

Learning to Localize and Align Fine-Grained Actions to Sparse Instructions

Anonymous ECCV submission

Paper ID 135

Abstract. Automatically generating textual video descriptions that are time-aligned with the video content is a long-standing goal in computer vision. The task is challenging due to the difficulty of bridging the semantic gap between the visual and natural language domains. This paper addresses the task of automatically generating an alignment between a set of instructions and a first person video demonstrating an activity. The sparse descriptions and ambiguity of written instructions create significant alignment challenges. The key to our approach is the use of egocentric cues to generate a concise set of action proposals, which are then matched to recipe steps using object detections and computational linguistic techniques. We obtain promising results on both the Extended GTEA Gaze+ dataset and the Bristol Egocentric Object Interactions Dataset.

Keywords: Action Proposal, Text to Video Alignment, Instructional Videos, Reference Resolution

1 Introduction

Written instructions, contained in recipe books, instruction manuals, how-to books, and so forth, along with visual recordings such as instructional videos, provide valuable sources of information that allow people to learn how to perform new tasks (e.g. cook a new dish) and gain new skills. In addition to their value to people, there is the potential for these online sources of knowledge about actions and activities to support new approaches to robot learning [1, 2]. Video and text are complementary sources of information. Text descriptions can provide semantic information that is missing in a video record. Moreover, text descriptions are compact and easily searchable. However, the translation from written description to motor program (in humans and machines) is challenging. Recipes and instructions are written for humans and therefore are naturally sparse and require inference of missing steps and details through NLP and task knowledge. For example, performing the recipe step “beat the egg whites until they are stiff” requires the ability to fill-in many missing fine-grained details. In contrast, videos contain a wealth of fine-grained detail, but suffer from an inherent linearity that makes it difficult to extract the “big picture” view of an activity or quickly find a specific step in a procedure. In particular, the lack of effective methods for action segmentation and localization in video make it

difficult to connect specific instructional steps to the appropriate segments of a continuous video.

Instructional videos can bridge the gap between video and text through an audio narration, which when transcribed brings a sequence of words into alignment with the video frames. While instructional videos are created for human consumption, several recent works have resulted in methods for automatically utilizing video narrations for activity and action analysis [3, 4]. But the scalability of the approach is limited by the relative scarcity of instructional videos, in comparison to non-narrated videos and text corpora. First Person (FP) videos are particularly attractive for activity understanding, as a headworn camera is ideally positioned to capture the actions of the camera wearer, and FP videos are becoming increasingly common in social media [5]. Recent work has demonstrated the utility of the first person approach in activity recognition [6, 7].

In this paper, we propose a novel task of egocentric *sparse alignment*, in which a *non-narrated* first person video recording of an activity is *aligned* with the steps in the recipe that describes it. The two key technical challenges of sparse alignment are 1) the extremely low ratio of text to video; and 2) the fact that no explicit correspondence is given at training or testing time between the recipe text and the video frames, making the alignment module unsupervised. In contrast, most previous works on video annotation or instructional video analysis either assume that frame-level text annotations are available [8–10] or infer them from dense text descriptions obtained from transcribing video narrations [3, 11]. The quantity of available text annotations is a key factor underlying the difficulty of the alignment problem. In order to quantify the *density* of text annotations, we measure the average number of words available per minute of video for different datasets. We evaluate our system over two egocentric datasets: Extended GTEA Gaze+ corpus (EGTEA Gaze+) [7] and the Bristol Egocentric Object Interactions Dataset [12]. We directly compare these datasets against other instructional video datasets [3, 4] and video captioning datasets [9, 10] in the graph in Figure 3. We can see clearly that the instructional video dataset from [3] has a far greater density than any of the other datasets. Because recipes tend to be written in short concise sentences, EGTEA has the lowest word density at 6.18 words/minute and is followed by the other recipe dataset YouCook2 with a word density of 11 words/minute. BEOID has a higher word density than the other recipe datasets at 32 words/minute. We hypothesize that word density is a key factor in how challenging a dataset is to align, and test this hypothesis in our experiments. Another important property of our approach is that it is designed to work well on continuous, unsegmented, long videos. As we can see in the graph in Figure 3, video clips in EGTEA are twice as long as the clips in other datasets. To address this challenge, we adopt a two-stage approach in which we first break videos into segments with a high probability of containing an action and then align the resulting action proposals with the steps in the recipe. Both stages are challenging problems in and of themselves, and our experimental results demonstrate that we achieve state-of-the-art performance in each. To overcome the sparsity of the text in the datasets that we employ,

we use the linguistic semantics of the observed objects in the video to match the video segments with the correct recipe steps. Word embeddings allow us to draw multiple concepts and relations from a single word. Therefore from a single action object pair we can match a recipe step to various segments containing relevant objects.

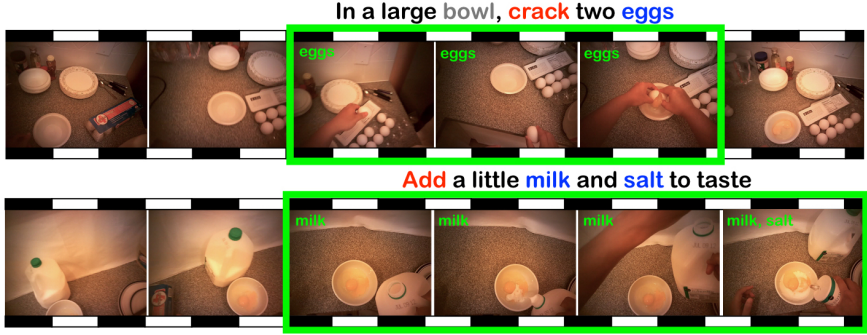


Fig. 1. The final results of the system on first person cooking videos from the EGTEA dataset. Video frames that are surrounded by a green box are part of a predicted action segment. The green words in the corner of the frames in the action segments are the recognized objects in the frame. The sentence above the action segment is the recipe step that the system automatically aligned the segment to. Each recipe step is colored according to how it was parsed by our alignment module. Blue words are primary objects, gray words are secondary objects and red words are actions.

This paper demonstrates that the challenge of action segmentation in FP video can be addressed by exploiting *egocentric cues*, such as hand and head movement along with eye gaze. Prior work [7] has demonstrated that these cues capture a powerful signal to recognize actions involving object manipulation. We show that using egocentric features within a LSTM-based method leads to significant improvements in the quality of action proposal generation in FP video. Proposals identify video segments where objects are manipulated by some action. In order to generate recipe alignments, we automatically detect these manipulated objects during the action segments. These detected objects provide a link between a video clip and the steps in the recipe. The sparseness of the recipe’s text descriptions leads to significant ambiguity in segment to step matching, which we address through the use of self-supervised linguistic reference resolution. We test the resulting system for aligning fine-grained actions to recipe steps by evaluating it on an multiple FP video datasets. We obtain promising results, outperforming both recent action proposal methods and prior video-text alignment pipelines. Our approach has the potential to unlock the vast repository of textual descriptions of actions by aligning them with the growing corpus of FP videos, enabling new approaches for human instruction and robot learning. Our system can also be used for video retrieval making instructional videos more accessible.

This paper makes three contributions: First, we present a system for automatically generating action proposals in FP videos which exploits egocentric cues for localizing the onset and offset of actions. Second, we create a novel method for aligning action proposals to steps in a recipe via action and object-derived features. Third, we annotate the popular EGTEA Gaze+ dataset and the Bristol Egocentric Object Interaction Dataset for the text-video alignment task and we show experimental results over them.

2 Related Works

Video-to-Text: The key difference between past work in video-to-text alignment [3, 8, 11, 13]. and our system is that these works use significantly denser text that is curated for each individual video, whereas our system is able to align multiple videos to the single recipe that describes the activity. One of the recent methods [3] identifies the key steps in an instructional video by clustering dense, transcribed narrations, and then aligns each step to a video in a weakly-supervised end-to-end method. [3] proves that narrations are an informative signal, however written narrations are not always available and spoken narrations are not always accurately transcribed by automatic speech recognition systems. [14] aims to resolve the noise in narrations using a visual-linguistic model but doesn't solve the alignment task. In fact the system in [14] relies on the narrations already being aligned to the frames. [11] does weakly-supervised alignment of videos to text using the TACoS [10] dataset, which has captions for each video. In contrast, in our dataset there is only a single recipe for all videos of that class rather than individual recipes for each video. This means that some videos may skip recipe steps or add activities that don't correspond to any recipe step. The problem of aligning videos to recipes is first addressed in [13], however they use written narrations in order to do the alignment. Using purely linguistic model [13] aligns the narrations to the recipe steps and only uses visual cues to refine the alignment. In contrast, we match videos to recipes without relying on additional narrations. For reference on the difference in word density as well as other statistics of the datasets used in previous works and our work look at Figure 3. The only other dataset with high sparsity is the recently released YouCook2 dataset [4]. It has not yet been used for text to video alignment. The work that introduces the dataset focuses on extracting action segments from the videos and does not use the recipe text at all [4].

Action Detection and Proposal Generation: One of the main contributions of this paper is the action segmentation module. Action proposal is a well known task in the computer vision field and is often used as part of action detection methods [15–17]. Our novel method for proposal generation in FP videos is most closely related to Deep Action Proposals (DAPs) [18]. We share with DAPs the need to generate proposals from extended continuous videos, and we also use features from 3D convolutional networks [19]. DAPs works best on long videos and isn't as effective at generating proposals for short videos, see §4.3 and Table 2. DAPs produces action proposals by sliding a window of a given

step size over video, inputting C3D features of the frames in the window into a unidirectional LSTM. Instead we treat the task as frame level classification and use a bidirectional LSTM to propagate information about adjacent frames. Additionally, DAPs outputs multiple overlapping candidate windows whereas our method does not allow overlapping action segments because our alignment system can only align each video frame to at most a single recipe step. To mediate this difference during our experimental evaluation we use non maximum suppression over the DAPs candidates to remove overlapping proposals. We differ from all previous action proposal systems by incorporating egocentric cues into our network. Also, no other action proposal systems aim to do video-to-text alignment or test on FP videos. The works [20, 21] do weakly supervised action labeling, which takes a temporally ordered set of action labels that occur in a video and align the video’s frames to the actions. Our task is different since we take in written recipes which have to be parsed and which do not explicitly state all actions that occur in a video. Also the recipes are not treated as a strictly ordered set since recipe steps can be done out of order.

First Person Vision: Our system is created for first person (FP) videos which have become more prevalent in the computer vision community in recent years [6, 7, 22–24]. We utilize the egocentric cues proposed by [7] in our method for action proposal generation. [7] focuses on action recognition over predefined ground truth action segments, in contrast, we automatically generate localized segments containing actions and then match them to recipe steps. Our use of gaze-related features connects to prior work on gaze prediction in FP video [25, 26].

3 Approach

Given a recipe in natural language text and a corresponding egocentric video clip containing the entire recipe, the goal of our system is to *align* the recipe and video by matching each recipe step to all of the video frames that correspond to its execution, as shown in Figures 1 and 4. Our approach has three main parts: 1) Identifying video segments that contain actions and are therefore candidates for matching; 2) Parsing the recipe text into a representation of objects and actions which supports matching; and 3) Establishing the correspondence between identified segments and recipe steps via the objects that are being used and manipulated. A diagram of the complete pipeline is shown in the Figure 2.

The main challenge of the alignment task is that recipe descriptions only provide sparse high-level descriptions of complex activities, e.g. “beat two eggs into a bowl”. This type of abbreviated description can map onto a large number of potential sequences of atomic actions. Instead of making a detector for each possible action, we explore the hypothesis that we can identify and align relevant video segments to their recipe steps with only *action proposals* and task-relevant object recognition. Our approach leverages the discriminative power of egocentric cues [7], such as hand configurations and head motion. These cues have been shown to be highly discriminative for action recognition and we show for the first time their usefulness in temporally localizing actions. After generating action

proposals for a video, we then automatically identify the main objects that are being manipulated during the actions. We use these detected objects as concrete reference points with which to establish a correspondence to specific recipe steps. The output of our pipeline is a set of video segments for each step in the recipe, where the segments delineate the actions described in that particular recipe step.

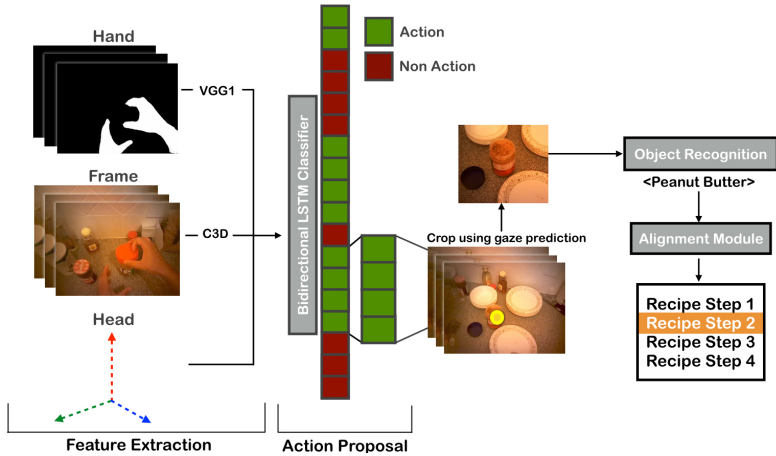


Fig. 2. Video to Recipe Alignment Pipeline: features extracted from head motion, hand masks and frames, are concatenated and fed into a bidirectional LSTM, which predicts whether a frame contains an action or not. Frames are then densely extracted from each action segment and fed one by one into the object recognition network, which builds a object histogram and passes the top k objects to the recipe alignment module. The recipe alignment module parses the recipe and extracts action-object pairs. It then uses the detected objects to align each action segment to a recipe step.

3.1 Generation of Action Proposals

The first component of our pipeline is a novel method for generating action proposals using egocentric cues. The goal at this stage is to identify which segments of an input video contain actions and which ones do not. We pose this as a supervised frame level classification problem, because the end goal of our system is to align each frame to a single recipe step. We employ a bidirectional LSTM recurrent neural network to perform the classification. We chose this approach because LSTMs store contextual information in their memory cells, allowing the system to store the temporal context of an action, and because bi-directionality propagates information about preceding and succeeding frames. This allows us to exploit the fact that the likelihood that a frame depicts an action is affected by whether the surrounding frames depict actions. Our method localizes segments which contain fine-grained actions, such as “take”, “put”, “close”, “open” and so on. The primary novelty in our approach is our success in harnessing egocentric cues in order to segment fine-grained actions which are hard to localize because

their short duration. The egocentric features can capture the subtle movement patterns of the hands and head which can indicate the start or end of an action, and which are quite challenging for other methods to capture using more general image feature representations. Our experiments provide evidence for this claim.

The LSTM network is fed both egocentric and spatio-temporal context features. We automatically extract the egocentric features of *head motion* and *hand pose* from the videos in the EGTEA and BEOID datasets, which were captured with head-mounted cameras. Hands are highly discriminative for actions, especially in activities of daily living which require substantial object manipulation. Hands help distinguish action segments, since a subjects hands are likely to be both visible and in motion when an action is being performed. From each video frame, we segment out the hand from the rest of the image using a deep semantic segmentation method [27]. Each segmented hand region is fed into a pre-trained VGG16 network and the last fully connected layer is extracted as the hand feature for the frame. Head motion in egocentric videos corresponds to camera motion. In contrast to the hands, head motion is usually at its peak right before and right after an action. To model camera motion we follow a similar procedure to [7]. We first match sparse interest points generated by ORB [28] between adjoining frames. All interest points that lie on the hand mask are deleted. Then we use RANSAC [29] to fit the points to a 9 dimensional homography. We set the homography to zeros for frames with too low of a quality to extract meaningful interest points. The spatio-temporal features we use are the C3D features [19] which have been shown to capture both appearance and temporal information. We test the combinations of features in a comprehensive ablation study in §4.1 and find that the concatenation of the hands, head, and C3D features yield the best performance.

3.2 Task-Specific Object Recognition

Our system uses the manipulated objects detected in a video segment as the main discriminatory signal for aligning the segment to a recipe step, as described in §3.3. We found that objects are more discriminative than actions, because the fine-grained action labels that make up our datasets provide a weaker signal. This is due to the fact that some of the most frequently occurring actions are “take”, “put”, and “close”. These actions can be applied to most objects and can be found in many recipe steps, making them too ambiguous to support accurate alignment.

Supervised methods for object classification [30–33] and detection [34–38] have obtained good performance on standard image datasets. However, training a state-of-the-art object detector would require annotated bounding boxes that are unavailable in EGTEA and BEIOD. Instead, we use gaze information obtained from a wearable eye tracker, which is available in both EGTEA and BEOID, coupled with a CNN model for object classification, to identify the objects being manipulated. For each video in the training set, we take the ground truth action segments and extract one in every five frames. These frames are taken as positive examples of the main object in the action segment and are added to the training

images. We have a total of 86 object classes for the EGTEA dataset and 22 object classes for BEOID. Both datasets include annotations of the gaze point of the user at every frame. In our experiments, we found that incorporating egocentric information at test time improved prediction accuracy for most objects. Subjects are almost always looking near the object that they are manipulating, therefore, we crop images around the gaze point of the user at test time in order to better disregard background objects. For FP videos that do not contain gaze points, we could potentially use techniques such as [25] to predict the gaze points. After extracting and cropping images for each action proposal segment, we feed them through our object classification network as depicted in Figure 2 to obtain a histogram of object predictions for every action segment. The three top elements of each histogram are passed to the alignment module. This approach gives us robust predictions of the objects that are being manipulated in the action segments, and allows us to filter spurious mistakes and false detections.

3.3 Recipe Parsing and Alignment

The final stage of our pipeline completes the alignment of each proposed video segment to a recipe step, based on the correspondence between the automatically detected video objects and the text description. Our alignment module is *not trained*, and does not require ground truth manual annotations. As a result, it is scalable and can be applied to diverse datasets. The alignment problem is *challenging* because of the mismatch between the objects that are present in a video and the manner in which objects are referred to in the concise natural language descriptions that comprise a recipe. We address this challenge using tools from natural language processing to identify and extract the key verbs and nouns from the recipe, and to resolve ambiguities and establish references between steps in a recipe.

First, each recipe step is analyzed with the Stanford Dependency Parser [39], which identifies the dependency relations between the words in a sentence. The key type of dependency relation that exists between an action and the object of the action, is called a direct object relation. For example, in the sentence “pour the milk, sugar and coffee into the mug” the only direct object is from “pour” to “milk.” From this example, we can see that the direct object relations obtained from the dependency tree will not encompass all of the action-object pairs in the sentence. Therefore, we perform additional processing on the results of the dependency tree. For lists of objects, we extract all of the nouns in the list by identifying the conjunction relations with the direct object noun. In the example sentence, this extracts the nouns “sugar” and “coffee.” In the example, “milk, sugar and coffee” are the primary manipulated objects and “mug” is a secondary manipulated object. Secondary objects are noun phrases which have a nominal noun modifier dependency relation with the main action. In the example, the nominal noun modifier relation is between “pour”, “in” and “mug”. Another challenge arises when pronouns such as “it” or “them” are used to reference items mentioned in previous steps. When this occurs, we use co-reference resolution to identify the noun that the pronoun is referring to. If an object in a recipe step

x is resolved to a previous recipe step y , then we create the constraint that the module must align action segments to step y before aligning action segments to step x , since the step y is required to complete step x . Our parsing algorithm effectively extracts action-object pairs from the recipe and resolves references, as illustrated in Figure 4.

The next step is to calculate an *alignment similarity score* between each frame of an action segment and each recipe step, which will be used to produce the final alignment. The score is a weighted average of three distances: the word vector similarity between the primary and secondary manipulated objects and the detected objects, the word vector similarity between the recipe action and the detected objects, and the temporal similarity of the video segment to the recipe step. To obtain word similarity we use the Word2Vec [40] embeddings and calculate the euclidean distance between vectors. Temporal similarity is measured by computing the difference between the ratio of time of the proposed segment to the length of the entire video and the ratio of the number of the recipe step to the total number of recipe steps. This imposes a loose temporal constraint on the alignment. This constraint makes the assumption that while people may do things a few steps out of order, it will be unlikely, for example that the first step preformed will be the last step in the recipe. As a final step, the scores for a segment are weighted using a normal distribution so that frames at the center of the segment contribute more. Note that we discard segments whose similarity score is below a certain threshold, which helps to prune incorrect or irrelevant proposals, since the videos can contain additional actions that do not correspond any recipe steps. We then produce the final alignment by combining the frames in each segment to identify the recipe step with the highest score, and assign the resulting step to all frames in the segment.

The novelty of our alignment module lies in the combination of parsing methods to extract primary and secondary manipulated objects as well as the utilization of word embeddings to infer correlations that are not explicitly stated between the video segments and the recipe steps. For instance the recipe step “chop the carrots” implies the actions of getting a cutting board and a knife but does not state them. Through our alignment module’s specific use of word vector similarities, we are able to align action segments such as retrieving the cutting board with the correct recipe step. By drawing on the linguistic semantic bank, we can infer what activities are being performed simply by the objects being manipulated and the order they are manipulated in.

4 Experiments

Datasets. We evaluate our system over two egocentric datasets: the Extended GTEA Gaze (EGTEA) dataset [7] dataset which is composed of first person cooking videos and the Bristol Egocentric Object Interaction Dataset (BEOID) [12] which contains first person videos of scripted daily activities. EGTEA contains 86 unique cooking videos from 32 subjects with an average length of 15 min. There are 7 different recipes, from making a pasta salad to cooking a bacon and

eggs breakfast. We performed additional annotations of EGTEA to support our evaluation by adding the number of the appropriate recipe step to each ground truth action segment. Actions that do not correspond to any recipe step were labelled as such. EGTEA is challenging because the videos are long and complex and the word to text density is quite low (see Figure 3.) While all EGTEA videos were recorded in the same indoor kitchen setting, the manipulated objects differ greatly between recipes, and most recipes require cooking multiple things. BEOID contains 58 unique daily activity videos from 8 participants with an average of one minute. There are 6 different indoor tasks, each performed in a different location, from operating gym equipment to making a cup of coffee. Using the descriptions given to participants, we created an instruction set (recipe) for each task and labeled the ground truth action segments appropriately. The supplementary material contains additional details. Note that all performance numbers are obtained using five-fold cross-validation.

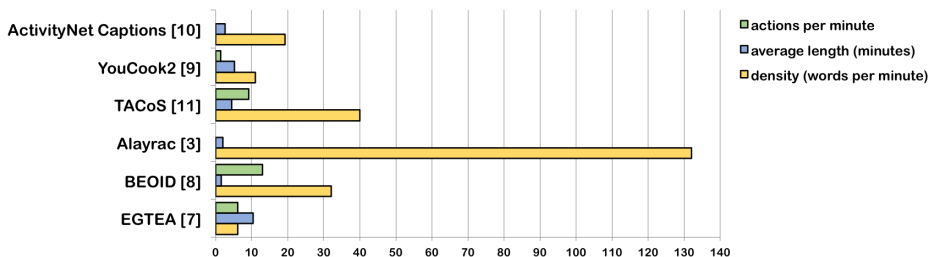


Fig. 3. Statistics of relevant instructional video and video captioning datasets.

4.1 Action Proposal Evaluation

Our action proposal system does frame level classification to temporally localize actions in videos. In our final system, egocentric features and C3D [19] features are fed into a bidirectional LSTM which determines whether or not an action is happening in the frame. Consecutive action frames are classified as a single action segment and then action segments are then smoothed. We experimented with multiple types features including traditionally features such as improved Dense Trajectories and generic features such as pre-trained VGG network extracted frame features. In our experimentation with the features, we found that the traditional features from improved Dense Trajectories (iDT) [41] did not work well for our task. We hypothesize this occurred due to over-fitting caused by the large size of the fisher vector encoding of iDTs. Similarly we believe that the C3D + hands + frame feature has a dip in performance because the feature vector is larger and wasn't able to optimize as well given the same amount of training data. We performed an extensive ablation study to examine the effectiveness of

each of the features which can be seen in Table 1. In the table, it is clear that the addition of the egocentric features significantly increases the IOU score of the proposals.

Our final model is a single layer bidirectional LSTM trained for 10 epochs at a learning rate of $1e-4$. Our input vector to the network is a 8201 dimensional and we use 300 hidden units. Both EGTEA and BEOID have frame level annotations of actions. The frames in EGTEA videos have a 30:70 split action to non-action and the BEOID has a 40:60 split. Each action segment is defined as multiple consecutive frames classified as having an action. We apply temporal smoothing in the form of a 10 frame box filter to make the action segments more fluid.

We calculate the IOU at three different α thresholds. We compare our action proposal system against the Deep Action Proposal (DAP) system [18], which uses C3D features and an LSTM to generate proposals. We retrain the DAP network for both datasets using activations of the last fully connected layer of the C3D network and we extracted the features from our videos at a 8-frame temporal resolution. In our model we use the 4096 dimensional vector extracted from C3D, however for the DAP network we follow their procedure and reduce the features using PCA to 500 dimensions. DAP outputs many proposals along with a confidence score. Our model does not allow for overlapping action proposals, so in order to do a fair comparison we run Non-Maximal Suppression to select the most confident DAP proposals as action segments to test.

Table 1. Action Proposal Evaluation: IOU values over three different thresholds are given for each proposal method. The results from inputting different feature combinations to the bidirectional LSTM are shown. *frame* denotes pretrained VGG16 features.

Action Proposal Methods	BEOID			EGTEA		
	.5	.4	.3	.5	.4	.3
Deep Action Proposals [18]	0.1824	0.2625	0.4129	0.1757	0.2691	0.3966
<i>Bidirectional LSTM:</i>						
iDT	-	-	-	0.1763	0.2371	0.2952
frame	0.1846	0.2957	0.4359	0.1647	0.3235	0.5177
C3D	0.3720	0.4538	0.5712	0.2599	0.3501	0.4529
C3D + hands	0.5091	0.6037	0.6832	0.4051	0.5262	0.6249
C3D + hands + frame	0.3899	0.5114	0.6380	0.2344	0.3750	0.6094
C3D + hands + head motion	0.5224	0.6117	0.7235	0.3848	0.5482	0.6399

4.2 Task-Specific Object Recognition Results

We test our object classification network as a standalone module using five-fold cross-validation on EGTEA Gaze+ and BEOID. We extract one in every five frames from all action segments of the videos and we label each frame as the main task-specific object in the scene. We augment our data by using randomly resized crops spanning aspect ratios of 0.7 to 1 and random horizontal flips. At

test time, we crop a square of dimensions 400×400 around the gaze point and resize it into a square image of size 224×224 . If gaze points are not obtainable we simply resize the image to 224×224 .

We use an ImageNet pre-trained ResNet-101 architecture [33] which we fine-tune on each datasets. We train the network until convergence and use a batch size of 64. The optimization algorithm used was Adam [42] with learning rate of $3e-4$, weight decay of $1e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$. We test our fine-tuned object recognition network using five-fold cross-validation on the EGTEA Gaze+ and BEOID datasets and present top-1 and top-3 accuracy. For BEOID top-1 accuracy was 0.756 and top-3 accuracy was 0.946. For EGTEA top-1 accuracy was 0.542 and top-3 accuracy was 0.732. We observe that task-specific object recognition on the EGTEA dataset is substantially harder than in the BEOID dataset. This is due in part to the long-tail distribution of objects that is present in the EGTEA dataset, with some object classes presenting much larger amounts of instances than others. For example there are 8,361 images for the “cucumber” class whereas “teabag” presents 548 images and “coffee” presents only 155.

4.3 Video-Recipe Alignment Results

Evaluation measure. Our system has three modules: action proposal, object recognition and text to action segment alignment. The effectiveness of each module is assessed by substituting it with ground truth results Table 2. To evaluate the final alignment, we use the same metric as [11], which computes the precision of the frame level alignment predictions. The reported score is the averaged score of all the videos in the test set.

Baseline. We compare our alignment module to the recent alignment method of [11], which is also designed to align video and text. To do a fair comparison of alignment modules we test the baseline both with our deep features and the original hand-crafted features employed in the paper [11], which are bag-of-words representations of improved dense trajectories [41] for HOG, HOF and MBH channels. To represent the recipe steps, we use simple bag-of-words of

Table 2. Recipe-Video Alignment Accuracy: AP is given for each alignment method. The last two rows are ablation studies using ground truth (GT) data.

Method	Average Precision	
	BEOID	EGTEA
(i) Bojanowski et al. [11] w/ original features	0.2828	0.2169
(ii) Bojanowski et al. [11] w/ our features	0.4069	0.2292
(iii) Detected Objects, DAPs Action proposal [18]	0.2549	0.3175
(iv) Detected Objects, Our Action proposal	0.7641	0.4905
(v) Detected Objects, GT action segments	0.8272	0.5658
(vi) GT objects, GT action segments	0.8425	0.6830

ROOT verbs as it was the best performing strategy (as observed in [11]). Hyper-parameters are chosen by cross validation over our splits.

Methods. All results are reported in Table 2. **(i)** corresponds to the baseline [11], which *does* not use action proposal techniques. **(ii)** corresponds to our pipeline but uses the action proposal method introduced in [18]. **(iii)** corresponds to our full pipeline. Finally the in ablation studies, **(iv)** and **(v)**, we replace parts of our pipeline with the ground truth information in order to assess where improvements have to be made. More precisely **(iv)** uses ground truth action segments and **(v)** uses both ground truth action segments and objects.

Result analysis. We can view the results that use ground truth (GT) labels as upper bounds as well as an ablation study of the individual modules of our system. The result of using the GT object labels and the GT action segments **(vi)** tests only the alignment module. We can see that there is an 15.25% difference in AP between the datasets. Despite the BEOID being taken in multiple different settings it still outperforms EGTEA for alignment. We can attribute this to the higher density of the recipes, directly proving the difficulty of the task of sparse text to video alignment.

The poor performance of the alignment baseline shown in **(i)** and **(ii)** can be explained by the fact that the task considered here is more challenging than usual text-to-video alignment tasks due to our datasets’ high text sparsity. More priors have to be provided in order to achieve better performance for the baseline. This is demonstrated with **(iv)** which uses action proposal in order to reduce the search space for actions. We can see a clear difference in performance between methods **(i)** and **(ii)**, this difference corroborates the power of the egocentric cues in our features. When comparing the action proposal baseline **(iii)** and our system **(iv)**, we can see a extremely large boost in performance (+35.34%) induced by our new action proposal which is more suited for FP videos. Unlike the other methods, baseline **(iii)** performs worse on BEOID than EGTEA. When looking at the results on individual videos in BEOID we found the bad performance was due to failure on the short videos. The method was producing too many proposed segments for the short videos leading to a low AP score that was not reflected in the low IOU.

We can gain some interesting insights by looking at the experiments using ground truth labels. Using ground truth action segments **(iv)**, brings a relatively small improvement (BEOID: 6%, EGTEA: 7%) over our generated action proposals **(iii)**, thus showing that our action proposal system works extremely well for this task. The difference between using ground truth objects **(v)** and using object recognition **(iv)** is also small, meaning that our alignment system is able to handle mislabeled objects without breaking down. The the drop between **(v)** and **(iv)** is smaller for BEIOD than for EGTEA because our object recognition model achieves better performance on the BEOID dataset described in §4.2.

The performance of our upper bound for the alignment **(v)**, demonstrates the difficulty of the video-to-recipe alignment task. The challenge arises from the sparse nature of the recipe instructions. A recipe step gives an overall description of some activity to complete, however it does not explicitly state the individual

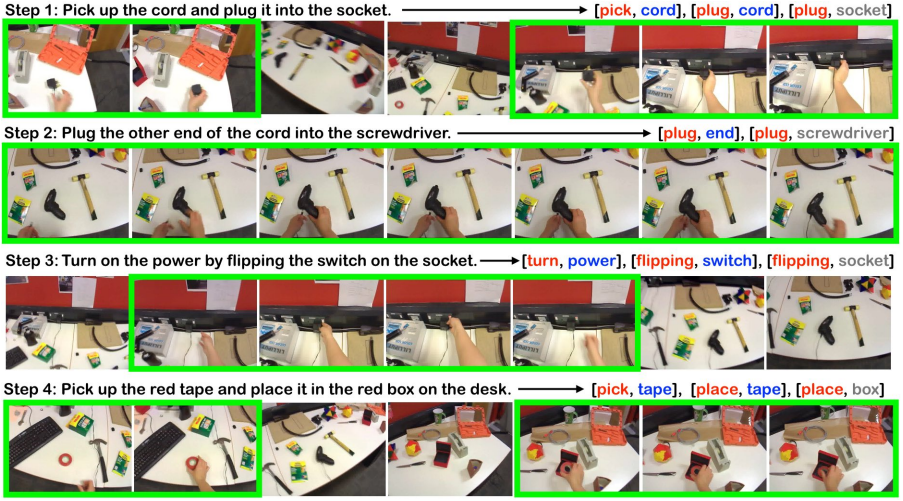


Fig. 4. Alignment results for BEOID. Each recipe step is linked to extracted action-object pairs, with actions in red, primary objects in blue, and secondary objects in gray. Green boxes identify action proposals which have been aligned to the step.

actions and objects required to complete this goal. For instance a recipe step such as “scramble 2 eggs over medium heat” must be mapped to the numerous actions of cracking two eggs, mixing them and putting them on a pan on the stove. We aim to learn these mappings and we supplement the learning with the semantic knowledge of language. The use of word embeddings in our alignment pipeline allows the system to extract contextual knowledge and find conceptual similarities between the recipe step and detected objects.

5 Conclusions

In this work we show a comprehensive method to align fine-grained actions in video to sparse recipe steps, shown in Figure 4. In order to do this we proposed a three-step pipeline which first predicts the onset and offset of action segments in the video, then retrieves task-specific objects that are manipulated during the action and finally uses this information to align the segment to a recipe step using natural language processing techniques. We present results of our recipe alignment pipeline on the new EGTEA dataset and the Bristol Egocentric Object Interaction dataset. For both datasets the system achieves an improved accuracy compared to a recent video and text alignment technique. Additionally, we have shown that by leveraging egocentric cues such as hand masks and head motion we can substantially improve upon state-of-the-art action proposal methods in first person videos.

References

1. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. *Robotics and autonomous systems* **57**(5) (2009) 469–483
2. Schaal, S.: Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences* **3** (1999) 233–242
3. Alayrac, J.B., Bojanowski, P., Agrawal, N., Sivic, J., Laptev, I., Lacoste-Julien, S.: Unsupervised learning from narrated instruction videos. In: *CVPR*. (2016) 4575–4583
4. Zhou, L., Xu, C., Corso, J.J.: Towards automatic learning of procedures from web instructional videos. *arXiv preprint arXiv: 1703.09788*
5. Nguyen, P.X., Rogez, G., Fowlkes, C., Ramanan, D.: The open world of micro-videos. *arXiv preprint arXiv:1603.09439* (2016)
6. Ma, M., Fan, H., Kitani, K.M.: Going deeper into first-person activity recognition. In: *CVPR*. (June 2016)
7. Li, Y., Ye, Z., Rehg, J.M.: Delving into egocentric actions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2015) 287–295
8. Xu, R., Xiong, C., Chen, W., Corso, J.J.: Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In: *AAAI*. Volume 5. (2015) 6
9. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: *Proceedings of the IEEE International Conference on Computer Vision*. Volume 1. (2017) 6
10. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: *Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE* (2013) 433–440
11. Bojanowski, P., Lajugie, R., Grave, E., Bach, F., Laptev, I., Ponce, J., Schmid, C.: Weakly-supervised alignment of video with text. In: *Proceedings of the IEEE ICCV*. (2015) 4462–4470
12. Damen, D., Leelasawassuk, T., Haines, O., Calway, A., Mayol-Cuevas, W.W.: You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In: *BMVC*. Volume 2. (2014) 3
13. Malmaud, J., Huang, J., Rathod, V., Johnston, N., Rabinovich, A., Murphy, K.: What’s cookin’? interpreting cooking videos using text, speech and vision. *NAACL* (2015)
14. Huang, D.A., Lim, J.J., Fei-Fei, L., Niebles, J.C.: Unsupervised visual-linguistic reference resolution in instructional videos. *arXiv preprint arXiv:1703.02521* (2017)
15. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: *CVPR*. (2016) 2678–2687
16. Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., Gould, S.: Dynamic image networks for action recognition. In: *CVPR*. (2016) 3034–3042
17. Singh, B., Marks, T.K., Jones, M., Tuzel, O., Shao, M.: 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*. (2016) 1961–1970
18. Escorcia, V., Heilbron, F.C., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: *ECCV*, Springer (2016) 768–784
19. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: *ICCV*. (2015) 4489–4497

20. Richard, A., Kuehne, H., Gall, J.: Weakly supervised action learning with rnn based fine-to-coarse modeling. arXiv preprint arXiv:1703.08132 (2017)
21. Huang, D.A., Fei-Fei, L., Niebles, J.C.: Connectionist temporal modeling for weakly supervised action labeling. In: European Conference on Computer Vision, Springer (2016) 137–153
22. Pirsiaavash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: CVPR, 2012 IEEE Conference on, IEEE (2012) 2847–2854
23. Kitani, K.M., Okabe, T., Sato, Y., Sugimoto, A.: Fast unsupervised ego-action learning for first-person sports videos. In: CVPR, IEEE (2011) 3241–3248
24. Zhang, Y.C., Li, Y., Rehg, J.M.: First-person action decomposition and zero-shot learning. In: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, IEEE (2017) 121–129
25. Li, Y., Fathi, A., Rehg, J.M.: Learning to predict gaze in egocentric video. In: ICCV. (2013) 3216–3223
26. Yamada, K., Sugano, Y., Okabe, T., Sato, Y., Sugimoto, A., Hiraki, K.: Attention prediction in egocentric video using motion and visual saliency. Advances in image and video technology (2012) 277–288
27. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015) 3431–3440
28. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE international conference on, IEEE (2011) 2564–2571
29. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Readings in computer vision. Elsevier (1987) 726–740
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2015)
33. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
34. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014) 580–587
35. Girshick, R.: Fast r-cnn. In: ICCV. (2015) 1440–1448
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., eds.: Advances in Neural Information Processing Systems 28. Curran Associates, Inc. (2015) 91–99
37. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016) 779–788
38. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV, Springer (2016) 21–37
39. De Marneffe, M.C., MacCartney, B., Manning, C.D., et al.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC. Volume 6., Genoa (2006) 449–454
40. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

41. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR, 2011 IEEE Conference on, IEEE (2011) 3169–3176
42. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)