Hierarchical Modeling for Task Recognition and Action Segmentation in Weakly-Labeled Instructional 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 [1] focuses on task recognition and action segmentation in weakly-labeled instructional videos, where only the ordered sequence of video-level actions is available during training. We propose a two-stream framework, which exploits semantic and temporal hierarchies to recognize top-level tasks in instructional videos. Further, we present a novel top-down weakly-supervised action segmentation approach, where the predicted task is used to constrain the inference of fine-grained action sequences. Experimental results on the popular Breakfast and Cooking 2 datasets show that our two-stream hierarchical task modeling significantly outperforms existing methods in top-level task recognition for all datasets and metrics. Additionally, using our task recognition framework in the proposed top-down action segmentation approach consistently improves the state of the art, while also reducing segmentation inference time by 80-90 percent.

1. Introduction

Millions of people watch instructional videos online every day, to learn to perform tasks such as cooking or changing a car tire. Also, new models of assistant robots [1] can learn from such videos how to assist humans in their daily lives. Hence, there has been extensive research in recent years on automated understanding of the top-level tasks and their sub-actions in such videos [8, 10, 44, 50].

From a theoretical point of view, instructional videos can be seen as videos illustrating hierarchical activities. Each instructional video illustrates a single top-level activity, for which we use the term "task" throughout this paper. Examples of such video-level tasks are "making coffee" or "cooking eggs". Each video-level task is composed of a sequence of lower level activities, such as "pouring milk" or "adding sugar". Throughout the paper, we will refer to such lower-level activities using the term "action". Consequently, using

---
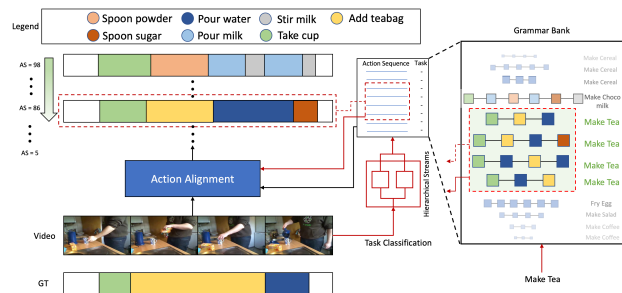[1]https://github.com/rezaghoddoosian/Hierarchical-Task-Modeling

Figure 1. The proposed segmentation approach. Initially, the weak understanding of duration leads to the alignment of irrelevant actions with an Alignment Score (AS) of 98. However, a prior prediction of the task allows the alignment module to infer the correct sequence of actions (outlines by red) despite the lower AS (86).

this terminology, each instructional video illustrates a task that consists of a sequence of actions.

For instructional videos, and hierarchical activity videos in general, we would like to have automated systems that both recognize the overall task and also understand what lower-level actions take place, and when those actions start and end. Fully-supervised training would require not only annotating the top level task, but also marking the start and end frame of each lower-level action. With the ever-growing size of instructional video datasets, manually annotating such start and end frames can quickly become a bottleneck. To address this issue, *weakly-supervised action segmentation* methods require, as training data, only the sequence of actions that takes place at each video, and no start/end frame information for those actions [4, 6, 7, 27, 42].

Our goal in this paper is to jointly address the problems of top-level task recognition and lower-level action segmentation, in the weakly supervised setting (given sequences of actions, not given start/end frames). The main novelty is a method for top-level task recognition that uses, in parallel, two different hierarchical decompositions of the problem. One module models the semantic hierarchy between the top-level task and lower-level attributes. These attributes correspond to either the set of actions, or the set of the

object/verb components of those actions, e.g., "take" and "cup" in the action "take cup". This module jointly learns to identify the presence of attributes in the video and to recognize the top-level task based on the estimated attributes.

A parallel second stream models the temporal hierarchy between the entire video and equal-duration subdivisions of the video. Tasks are usually performed in a relative order of stages, and some stages are particularly useful for distinguishing tasks from each other. For example, preparing tea typically involves three stages: taking a cup, adding a tea bag, and pouring water from the kettle. The first and last stages are visually similar with corresponding stages of the "preparing coffee" task. The temporal hierarchy module can capture the importance of adding the tea bag in distinguishing the "preparing tea" task from the "preparing coffee" task. This module learns the relation between stages and their importance in classifying the video task.

We also propose a novel top-down approach for action segmentation (i.e., frame-level action labeling), that combines our task recognition method with existing weakly-supervised segmentation methods. In this approach (Fig.1), the video-level task is estimated first, and is subsequently used to constrain the search space for action segmentation. In summary, the contributions of this paper are these:

1) We introduce a two-stream framework that exploits both semantic and temporal hierarchies to recognize tasks in weakly-labeled instructional videos.

2) We provide specific, non-trivial implementations of these two streams. Our ablation studies demonstrate that our implementation choices have a significant impact on performance. A highlight of such an implementation choice is using TF-IDF weights to model the discriminative power of each attribute for each task (see Section 3.2.2).

3) We present a novel top-down approach for weakly-supervised action segmentation, where the video-level task is used to constrain the segmentation output.

4) We present results on two benchmark datasets: Breakfast [21] and MPII Cooking 2 [44]. In top-level task recognition, our method significantly outperforms the state of the art on both datasets for all metrics. For weakly supervised action segmentation (frame-level labeling), applying the proposed top-down approach on top of existing methods [27, 42] again leads to state of the art results, and also cuts the inference time by 80-90 percent.

2. Related Work

Instructional Video Analysis. In recent years, untrimmed instructional videos have been studied in areas like video retrieval [8, 31, 32], quality assessment [9, 36], future action planning [5], and key-step segmentation [10, 24, 28, 41, 44, 50, 56]. Fully-supervised action segmentation methods [12, 22, 23, 26, 39, 43, 46] learn to identify action segments in the presence of frame-level

ground-truth. For example, [50] use a bottom-up technique to aggregate initial action proposal scores to classify the top-level video task, before modifying its preliminary frame labels in a fully supervised way. Also, [44] analyze a host of holistic and regional features to train shared low-level classifiers to recognize tasks and detect fine-grained actions.

Recently, unsupervised learning of instructional videos has seen increased attention [10, 11, 24, 45]. In [24], an unsupervised approach performs video segmentation and task clustering through learned feature embeddings. In [10], a network is trained using only video task labels, for unsupervised discovery of procedure steps and task recognition.

The above-mentioned methods are either fully supervised or unsupervised, and thus they are not direct competitors for our method, which uses weak labels.

In the scope of activity recognition, most works [13, 18, 53] study short-range or trimmed videos. Our work is closest to [16, 17, 54], where the focus is recognizing minutes-long activities. However, unlike them, our paper is on instructional videos, and on how recognition can aid segmentation, so it relies on hierarchical activity labels (top-level task, lower-level attributes as targets for segmentation).

Weakly-Supervised Key-Step Localization. In the context of weakly-labeled instructional videos, many methods [2, 25, 30, 57] are trained under the supervision of narration and subtitle. Directly relevant to our work are [4, 6, 7, 15, 27, 40, 42], where, as in our method, only the sequence of actions is known for each training video. In particular [27, 42] deploy a factorized probabilistic model to tackle the segmentation problem using dynamic programming. Also, [4] formulate a differential dynamic programming framework for end-to-end training of their model.

Recent weakly-supervised segmentation methods [4, 6, 27, 42, 48, 47] are formulated to identify the action taking place at each frame, and not the top-level video task. At the same time, the output of these methods implicitly specifies the top-level task, because only one task is compatible with the detected sequence of actions. We use these implicit task predictions of [27, 42] to compare those methods to ours on task recognition accuracy. In contrast to these bottom-up approaches (going from actions to task), our method explicitly learns to classify video-level tasks, and this classification is used in a top-down fashion (from task to actions) to constrain the detected action sequence.

We should also mention the methods in [33, 35, 37, 52], which perform weakly-supervised action *detection*. These methods identify and localize occurrences of, typically, a single action in the input video. For completeness, we evaluate extensions of these methods to task classification.

3. Hierarchical Task Modeling Method

In this section, we present an overview of our two-stream hierarchical task modeling. Full details of our implementa-

tion choices and architecture are provided in Sec. 3.2.

As our formulation uses many terms and symbols, the supplementary material provides a glossary of terms and a table of all symbols we use.

3.1. Method Overview

Problem Definition. The training set $\mathbb{V} = \{\mathbf{v}_i\}_{i=1}^N$ consists of N videos $\mathbf{v}_i$. From each $\mathbf{v}_i$ we extract a feature vector $\mathbf{x}_i \in \mathbb{R}^{F \times T_i}$, that consists of T_i frames of F-dimensional features. We denote by $\mathbb{C} = \{c_i\}_{i=1}^{|\mathbb{C}|}$ the set of all top-level task labels, and by $\mathbb{A} = \{a_j\}_{j=1}^{|\mathbb{A}|}$ the set of all lower-level attribute labels. As an implementation choice, these attributes can be the set of actions in the dataset, or the set of verb/object components of those actions. Each video $\mathbf{v}_i$ is labeled by a task $c_i \in \mathbb{C}$, and also by a set $\mathbb{A}_i \subseteq \mathbb{A}$ of M_i attributes, so that $\mathbb{A}_i = \{a_{i,j}\}_{j=1}^{M_i}$. At test time, given an input video, the system estimates the top-level task.

Semantic Hierarchy Stream (SHS). To recognize the task, one approach is to directly estimate $p(c_i|\mathbf{x}_i)$. However, this approach is prone to overfitting when the number of video samples per task is limited. As attributes can be shared among tasks, the average number of training videos per attribute is typically greater than the average number of videos per task. Using attribute information also helps the model learn similarities and differences of spatio-temporal patterns in different tasks.

Thus, we model task recognition as $p(c_i|\psi_i^a).p(\psi_i^a|\mathbf{x}_i)$, where ψ_i^a is an intermediate vector of attribute scores that is computed for each $\mathbf{x}_i$. The system learns a mapping function $\mathcal{M}_x^a : \mathbb{R}^{F \times T_i} \rightarrow \mathbb{R}^{|\mathbb{A}|}$, that maps each vector $\mathbf{x}_i$ to attribute score vector ψ_i^a. It also learns a function $\mathcal{M}_a^c : \mathbb{R}^{|\mathbb{A}|} \rightarrow \mathbb{R}^{|\mathbb{C}|}$, that maps each attribute score vector ψ_i^a to a task score vector ψ_i^c (Fig.2).

Temporal Hierarchy Stream (THS). Tasks in instructional videos are usually performed in a relative order of steps. Understanding the task-discriminative stages of a video is essential in distinguishing tasks that share similar-looking actions. Thus, we divide each video into K stages of equal duration, and train a classifier $\mathcal{S}_\kappa : \mathbb{R}^{F \times \frac{T_i}{K}} \rightarrow \mathbb{R}^{|\mathbb{C}|}$ for each stage κ. The system also learns an aggregation function $\mathcal{T} : \mathbb{R}^{K|\mathbb{C}|} \rightarrow \mathbb{R}^{|\mathbb{C}|}$, that maps stage-wise predictions to classification scores $\vartheta_{i,total}$ of the entire video.

Stream Fusion Module. In the end, we fuse the predictions of the SHS and THS streams to output the final task prediction scores f_i^c of the entire model. A high-level diagram of the overall network is shown on Fig.2. The network is optimized using a loss function for the fusion module, as well as separate loss functions for the SHS and THS streams.

3.2. Detailed Architecture

In this section we explain in detail the architecture of our two-stream hierarchical model (Fig.3), and we derive the
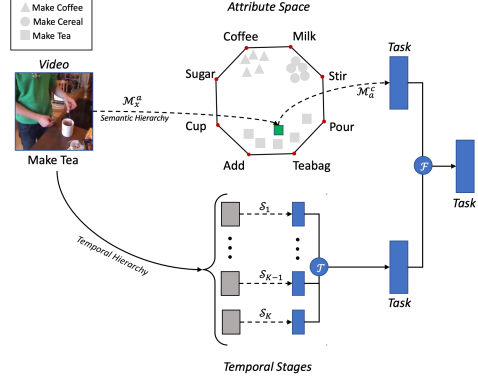

Figure 2. An overview of how our recognition model exploits the semantic and temporal hierarchies of tasks. The attribute representation of videos are formed by their discriminative attributes.

three proposed loss functions.

3.2.1 Feature Extraction

Video task recognition is highly dependent not only on motion patterns, but also on object appearance. Ignoring object appearance can lead to misclassifications when the motion patterns of two tasks are very similar, e.g., making coffee and making tea. Hence, instead of the mostly motion-based iDT features[51] used in [4, 27, 42], we adopt the I3D network, pre-trained on the Kinetics dataset[3]. I3D extracts, for each frame, 1024-dimensional feature vectors respectively from the RGB and optical flow channels. We use PCA separately on RGB and flow features, to reduce the dimensions from 1024 to 128.

The 256-dimensional concatenated RGB and optical flow features of each frame are stored in video-level feature vector $\mathbf{x}_i \in \mathbb{R}^{256 \times T_i}$, where T_i is the total number of frames in video $\mathbf{v}_i$. In principle, any spatio-temporal network can be used instead of I3D. In the supplementary material, we show that I3D outperforms iDT for task recognition.

3.2.2 Semantic Hierarchy Loss

In order to obtain a data-specific representation of video-level feature vector $\mathbf{x}_i$, we pass each frame-level subvector of $\mathbf{x}_i$ through a fully-connected layer g with bias and output dimension of 256, then apply temporal convolution to the output of g. Using such 1D temporal convolutions with a set of F learnable kernels $\mathbf{k}_\phi \in \mathbb{R}^{L \times 256}$ of size L, $\mathbf{x}_i$ is eventually mapped to a F-dimensional feature encoding $\phi(\mathbf{x}_i) \in \mathbb{R}^{F \times T_i}$.

Let $|\mathbb{A}|$ be the number of unique attributes in the dataset. We pass each frame-level subvector of $\phi(\mathbf{x}_i)$ through a fully connected layer with bias and output dimension of $|\mathbb{A}|$ to obtain $\Psi_i^a \in \mathbb{R}^{|\mathbb{A}| \times T_i}$, which is the sequence of attribute
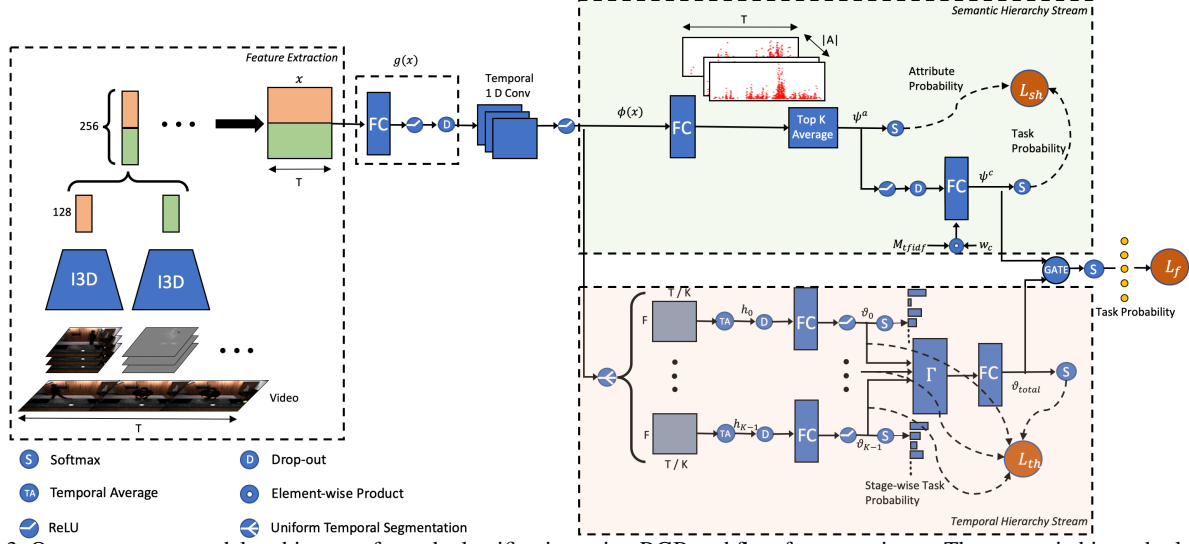
Figure 3. Our two-stream model architecture for task classification using RGB and flow frames as input. The semantic hierarchy loss $\mathcal{L}_{sh}$ ensures task classification after clustering videos based on their shared weighted attributes through the TF-IDF mask $\mathbf{M}_{tfidf}$. The THS stream learns to aggregate stage-wise task predictions by $\mathcal{L}_{th}$, and a third loss ($\mathcal{L}_f$) optimizes the fused results of both streams.

scores for each of the T_i frames. Ψ_i^a is also known as *temporal class activation map* (T-CAM)[34].

Similar to [37, 38], we compute a video-level attribute score vector ψ_i^a by average-pooling the highest $k_i = \lfloor \frac{T_i}{s} \rfloor$ T-CAM scores of each attribute separately over time, where s is a hyperparameter:

$$\psi_i^a[j] = \frac{1}{k_i} \sum_{i=0}^{k_i-1} \text{top}_k\{\Psi_i^a[j,:]\} \tag{1}$$

Intuitively, these selected k_i scores highlight the most important parts of a task in video i.

To denote the set $\mathbb{A}_i$ of attributes present in video $\mathbf{v}_i$, we define a multihot ground-truth attribute vector $\vec{a}_i \in \{0,1\}^{|\mathbb{A}|}$, where for every $j \in \{0,1,...,|\mathbb{A}|-1\}$, $\vec{a}_{i,j} = 1$ if $a_j \in \mathbb{A}_i$, otherwise $\vec{a}_{i,j} = 0$. However, this representation fails to capture the fact that different attributes have different levels of relevance for recognizing each task. For example, attributes "take out" and "open" are present in most videos, and thus not discriminative. As a second example, for the "preparing avocado" task, "avocado" is a more informative attribute compared to "knife".

Inspired by text retrieval methods [19, 29], we compute the TF-IDF weight matrix $\mathbf{W}_{tfidf} \in \mathbb{R}^{|\mathbb{A}| \times |\mathbb{C}|}$, so that $\mathbf{W}_{tfidf}(j,\tau)$ captures the importance of attribute j for task τ. Initially, we formulate $\text{TF} \in \mathbb{R}^{|\mathbb{A}| \times |\mathbb{C}|}$ and $\text{IDF} \in \mathbb{R}^{|\mathbb{A}|}$ as follows:

$$\text{TF}(j,\tau) = \frac{\sum_{i=0}^{N-1} \vec{a}_{i,j} \cdot \mathbb{1}(\tau = c_i)}{\sum_{i=0}^{N-1} \mathbb{1}(\tau = c_i)} \tag{2}$$

$$\text{IDF}(j) = \log\left(\frac{|\mathbb{C}|}{|\{\tau \in \mathbb{C} | \text{TF}(j,\tau) > 0\}|}\right) \tag{3}$$

where $\mathbb{1}()$ denotes the indicator function and $\text{TF}(j,\tau)$ and $\text{IDF}(j)$ are, respectively, the percentage of times attribute

a_j is present in videos of task $\tau \in \mathbb{C}$, and the log inverse of percentage of all tasks that entail attribute j in at least one of their videos. We then define the elements $\mathbf{W}_{tfidf}(j,\tau)$ of the TF-IDF weight matrix as:

$$\mathbf{W}_{tfidf}(j,\tau) = \frac{\text{TF}(j,\tau) \cdot \text{IDF}(j)}{\epsilon + \sum_{k=0}^{|\mathbb{A}|-1} \text{TF}(k,\tau) \cdot \text{IDF}(k)} \tag{4}$$

with ϵ set to a very small value to avoid division by zero.

Using these TF-IDF weights, we intorduce the TF-IDF-weighted attribute ground-truth vector $\vec{a^w}_i \in \mathbb{R}^{|\mathbb{A}|}$ as:

$$\vec{a^w}_{i,j} = \frac{\vec{a}_{i,j} \cdot \mathbf{W}_{tfidf}(j,c_i)}{\sum_{k=0}^{|\mathbb{A}|-1} \vec{a}_{i,k} \cdot \mathbf{W}_{tfidf}(k,c_i)} \tag{5}$$

We also define a TF-IDF mask $\mathbf{M}_{tfidf} \in \mathbb{R}^{|\mathbb{A}| \times |\mathbb{C}|}$, where $\mathbf{M}_{tfidf}(j,\tau)$ is 1 if the corresponding TF-IDF weight $\mathbf{W}_{tfidf}(j,\tau)$ is nonzero, otherwise it is 0. We use the TF-IDF mask to form a mapping from attribute score vectors ψ_i^a to task probability scores $\hat{\psi}_i^c \in \mathbb{R}^{|\mathbb{C}|}$, as:

$$\hat{\psi}_i^{\,c} = \mathfrak{s}[(\mathbf{w}_c \odot \mathbf{M}_{tfidf}^T)\text{ReLU}(\psi_i^a)] \tag{6}$$

In the above, $\mathfrak{s}[]$ and $\odot$ mean the softmax and element-wise product operations respectively, and $\mathbf{w}_c \in \mathbb{R}^{|\mathbb{C}| \times |\mathbb{A}|}$ are weights to be learned. Using the TF-IDF mask allows the model to focus only on relevant attributes for each task.

Let $\vec{c}_i$ be the one-hot task ground-truth vector and $\hat{\psi}^a = \mathfrak{s}[\psi^a]$. The semantic hierarchy loss $\mathcal{L}_{sh}$ is then defined as:

$$\mathcal{L}_{sh} = -\lambda \mathbb{E}[\vec{a^w}_i^T log(\hat{\psi}_i^a)] - (1-\lambda)\mathbb{E}[\vec{c}_i^T log(\hat{\psi}_i^{\,c})] \tag{7}$$

$\mathbb{E}$ denotes "expected value", and λ is a design parameter that decides how fast each term is trained comparatively.

3.2.3 Temporal Hierarchy Loss

We model the temporal hierarchy by dividing each video into K stages of equal duration d, and training a classifier for each stage. Formally, given the frame-level feature encoding $\phi(\mathbf{x}_i)$ of video i, we define $h_{i,\kappa} \in \mathbb{R}^F$ as the feature summary of the κ-th stage and produce unnormalized task scores (logits) $\vartheta_{i,\kappa} \in \mathbb{R}^{|\mathbb{C}|}$:

$$h_{i,\kappa} = \frac{\sum_{t=\kappa d}^{[(\kappa+1)d]-1} \phi(\mathbf{x}_i)[:,t]}{d} \qquad (8)$$

$$\vartheta_{i,\kappa} = \mathbf{w}_\kappa h_{i,\kappa} + b_\kappa \qquad (9)$$

where $\mathbf{w}_\kappa \in \mathbb{R}^{|\mathbb{C}| \times F}$ and $b_\kappa \in \mathbb{R}^{|\mathbb{C}|}$ are parameters of each stage. During the training process, for each stage, the loss function $\mathcal{L}_\kappa = \mathbb{E}[\bar{c}_i^T log(\hat{\vartheta}_{i,\kappa})]$ is defined on the softmax of the stage-wise task prediction logits $\hat{\vartheta}_{i,\kappa}$.

Stage Aggregation Function. As mentioned earlier, certain stages of a task are more discriminative than others. We define the auxiliary function $\Gamma([\vartheta_0 \; \vartheta_1 \; ... \; \vartheta_{K-1}])$, that maps stage-level task score vectors to a video-level task score vector. While training, $\Gamma(\;)$ randomly masks out one of its K input prediction vectors ϑ_κ entirely and multiplies the rest of the input predictions by $\frac{K}{K-1}$. $\Gamma(\;)$ acts similar to the spatial drop out operation by promoting independence between predictions of each stage, to avoid overfitting to a single stage. We form our stage aggregation function using $\Gamma(\;)$ and aggregation parameters $\mathbf{w}_{total} \in \mathbb{R}^{K|\mathbb{C}| \times |\mathbb{C}|}$ to produce video-level task probability values $\hat{\vartheta}_{i,total} \in \mathbb{R}^{|\mathbb{C}|}$:

$$\hat{\vartheta}_{i,total}^T = \Gamma(\text{ReLU}([\vartheta_{i,0}^T \; \vartheta_{i,1}^T \; ... \; \vartheta_{i,K-1}^T])) \, \mathbf{w}_{total} \qquad (10)$$

Finally, we present our temporal hierarchy loss $\mathcal{L}_{th}$ to incorporate the aggregated and stage-wise predictions:

$$\mathcal{L}_{th} = -\mathbb{E}[\bar{c}_i^T log(\hat{\vartheta}_{i,total})] - \sum_{\kappa=0}^{K-1} \mathcal{L}_\kappa \qquad (11)$$

3.2.4 Stream Fusion Loss

We explore three different mechanisms for fusing the predictions of the SHS and THS streams, to produce the final task prediction logits f_i^c. We provide experimental results of each in Section 5.2.

Average Fusion. Here, we treat results of semantic and temporal hierarchies equally, and we backpropagate the same gradient to both streams at training time.

$$f_i^c = 0.5(\vartheta_{i,total} + \psi_i^c) \qquad (12)$$

Weighted Average Fusion. Here, the final prediction is a linear combination of streams, whose predictions are weighted by learned weights $\mathbf{w_1}, \mathbf{w_2} \in \mathbb{R}^{|\mathbb{C}| \times |\mathbb{C}|}$. .

$$f_i^c = \mathbf{w_1}\vartheta_{i,total} + \mathbf{w_2}\psi_i^c \qquad (13)$$

Task-wise Switching Gates. Sometimes, wrong predictions of one stream can negatively impact the final fused classification scores f_i^c at test time. We introduce task-wise switching gates to allow the system enough freedom to learn, for each task independently, to switch between stream predictions. We define switching gate $\alpha = \sigma(w_\alpha)$ as the sigmoid function $\sigma()$ of learnable parameters $w_\alpha \in \mathbb{R}^{|\mathbb{C}|}$. The sigmoid function makes sure our gates stay in the range of 0 to 1. Then, for training and test time, f_i^c is defined as:

$$f_i^c = \begin{cases} \alpha \odot \vartheta_{i,total} + (1-\alpha) \odot \psi_i^c, & \text{training} \\ \mathcal{H}_{0.5}(\alpha) \odot \vartheta_{i,total} + (1 - \mathcal{H}_{0.5}(\alpha)) \odot \psi_i^c, & \text{test} \end{cases} \qquad (14)$$

where $\mathcal{H}_x()$ denotes the Heaviside step function shifted to x. At test time, given task τ, our final prediction is discretely chosen from the SHS stream if $\alpha_\tau < 0.5$ or is selected from the THS stream otherwise.

In cases of the weighted average fusion and switching gates, our fusion loss $\mathcal{L}_f = -\mathbb{E}[\bar{c}_i^T log(\hat{f}_i^c)]$ is added to the previous losses to form our final loss $\mathcal{L}$ with the design parameter β. Finally we train the whole network end-to-end but stop the gradients of $\mathcal{L}_f$ flowing back to the streams to isolate the fusion module from the rest.

$$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_{sh} + \beta\mathcal{L}_{th} \qquad (15)$$

4. Top-Down Action Segmentation

We now present our top-down segmentation approach. In the segmentation problem, the goal is to partition a video temporally into a sequence of S action labels $\boldsymbol{\delta}_1^S$ and their corresponding durations $\mathbf{l}_1^S$. The input in our approach is a video of T_v frames, represented by $\mathbf{x}_1^{T_v}$, as the sequence of per-frame features. Let $\Pi(\tau)$ be the set of all action sequences in the training set given the top-level task τ. Then, grammar $\boldsymbol{\pi} \in \Pi(\tau)$ lists an ordered sequence of S action labels taking place in the video of task τ. The goal is to identify the most likely sequence of action labels $\bar{\boldsymbol{\delta}}_1^S$ and their durations $\bar{\mathbf{l}}_1^S$ associated with a specific grammar $\boldsymbol{\pi}$:

$$(\bar{\boldsymbol{\delta}}_1^S, \bar{\mathbf{l}}_1^S, \bar{\tau}) = \underset{\boldsymbol{\delta}_1^S, \mathbf{l}_1^S, \tau}{\text{argmax}} \; p(\boldsymbol{\delta}_1^S, \mathbf{l}_1^S, \tau | \mathbf{x}_1^{T_v}) \qquad (16)$$

$$= \underset{\boldsymbol{\delta}_1^S \in \Pi(\tau), \mathbf{l}_1^S, \tau}{\text{argmax}} \; p(\mathbf{x}_1^{T_v}|\boldsymbol{\delta}_1^S)p(\mathbf{l}_1^S|\boldsymbol{\delta}_1^S)p(\boldsymbol{\delta}_1^S|\tau)p(\tau) \qquad (17)$$

where $p(\mathbf{x}_1^{T_v}|\boldsymbol{\delta}_1^S)$ is modeled by a neural network and the Bayes rule as in [40, 42], and $p(\mathbf{l}_1^S|\boldsymbol{\delta}_1^S)$ is any given duration model, e.g., Poisson[42] or DurNet[14].

Eq.17 is formulated similarly to the probabilistic model in [42]. However, we explicitly integrate the task variable $\tau \in \mathbb{C}$ into this equation, which dictates the choice of the fine-grained action sequence $\boldsymbol{\delta}_1^S$. Specifically we introduce the task model $p(\tau)$ as the probability output of a task classification network. Without loss of generality, we used the output of our two-stream hierarchical network $\hat{f}^c$, so that $p(\tau) = 1$ for the predicted task $\tau = \text{argmax}(\hat{f}^c)$ and

$p(\tau) = 0$ otherwise. In [42], the task is a by-product of the inferred segments $(\overline{\boldsymbol{\delta}}_1^S, \overline{\mathbf{l}}_1^S)$. In contrast, our proposed approach eliminates all segmentations whose inferred actions do not belong to $\Pi(\tau)$ of the predicted task τ by setting $p(\boldsymbol{\delta}_1^S|\tau)$ to 0 for those segmentations, and to 1 otherwise. We follow the Viterbi algorithm in [42] to solve Eq. 17.

5. Experiments

We compare our method to several existing methods on two popular instructional video datasets, both for task classification and for action segmentation using weakly-labeled videos as training. Further, in ablation studies we evaluate the contribution of each component of our model.

Datasets. 1) *The Breakfast Dataset (BD)* [21] consists of around 1.7k untrimmed cooking videos of few seconds to over ten minutes long. There are 48 action labels demonstrating 10 breakfast dishes with a mean of 4.9 actions per video, and the evaluation metrics are conventionally calculated over four splits. 2) *The MPII Cooking 2 (C2)* [44] has training and test subject-wise splits of 201 and 42 long and high quality videos respectively. Particularly, these videos are 1 to 40 mins long adding to 27 hours of data from 29 subjects who prepare 58 different dishes. This dataset offers different challenges compared to the *BD* dataset for two main reasons; First, the annotated 155 objects and 67 actions (verbs) are extremely fine-grained, so that there are on average 51.8 non-background action segments per video. Second, despite the great number of frames in the dataset, the number of samples per class is unbalanced and limited.

Metrics. We evaluate task classification performance using two metrics: 1) *t-acc* is the standard mean task accuracy over all videos. 2) *t-mAP* denotes the mean Average Precision of task predictions. *mAP* is used in [44] to assign soft class-wise scores to give insight about how far off the wrong predictions are. Further, we use four metrics as [7] to measure the segmentation results: *acc* and *acc-bg* are the frame-level accuracies with and without background frames, while *IoU* and *IoD* define the average non-background intersection over union and detection, respectively.

Implementation. We extracted I3D features on the *C2* dataset using TV-L1 optical flow [55] on a moving window of 32 frames with stride 2, and the pre-computed I3D features of the *BD* dataset were obtained from [12]. We noticed that it is not necessary to process the whole video. Instead, we followed the sampling strategy in [37] to maintain the length of the videos in a batch to be less than a pre-defined length $T \approx 9$ mins while training. This approach speeds up training, lowers memory demands, and applies temporal augmentation. Also, we divided videos into $K = 3$ stages for the THS stream (analysis in Section 5.2).

Our model is trained with a batch size of 10 using the Adam [20] optimizer with 10^{-3} learning rate and 0.005

Table 1. Task classification results of state-of-the-art methods on two main datasets. Best results reported out of I3D ([†]) or iDT ([‡]) features (more in supplementary material and [49]).* results obtained using the author's source code. [24] results for 10 classes.

Supervision	Models	Breakfast (%)		Cooking (%)	
		t-acc	t-mAP	t-acc	t-mAP
Full	Rohrbach *et al.*[44]	-	-	-	57.40
Unsupervised	CTE[24]	31.80[‡]	-	-	-
Weak	NNViterbi[42]*	70.98[‡]	-	23.80[†]	-
	CDFL[27]*	74.86[‡]	-	28.57[†]	-
	W-TALC[37]*	76.19[†]	80.98[†]	33.33[†]	43.07[†]
	3C-Net[33]*	75.23[†]	80.99[†]	30.95[†]	46.30[†]
	Timeception[16]*	76.37[†]	80.80[†]	21.43[†]	25.14[†]
	VideoGraph[17]*	78.70[†]	-	23.80[†]	-
	Our Method	**80.04**	**86.36**	**45.24**	**54.49**

weight decay for 20k iterations. For both datasets, we adjust λ to 0.9, and β is set to 0.25 and 0.01 for the *BD* and *C2* datasets, respectively. The 1D convolutions are done with $F = 64$ as the number of kernels, and $L = 15$ as their size. $s = 8$ and we use a drop-out keep rate of 0.3. The set of verbs and objects are used as our attributes.

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

We used the standard dish labels in both datasets as task labels. All experiments on the *BD* dataset for all models, except the unsupervised CTE [24], were done for 9 tasks after we combined the two dishes of *frying* and *scrambling eggs* as the top-level task of *making eggs*, because both share almost the same set of actions. For CTE, we report the results on the original 10 classes, as given by the authors. We note that CTE is unsupervised and not a direct competitor.

Task Classification. Table 1 shows quantitative results on task recognition, for our method as well as other methods that use different types of supervision. Particularly, [27, 42] are the state-of-the-art open-source weakly-supervised segmentation methods. They implicitly identify the task corresponding to the inferred sequence of actions during inference. Our explicit task modeling significantly outperforms them in accuracy by around 5 to 9 percent on the *BD* dataset, and by 16 to 21 percent in the 58 tasks of the *C2* dataset.

Originally, [33, 37] are the state-of-the-art open-source weakly-supervised methods with specific loss functions to classify and localize sparse action instances in videos. To compare with them, we trained both to classify tasks. Also, [16] and [17] classify tasks in long videos by training multi-scale temporal convolutions and graph based representations respectively. Both networks make heavy use of memory and suffer from overfitting specifically in the *C2* dataset, where using low-level attributes is key. While such direct task modelings under weak supervision prove to be more effective than the implicit classification using fine-grained action segments [27, 42], our hierarchical approach outperforms all competitors considerably in all metrics and

Table 2. Consistent performance gain in weakly-supervised action segmentation following our proposed top-down approach. I3D and iDT features used for experiments on the *C2* and *BD* datasets, respectively. (* as specified in Table 1, **: no source code).

Models	Breakfast (%)				Cooking (%)			
	acc	acc-bg	IoU	IoD	acc	acc-bg	IoU	IoD
TCFPN[7]*	38.4	38.4	24.2	40.6	26.9	30.3	9.5	**17.0**
D3TW[4]**	45.7	-	-	-	-	-	-	-
DP-DTW[6]**	50.8	-	**35.6**	45.1	-	-	-	-
NNViterbi[42]*	43.6	42.5	27.8	39.2	23.5	21.2	7.7	10.9
CDFL[27]*	50.2	50.4	33.5	45.6	29.9	32.2	11.0	13.8
NNViterbi+Ours	46.2	46.1	30.2	42.2	26.9	25.0	9.6	12.7
CDFL+Ours	**51.4**	**52.0**	34.5	**46.7**	**31.3**	**34.5**	**12.8**	15.6
CDFL+GT	*59.8*	*63.0*	*41.3*	*55.2*	*35.0*	*39.7*	*14.4*	*17.6*

Table 3. Inference run time (minutes) improvement of state-of-the-art following the proposed top-down approach for segmentation.

Models	Breakfast (split 4)	Cooking
NNViterbi[42]	100	840
CDFL[27]	144	1070
NNViterbi+Ours	21	64
CDFL+Ours	25	110

datasets. Table 1 shows our *t-mAP* on the *C2* dataset comes close to the fully-supervised baseline [44], which is trained on frame-level action ground-truth. Comparison results on 10 classes of the *BD* dataset are in the supp. material.

Action Segmentation. Table 2 shows results for action segmentation. In our experiments, we combined our two-stream task prediction framework on top of the state-of-the-art weakly-supervised segmentation methods [27, 42] and achieved new state-of-the art results on both datasets, manifested more vividly in *acc-bg*, because background frames are independent of the task. Therefore, excluding background frames highlights the contribution of the correct task label in segmentation. This consistent improvement in all metrics, while decreasing the inference time by 80-90% (Table 3), demonstrates the potential of the proposed top-down approach for weakly-supervised segmentation. Moreover, *CDFL+GT* in Table 2 represents CDFL segmentation results constrained by ground-truth task labels, which serves as an upper bound for our proposed top-down model.

Table 4. Stream-specific ablation study for task classification.

Stream	Breakfast (%)		Cooking (%)	
	t-acc	t-mAP	t-acc	t-mAP
Semantic hierarchy	73.1	77.2	42.9	52.6
Temporal hierarchy-stage 1	62.7	-	16.7	-
Temporal hierarchy-stage 2	68.9	-	28.6	-
Temporal hierarchy-stage 3	64.7	-	23.8	-
Temporal hierarchy-aggregated	80.0	86.4	31.0	45.2
Two streams fused	**80.0**	**86.4**	**45.2**	**54.5**

Table 5. The effect of sharing attributes (a) and using TF-IDF weights (b) in the SHS stream (Cooking dataset).

(a)			(b)		
Setting	t-acc	t-mAP	Setting	t-acc	t-mAP
Without attributes	33.3	45.4	Without TF-IDF	38.1	49.4
With attributes	**45.2**	**54.5**	With TF-IDF	**45.2**	**54.5**

Table 6. Evaluation of our model under different choices of K on the *BD*.

# Stages	t-acc	t-mAP
1	74.2	82.1
2	79.6	86.4
3	80.0	86.4
4	80.0	86.4
5	79.2	85.8

Table 7. Comparison between different fusion mechanisms on the *BD*.

Fusion Type	t-acc	t-mAP
Average	77.2	84.2
Weighted Average	78.4	84.9
Switching Gate	**80.0**	**86.4**

5.2. Analysis and Ablation Study in Task Modeling

Stream-Specific Results. We evaluated the contribution of the SHS and THS streams separately in Table 4. The SHS stream is more effective on the *C2* dataset because of two main reasons: First, the average number of videos per task (3.4) is low compared to that of videos per attribute (28.4), so any direct way of task modeling is prone to overfitting. Second, the large number of attributes per task allows the learning of a discriminative attribute-to-task mapping. Meanwhile in the THS stream, despite the weak classification power of the stage-specific classifiers, our hierarchical modeling is able to aggregate stage-wise predictions effectively and produce significantly superior results. This shows that different stages provide complimentary information. Note that the THS stream alone achieves state-of-the-art on the *BD* dataset with only task label supervision.

Semantic Hierarchy Ablation. As shown in Table 5a, removing the attributes from the semantic hierarchy loss (Eq.7), and directly classifying tasks from the feature encoding $\phi(\mathbf{x})$, leads to around 12% drop in *t-acc* on the *C2* dataset. Simply sharing low-level attributes among tasks is beneficial, and using the TF-IDF weights led to an additional 7% difference in *t-acc* (Table 5b).

Temporal Hierarchy Analysis. Modeling tasks as a temporal hierarchy of multiple stages improves the performance compared to the single-stage approach. Furthermore, as indicated by Table 6, such an approach is not sensitive to the number of stages ($K > 1$) in the hierarchy. This concludes that these stages provide complimentary information for the stage-aggregation function regardless of their exact positioning or duration in the video.

Comparison of the Stream Fusion Mechanisms. Table 7 compares different mechanisms to fuse the predictions of SHS and THS streams. Specifically, the Task-wise Switching Gates are trained to identify the stronger stream per task and perform best, while the vanilla and weighted averaging compromise between both streams and produce sub-optimal

Figure 4. Qualitative task classification results of our THS stream on sample challenging videos from the *BD* dataset. For each video, we show the ground-truth task label, predictions from the single-stage baseline, and the stage-wise and stage-aggregated recognition result. Also, stage-wise accuracy for each of the four tasks is presented on the right side. In the first video, the subject sequentially takes a cup, pours coffee and milk, adds sugar and stirs coffee. In the second video, the subject takes a cup, adds teabag, pours hot water, spoons sugar and stirs. In the third video, the subject takes a cup, pours milk, spoons choc. powder, adds sugar and stirs. In the fourth video, the subject pours cereal, spoons choc. powder, pours milk, and then stirs. Red and green boxes denote wrong and correct predictions, respectively.

results. In the *BD* dataset these gates propagate the results of the THS stream, whereas in the *C2* dataset they switch between both for different tasks and combine predictions (see row 1, 5 and 6 of Table 4).

5.3. Qualitative Results

Fig.4 compares results of the THS stream with the single-stage baseline on four challenging videos from the *BD* dataset. Due to similar-looking actions shared between tasks, the single stage baseline misclassifies the video task, whereas our THS stream outputs the correct task under different stage-wise settings. Specifically, in the first and third videos, our model classifies the task correctly although only one of the stage predictions is correct. For example, according to the stage-wise accuracy of the task *making chocolate milk* in Fig.4, the last stage of this task is the most discriminative one. Thus, our model learns to put more weight on the predictions of this stage, which compensates for the first two stages outputting the wrong class of *making cereal*.

The two tasks of *making coffee* and *making tea* share similar-looking actions in the first and second videos, so analyzing the entire video in one step produces wrong predictions for both cases. The second video, in particular, provides an interesting case where similar visuals of the action *taking cup* between tasks of *making chocolate milk* and *making tea* led to confusion of the first stage. Also, the later two stages mistakenly predicted the task of *making coffee*,

because the two actions of *pouring* and *stirring* are shared between both tasks of *making coffee* and *making tea*. Although all three stage-wise predictions are wrong, the aggregated result of those stages is correct. This shows that the proposed hierarchical model not only considers the predicted class of each stage, but also learns the relationship between stages and their fine-grained prediction scores.

In a given stage, the short discriminative part may be dominated by the longer ambiguous section. For example, the final stage of the last video depicts how *stirring* while occluding the bowl dominates the shorter and more discriminative action of *pouring milk*. This effectively resembles the appearance and motion of *flipping a pancake by spatula*, but the complementary information of the first two stages eventually results in the correct aggregated recognition.

6. Conclusion

We have introduced a two-stream framework, that exploits semantic and temporal hierarchies to recognize tasks in weakly-labeled instructional videos. We have also proposed a novel top-down segmentation approach, where the predicted task constrains the fine-grained action labels. We report experimental results on two public datasets. Our two-stream task recognition method outperforms existing methods. Similarly, our top-down segmentation approach improves the accuracy of existing state-of-the-art methods, while simultaneously improving runtime by 80-90%.

References

[1] 'Robotic kitchen assistant on a rail', May 2020. [Online]. Available: `https://www.roboticsresear.ch/articles/19606/robotic-kitchen-assistant-on-a-rail`. [Accessed: 15- November- 2020].

[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] 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.

[4] 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.

[5] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. *arXiv preprint arXiv:1907.01172*, 2019.

[6] Xiaobin Chang, Frederick Tung, and Greg Mori. Learning discriminative prototypes with dynamic time warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8395–8404, 2021.

[7] 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.

[8] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. Action modifiers: Learning from adverbs in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 868–878, 2020.

[9] Hazel Doughty, Walterio Mayol-Cuevas, and Dima Damen. The pros and cons: Rank-aware temporal attention for skill determination in long videos. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7862–7871, 2019.

[10] Ehsan Elhamifar and Dat Huynh. Self-supervised multi-task procedure learning from instructional videos. *European Conference on Computer Vision*, 2020.

[11] Ehsan Elhamifar and Zwe Naing. Unsupervised procedure learning via joint dynamic summarization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6341–6350, 2019.

[12] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.

[13] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[14] Reza Ghoddoosian, Saif Sayed, and Vassilis Athitsos. Action duration prediction for segment-level alignment of weakly-labeled videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2053–2062, 2021.

[15] 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.

[16] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2019.

[17] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. *arXiv preprint arXiv:1905.05143*, 2019.

[18] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2000–2009, 2019.

[19] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

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

[21] 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.

[22] 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.

[23] 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.

[24] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12066–12074, 2019.

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

[26] 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.

[27] 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.

[28] Jun Li and Sinisa Todorovic. Set-constrained viterbi for set-supervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10820–10829, 2020.

[29] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[30] 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.

[31] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[32] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE international conference on computer vision*, pages 2630–2640, 2019.

[33] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8679–8687, 2019.

[34] 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.

[35] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5502–5511, 2019.

[36] Paritosh Parmar and Brendan Tran Morris. What and how well you performed? a multitask learning approach to action quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–313, 2019.

[37] 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.

[38] Maheen Rashid, Hedvig Kjellstrom, and Yong Jae Lee. Action graphs: Weakly-supervised action localization with graph convolution networks. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 615–624, 2020.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8368–8376, 2018.

[46] 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.

[47] Yaser Souri, Yazan Abu Farha, Fabien Despinoy, Gianpiero Francesca, and Juergen Gall. Fifa: Fast inference approximation for action segmentation. *arXiv preprint arXiv:2108.03894*, 2021.

[48] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. Fast weakly supervised action segmentation using mutual consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[49] Yaser Souri, Alexander Richard, Luca Minciullo, and Juergen Gall. On evaluating weakly supervised action segmentation methods. *arXiv preprint arXiv:2005.09743*, 2020.

[50] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019.

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

[52] 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.

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[54] Tianshu Yu, Yikang Li, and Baoxin Li. Rhyrnn: Rhythmic rnn for recognizing events in long and complex videos. In *European Conference on Computer Vision*, pages 127–144. Springer, 2020.

[55] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007.

[56] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. *arXiv preprint arXiv:1703.09788*, 2017.

[57] 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.