A New Dataset and Approach for Timestamp Supervised Action Segmentation Using Human Object Interaction

Saif Sayed* Reza Ghoddoosian* Bhaskar Trivedi Vassilis Athitsos
University of Texas at Arlington
{saififtekar.sayed, reza.ghoddoosian, bhaskarchandra.trivedi}@mavs.uta.edu, athitsos@uta.edu

Abstract

This paper focuses on leveraging Human Object Interaction (HOI) information to improve temporal action segmentation under timestamp supervision, where only one frame is annotated for each action segment. This information is obtained from an off-the-shelf pre-trained HOI detector, that requires no additional HOI-related annotations in our experimental datasets. Our approach generates pseudo labels by expanding the annotated timestamps into intervals and allows the system to exploit the spatio-temporal continuity of human interaction with an object to segment the video. We also propose the (3+1)Real-time Cooking (ReC)[1] *dataset as a realistic collection of videos from 30 participants cooking 15 breakfast items. Our dataset has three main properties: 1) to our knowledge, the first to offer synchronized third and first person videos, 2) it incorporates diverse actions and tasks, and 3) it consists of high resolution frames to detect fine-grained information. In our experiments we benchmark state-of-the-art segmentation methods under different levels of supervision on our dataset. We also quantitatively show the advantages of using HOI information, as our framework improves its baseline segmentation method on several challenging datasets with varying viewpoints, providing improvements of up to 10.9% and 5.3% in F1 score and frame-wise accuracy respectively.*

1. Introduction

Action segmentation is the task of temporally segmenting untrimmed videos and producing an action label for every frame [9, 20, 23, 44]. Fully supervised action segmentation methods require as training data the start and end frame of each action in each training video. However, manually annotating these action boundaries is time-consuming and simply not scalable to large datasets.

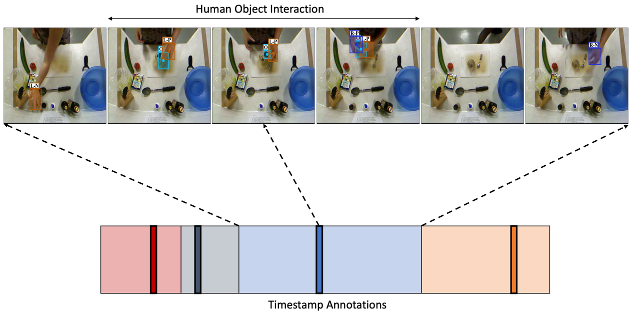

Figure 1. The continuity of human object interaction carries important information about the continuity of an action. The blue bounding boxes in the video indicate the spatial locations of objects that the human is interacting with. In timestamp supervision only one arbitrary frame per action segment is known (indicated by vertical bars in the segmented video), but the action label of that frame can be propagated to neighboring frames based on patterns of human-object interaction around that frame.

To alleviate the manual annotation bottleneck, weakly supervised action segmentation approaches [4, 8, 15, 21, 22, 25, 35, 40] utilize the ordered sequence of action labels present in the video, without specifying the start and end frames of each action. Similarly [11, 26, 36] use action sets to segment the video temporally. While these methods have significantly lighter annotation requirements, they attain much lower accuracy than their fully supervised counterparts. This gap in accuracy has led to an alternative type of supervision called time-stamp supervision [31] where, in addition to the ordered sequence of actions, the training data also contains a single frame number for each action, thus placing significant constraints on when each activity may be happening.

In this paper, we focus on timestamp supervision, given its promising combination of lighter annotation requirements and accuracy that is closer to that of fully-supervised methods. Within that context, we propose extracting and using human object interaction information to improve accuracy. Our approach extends the supervisory signal of single-

*These authors contributed equally to this work

[1] https://github.com/saifsayed/rec-dataset

frame timestamps to intervals around those timestamps, by identifying neighboring frames where human object interaction occurs continuously, and labeling such frames with the same action. Figure 1 illustrates this idea using an example. In that figure, for the action of *add pepper* in a video, the human takes the container, adds the pepper and puts it back. Detecting the time interval of interaction between the human and the pepper container allows us to propagate the *add pepper* action label from the single frame included in the training data to all frames in that interval.

As another contribution of this paper, we introduce and will publicly release the (3+1)Real-time Cooking (ReC) dataset, which consists of synchronized egocentric (1ReC) and three third person (3ReC) view-points. The (3+1)ReC dataset is a 109 hr collection of 1799 diverse videos, where 30 participants prepare 15 breakfast foods and drinks. We refer to these drinks and foods as dish. Our proposed dataset is an effort to create a benchmark to study human object interaction more effectively in instructional videos, and has three main properties: 1) It includes high quality videos with a resolution of 1920×1080. 2) To the best of our knowledge, it is the first public dataset to offer synchronized third person and first person view-points. 3) It incorporates 102 types of actions that are diverse in their motion, appearance, interaction with objects, and duration. In order to encourage further studies on our dataset, we benchmark segmentation results of state-of-the-art (SOTA) for each of weak, full and time-stamp supervision levels. The main contributions of the paper are as follows:

1) A key novelty is the idea of using HOI information to improve action segmentation accuracy. Furthermore, we show that in practice this idea does not require any extra training data for new action recognition datasets. The proposed framework demonstrates the feasibility and benefits of using HOI information in action segmentation.

2) We propose, and benchmark the (3+1)ReC dataset as the first instructional video dataset with synchronized third person and egocentric views. As an advantage, our dataset can be used to study the effect of human object interaction by providing high quality videos and a diverse set of actions.

3) The proposed framework outperforms its baseline action segmentation method using timestamped supervision in four out of five following datasets: 1ReC, 3ReC, 50salads [41], MPII Cooking 2 [37], and GTEA [10]. The system can be applied to varying environments and viewpoints.

In principle, our idea of using HOI information requires additional, HOI-specific training data in order to train an HOI detector. In practice, we have used the same pretrained off-the-shelf HOI detector in all our experimental datasets. Thus, these extra HOI-specific annotations can be treated as a one-off cost (that has already been paid if one uses an off-the-shelf HOI detector), as opposed to being an additional cost for each new action recognition dataset. The source code and extensive documentation will be made public.

2. Related Work

Timestamp Supervised Action Segmentation. Timestamp supervision [1, 29] has recently been explored as a way to bridge the accuracy gap between weakly and fully supervised methods while still not requiring the same annotation burden as full supervision. [31] trained a fine-grained acition classifier by employing a plateau function sampling distribution centered around temporal timestamp annotations. This work showed promising result on action localization for trimmed videos. Later, [24] mined action and background frames to extend the action localization system. Recently, [1] proposed a constrained k-medoids algorithm to generate pseudo-labels. Additionally, Li [29] introduced a timestamp supervision method which uses the model predictions and the annotated timestamps to estimate action change. [29] also proposed a confidence loss that forces model confidence to monotonically decrease as the distance to timestamp increases. The approach of [29] led to improved results compared to weakly supervised methods, and it serves as the baseline in our experiments.

Human Object Interaction. The task of human object interaction(HOI) detection is to localize a human and an object in their respective bounding boxes and then to specify the interaction between them, by outputting a tuple <human bounding box,object bounding box, object class, action class> given an image. This is an active research area [5, 6, 14, 42] and further literature on image based HOI can be found in HOI papers [42].

In the video domain, [13] formulated a Bayesian approach that integrates various perceptual tasks involved in understanding HOI. Also, [17] formulated the problem as a graph where the edges represented affordance and relation between human actions and objects and nodes represented objects. Environment affordance [32] was utilized in applications involving action anticipation [33]. Another method [38] on image-level HOI detection detects hands and objects when they are in contact. That system not only predicts the hand in contact with the object, but also finds the bounding box of the object in contact. This system is technically related to [12] but instead of predicting triplets <human, verb, object>, they propose an alternative representation based on physical contact and interaction. The system is trained to recognize hands and active objects irrespective of object or activity class and thus can be generalized to other domains. However, these approaches work on single images or trimmed videos, and no prior work has used HOI for action segmentation.

The Proposed Method in the Context of Related Methods. With respect to the action segmentation methods discussed above, our method falls under timestamp supervi-

Table 1. Real-time instructional video dataset comparison. * indicates approximation due to a hidden test set. “Views” refers to 3^{rd} person. Some statistical discrepancies between 1ReC and 3ReC is due to frame loss in some videos. “Envir.” includes various camera setups.

Dataset	#Subj.	#Envir.	#Vids	Dur.	#Tasks	#Actions	#Transcripts	Mean Trans. Len.	Mean Vid. Dur.	#Views	Egocentric	Vid. Res.
EPIC 100 [7]	37	45	700	100 hr	NA	$\approx$3.8k*	700	128*	8.6* min	×	✓	1920×1080
GTEA [10]	4	1	28	0.6 hr	7	10	28	33	1.2 min	×	✓	720×404
EGTEA+ [28]	32	1	86	28 hr	7	106	86	239	$>$ 19 min	×	✓	1280×960
CSV [34]	82	1	1940	11.1 hr	14	18	70	9.5	0.3 min	×	✓	1920×1080
Cooking2 [37]	29	1	273	8 hr	58	87	272	95	6 min	1	×	1624×1224
50 Salad [41]	25	1	50	4.5 hr	1	19	50	20	6.4 min	1	×	640×480
Breakfast [18]	52	18	1712	77 hr	10	47	256	6.9	2.3 min	2-5	×	320×240
IKEA [2]	48	5	1113	35 hr	4	33	359	22.7	1.9 min	3	×	1920×1080
1ReC	30	8	450	27 hr	15	102	418	11.7	3.6 min	×	✓	1920×1080
3ReC	30	10	1349	82 hr	15	102	441	11.7	3.6 min	3	×	1920×1080
(3+1)ReC	30	10	1799	109 hr	15	102	444	11.7	3.6 min	3	✓	1920×1080

sion. The key feature differentiating our method from existing action segmentation methods is the use of information from human object interaction. Our method integrates HOI information within the timestamp supervised action segmentation framework of Li *et al.* [29], and the experiments show that using HOI information leads to improved accuracy compared to the original results of [29] in most cases. The proposed method uses an HOI detection module as a black box, so any HOI method can be plugged in. Our implementation uses the off-the-shelf pre-trained system described in [38]. Consequently, our method can be applied to novel action recognition datasets without needing any additional HOI annotations for those datasets.

Real-time Instructional Video Datasets. Among real-time instructional video datasets, [7, 10, 28, 34] focus on only egocentric while [2, 18, 37, 41] include only third person videos. Notably, [39] has pairs of egocentric and third person recordings, but each view-point is recorded separately and video pairs are not synchronized. On the other hand, our proposed dataset contains both egocentric and third person views, which are also synchronized unlike previous work. Among third person datasets, [18] contains low resolution videos, and restricts the performance of human object interaction models. While videos in [37, 41] and [2] are of high quality, they are limited in sample size and diversity of tasks respectively. In comparison, our dataset offers high quality videos, and diverse actions with many instances for each. This allows for a better study of human object interaction in action segmentation and other long-range video understanding problems for the community. Refer to Table 1 for a direct comparison with existing datasets.

3. (3+1)Real-time Cooking (ReC) Dataset

3.1. Data Collection

The (3+1)ReC dataset consists of 1799 ego enteric and third person videos. Specifically, we recorded 30 participants cooking 15 breakfast dishes by three fixed Lorex wifi security cameras and a GoPro HERO 7, where almost[2] all videos are synchronized with a delay of maximum one second (Fig. 2). Following IRB protocols, our human subjects included volunteers and students who were given extra credit for their participation. All videos are muted and recorded with a resolution of 1920×1080. In order to avoid unexpected actions, we instructed all participants to follow our verbal instructions of actions and scripted transcripts to prepare each dish. In total, recording was done in 8 unique kitchen environments using 10 different positional configurations of the camera set. In the supp. material, we provide a detailed list of videos, where we indicate their recorded fps, and whether they lose any frames, and consequently are not in sync with videos of other view points.

3.2. Statistics

The diversity of our dataset stems from cooking 15 different dishes and their constituent 102 low-level actions. Furthermore, each action can be divided into verb and object components resulting in overall 23 verbs and 57 objects. For example, actions *cut lemon* and *pour sugar* take place while making *lemonade*. We made sure all actions are frequently represented in our dataset. In particular, 984 (*take spoon*) and 60 (*pour egg to pan*) are, respectively, the maximum and minimum numbers of samples for an action class in the (3+1)ReC dataset. The duration of videos range from 0.3 to 10.3 mins with a mean of 3.6 mins. There are in total 109 hrs of videos, and 444 unique transcripts with 11.7 actions per transcript on average (Table 1).

3.3. Annotations

Temporal annotations of action segments were done in two stages. Firstly, we divided the videos into three groups, and the label, start and end frames of all action segments in each group of videos were marked by a separate expert annotator. Secondly, the three annotators cross checked labeling of other annotators to remove inconsistencies and mistakes. Also, in order to alleviate annotator subjectivity, an-

[2] About 7% of videos are not in sync due to frame loss

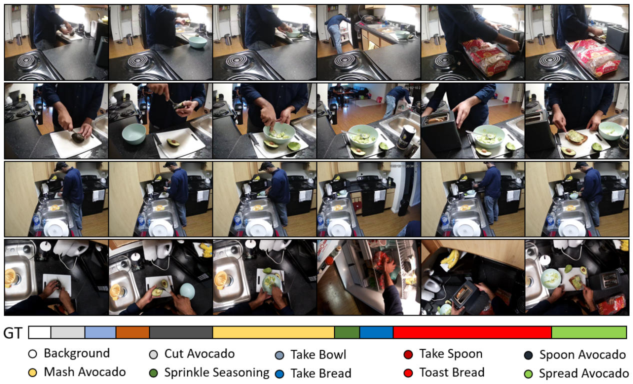

Figure 2. Sample video set from the (3+1)ReC dataset showing synchronized third person and egocentric views. The figure illustrates the sequence of actions taken for making *avocado toast*.

notating each action in videos was done based on when a participant **intends** to start and finish an action. For example, the action *microwave bowl* while making *oatmeal* starts from the moment the subject aims to pick up the bowl, includes the waiting time while the oatmeal is cooked in the microwave, and it ends when the bowl is taken out of microwave and placed on the counter. As a result, recognition of some actions in our dataset requires a long range understanding of context and human object interaction in the video.

4. Temporal Action Segmentation

Given a sequence of video frames $X = [x_1, ..., x_T]$ where T is the length of the video, the goal in temporal action segmentation is to predict action class label $a_{1:T} = [a_1, ..., a_T]$ for each frame. In Section 4.1 we explain the problem formulation for action segmentation using timestamp supervision. In Section 4.2 we describe the proposed framework for learning from timestamp supervision using Human Object Interaction. Then we provide the details of loss function in Section 4.2.3.

4.1. Timestamp Supervision

In a fully supervised setup, each training video $X = [x_1, ..., x_T]$ is accompanied by frame-wise labels $[a_1, ..., a_T]$. However for timestamp supervision, the model is only provided with a single frame annotation per action segment during training. For a training video X containing T frames and N action segments, where $N << T$, labels $A_{TS} = [a_{t_1}, ..., a_{t_N}]$ specify one frame for each of the N segments. It is reported in [30] that it is 6 times faster to annotate a single frame per action than to annotate the start and end frames of each action.

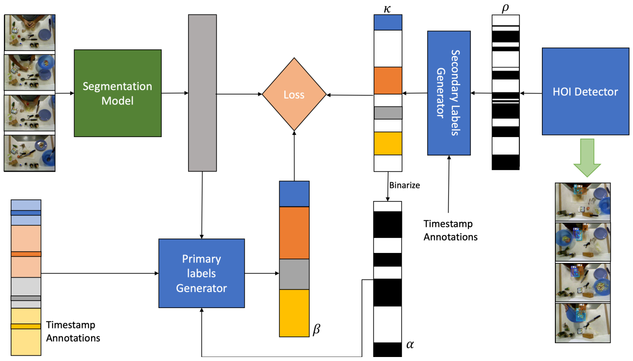

Figure 3. The proposed training framework. The secondary labels generator creates new pseudo ground-truth, κ using the HOI detections ρ and existing timestamp annotations. The binarized pseudo ground-truth(α) also provides new supervisory signal to the primary label generator for generating frame-wise labels β.

4.2. Action Segmentation and HOI

Compared to other weaker forms of supervision such as transcripts (i.e., sequences of actions), timestamps provide not only the action class label but also a concrete temporal location when the activity is happening. This information allows us to explore and exploit patterns around that time frame. Commonly used datasets [10, 19, 37, 41] all display a human performing activities that involve interacting with objects. If we detect an interval of continuous human object interaction around a specific timestamp, we can assume that all frames in that interval belong to the same action as the timestamped frame. This approach creates HOI-influenced pseudo-groundtruth that enhances any other available real or pseudo-ground truth.

Many HOI detectors predict the action verb and spatial location of the interaction. There may be benefits to using the action verb information, but that may also require HOI training data more related to the specific action recognition dataset. To keep training requirements minimal, our current method does not use any action verb labels, and therefore does not require the HOI module to produce such labels.

In our implementation, we use the off-the-shelf pretrained HOI detector of Shan *et al.* [38]. Given an image, the model predicts hand sides and contact states either with the hands or surrounding objects. Hand side values are *left* or *right*, and hand state is represented as a 2D one-hot vector. There are five contact states: *none*, *self*, *other*, *portable*, and *non-portable*. The contact state is represented as a 5D one-hot vector. Alongside these categorical outputs, the model also produces bounding boxes around the hands and the interacting objects. In our method, we considered only those frames which had an interaction with a portable object. So, every frame with a detected contact state of portable is considered as a valid HOI frame, and the object bounding box b_t is stored. Here $t \in [1, T]$ and T is the length of the video.

4.2.1 HOI-Influenced Pseudo-Ground Truth

In the architecture diagram on Figure 3, the secondary label generator uses HOI information to generate pseudo-ground truth action labels. In this subsection we describe how the secondary label generator works.

The inputs are a video X, single-frame timestamp annotations $A_{TS} = [a_{t_1}, ..., a_{t_N}]$, and a sequence of frame-level HOI predictions ρ. The output is pseudo-ground truth κ. As shown in Figure 4, we start with a window of τ frames around a given timestamp frame t_i. We denote by b_{anchor} the mean center location of the detected object bounding boxes within that window of τ frames. Point b_{anchor} provides an approximate location of the human object interaction around timestamp t_i. Neighboring frames will be labeled with the same action if the location of the detected human object interaction in those frames stays close to b_{anchor}.

Frame-wise labels κ are initialized to ground-truth single-frame timestamp action labels a_{t_i} for a video. Then, for each anchor location t_i, the algorithm considers adjacent intervals forward and backward in time, with a hop of w frames at a time, to decide whether to propagate label a_{t_i} to each of those intervals. We denote by $b_{i,j}$ the mean location of the object bounding box in frames $x_i, x_{i+1}, \ldots, x_j$. We denote by $\delta_{i,j}$ the distance between locations b_{anchor} and $b_{i,j}$. Given this notation, for a hop index h starting from 0 which increments by 1, $h \in \mathbb{R}$ and spatial threshold σ in pixels, the forward expansion of timestamp action a_{t_i} proceeds as follows:

$$\kappa_{[t_i+hw, t_i+(h+1)w]} = a_{t_i}, \text{ if } \delta_{[t_i+hw, t_i+(h+1)w]} < \sigma \tag{1}$$

The forward search terminates if $\delta_{[t_i+hw, t_i+(h+1)w]}$ for a hop h is greater than σ, if no valid HOI frames have been detected in hop h, or if the time search range reaches the end of the video.

Similarly the backward expansion of timestamp action a_{t_i} is as follows:

$$\kappa_{[t_i-hw, t_i-(h+1)w]} = a_{t_i}, \text{ if } \delta_{[t_i-hw, t_i-(h+1)w]} < \sigma \tag{2}$$

Once the forward and backward expansion of action timestamp a_{t_i} terminate, the next timestamp $a_{t_{i+1}}$ is considered for forward and backward expansion following the same logic.

4.2.2 Fine-tuning Action Changes

In the architecture diagram on Figure 3, the primary label generator, given a video X and timestamp annotations

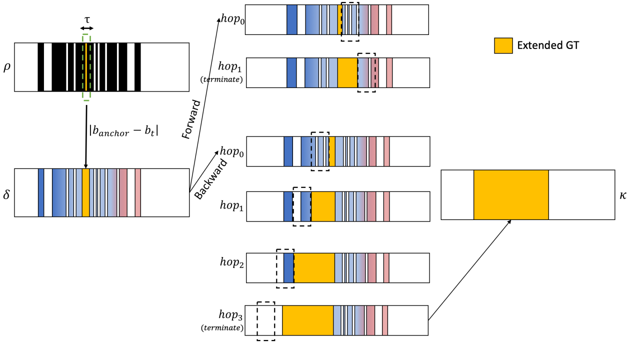

Figure 4. The proposed pseudo-ground truth generation method for a given action segment in a video. Timestamps are indicated in yellow. The black section in ρ indicates the frames where HOI was detected. After subtracting b_{anchor} from the bounding boxes of the neighbouring frames, the color spectrum in δ indicates magnitude difference from blue(low) to red(high) . hop_h indicates the progression of search window in forward and backward direction. Final pseudo ground-truth is indicated by the block κ.

$A_{TS} = [a_{t_1}, ..., a_{t_N}]$, generates frame-wise labels $\hat{A} = [\hat{a_1}, ..., \hat{a_T}]$ such that $\hat{a}_{t_i} = a_{t_i}$ for $i \in [1, N]$ where N is the number of segments. In this subsection we describe the operation of the primary label generator.

Our formulation for this module builds on the method of [29], which trains a TCN model M for action segmentation. That TCN model is referred to as "segmentation model" in Fig. 3. To generate frame-wise labels, the method of [29] estimates the time t_{b_i} of action change between two consecutive timestamps t_i and t_{i+1}, as follows:

$$t_{b_i} = \underset{\hat{t}}{\operatorname{argmin}} \sum_{t=t_i}^{\hat{t}} d(h_t, c_i) + \sum_{t=\hat{t}+1}^{t_{i+1}} d(h_t, c_{i+1}) \tag{3}$$

$s.t.$

$$c_i = \frac{1}{\hat{t} - t_i + 1} \sum_{t=t_i}^{\hat{t}} h_t, \tag{4}$$

$$c_{i+1} = \frac{1}{t_{i+1} - \hat{t}} \sum_{t=\hat{t}+1}^{t_{i+1}} h_t \tag{5}$$

In the above, $d(.,.)$ signifies the Euclidean distance and h_t is the output of the penultimate layer of the TCN at time t. Intuitively, the algorithm divides the the frames between timestamps t_i and t_{i+1} into two clusters by finding the location t_{b_i} such that the average distance between the frame outputs and cluster centers is minimized.

In [29], this approach is implemented using a forward-backward algorithm. In the forward direction, frames from the last computed boundary $t_{b_{i-1}}$ to the timestamp t_i are assigned action label a_{t_i}, and these frames are used in estimating the next action boundary $t_{b_i,FW}$. For the backward direction, boundary estimate $t_{b_{i+1}}$, is used to predict the previous boundary $t_{b_i,BW}$. The average of the 2 estimates is used to find the final estimate t_{b_i}. As initial conditions, $t_{b_0} = 1$ and $t_{b_N} = T$, where T is the number of frames.

$$t_{b_{i,FW}} = \underset{\hat{t}}{\text{argmin}} \sum_{t=t_{b_{i-1}}}^{\hat{t}} d(h_t, c_i) + \sum_{t=\hat{t}+1}^{t_{i+1}} d(h_t, c_{i+1}) \quad (6)$$

$$t_{b_{i,BW}} = \underset{\hat{t}}{\text{argmin}} \sum_{t=t_i}^{\hat{t}} d(h_t, c_i) + \sum_{t=\hat{t}+1}^{t_{b_{i+1}}} d(h_t, c_{i+1}) \quad (7)$$

$$p = \frac{t_{b_{i,FW}} + t_{b_{i,BW}}}{2} \quad (8)$$

In [29], the value of p from Eq. 8 is used as the estimate for t_{b_i}. This is where our method diverges, and uses human-object interaction information to improve upon this estimate. Our modification is formulated as follows:

$$t_{b_i} = \begin{cases} p, & \alpha_p = 0 \\ f(p, G), & \alpha_p = 1 \end{cases} \quad (9)$$

$$f(p, G) = \begin{cases} min(G), & G \neq \emptyset \\ p, & G = \emptyset \end{cases} \quad (10)$$

$$G = \{t | t \in [t_{b_{i,FW}}, t_{b_{i,BW}}], \alpha_t = 0\} \quad (11)$$

Here $\alpha_t \in \{0, 1\}$, for $t \in [0, T]$, indicates the interaction label obtained by binarizing the pseudo ground-truths κ at time t. Value 1 signifies interaction and 0 as no interaction. Figure 3 illustrates the binarized results α where the black segments indicate interaction and white segments indicate no interaction. Thus, we improve upon the architecture by adding a constraint that the detected boundary t_{b_i} is invalid if there is an ongoing human object interaction at that time. The boundary is re-adjusted to a temporal location where there is no interaction. During training, the final estimate t_{b_i} is estimated by the interaction label α_p. If interaction exists at time p then a subset of interaction values $\alpha_{[t_{b_{i,FW}}, t_{b_{i,BW}}]}$ is used to find a new action boundary. In the subset, the first time frame when there is no interaction is assigned as the new t_{b_i}. If there is interaction happening in all the frames in $\alpha_{[t_{b_{i,FW}}, t_{b_{i,BW}}]}$, then $t_{b_i} = p$.

4.2.3 Loss Functions

We use the already successful combination of classification loss and smoothing loss used in traditional action segmentation techniques [9, 16, 43] and the novel confidence loss [29].

Classification Loss. For classification loss, we employed a cross entropy loss that computes the loss between the prediction action probabilities and the generated target labels as well as the generated pseudo ground-truths using HOI. Here $\tilde{y}_{t,\hat{a}}$ is the predicted probability from the model for target action label $\hat{a}$ at time t.

$$\mathcal{L}_{cls} = \frac{1}{T} \sum_{t} -log(\tilde{y}_{t,\hat{a}}), \quad (12)$$

Smoothing Loss. To penalize for local inconsistencies in the the predicted action class probabilities we adopted the truncated mean square error as a smoothing loss [9]. This loss encourages the network to avoid over-segmentation errors.

$$\mathcal{L}_{T-MSE} = \frac{1}{TC} \sum_{t,a} \tilde{\Delta}^2_{t,a}, \quad (13)$$

$$\tilde{\Delta}_{t,a} = \begin{cases} \Delta_{t,a}, & \Delta_{t,a} \leq \tau \\ \tau, & otherwise \end{cases} \quad (14)$$

$$\Delta_{t,a} = |log\tilde{y}_{t,a} - log\tilde{y}_{t-1,a}| \quad (15)$$

Where C is the number of action classes, $\tilde{y}_{t,a}$ is the action a probability at time t.

Confidence Loss. The confidence loss [29] enforces monotonicity on the model confidence as defined below:

$$\mathcal{L}_{conf} = \frac{1}{T'} \sum_{a_{t_i} \in A_{TS}} (\sum_{t=t_{i-1}}^{t_{i+1}} \delta_{a_{t_i},t}), \quad (16)$$

$$\delta_{a_{t_i},t} = \begin{cases} max(0, log\tilde{y}_{t,a_{t_i}} - log\tilde{y}_{t-1,a_{t_i}}), \; if \; t \geq t_i \\ max(0, log\tilde{y}_{t-1,a_{t_i}} - log\tilde{y}_{t,a_{t_i}}), \; if \; t < t_i \end{cases} \quad (17)$$

Using this loss, the low confident regions which are surrounded by higher probability regions are encouraged to produce higher probabilities. This loss also penalizes outlier frames carrying high probabilities that are far from the annotated timestamp and that are not surrounded by high confidence regions.

The final loss of the action segmentation model is:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \alpha \mathcal{L}_{T-MSE} + \beta \mathcal{L}_{conf} \quad (18)$$

Here α and β are the hyper-parameters that guide the contribution of each loss.

5. Experiments

In this section, we compare our method with the first systems for action segmentation using timestamp supervision. We also show the contribution of each component quantitavely and qualitatively. We also benchmarked a fully supervised [27] and weakly supervised [40] method that can act as a baseline for further research on (3+1)ReC. Results on the (3+1)ReC dataset are based on 6 fold-cross validation, where each fold includes five independent subjects.

Datasets. In our experiments, in addition to (3+1)ReC, we have used three public datasets commonly used for evaluating action segmentation methods:

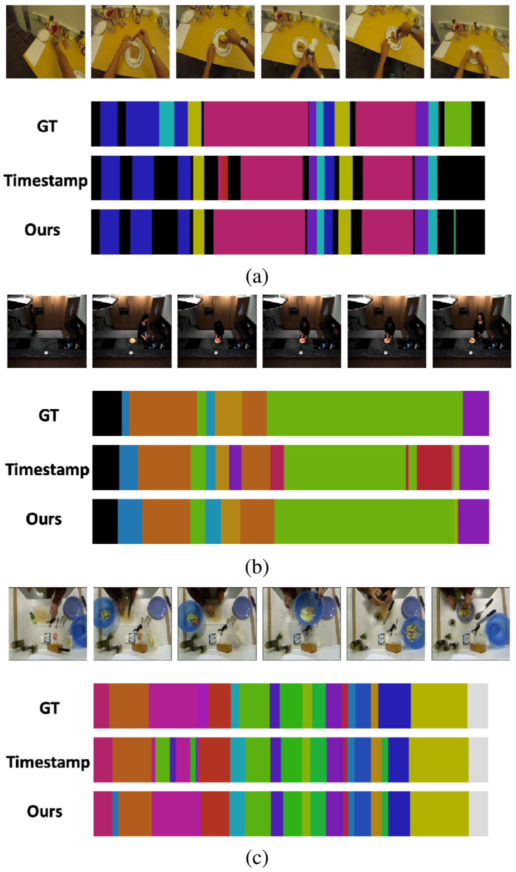

Figure 5. Qualitative results on (a) 50Salads, (b) MPII Cooking2 and (c) GTEA datasets. The baseline method suffers from over-segmentation, while our approach alleviates this issue by utilizing the continuity in human object interaction.

1) The *50Salads* dataset [41] contains 50 videos and 17 fine-grained action classes. Each video on average contains 20 fine-grained action instances and is 6.4 minutes long. The videos display human subjects preparing different types of salads. There are 25 video-level class labels (different salads) overall, and every actor prepares two different salads. 2) The *GTEA* dataset [10] contains 28 egocentric videos and 11 fine-grained action classes. There are 7 different video-level classes such as “preparing tea” and “hot dog”, performed by 4 subjects. Each video contains 20 fine-grained action instances on average. 3) The *MPII Cooking 2* [37] contains 243 high quality videos, ranging in length from 1 minute to 40 minutes, and 67 fine-grained action classes. It includes 29 subjects who prepare 58 different dishes (video-level class labels) like “making pizza”.

Evaluation Metrics. We use evaluation metrics commonly used in action segmentation tasks [9, 16, 43]: frame-wise accuracy (Acc), segmental edit distance (Edit) and segmental F1 score at overlapping thresholds of 10%, 25% and 50%, denoted as F1@{10,25,50}. While frame-wise accuracy is the most commonly used metric in action segmentation research, it naturally places more importance on long-duration actions over shorter actions, and it lacks an explicit penalty for over-segmentation errors. Segmental edit score and F1 score penalize the over-segmentation errors and treat shorter and longer duration actions as equally important.

Implementation Details. For the action segmentation module of Fig. 3 we use the multi-stage temporal convolution network of Li *et al.* [29]. For HOI detection, if there are multiple objects detected, where the human is interacting with an object in each hand, the bounding boxes are merged to a bigger bounding box. We trained for 70 epochs using Adam optimizer. The learning rate is 0.0005 and the batch size is 8. For the loss function, we used $\tau = 4$, $\alpha = 0.15$ and $\beta = 0.075$. We used the same I3D [3] features as in [9]. We trained all models using the same timestamp annotations as Li *et al.* [29], for fair comparison with other methods. Further implementation details, and all parameters can be obtained in the supplementary material.

5.1. Results

5.1.1 Comparison with other segmentation baselines

In Table 2, we compare our method with timestamp supervised baselines [29] and [31] for action segmentation. Compared to [29], our approach consistently attains higher accuracy in four datasets for all metrics. For GTEA, the F1 score at 50% overlapping threshold improves by 10.9%. The frame-wise accuracy improves by 5.3% when compared to [29] and is now 92.5% of the fully supervised approach. For 50Salads, the F1 score at 50% overlapping threshold improves by 5.9% and the frame-wise accuracy improves by 0.4% when compared to [29] and is now 97.45% of the fully supervised approach. For MPII cooking2, the F1 score at 25% overlapping threshold improves by 4.5% and the frame-wise accuracy is improved by 2.3% when compared to [29] and is now 95.1% of the fully supervised approach. We include more results of fully supervised methods on GTEA and 50Salad datasets in the supp. material.

Table 2 also shows action segmentation results, separately, on third-person (3ReC) and egocentric (1ReC) videos of our proposed dataset. It can be seen for Egocentric views, the performance of the baseline timestamp segmentation is improved significantly in all metrics when we utilized HOI in our method. However, the inferior performance of our approach in the third person setting (3ReC) shows our method's limitation in tracking the HOI in more complicated scenarios of the 3ReC dataset. We associate this with the nuances introduced in our third person dataset, *e.g.*, objects are occasionally occluded or interaction with objects is not always tactile when an item is in the microwave/toaster. Furthermore, in Table 2, we benchmark results of sample SOTA methods under full [27] and weak [40] supervision using their public source codes. Results in-

	F1 @ {10, 25, 50}			Edit	Acc
50Salads					
Seg model + plateau [31]	71.2	68.2	56.1	62.6	73.9
Timestamp [29]	73.9	70.9	60.1	66.8	75.6
Ours	**77.3**	**75.2**	**63.6**	**69.8**	**75.8**
Full Supervision*	70.8	67.7	58.6	63.8	77.8
GTEA					
Seg model + plateau [31]	74.8	68.0	43.6	72.3	52.9
Timestamp [29]	78.9	73.0	55.4	72.3	66.4
Ours	**82.1**	**78.7**	**63.0**	**74.8**	**70.4**
Full Supervision*	85.1	82.7	69.6	79.6	76.1
MPII Cooking2					
Timestamp [29]	42.7	38.7	28.7	41.1	50.1
Ours	**44.9**	**40.6**	**28.8**	**43.5**	**51.3**
Full Supervision*	45.5	42.1	32.5	43.2	54.0
1ReC (Egocentric (3+1)ReC)					
Timestamp [29]	34.1	24.9	10.2	36.6	27.5
Ours	**37.5**	**28.2**	**12.3**	**39.2**	**28.9**
Weak Supervision [40]	37.7	31.2	19.5	43.7	22.3
Full Supervision [27]	46.0	42.4	34.9	44.8	46.9
3ReC (Third Person (3+1)ReC)					
Timestamp [29]	**46.8**	**42.2**	**32.5**	**46.1**	**38.8**
Ours	38.9	29.1	12.9	41.7	25.9
Weak Supervision [40]	38.7	32.6	21.3	43.5	21.5
Full Supervision [27]	42.8	39.7	32.8	43.9	46.3

Table 2. Comparison between our method and other action segmentation baselines under different supervision levels. * indicates training [29] with real ground-truth as opposed to pseudo labels.

	F1 @ {10, 25, 50}			Edit	Acc
50Salads					
loss [29]	73.9	70.9	60.1	66.8	75.6
loss+pg	76.5	74.4	62.6	69.3	75.7
loss+pg+ft	**77.3**	**75.2**	**63.6**	**69.8**	**75.8**
GTEA					
loss [29]	78.9	73.0	55.4	72.3	66.4
loss+pg	79.9	75.5	58.1	74.2	68.2
loss+pg+ft	**82.1**	**78.7**	**63.0**	**74.8**	**70.4**
MPII Cooking2					
loss [29]	42.7	38.7	28.7	41.1	50.1
loss+pg	44.4	40.0	28.3	42.1	50.5
loss+pg+ft	**44.9**	**40.6**	**28.8**	**43.5**	**51.3**

Table 3. Contribution of the original loss of [29], new pg generation and fine-tuning(ft) of the action change using HOI.

	F1@{10,25,50}			Edit	Acc
GTEA					
loss [29]	78.9	73.0	55.4	72.3	66.4
loss+ft	78.6	74.5	57.6	72.0	67.9
Improvement	-0.3	1.5	2.2	-0.3	1.5
loss+pg	79.9	75.5	58.1	74.2	68.2
loss+pg+ft	82.1	78.7	63.0	74.8	70.4
Improvement	2.3	3.1	4.9	0.6	2.2

Table 4. Improvement in performance for GTEA using labels generated by adding constraint of HOI to detect action change.

dicate the challenging nature of action segmentation on our dataset and the potential to study further improvements.

5.1.2 Impact of loss with HOI

Table 3 shows the benefits of using HOI information. We show results using the original loss function of [29], and results obtained by incorporating two changes proposed in this paper: "pg" denotes the pseudo-ground truth generated using HOI, as described in Sec. 4.2.1. By "ft" we denote detecting action boundaries using the proposed "fine-tuning" equations 9-11 of Sec. 4.2.2, whereas versions not marked with "ft" detect action boundaries as described in [29].

For 50Salads, the F1 score @50% overlap increased by 2.5% when compared to [29] when adding the "pg" component, and increased further by 0.6% when using the "ft" approach. The qualitative results showcase how our approach corrected some of the over-segmentation errors in [29]. Similar improvements were seen in GTEA, where the F1 score @50% increased by 2.7% by using just pseudo-ground truth and by 4.9% with fine-tuning action changes using HOI. Similar gains were seen in MPII Cooking 2.

5.1.3 Impact of fine-tuning.

Table 4 illustrates the benefits of re-adjusting the action change boundaries using HOI information. The terms "loss", "pg" and "ft" have the same meanings that we defined in discussing Table 3. Table 4 shows that, for the GTEA dataset, our proposed improvements lead to higher accuracy in almost all metrics. There are only two entries in that table (out of a total of 10) where the proposed components do not improve accuracy, but in both those cases the drop is marginal (0.3%). In the other eight entries, our components lead to improvements ranging from 0.6% to 4.9%.

6. Conclusion

We showed in this paper, that information from human-object interaction can be used to improve action segmentation accuracy under timestamp supervision. Our model extends the single frame timestamp annotations using the frame level predictions of a human-object interaction detector. We improve the segmentation results by adding a constraint that an action boundary cannot exist around frames where the human is continuously interacting with the object. We also proposed the (3+1)ReC dataset as a diverse and high resolution instructional video dataset with synchronized third and first person views. Results on public datasets show that the key idea of using HOI information can indeed improve action segmentation accuracy in most cases and close the gap with fully-supervised models.

References

[1] Nadine Behrmann, S Alireza Golestaneh, Zico Kolter, Jürgen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 52–68. Springer, 2022. 2

[2] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021. 3

[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. 7

[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/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3555, 2019. 1

[5] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *2018 ieee winter conference on applications of computer vision (wacv)*, pages 381–389. IEEE, 2018. 2

[6] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. Hico: A benchmark for recognizing human-object interactions in images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1017–1025, 2015. 2

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. 3

[8] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6508–6516, 2018. 1

[9] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019. 1, 6, 7

[10] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011. 2, 3, 4, 7

[11] Mohsen Fayyaz and Jurgen Gall. Sct: Set constrained temporal transformer for set supervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–510, 2020. 1

[12] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8359–8367, 2018. 2

[13] Abhinav Gupta, Aniruddha Kembhavi, and Larry S Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(10):1775–1789, 2009. 2

[14] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015. 2

[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. 1

[16] Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. Alleviating over-segmentation errors by detecting action boundaries. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2322–2331, 2021. 6, 7

[17] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2015. 2

[18] 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. 3

[19] 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. 4

[20] 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. 1

[21] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 1

[22] Hilde Kuehne, Alexander Richard, and Juergen Gall. A hybrid rnn-hmm approach for weakly supervised temporal action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):765–779, 2018. 1

[23] 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. 1

[24] Pilhyeon Lee, Jinglu Wang, Yan Lu, and Hyeran Byun. Weakly-supervised temporal action localization by uncertainty modeling. *arXiv preprint arXiv:2006.07006*, 2020. 2

[25] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6243–6251, 2019. 1

[26] 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. 1

[27] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 6, 7, 8

[28] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635, 2018. 3

[29] Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8365–8374, 2021. 2, 3, 5, 6, 7, 8

[30] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. In *European conference on computer vision*, pages 420–437. Springer, 2020. 4

[31] Davide Moltisanti, Sanja Fidler, and Dima Damen. Action recognition from single timestamp supervision in untrimmed videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9915–9924, 2019. 1, 2, 7, 8

[32] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8688–8697, 2019. 2

[33] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172, 2020. 2

[34] Yicheng Qian, Weixin Luo, Dongze Lian, Xu Tang, Peilin Zhao, and Shenghua Gao. Svip: Sequence verification for procedures in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19890–19902, 2022. 3

[35] 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. 1

[36] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5987–5996, 2018. 1

[37] 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. 2, 3, 4, 7

[38] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9869–9878, 2020. 2, 3, 4

[39] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7396–7404, 2018. 3

[40] 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. 1, 6, 7, 8

[41] Sebastian Stein and Stephen 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, 2013. 2, 3, 4, 7

[42] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10410–10419, 2021. 2

[43] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. Boundary-aware cascade networks for temporal action segmentation. In *European Conference on Computer Vision*, pages 34–51. Springer, 2020. 6, 7

[44] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 1