Weakly-Supervised Action Segmentation with Iterative Soft Boundary Assignment

Li Ding*
Massachusetts Institute of Technology
liding@mit.edu

Chenliang Xu
University of Rochester
chenliang.xu@rochester.edu

Abstract

In this work, we address the task of weakly-supervised human action segmentation in long, untrimmed videos. Recent methods have relied on expensive learning models, such as Recurrent Neural Networks (RNN) and Hidden Markov Models (HMM). However, these methods suffer from expensive computational cost, thus are unable to be deployed in large scale. To overcome the limitations, the keys to our design are efficiency and scalability. We propose a novel action modeling framework, which consists of a new temporal convolutional network, named Temporal Convolutional Feature Pyramid Network (TCFPN), for predicting frame-wise action labels, and a novel training strategy for weakly-supervised sequence modeling, named Iterative Soft Boundary Assignment (ISBA), to align action sequences and update the network in an iterative fashion. The proposed framework is evaluated on two benchmark datasets, Breakfast and Hollywood Extended, with four different evaluation metrics. Extensive experimental results show that our methods achieve competitive or superior performance to state-of-the-art methods.

1. Introduction

One of the major challenges in video understanding [7, 21, 19, 23, 24, 20, 5] is to localize and classify human actions in long, untrimmed videos, of which usually the requirement is to predict per-frame semantic labels. Recently, many approaches [22, 9, 12, 17, 4] have been introduced to address this problem in fully-supervised setting, relying on the densely-annotated video data such as [8, 2]. However, such data is usually too tedious to obtain, which makes these methods not scalable under real-world condition. Furthermore, annotating the precise temporal boundaries between actions is also a challenge for humans; the resulting data will be inconsistent and less likely to be relied on for learning in larger scale.
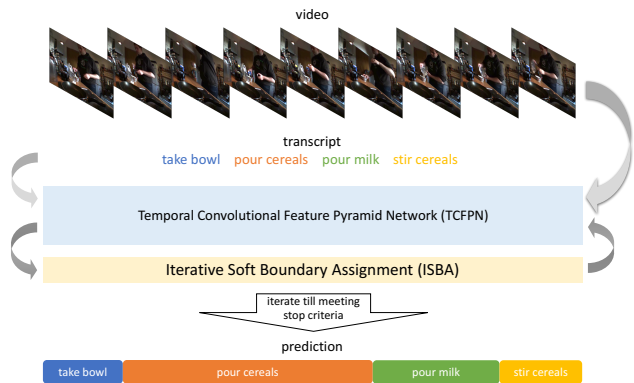

Figure 1. Overview of the proposed framework with Iterative Soft Boundary Assignment (ISBA) and Temporal Convolutional Feature Pyramid Network(TCFPN).

As a result, there is an increasing attention on methods [16, 10, 6, 2] that focus on solving the problem under weaker supervision, *e.g.* with *action transcripts*. In this case, an *action transcript* refers to a set of action units organized by their occurrence ordering in a video without precisely locating their temporal boundaries. This kind of labeling is much easier to obtain, and can even be automatically generated from other meta information [1, 11, 14]. Our paper aligns with this set of weakly-supervised works and assumes only access to action transcripts of training.

However, current methods stick to use Recurrent Neural Networks (RNN) to encode the video data and use different algorithms, such as Extended Connectionist Temporal Classification (ECTC) [6] and Hidden Markov Model (HMM) [16, 10], to go through all the possible action sequences and find the one with maximal likelihood. Both parts, i.e., network and learning model, suffer from expensive computational cost. For example, in [16], videos have to be cut into small chunks to enable RNN training. Besides, [16, 10] also introduce iterative learning approaches that train until converge, which do not consider the condition of overfitting.

---

*Work performed when the author was with University of Rochester.

To overcome the above collective limitations, we propose a novel action segmentation framework for weakly supervised learning, which consists of a new temporal convolutional network, named Temporal Convolutional Feature Pyramid Network (TCFPN), for predicting frame-wise action labels, and a novel training strategy, named Iterative Soft Boundary Assignment (ISBA), to align action sequences and update the network in an iterative fashion (see Fig. 1 for an overview). Both parts feature efficient and highly parallelizable computation, e.g., not using any recurrency or Markovian process. We also propose a specific stop criteria that can effectively evaluate the current training process and prevent overfitting.

Concretely, the ISBA allows us to iteratively train a temporal segmentation network with the training target generated from action transcript, and refine the action transcript based on the inference of current network. Each time, the proposed ISBA looks at the boundary frames between two actions and decides an update to the action boundary based on the current inference result. This leads to increased efficiency than other methods [16, 6], because unlike these methods that try to find the optimal action sequence with maximal likelihood after going through all the candidates during each iteration, ISBA uses a simple-yet-effective algorithm and tries to form a reasonable probabilistic distribution of different actions. Such process takes the idea from Expectation-Maximization (EM) algorithm [3], but features a more randomized behavior.

We also propose a soft boundary mechanism that puts weaker supervision on the boundary frames, which can further help the network find most discriminative patterns for learning an action model. Our proposed stop criteria is based on video-level recognition loss and prevents a network from overfitting to the iterative training in ISBA.

In addition, we propose TCFPN, a convolution-based neural network that achieves state-of-the-art performance on temporal modeling with exceeding speed than RNN-based methods. It has a pyramid structure to make use of both low-level and high-level features, with lateral connection [13] to reduce computation cost.

For evaluation, we use two benchmark datasets, Breakfast and Hollywood Extended, both of which are among the largest datasets for evaluating action segmentation. Four different metrics are used to comprehensively evaluate the performance. Extensive experimental results show that our methods achieve competitive or superior results than state-of-the-art methods on both datasets.

Our contributions are three-fold. First of all, we propose ISBA training strategy with novel stop criteria for weakly-supervised action segmentation and alignment. The ISBA is independent from a chosen segmentation network. Secondly, we propose TCFPN, a new temporal convolutional network for action segmentation. Thirdly, our whole system combining TCFPN and ISBA achieves state-of-the-art performance on weakly-supervised action segmentation and alignment, with exceeding efficiency and scalability.

2. Related Work

We organize the related work on video action modeling into two sets: fully-supervised methods and weakly-supervised methods, and discuss them next.

Fully-Supervised Methods. Many existing works in this category use frame-level features as input and then build temporal models on the whole video sequence. Yeung *et al*. [22] propose an attention LSTM network to model the dependencies of the input frame features in a fixed-length window. Singh *et al*. [17] present a multi-stream bi-directional recurrent neural network for fine-grained action detection task. Kuehne *et al*. [9] introduce an end-to-end generative framework for action segmentation using the HTK system, and focus on the part of feature extraction. Lea *et al*. [12] devise two temporal convolutional networks for action segmentation and detection, transforming successful approaches from speech recognition. Ding *et al*. [4] further introduce a hybrid temporal convolutional and recurrent network that also learns action ordering, but suffers high computation cost.

Weakly Supervised Methods. A variety of different approaches have been explored for the task of weakly supervised action labeling. Bojanowski *et al*. [2] formulate the temporal assignment problem and propose Ordering Constrained Discriminative Clustering (OCDC), with the introduction of Hollywood extended dataset. Huang *et al*. [6] propose ECTC which enforces the action alignments to be consistent with frame-wise visual similarities. Kuehne [10] use HMM to model the action and set the ground truth to be the sequence that maximizes the likelihood of all possible sequences. They iteratively refine the segmentation. Following a similar pipeline, Richard *et al*. [16] propose an iterative fine-to-coarse sub-action modeling mechanism with RNN and HMM. Our method is in this category. Comparing to previous works, our method runs fast and achieves state of the art performance, which we show in Sec. 5.

3. Temporal Convolutional Feature Pyramid Network (TCFPN)

In order to counter the weakly supervised action segmentation task, a good temporal segmentation network is essential to the whole system. We set the requirements for a *good* segmentation network for our task: (1) it should be able to learn from the coarse ground truth; and (2) it needs to run efficiently enabling the iterative training process. As a result, we adopt the Encoding-Decoding Temporal Convolutional Network (ED-TCN) [12] as our baseline and propose an improved structure: TCFPN. We achieve this by adding
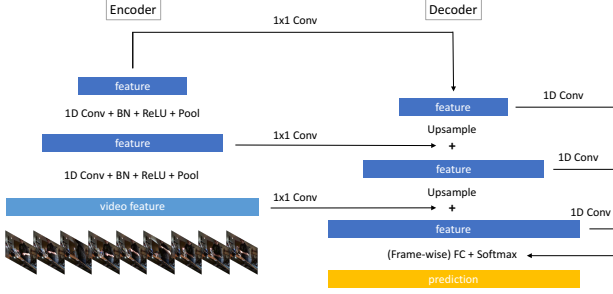
Figure 2. Structure overview of TCFPN. The proposed network extends the original ED-TCN [12] by adding lateral connections [13] between encoder and decoder.
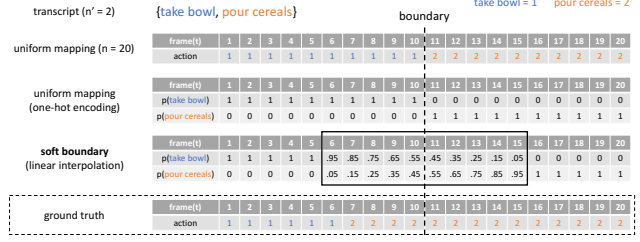

Figure 3. Example case of the proposed soft boundary mechanism. By using linear interpolation of probabilities, the temporal boundary between two different actions becomes coarse, which makes the generated target more robust and reliable.

the lateral connection mechanism proposed in [13], which, we note, is for a different task, i.e., object detection. We adapt it here for action segmentation. Figure 2 shows an overview of the structure of our proposed network.

TCFPN retains an encoder-decoder architecture. Both encoder and decoder parts consist of K layers of features. We define the encoding layers as $L_E^{(i)}$, and the decoding layer as $L_D^{(i)}$, for $i = 1, 2, \ldots, K$, where $K > 0$ is the depth parameter that can vary based on the size of dataset. Empirically, we set $K = 3$ for all of our experiments.

In the encoder part, $L_E^{(1)}$ is the video feature extracted from each frame. For $i > 1$, each layer $L_E^{(i)}$ is the output of a combination of operations, *i.e.*, temporal (1D) convolution, batch normalization, ReLU nonlinearity, and max pooling across time.

In the decoder part, $L_D^{(1)}$ is computed from $L_E^{(K)}$ with a 1×1 convolution with a desired number of filters, which serves mainly as dimension reduction. For $i > 1$, each layer $L_D^{(i)}$ is computed by the lateral connection with $L_E^{(K+1-i)}$, which is the element-wise sum of up-sampled $L_D^{(i-1)}$ and the result of applying 1×1 convolution on $L_E^{(K+1-i)}$. Note that all the 1×1 convolution operations in the decoder part have the same number of filters. This is required for element-wise sum in lateral connection and can reduce the dimension of high-level features. The key idea of decoder part is to add high-level semantic information into low-level dense feature maps.

Finally, each decoder layer $L_D^{(i)}, i = 1, 2, \ldots, K$ is operated by another temporal convolution to reduce the aliasing effect of upsampling. A frame-wise fully-connected layer with softmax activation is used to output the class probabilities at each time step for all K layers. The final prediction is averaged through all these layers. Such design naturally combines coarse, semantically-strong features with fine, semantically-weak features in a pyramidal hierarchy, with little extra computation expense.

4. Iterative Soft Boundary Assignment (ISBA)

In this section, we introduce the overall procedure of our weakly supervised learning mechanism, named Iterative Soft Boundary Assignment (ISBA), which mainly consists of two parts, target generation from transcript and iterative transcript refinement. An overview of the whole procedure is shown in Fig. 4.

Given a video of *making cereals* as an example, in the training process, we are given the ground-truth action transcript, *i.e.* {*take_bowl, pour_cereals, pour_milk, stir_cereals*}. Notice that, we have only access to the labels and their orders, but not their temporal boundaries in a video. The goal is to localize each action unit in the training video with the given transcript (weakly-supervised action alignment), as well as to predict the actual frame-wise labels for unseen testing videos (weakly-supervised action segmentation).

4.1. Target Generation with Soft Boundary Assignment

We start with a linear mapping from the action transcript to video frames. Given a video of length n and its action transcript of length n', usually the linear mapping assign each action label in the transcript $\{A_1, A_2, ..., A_{n'}\}$ to its corresponding n/n' frames as a hard assignment and thus form a target with length n. However, this is not an ideal setting for weakly-supervised tasks because such mapping may fail to serve as a good target as the actual lengths of actions vary. In order to encounter this problem, we propose a soft boundary assignment mechanism to set the target. The soft boundary between different actions is defined by a temporal linear interpolation of the probabilities of two labels. As shown in Fig. 3, after uniform initialization with linear mapping and soft boundary assignment, the current target is a sequence with mixed probabilities that generated from linear interpolation. That is to say, we set a *coarse* boundary between different action units to let frames that are close to the boundary have mixed probabilities to have either label.
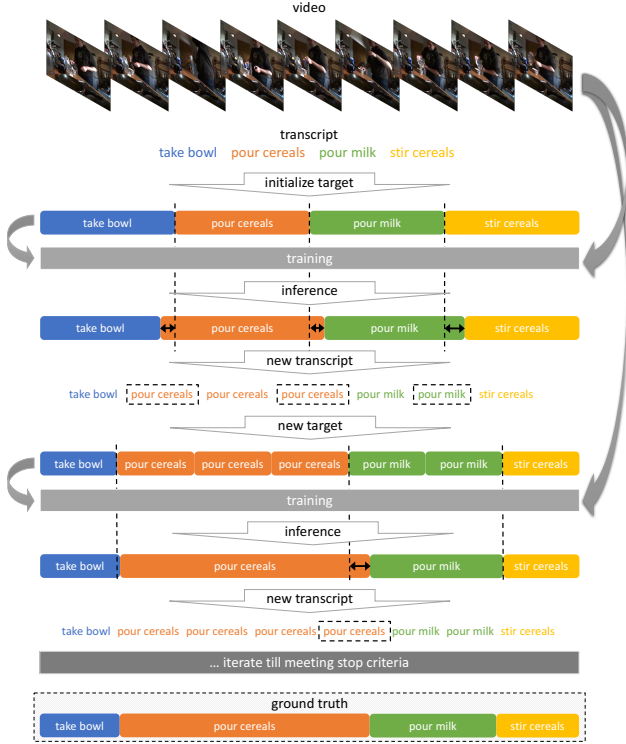
Figure 4. Overview of the proposed Iterative Soft Boundary Assignment (ISBA) strategy. Under weak supervision, we start from the action transcript and initialize the training target by uniform mapping. A temporal segmentation model is then trained and with which we infer the result on the training set. The action transcript is then refined by the training result and the next iteration starts. Finally until meeting the stop criteria, the final model is used to infer on the training set for the alignment task, and predict on the testing set for the segmentation task.

One significant advantage of the proposed soft boundary assignment is that, during training, we guide the model to learn discriminative patterns of an action in a smaller temporal interval that is more likely to have the same real ground-truth action label. It works similar to a weighted loss that does not care too much about the boundary, but more specifically let the model choose from the two consecutive actions and thus preserve the ordering information.

We note that, although our change is simple, it is effective. During experiments, we found such setting usually gives a bonus on performance than simply up-sample transcript to video length by hard assignment. A detailed ablation study is performed in Sec. 5.4. We have also tried other scaling methods such as cubic interpolation, but since the sum of probability should be equal to one as being consistent with the softmax output, and all the probabilities should be greater than zero, the generated target is thus similar to linear interpolation after such normalization.

4.2. Transcript Refinement with Iterative Training and Inference

After each training procedure, a probability sequence is calculated by running inference of the current model on each training sample. As shown in Fig. 5, the predicted probability is likely to be different from the current target. We use an insertion strategy to refine each transcript by utilizing the information learned by the currently-trained model. This operation pushes each action unit to be closer to its possible ground-truth position and its possible length, while preserving their orderings as the weak but exact ground-truth.

Specifically, for a training video of length n and its action transcript of length n', i.e. $\{A_1, A_2, ..., A_{n'}\}$, at each action boundary $i \in [1, n' - 1]$ between different action units, frame $t = (n/n') \cdot i$ is the temporal boundary between action A_i and action $A_{(i+1)}$. If $A_i \neq A_{(i+1)}$, we then observe the inferred probability of both action classes, $P_{A_i}(t)$ and $P_{A_{(i+1)}}(t)$, predicted by the current model at that frame. By setting up a threshold $\rho \in (0, 1)$, if:

$$|P_{A_i}(t) - P_{A_{(i+1)}}(t)| > \rho \ , \qquad (1)$$

we insert an action label chosen from $\{A_i, A_{(i+1)}\}$ with the higher probability into the action transcript at a location corresponds to that boundary. Suppose $P_{A_i}(t) > P_{A_{(i+1)}}(t)$, then A_i is the one to insert and the transcript now becomes $\{..., A_i, \boldsymbol{A_i}, A_{(i+1)}, ...\}$. Alg. 1 describes the whole iterative process in detail. The runtime of this algorithm is almost linear to different actions in the whole dataset, which is the size of all original transcripts. It is very fast with the simple insertion operation.

Besides, in order to make the process more robust, we also include a randomness parameter $\theta \in (0, 0.5]$, according to which the label is chosen from a Bernoulli distribution of the two labels with $p = \theta$. In this case, the probability of insert A_i into transcript is $(1 - \theta)$. This idea is inspired from taking random steps used in deep reinforcement learning. Finally, after using the above method to refine all the training transcripts, the next training iteration starts from the new target generation process with the refined transcript.

The intuition behind this transcript refinement strategy is to iteratively transform the initial action transcript into the *pesudo-supervision*. It can adjust the length of each action unit during every iteration while preserving the ground-truth action orderings. Throughout the refinement, the action transcript becomes less coarse and thus the next target generated from transcript will have *stronger* supervision for training. The most ideal case is that the transcript will get to be more accurate overtime, and at the same time, the training model during each iteration becomes better as the target is more accurate, thus can infer an even better transcript.
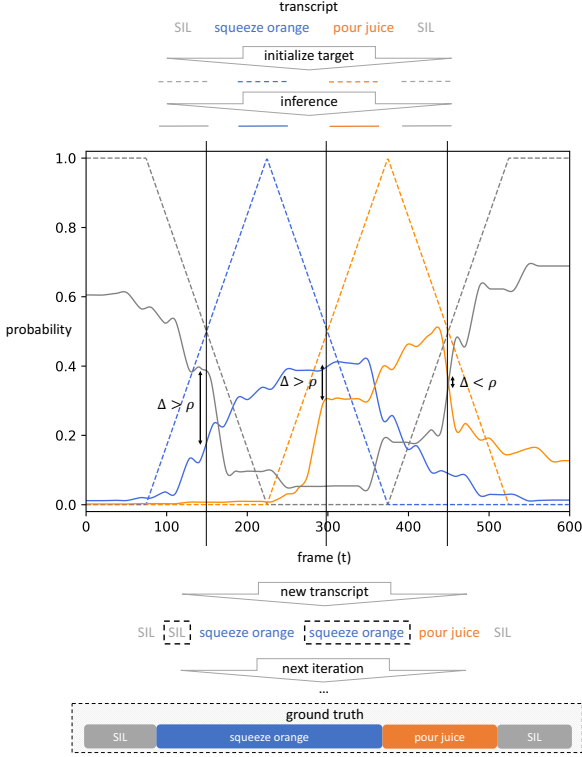
Figure 5. Example case of the proposed transcript refinement method. Given the initial transcript {*SIL, squeeze orange, pour juice, SIL*}, we first generate the target with soft boundary, train the model, and infer the probability. At each temporal boundary between different actions, *i.e.* $t = 150, 300, 450$, if $\Delta > \rho$ where Δ is the difference of probabilities and ρ is a threshold, we then insert the action with higher probability into the transcript at corresponding location. In this case, we insert two actions and the new transcript becomes {*SIL, SIL, squeeze orange, squeeze orange, pour juice, SIL*}. As a result, the transcript is gradually being refined to mimic the unknown ground truth.

4.3. Stop Criteria

A proper stop criteria is essential to our approach, as it is very likely to gradually make the refined training target overfit the behavior of the network, which might not really explain the data, and thus miss to fit the unknown ground truth. Also since ISBA itself features a converging behavior (see Alg. 1), other methods such as threshold on frame change [16] are not ideal for this case because we can always meet that threshold after a certain number of iterations. In this case, we propose another performance monitoring method to solve this problem.

Besides the ordering of actions, what we can also obtain under weak supervision is the occurrence of actions in each video. We utilize this as a measurement to evaluate how the model has learned to recognize actions on video-level. After training and inference, for each training video,

---

Algorithm 1 Transcript Refinement
___
Given $transcript$ in training set
for each $action$ in length($transcript$) **do**
   $t = boundary\_frame$
   if ($action \neq next\_action$) **then**
      if $P_t(action) - P_t(next\_action) > \rho$ **then**
         insert $action$ into $transcript$ at current location
      else if $P_t(next\_action) - P_t(action) > \rho$ **then**
         insert $next_action$ into $transcript$ at current location
      end if
   end if
end for
___

we use a global max-pooling through time to get the maximal probability of each actions in video. We then propose a video-level recognition loss, calculated as the binary cross-entropy loss against the ground-truth occurrence of actions, which can be obtained from the action transcript.

Concretely, given a video of length n and the action set with k classes, the inference result P at frame-level has dimension (n, k). The global max-pooling shrinks P into P' with dimension $(1, k)$ as the maximal probability of each action class throughout the whole video. Given the ground-truth action occurrence for the same sequence as Y, the binary cross entropy loss is calculated as:

$$\mathcal{L}_{\text{video}} = \sum_{i=1}^{k}[Y_i \log(P'_i) + (1 - Y_i) \log(1 - P'_i)] \ . \quad (2)$$

The final loss is averaged over all the samples. In this work, we stop training if this recognition loss does not decrease for three iterations. We then choose the result at the iteration with minimal video-level loss as the final result for a video.

There are several reasons for using such validation. First, it can measure how the model learns to recognize the actions in the video, despite the ordering. It is essential that the model should first recognize the action in order to give the correct ordering and temporal location. Second, its real ground truth can be obtained from transcript under the weakly-supervised setting. Third and most importantly, since we do not use that loss directly for back propagation during training, it is less likely that the model will overfit that loss. Given the fact that TCFPN does classification on each frame, the video-level context is harder to be learned from frame-level loss.

Generally speaking, we use this sequence recognition loss to find out the best condition that the model learns to model discriminative patterns of different actions during the whole iterative training process, although the training targets are generated from different transcripts.
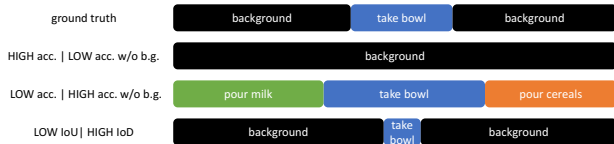
ground truth	background	take bowl	background
HIGH acc. | LOW acc. w/o b.g.	background		
LOW acc. | HIGH acc. w/o b.g.	pour milk	take bowl	pour cereals
LOW IoU| HIGH IoD	background	take bowl	background

Figure 6. Comparison among different metrics.

5. Experiments

In this section, we describe in detail the experimental results. We experiment with three different tasks, e.g., fully-supervised action segmentation, weakly-supervised action segmentation, and weakly-supervised action alignment, on two datasets, e.g., Breakfast [8] and Hollywood Extended [2]. The Breakfast dataset has over 1.7k video sequences of cooking in the kitchen environment with an overall duration of 66.7h. Each video is labeled with a set of 48 action classes with a mean of 4.9 instances per video. The Hollywood Extended dataset contains 937 video sequences from different Hollywood movies. It features a set of 16 different action classes overall and a mean of 2.5 action instances per video. Surpassing [15, 18], these two datasets are among the largest datasets for evaluating action segmentation.

We use four different metrics to evaluate the performance, making it more general and easy to compare with previous and future work on this topic. Besides, we also do an ablation study to further explore the proposed method.

5.1. Metrics

Performance metric is very task- or need-specific for video understanding problems. Most of other works propose or stick to only one or two specific metrics. In this work, we use four different metrics to evaluate qualitative results on all three tasks. Examples of comparison among different metrics are shown in Fig. 6.

Frame-wise accuracy (**Acc.**) is a common metric for action segmentation, which directly evaluates how many frames are correctly labeled. A drawback of this metric is that, if frames in a are dominated by a single class, *e.g. background*, one can achieve high accuracy by simply classifying everything as that class. To overcome the limitations, we propose another metric, frame-wise accuracy without background (**Acc.−b.g.**), which computes the accuracy for all frames except background frames. In this case, the metric emphasizes on evaluating frames with a real action label.

Another set of metrics are Jaccard measures, which observe how the prediction and ground truth overlaps each other. In this work, we obtain results of both intersection over union (**IoU**) and intersection over detection (**IoD**). Given a ground-truth action interval I^* and a prediction interval I, IoU is measured as $|I \cap I^*|/|I \cup I^*|$ and IoD is

Table 1. Results of **fully-supervised action segmentation**. The proposed network TCFPN outperforms state-of-the-art methods on both datasets with all four metrics. (*result obtained from the author of [16]; **our implementation)

Breakfast	Acc.	Acc.−b.g.	IoU	IoD
HTK(64) [9]	56.3	-	-	-
HTK [10]	50.7	-	36.1	-
GRU [16]*	**60.6**	-	-	-
ED-TCN [12]**	43.3	43.6	29.4	42.0
TCFPN	52.0	**52.6**	**36.7**	**54.9**

Hollywood Ext.	Acc.	Acc.−b.g.	IoU	IoD
HTK [10]	39.5	-	8.4	-
ED-TCN [12]**	36.7	27.3	10.9	13.1
TCFPN	**54.8**	**33.1**	**20.4**	**28.8**

measured as $|I \cap I^*|/|I|$. IoU requires accurate temporal segmentation to achieve high score, while IoD requires the prediction to be included in the ground truth. Thus, IoD is more related to the detection task instead of action segmentation. Since previous work [6, 2, 16] tend to use IoD as a measure, we also include the IoD metric for comparison although it is not so relevant to the segmentation task.

5.2. Implementation Details

In order to make the results comparable, for both datasets we obtain features used in previous work [10, 9] from the authors, which are computed using improved dense trajectories (iDT) and Fisher vectors (FV) with PCA and GMM, as described in [9]. Frames are down-sampled to about 1 fps. We will release our implementations along with all parameters upon acceptance to facilitate future research.

5.3. Comparing to State-of-the-Art Methods

We compare the proposed methods under three different tasks to the state-of-the-art. Since other work usually only use one or two metrics, we leave the blank for those missing evaluations.

Fully-Supervised Action Segmentation
We first evaluate the proposed temporal segmentation network TCFPN in fully-supervised setting. We also implement the ED-TCN [12] as our baseline method, which has not been implemented on the two datasets yet. As shown in Table 1, TCFPN outperforms ED-TCN to a large extent and shows competitive results to state-of-the-art methods. The HTK(64) [9] uses the same HTK system as [10], but a better feature with GMM size of 64. Besides, the proposed method does not contain recurrent connections, e.g., LSTMs, and thus the training can be very efficient with highly-parallelized computation.

Table 2. Results of **weakly-supervised action segmentation**. The proposed TCFPN + ISBA outperforms other state-of-the-art methods on both datasets with most of the metrics. (*from [6])

Breakfast	Acc.	Acc.−b.g.	IoU	IoD
OCDC [2]*	8.9	-	-	-
HTK [10]	25.9	-	9.8	-
ECTC [6]	27.7	-	-	-
GRU reest. [16]	33.3	-	-	-
ED-TCN + ISBA	32.0	28.8	18.4	30.6
TCFPN + ISBA	**38.4**	**38.4**	**24.2**	**40.6**

Hollywood Ext.	Acc.	Acc.−b.g.	IoU	IoD
HTK [10]	**33.0**	-	8.6	-
GRU reest. [16]	-	-	11.9	-
ED-TCN + ISBA	27.8	29.4	8.9	10.7
TCFPN + ISBA	28.7	**34.5**	**12.6**	**18.3**

Table 3. Results of **weakly-supervised action alignment** on training set. The proposed TCFPN + ISBA outperforms other state-of-the-art methods on both datasets with most of the metrics. (*from author's plot)

Breakfast	Acc.	Acc.−b.g.	IoU	IoD
OCDC [2]	-	-	-	23.4
HTK [10]	43.9	-	26.6	42.6
ECTC [6]*	∼35	-	-	∼45
GRU reest. [16]	-	-	-	47.3
ED-TCN + ISBA	52.7	50.5	33.5	51.4
TCFPN + ISBA	**53.5**	**51.7**	**35.3**	**52.3**

Hollywood Ext.	Acc.	Acc.−b.g.	IoU	IoD
OCDC [2]	-	-	-	43.9
HTK [10]	49.4	-	29.1	**46.9**
ECTC [6]*	-	-	-	∼41
GRU reest. [16]	-	-	-	46.3
ED-TCN + ISBA	50.3	32.4	26.2	34.8
TCFPN + ISBA	**57.4**	**36.1**	22.3	39.6

Weakly-Supervised Action Segmentation

We evaluate the proposed ISBA strategy with both ED-TCN and TCFPN on the weakly-supervised action segmentation, which is the main task of this work. As shown in Table 2, the proposed TCFPN+ISBA outperform other state-of-the-art methods to a large extent on almost every metric. The only exception is for the frame-wise accuracy on Hollywood Extended dataset, which is due to a large amount of `background` frames. Although the HTK [10] has a better frame-wise accuracy, its IoU is much lower than other state-of-the-art methods and our proposed method. Our method outperforms HTK significantly on Breakfast dataset.

Another observation, when jointly considering the result from previous fully-supervised action segmentation, is that when TCFPN shows competitive or slightly better performance than other methods, TCFPN + ISBA shows much stronger performance on weakly-supervised task. More specifically, for Breakfast dataset, TCFPN is 5.8% worse than GRU on frame-wise accuracy if fully supervised. However when it comes to weakly-supervised task, TCFPN + ISBA is 5.1% better than GRU and HMM with re-estimation [16]. Thus, we regard the proposed ISBA mechanism as the main boost in the weakly supervised action segmentation task.

Weakly-Supervised Action Alignment

We also show the weakly-supervised action alignment task, which aims to align the given transcript to its proper temporal location. This task is in parallel with segmentation because we iteratively refine the transcript and thus have a better segmentation model with improved training target. We report the alignment result on the testing set by iteratively refine the testing scripts for 10 iterations with the best model determined by stop criteria.

As shown in Table 3, the proposed TCFPN+ISBA again shows superior result on Breakfast dataset. For Hollywood Extended dataset, many works choose to use IoD as the only metric to evaluate the detection performance. In this work, we assume the video is well labeled with action transcript and there are less background or meaningless labels. As a result, we achieve a better IoU but a lower IoD, because the network is less likely to label the majority of frames as background.

5.4. Analysis and Ablation Study

In this section, several experiments are done to further evaluate the proposed method. First, we evaluate the speed of the proposed method. Second, we investigate the effectiveness of the proposed soft boundary mechanism. Third, we visualize detailed result of an example training process with all the iterations included, showing how the method learns from weak labels and how the proposed stop criteria works. Experiments give more direct insights about performance and capability of the proposed method.

Speed. On Breakfast dataset, TCFPN+ISBA takes only 4 minutes for one iteration of training including inference and prediction on the training set, averagely around half an hour for the whole training process. Comparing to previous work, OCDC [2] takes about 2 hours to train; HTK [10] takes 23.7 minutes just for segmentation prediction.

Soft Boundary. In addition to experiments described in the previous section, we train the TCFPN+ISBA model without soft boundary on Breakfast dataset. In this case, the transcript is simply repeatedly up-sampled to the video

Table 4. **Ablation study** of weakly-supervised action segmentation and alignment on Breakfast dataset. By adding the soft boundary mechanism, the final results gain improvement on most of the metrics.

Breakfast (seg.)	Acc.	Acc.−b.g.	IoU	IoD
TCFPN + ISBA	**38.4**	**38.4**	**24.2**	40.6
Above w/o soft b.d.	37.8	38.1	24.1	**41.8**

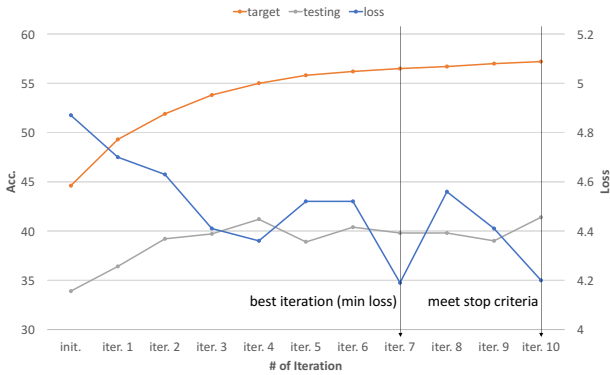| Breakfast (align.)	Acc.	Acc.−b.g.	IoU	IoD
TCFPN + ISBA	**56.7**	**55.9**	**38.7**	**54.0**
Above w/o soft b.d.	55.7	55.8	38.1	53.0

Figure 7. Example training process on the first split of Breakfast dataset. Stop criteria is met at iteration 10 as the loss does not decrease for three iterations from iteration 7, which is the best iteration and used as the final result.

length to generate the training target. The comparison result is shown in Table 4. By adding the soft boundary mechanism, the final results gain improvement on most of the metrics. Although the margin is small, the improvement is consistent and can be easily applied to other weakly-supervised systems. Another benefit from soft boundary is that ISBA converges more quickly.

Overall, the soft boundary mechanism stands for the idea of using a mixture of probabilities to represent the boundary between two different actions. We believe such representation and mechanism are important in the field of action modeling, where you can not always tell the exact changing point of an action, and thus worths further exploration.

Stop Criteria. We visualize the whole training process on the first split of Breakfast dataset. Fig. 7 shows the frame-wise accuracy for both alignment task on training set and segmentation task on testing set, together with the recognition loss used as our stop criteria. One thing to mention is that we can only access the video-level recognition loss during training, among these three values. By using the proposed stop criteria, we can quickly find the best recognition model without extra training.
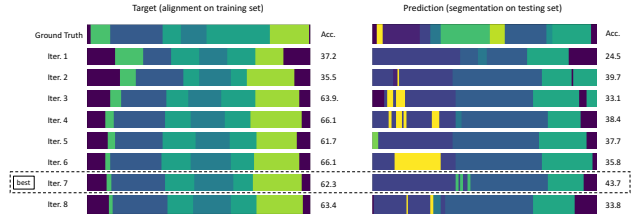

Figure 8. Two example videos in training and testing set of Breakfast dataset during the iterative refinement. On the left-hand-side, the training target starts from a unique mapping and gets refined over time. Note that the targets are probabilities generated with soft boundary, and visualization shows only the label with largest probability. On the right-hand-side, the testing sample is predicted by models trained during each iteration.

As we see, although the alignment accuracy keeps being improved, the testing accuracy begins to converge after a few iterations. The proposed recognition loss aligns well against the actual testing accuracy. We also do additional experiment using the stop criteria from [16] on the first split of Breakfast dataset, resulting 21 iterations with 38.7 Acc., which means almost twice iterations with lower performance, comparing to the 10 iterations with 39.8 Acc. from Fig. 7.

Fig. 8 shows example results on training and testing set, respectively. As the training target gradually being refined to its proper length and location, the testing result also gets better as the model becomes more precise. The proposed method is designed to find useful information throughout the whole dataset in an iterative fashion.

6. Conclusion

In this work, we propose ISBA as a novel strategy for weakly-supervised action segmentation and alignment. It features an iterative training procedure with transcript refinement and soft boundary assignment, together with a video-level loss metric proposed as the stop criteria. We also propose TCFPN, a new temporal convolutional network for supervised action segmentation, which can be fast-trained and shows competitive performance to state-of-the-art methods. The whole system TCFPN+ISBA outperforms state-of-the-art on both weakly-supervised action segmentation and alignment. Our training strategy is general and can be integrated with other work or used in other tasks, to facilitate future research in video action understanding.

References

[1] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 1

[2] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014. 1, 2, 6, 7

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. 2

[4] L. Ding and C. Xu. Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation. *arXiv preprint arXiv:1705.07818*, 2017. 1, 2

[5] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *IEEE International Conference on Computer Vision*, 2017. 1

[6] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *European Conference on Computer Vision*, pages 137–153. Springer, 2016. 1, 2, 6, 7

[7] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *IEEE International Conference on Computer Vision*, 2017. 1

[8] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 1, 6

[9] H. Kuehne, J. Gall, and T. Serre. An end-to-end generative framework for video segmentation and recognition. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE, 2016. 1, 2, 6

[10] H. Kuehne, A. Richard, and J. Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 2017. 1, 2, 6, 7

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[12] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3, 6

[13] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 3

[14] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2929–2936. IEEE, 2009. 1

[15] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 756–763, 2014. 6

[16] A. Richard, H. Kuehne, and J. Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 5, 6, 7, 8

[17] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1961–1970, 2016. 1, 2

[18] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738. ACM, 2013. 6

[19] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[20] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso. Can humans fly? action understanding with multiple classes of actors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[21] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[22] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, pages 1–15, 2015. 1, 2

[23] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[24] L. Zhou, C. Xu, and J. J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2018. 1