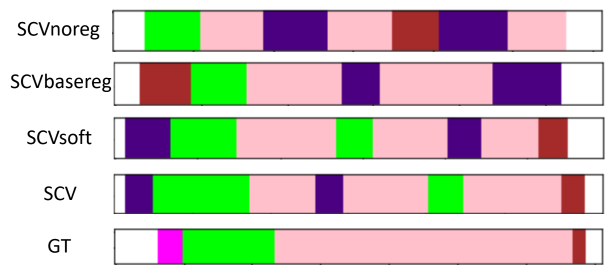


Figure 5. Our ablation of regularization on a sample test video *P03_webcam01_P03_friedegg* from Breakfast. The rows from top to bottom show action segmentation of: SCVnoreg, SCVbasereg, SCVsoft, SCV, and the ground-truth color-coded action sequence {crack_egg,fry_egg,take_plate,put_egg2plate}. The background frames are marked white. SCV gives the best performance.

training outperforms all of the other types of regularization. From Tab. 4, SCV outperforms SCVbasereg, which demonstrates that it is critical to maximize the discrimination margin between features of shared classes and features of non-shared classes, rather than only maximize affinity between features of shared classes. Fig. 5 illustrates our ablation of regularization on an example test video *P03_webcam01_P03_friedegg* from Breakfast. In general, SCV gives the best segmentation results.

6.3. Action Alignment given Action Sets

The task of action alignment provides access to the ground-truth unordered set of actions C present in a test video. Thus, instead of uniformly sampling sets of actions in inference on the test video, for alignment, we take the provided ground-truth C, and continue to perform Monte Carlo sampling of action sequences $\boldsymbol{c}$ that are restricted to come from C, as described in Sec. 5. In comparison with the state of the art, Tab. 5 shows that SCV improves action alignment by 12.4% for Breakfast, 4.5% for Cooking 2, 11.3% for Hollywood Ext. Also, SCV achieves comparable results to some recent approaches that use the stronger transcript-level supervision in both training and testing. These results demonstrate that SCV is successful in estimating an optimal legal action sequence for a given ground-truth action set. This improved effectiveness of SCV over the state of the art justifies our reasonable increase in training complexity. Fig. 6 illustrates action alignment on a sample test video *1411* from Hollywood Ext. From the figure, SCV successfully aligns the actions, but may incorrectly detect their true locations.

7. Conclusion

We have addressed set-supervised action segmentation and alignment, and extended related work that trains their framewise classifiers with a pseudo ground truth generated by independently labeling video frames. We have made two contributions: (1) Set Constrained Viterbi algorithm aimed at generating a more accurate pseudo ground truth for our set-supervised training by efficiently approximating the NP-hard all-color shortest path problem; (2) Regularization of learning with the new n-pair loss aimed at maximizing a distance margin between video features of shared classes and video features of non-shared classes. Our evaluation demonstrates that we outperform the state of the art on three benchmark datasets for both action segmentation and alignment. This justifies an increase of our training complexity relative to that of prior work. Also, our approach gives comparable results to some related methods that use stronger transcript-level supervision in training. Our tests of various ablations and comparisons with reasonable baselines demonstrate effectiveness of individual components of our approach.

Model	Breakfast (*Mof*)	Cooking 2 (*midpoint hit*)	Holl.Ext. (*IoD*)
(Set-supervised)			
Action Set [16]	28.4	10.6	24.2
SCV	**40.8**	**15.1**	**35.5**
(Transcript-supervised)			
ECTC [7]	∼35	-	∼41
HTK [12]	43.9	-	42.4
OCDC [2]	-	-	43.9
HMM+RNN [15]	-	-	46.3
TCFPN [5]	53.5	-	39.6
NN-Viterbi [17]	-	-	48.7
D3TW [3]	57.0	-	50.9
CDFL [13]	63.0	-	52.9

Table 5. Action alignment given ground-truth action sets of test videos. SCV outperforms the state of the art on the three datasets. The transcript-supervised prior work uses stronger supervision in training, yet SCV gives comparable results. The dash means that the prior work did not report the result.

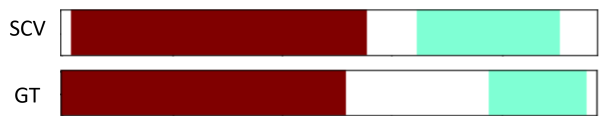


Figure 6. Action alignment on a sample test video *1411* from Hollywood Ext. Top row: SCV result. Bottom row: ground truth with the color-coded action sequence {Run, ThreatenPerson}. The background frames are marked white. In general, SCV successfully aligns the actions, but may incorrectly detect their locations.

Acknowledgement. This work was supported by DARPA N66001-17-2-4029 and DARPA N66001-19-2-4035 grants.

References

[1] Yunus Can Bilge, Dogukan Çagatay, Begüm Genç, Mecit Sari, Hüseyin Akcan, and Cem Evrendilek. All colors shortest path problem. *CoRR*, abs/1507.06865, 2015. 1, 4

[2] 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. 2, 6, 7, 8

[3] 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, 8

[4] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. *arXiv preprint arXiv:1901.02598*, 2019. 2

[5] 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. 2, 6, 7, 8

[6] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9346–9355, 2019. 1

[7] 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. 2, 6, 7, 8

[8] H Jhuang, H Garrote, E Poggio, T Serre, and T Hmdb. A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, volume 4, page 6, 2011. 2

[9] Oscar Koller, Hermann Ney, and Richard Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3793–3802, 2016. 2

[10] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4297–4305, 2017. 2

[11] 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. 6

[12] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 2, 7, 8

[13] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 7, 8

[14] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018. 2

[15] 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. 2, 6, 7, 8

[16] 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. 1, 2, 3, 5, 6, 7, 8

[17] 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. 2, 7, 8

[18] 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. 6

[19] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373, 2016. 6

[20] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *The European Conference on Computer Vision (ECCV)*, 2018. 1

[21] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–171, 2018. 2

[22] 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. 2

[23] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[24] Chen Sun, Sanketh Shetty, Rahul Sukthankar, and Ram Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 371–380. ACM, 2015. 2

[25] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 6

[26] 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. 2

[27] Zhi-Hua Zhou and Min-Ling Zhang. Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology, Beijing, China*, pages 455–459, 2002. 2