

# Learning to Recognize Objects in Egocentric Activities

Alireza Fathi<sup>1</sup>, Xiaofeng Ren<sup>2</sup>, James M. Rehg<sup>1</sup>

<sup>1</sup> College of Computing  
Georgia Institute of Technology

{afathi3, rehg}@cc.gatech.edu

<sup>2</sup> Intel Labs Seattle

xiaofeng.ren@intel.com

## Abstract

*This paper addresses the problem of learning object models from egocentric video of household activities, using extremely weak supervision. For each activity sequence, we know only the names of the objects which are present within it, and have no other knowledge regarding the appearance or location of objects. The key to our approach is a robust, unsupervised bottom up segmentation method, which exploits the structure of the egocentric domain to partition each frame into hand, object, and background categories. By using Multiple Instance Learning to match object instances across sequences, we discover and localize object occurrences. Object representations are refined through transduction and object-level classifiers are trained. We demonstrate encouraging results in detecting novel object instances using models produced by weakly-supervised learning.*

## 1. Introduction

This paper is motivated by the desire to automatically learn rich models of human activities and behavior from weakly-labeled video sequences. The ability to provide fine-grained annotations of an individual's behavior throughout their daily routine would be extremely interesting in a range of healthcare applications, such as assessing activities of daily living in elderly populations. However current approaches to activity recognition depend upon highly-structured activity models and large amounts of labeled training data to obtain good performance [11]. Recently some authors have demonstrated the ability to automatically acquire labels for simple actions and sign language using scripts and close-caption text [5, 14]. While these are promising approaches, the transcripts they require are not generally available for home video.

Our approach is based on the observation that many household activities involve the manipulations of objects, and that a simple but effective activity model can be con-

structed from patterns of object use [24]. However previous work in this area required the ability to identify when a particular object is being manipulated by the user (e.g. by means of an RFID sensor) in order to collect training data. This paper explores the hypothesis that object instances can be detected and localized simply by exploiting the co-occurrence of objects within and across the labeled activity sequences. We assume that we are given a set of training videos which are coarsely labeled with an activity and the list of objects that are employed, but without any object localization information. The difficulty of this learning problem stems from the fact that there are many possible candidate regions which could contain objects of interest, and a standard learning method could succeed only if an extremely large amount of diverse training examples were available.

We propose to address the problem of limited training data by adopting the paradigm of egocentric or first-person video (i.e. video captured from a wearable camera that images the scene directly in front of the user at all times). In contrast to the established third-person video paradigm, the egocentric paradigm makes it possible to easily collect examples of natural human behaviors from a restricted vantage point. The stability of activities with respect to the egocentric view is a potentially powerful cue for weakly-supervised learning. Egocentric vision provides many advantages: (1) there is no need to instrument the environment by installing multiple fixed cameras, (2) the object being manipulated is less likely to be occluded by the user's body, and (3) discriminative object features are often available since manipulated objects tend to occur at the center of the image and at an approximately constant size. In this paper we will show how the domain knowledge provided by egocentric vision can be leveraged to build a bottom-up framework for efficient weakly-supervised learning of models for object recognition.

Our method consists of two main stages. In the first stage, our goal is to segment the active objects and hands

from unimportant background objects. As humans, we can easily differentiate between background objects and the ones we are attending to during the course of an activity. Likewise, our learning method must be able to focus only on the objects being manipulated by our hands in order to be able to accurately distinguish different daily activities. Weakly supervised learning of objects will not be feasible unless we are able to ignore the dozens of unrelated and potentially misleading objects which occur in background. In the second stage of the method, we learn object appearances based on the patterns of object use provided as a weak source of information with the training data. We first use a MIL framework to initialize a few regions corresponding to each object type, and then we propagate the information to other regions using a semi-supervised learning approach.

## 2. Previous Work

**Egocentric Vision:** Recently there has been an increasing interest in using wearable cameras, motivated by the advances in hardware technology. Early studies of wearable cameras can be found in [15, 18]. Spriggs et. al [20] address the segmentation and activity classification using the first-person sensing. Ren and Gu [16] showed that figure-ground segmentation significantly improves object recognition results in egocentric video. In contrast, we show how the egocentric paradigm can be leveraged to learn object classification and segmentation with very weak supervision.

**Weakly Supervised Recognition:** Reducing the amount of required supervision is a popular topic in computer vision, given the expense of labeled image data. Recent works have tried to provide different sources of automatic weakly supervised annotations by using web data [4] or cheap human annotation systems such as Amazon’s Mechanical Turk. Others have studied probabilistic clustering methods such as pLSA and LDA for unsupervised discovery of object topics from unlabeled image collections [19].

Unsupervised methods are not necessarily appropriate for learning object categories, since they have no guarantee of finding topics corresponding to object classes. An alternative approach is to expand a small set of labeled data to the unlabeled instances using semi-supervised learning. For example, Fergus et. al [7] leverage the semantic hierarchy from WordNet to share labels between objects.

More recently, Multiple Instance Learning (MIL) has shown great promise as a method for weakly supervised learning in computer vision communities [1, 9, 5, 23, 6]. In MIL, labels are provided for bags containing instances (e.g. images containing objects). These information are then leveraged to classify instances and bags. Buehler et. al [5] localize signs in footage recorded from TV with a given script. Vijayanarasimhan and Grauman [23] learn discriminative classifiers for object categories given images returned from keyword-based search engines.

In this paper, we show that egocentric video provides a new paradigm for weakly-supervised object learning. We present results on reliably carving object classes out of video of daily activities by leveraging the constraints provided by the egocentric domain.

**Objects and Activities:** Li and Fei-Fei [12] use the object categories that appear in an image to identify an event. They provide ground truth object labels during learning in order to categorize object segments. Ryoo and Aggarwal [17] combine object recognition, motion estimation and semantic information for the recognition of human-object interactions. Their experiments involve four categories of objects (including humans). Gupta et. al [8] use a Bayesian approach to analyze human-object interactions, with a likelihood model that is based on hand trajectories.

Wu et. al [24] perform activity recognition based on temporal patterns of object usage, but require RFID-tagged objects in order to bootstrap appearance-based classifiers. In contrast to these methods we recognize and segment foreground objects from first-person view given a very weak amount of supervisory information.

**Video Segmentation:** Background subtraction is a well addressed problem for fixed-location cameras. Various techniques have been developed, such as adaptive mixture-of-Gaussian model [21]. However, problem is much harder for a moving camera and is usually approached by motion segmentation given sparse feature correspondences (e.g. [25]). The most relevant work to our background subtraction section is Ren and Gu [16]. Given ground-truth segmentations, they learn a classifier on motion patterns and foreground object prior location, specific to their egocentric camera. In comparison, our Segmentation method is completely unsupervised and achieves a higher accuracy.

## 3. Segmentation

In this Section we describe a bottom-up segmentation approach which leverages the knowledge from egocentric domain to decompose the video into background, hands and active objects. We first segment the foreground regions containing hands and active objects from background as described in Sec 3.1. Then in Sec 3.2 we learn a model of hands and separate them from objects, and further refine them into left and right hand areas. In Section 4 we show this step is crucial for weakly supervised learning of objects.

### 3.1. Foreground vs Background Segmentation

Our foreground segmentation method is based on a few assumptions and definitions: (1) we assume the background is static in the world coordinate frame, (2) we define foreground as every entity which is moving with respect to the static background, (3) we assume background objects are usually farther away from the camera than foreground objects and (4) we assume we can build a panorama of the

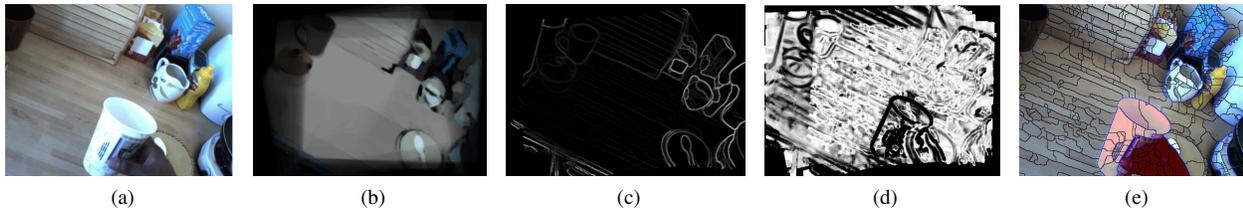


Figure 1: The Background Model. (a) A sample frame from Intel dataset [16], (b) mean color of color-texture background model, (c) mean intensity of background boundary model, (d) the edges corresponding to the object boundary in the sample image do not match the background model, (e) foreground segment is depicted in red.

background scene by stitching the background images together using an affine transformation. The fourth assumption is basically assuming that the background is roughly on a plane or far enough from the camera. An object will be moving with respect to the background when it is being manipulated by hands. When the subject finishes a sub-task and stops manipulating the object, the object will become a part of background again.

Our segmentation method is as follows. We first make an initial estimate of background regions in each image by fitting a fundamental matrix to dense optical flow vectors. We make temporally local panoramas of background given our initial background region estimates. Then we register each image into its local background panorama. The regions in the image which do not match the background scene are likely to be parts of foreground. We connect the regions in sequence of images spatially and temporally and use graph-cut to split them into foreground and background segments.

We split the video into short intervals and make local background models for each. The reason is that the background might change over time, for example the subject might finish manipulating an object and leave it on the table, letting it become a part of background. We initially approximately separate foreground and background channels for each image by fitting a fundamental matrix to its optical flow vectors. We compute the flow vectors to its few adjacent frames. For each interval we choose a reference frame whose initial background aligns the best to other frames.

We build two kinds of temporally local models for background (panoramas): (1) a model based on color and texture histogram of regions and (2) a model of region boundaries. To build these models, we fit an affine transformation to the initial background SIFT feature correspondences of each frame in the interval, and the reference frame. We stitch these images using affine transformation. After fixing the images to the reference frame coordinate, we build the color-texture and boundary background models. This is by computing a histogram of values extracted from interval images corresponding to each location in the background panorama. Here we describe these two background models in more details:

**Color-Texture Background Model:** We segment each image into small super-pixels [2], as shown in Fig 1(e).

We represent each super-pixel with a color and texture histogram. We compute texture descriptors [22] for each pixel and quantize them to 256 kmeans centers to produce the texture words. We sample color descriptors for each pixel and quantized them to 128 kmeans centers. We cluster the super-pixels by learning a metric which forces similarity and dissimilarity constraints between initial foreground and background channels. Euclidean distance between super-pixels might not be a good measure, since for example the color of a region on hand might look very similar to a super-pixel corresponding to background. As a result, we learn a metric on super-pixel distance which satisfies the following properties: (1) the distance between two spatially adjacent super-pixels in background is low, (2) the distance between temporally adjacent super-pixels with strong optical flow link is low and (3) the distance between a super-pixel in foreground and a one in background is high. We use the EM framework introduced in Basu et. al [3] to cluster the super-pixels into multiple words using the mentioned similarity and dissimilarity constraints.

We build a histogram of words for each location in the background model from the values that correspond to that location in each interval image. We have depicted the mean color of an example background model in Fig 1(b). Given the computed background model, we estimate the probability of image super-pixels belonging to background by intersecting their color-texture word with the histogram of their corresponding region in background model.

**Boundary Background Model:** The hierarchical segmentation method of [2] provides a contour significance value for pixels of each image. We transform contour images of each interval to the reference coordinate. For each pixel in the background model we build a histogram of contour values. We have shown average contour intensity for an example image in Fig 1(c). For each super-pixel we measure how well its contour matches to the background model as shown in Fig 1(d). For the super-pixels corresponding to the background, their edges will match the background model with a high probability (some times object edges might create occlusions on background regions), while the super-pixels corresponding to foreground region usually do not match with the background model.

Now that the foreground and background priors are com-

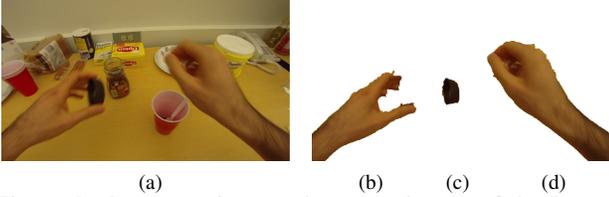


Figure 2: Segmentation results on an image of the Instant Coffee making activity is shown: (a) original image, (b) left hand segment, (c) object segment and (d) right hand segment.

puted for each super-pixel, we connect the super-pixels to each other both spatially and temporally. We connect adjacent super-pixels in each image and set the connection weight based on their boundary significance. We further connect the super-pixels in adjacent frames based on optical flow and SIFT correspondences [13]. We use a Markov Random Field model to capture the computed foreground and background priors, as well as spatial and temporal connections between super-pixels. We solve this MRF using graph-cut.

### 3.2. Hands vs Objects Segmentation

We use the fact that the hand presence is dominant in foreground to learn a model for hands. As objects are manipulated over time, they become a temporary part of foreground, while hands are present most of the time. Given the foreground/background segmentation computed in Section 3.1, we build color histograms for foreground and background regions through out each activity.

A super-pixel which has a very high similarity to foreground color histogram is more likely to belong to one of the hands. To set the hand prior for a super-pixel we intersect its color histogram with foreground color histogram and divide it to its intersection score with background color histogram. We set the prior of being object to the median of super-pixel priors for hands. We use graph-cut to segment the foreground into hands and objects.

Given the hand regions extracted in previous step, we segment left and right hands. We use the prior information that in egocentric camera left hand tends to appear in left side of image and right hand usually appears in the right. We set priors on super-pixels based on their location on horizontal axis of image and use graph-cut to segment them into two regions. An example of hand vs objects segmentation is shown in Fig 2.

## 4. Automatic Object Extraction

Given a thousands frame long activity image sequence, our goal is to carve out and label the few participating object categories without having any prior information on their spatial or temporal location. This is a fundamental and challenging problem. However, it becomes feasible given the knowledge and constraints existing in egocentric video. The key idea is that each object is used only in a subset of

activities and is not included in the rest. An object might be present in the background region of all activity videos, however we use our capability to segment the active object regions to remove the background noise.

We split the active object mask into multiple fine regions. Our goal is to learn an appearance model for each object type, and based on that assign each fine region to an object category. To solve this problem, we first initialize each object class by finding a very few set of fine regions corresponding to it. For this purpose, we extend the diverse density based MIL framework of [6] to infer for multiple classes. In our problem instances represent object regions and bags represent the set of all regions in video. The MIL framework finds patterns in regions that occur in positive bags (the sequences containing the object of interest) but not in negative ones. Positive bags are the sequences in which the object of interest exist. We need to infer for multiple classes simultaneously in order to discriminate different objects. We further use equality constraints to assign the same object category label to regions with significant temporal connections (with corresponding SIFT feature). These constraints help our method to assign a region to an object class based on the majority votes from its connections.

Given a few regions corresponding to each object class, we propagate these labels to unlabeled foreground regions. Then we learn a classifier for each object class in order to recognize regions in test activities.

We believe egocentric domain makes the object extraction step feasible. In egocentric domain, we are able to segment the active object region from background and extract regions consistent in shape, size and appearance corresponding to the same object instance from various activities.

### 4.1. Object Initialization

Chen et. al [6] extend ideas from the diverse density framework to solve the MIL problem. Here we further extend their method to (1) handle multiple instance labels simultaneously and (2) apply mutual equality constraints among some instances in each bag. They find a similarity measure between every instance  $x_i$  and every bag  $B_j$ , using the following equation

$$\begin{aligned} Pr(x_i|B_j) &= s(x_i, B_j) \\ &= \max_{x_k \in B_j} \exp\left(-\frac{\|x_i - x_k\|^2}{\sigma^2}\right) \end{aligned}$$

where  $\sigma$  is a pre-defined scaling factor. Given  $m$  instances in total and  $l$  bags, the following  $m \times l$  matrix is built:

$$\begin{bmatrix} s(x_1, B_1) & \dots & s(x_1, B_l) \\ s(x_2, B_1) & & s(x_2, B_l) \\ \vdots & \vdots & \vdots \\ s(x_m, B_1) & \dots & s(x_m, B_l) \end{bmatrix} \quad (1)$$

In this matrix each row corresponds to similarities of an instance to all the bags, and each column captures the similarity of each bag to all the instances. For our task, the instances correspond to image regions and bags correspond to activities. Our objective function is to find a sparse set of instances corresponding to each object category which have a high similarity to positive bags and a low similarity to negative bags.

We extend their formulation to infer multiple instance classes simultaneously. Instead of minimizing for a single vector  $w$ , we are looking for  $r$  sparse vectors  $w_1, w_2, \dots, w_r$  where each  $w_c$  is  $m$  dimensional and is positive at a few representative instances of object class  $c$  and is zero every where else.

We further add equality constraints  $w_c(p) = w_c(q)$  between a pair of elements  $(p, q)$  in all  $w_c, c = \{1, \dots, r\}$  if there is a temporal link between regions corresponding to instances  $p$  and  $q$ . We optimize the following linear program which minimizes the L1-norm of  $w_c$  vectors and returns sparse vectors:

$$\min_{w, b, \xi} \left\{ \sum_{c=1}^r |w_c| + C \sum_{j=1}^l \xi_j \right\} \quad (2)$$

$$w_c \cdot s(:, j) \geq w_{c'} \cdot s(:, j) + \delta_{c, c', B_j} - \xi_j \text{ s.t. } \forall c, c' = \{1, \dots, r\}$$

$$w_c(p) = w_c(q) \text{ s.t. } \forall c = \{1, \dots, r\}, (p, q) \in \mathcal{C}$$

$$\xi_j \geq 0 \text{ s.t. } \forall j = \{1, \dots, l\}$$

where  $\xi_j$  is slack variable for bag  $j$ ,  $C$  is a constant,  $s(:, j)$  contains the similarity vector of instances to bag  $j$ ,  $\delta_{c, c', B_j}$  is 1 if bag  $B_j$  is positive for class  $c$  and negative for class  $c'$  and 0 otherwise, and  $\mathcal{C}$  contains the set of equality constraints between instances.

We describe each region with a 32 dimensional feature vector by compressing its color and texture histograms using PCA. This representation is able to both describe objects with and without texture. The similarity between a region  $x_i$  and a bag  $B_j$  is computed based on the distance between  $x_i$ 's feature vector and its closest neighbor among all regions in  $B_j$  as in Eq 1. We observe that taking region shape and sizes into account enhances the performance. Regions corresponding to different objects might have similar texture and color appearance. For instance, there are white regions corresponding to spoon, sugar and tea bag, but their sizes and shapes are different. To take region shapes and sizes into account, we fit an ellipse to each region and reweight the computed distances based on the relative ratio of ellipse axis for matched regions. We then optimize the multi-class L1-SVM in Eq 2 to find a few positive instances for each object class.

## 4.2. Object Classification

Our goal is to automatically assign object labels to all foreground regions, while initially we have only a few labeled ones. To do so, we first propagate the labels using the region connectivities in video. For each activity sequence we build a pairwise adjacency matrix  $W$  by connecting its regions to their spatial and temporal neighbors. We set the class label of the regions which were initialized in previous step. To expand the label set, we minimize the following objective function

$$E(y) = \frac{1}{2} \sum_{i,j} w_{ij} \delta(y_i - y_j)^2$$

where  $y_i$  is the label of region  $i$  and  $w_{ij}$  is the similarity weight connecting regions  $i$  and  $j$  in computed adjacency matrix  $W$ . We estimate  $y$  for unlabeled regions by computing the harmonic function  $f = \operatorname{argmin}_f |f_L| E(f)$  as described in [26]. Harmonic solution  $f$  is a  $m \times r$  matrix where  $m$  is the number of regions and  $r$  is the number of labeled classes and can be computed in polynomial time by simple matrix operations. We fix the label of an unlabeled region  $i$  to  $c$ , if  $f(i, c)$  is greater than  $f(i, c')$  for  $\forall c' = \{1, \dots, r\}$ , and further  $f(i, c)$  is greater than a threshold.

After expanding the initial labels, we learn a classifier for each object class using Transductive SVM (TSVM) [10]. To train a classifier for a particular object category, we set the label of its assigned regions to 1, the label of regions in foreground regions of negative bags to  $-1$  and the label of regions assigned to other object classes to  $-1$  as well. We set the label of unlabeled regions to 0. TSVM as described in [10], iteratively expands the positive and negative classes until convergence.

## 5. Experiments and Datasets

In this Section we present a new egocentric daily activity dataset on which we validate our results.

### 5.1. Dataset

We collect a dataset of 7 daily activities from egocentric point of view performed by 4 subjects. We mount a GoPro camera on a baseball cap, which is positioned so as to cover the area in front of the subject's eyes. The camera is fixed and moves rigidly with the head. The camera captures and stores a high definition  $1280 \times 720$ , 30 frame per second 24-bit *RGB* image sequence. We extract frames with a 15 fp rate from the recorded videos. The total number of frames in the dataset are 31,222.

Our dataset contains the following activities: Hotdog Sandwich, Instant Coffee, Peanut Butter Sandwich, Jam and Peanut Butter Sandwich, Sweet Tea, Coffee and Honey, Cheese Sandwich. In Table 1 we have listed the activities and their corresponding objects. We use activities of subjects (2,3,4) as training data to learn object classifiers, and test on the activities of the subject 1. The set of objects

Activities	Objects
Hotdog Sandwich	Hotdog, Bread, Mustard, Ketchup
Instant Coffee	Coffee, Water, Cup, Sugar, Spoon
Peanut-butter Sandwich	Peanut-butter, Spoon, Bread, Honey
Jam Sandwich	Jam, Chocolate Syrup, Peanut-butter, Bread
Sweet Tea	Cup, Water, Tea bag, Spoon
Cheese Sandwich	Bread, Cheese, Mayonnaise, Mustard
Coffee and Honey	Coffee, Cup, Water, Spoon, Honey

Table 1: Our dataset consists of 7 activities and 16 objects.

appearing in each activity is known for training sequences, while for the test sequence they are unknown.

To validate our object recognition accuracy, we manually assign one object label to each frame of the test activities. In case of more than one foreground object, we assign the label to the object we think is the most salient. We later use these ground-truth annotations to measure our method’s performance.

## 5.2. Results

In this section, we present results that demonstrate the effectiveness of our method in segmenting and labeling ego-centric videos of daily activities.

**Segmentation:** We compare the accuracy of our foreground background segmentation approach to Ren and Gu [16]. Our method is completely unsupervised while Ren and Gu use an initial ground-truth segmentation set of images to learn priors on hand and object locations, optical flow magnitude and other features. To compare our results, we manually annotated the foreground segmentations for 1000 frames in the first sequence of Intel dataset introduced in [16] using the interactive segmentation toolkit of Adobe After Effects. Our method achieves 48% segmentation error rate and outperforms their method which results in 67% error. We calculate individual image errors by dividing the difference between the areas of ground-truth and results to the area of ground-truth foreground region. We average these numbers over the image sequence.

**Object Recognition:** Our method first initializes a few instances of each object in Section 4.1. Our object initialization results have a very high precision. We have shown 2 representative examples for each object category in Fig 3. There are 4 pair of mutually co-occurring object instances. As a result our method is not able to distinguish between them. We merge each pair into one class and reduce the number of object classes from 16 to 12.

In Section 4.2, we expand the initial object regions and learn a classifier for each object from the training sequences. We test our method on the test sequences as follows. We use our segmentation method to automatically segment the active object area in test images as described in Section 3. This area might contain more than one active object. We classify each region in the active object area using our learned object appearance models. Examples are shown

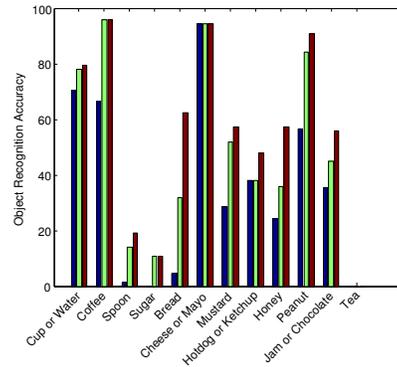


Figure 6: Object recognition accuracy. Random classification chance is 8.33%. Blue bars show how well the highest score detection in each frame matches the ground-truth object label. Green and red bars, depict these results for any of the 2 and 3 highest score detections. We provide these results since there might be more than one active object in a frame but the ground-truth provides only one label per frame.

in Fig 4. In Fig 5 we have shown a few interesting failures. We compare the labeling accuracy of our algorithm with the ground-truth object labels in Fig 6.

Activity	KNN(Active Objects Histogram), k=1,2,3			KNN(All Objects Histogram), k=1,2,3		
Hotdog	Hotdog	Hotdog	Tea	JamNut	Peanut	Hotdog
Coffee	Coffee	CofHoney	Coffee	CofHoney	Tea	Tea
Peanut	Peanut	Peanut	JamNut	Cheese	CofHoney	Tea
JamNut	JamNut	Peanut	Peanut	Cheese	Cheese	Hotdog
Tea	Coffee	Tea	Coffee	Tea	Coffee	Tea
CofHoney	Coffee	CofHoney	Coffee	CofHoney	Tea	Tea
Cheese	Cheese	Cheese	Cheese	Cheese	Peanut	Cheese

Table 2: We represent each video with either the histogram of its active objects or the histogram of its all objects both in foreground and background. Given the computed histograms, in each case we find the first 3 nearest training activities to each test activity.

In Fig 7, we show that our learning method is more capable in comparison to a general SVM-based MIL [1].

It is shown that activities can be categorized based on their object use patterns [24]. Segmenting the active object out of background is a crucial step, without which activity comparison based on all the objects appearing in video returns poor results. In Table 2 we show that activities can be reliably compared based on the histogram of active objects found by our method over time. In comparison, we show that building a histogram of all object detections both in foreground and background doesn’t perform as good.

## 6. Conclusion

We have developed a weakly supervised technique able to recognize objects by carving them out of large egocentric activity sequences. This is an intractable task in a general setting, however our algorithm utilizes the domain specific

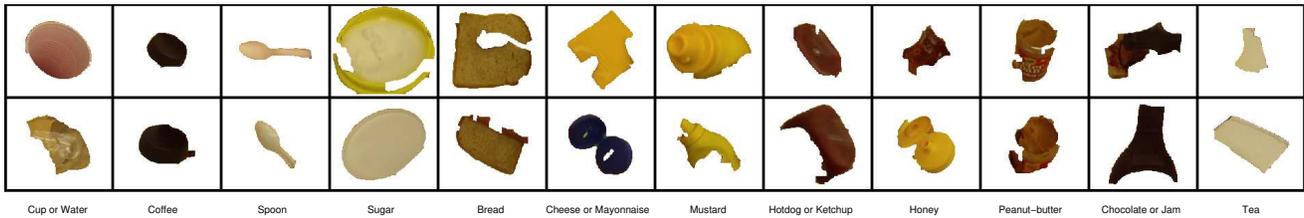


Figure 3: We first automatically initialize a few object regions corresponding to each class as described in Section 4.1. Two representative initialized regions are shown for each object category.

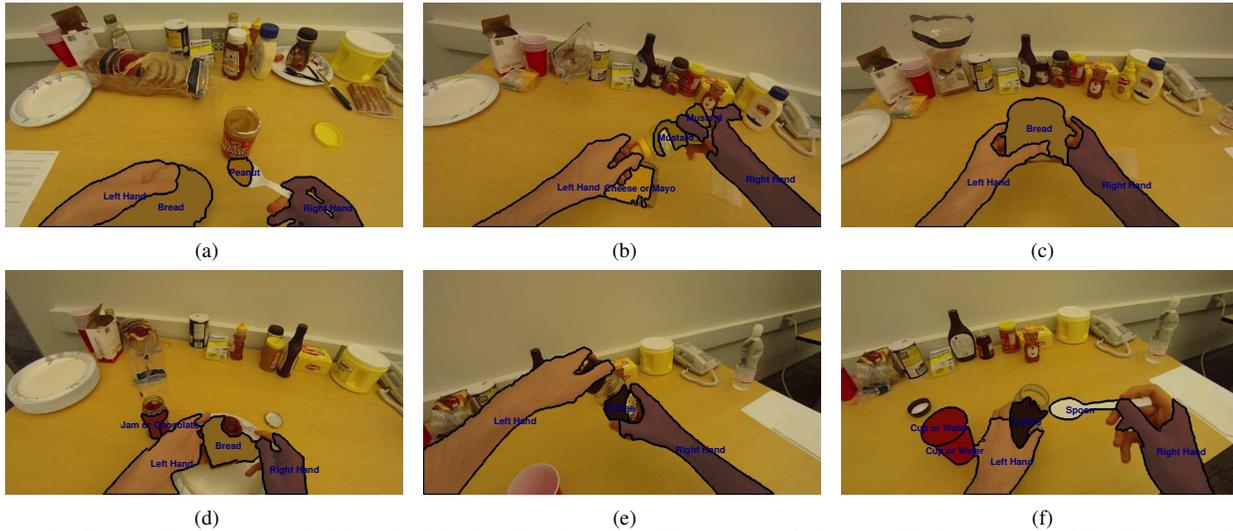


Figure 4: Our method extracts left hand, right hand and active objects at each frame. We learn a classifier for each object class and assign each non-hand region in foreground segment to the class with the highest response.

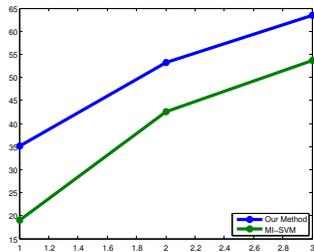


Figure 7: The object regions are sparse in the foreground and each object might contain regions with completely different appearance. As a result an algorithm such as MI-SVM [1] which doesn't take these considerations into account results in lower recognition accuracy. Random accuracy is 8.33%. We compare the results by matching the highest 1, 2 and 3 highest detection scores to ground-truth annotations.

knowledge from the first-person view to make it feasible.

Our method automatically segments the active object regions, assigns a few regions to each object and propagates its information using semi-supervised learning. We show that our method can reliably compare activity classes based on their object use patterns. We believe promising future directions for research involves combining the object and actions as context to each other in order to enhance activity recognition results. We have released our dataset at <http://cpl.cc.gatech.edu/projects/GTEA/>.

## 7. Acknowledgments

We would like to thank Jessica Hodgins and Takaaki Shiratori from Disney Research for providing us the head-mounted camera system, Priyal Mehta for helping in annotating the data and Ali Farhadi for useful discussions. Portions of this research were supported in part by NSF Award 0916687 and ARO MURI award 58144-NS-MUR.

## References

- [1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2003. 3282, 3286, 3287
- [2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: an empirical evaluation. In *CVPR*, 2009. 3283
- [3] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *International Conference on Knowledge Discovery and Data Mining*, 2004. 3283
- [4] T. L. Berg and D. A. Forsyth. Animals on the web. In *CVPR*, 2006. 3282
- [5] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching tv (using weakly aligned subtitles). In *CVPR*, 2009. 3281, 3282

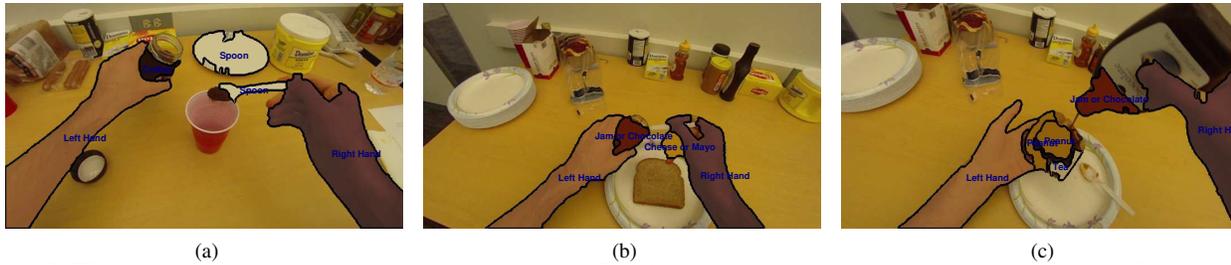


Figure 5: We show a few interesting failures of our method. In (a), the plate is mistakenly segmented as a part of active object regions. The spoon classifier fires on plate region as a result of their similar appearance. In (b), a small region belonging to hand is classified as cheese. In (c), a small part of bread is labeled as tea.

- [6] Y. Chen, J. Bi, and J. Z. Wang. Miles: multiple-instance learning via embedded instance selection. In *PAMI*, 2006. 3282, 3284
- [7] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic label sharing for learning with many categories. In *ECCV*, 2010. 3282
- [8] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: using spatial and functional compatibility for recognition. In *PAMI*, 2009. 3282
- [9] N. Ikidler-Cinbis and S. Sclaroff. Object, scene and actions: combining multiple features for human action recognition. In *ECCV*, 2010. 3282
- [10] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999. 3285
- [11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 3281
- [12] L. J. Li and L. Fei-Fei. What, where and who? classifying event by scene and object recognition. In *CVPR*, 2007. 3282
- [13] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 3284
- [14] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 3281
- [15] A. Pentland. Looking at people: sensing for ubiquitous and wearable computing. In *PAMI*, 2000. 3282
- [16] X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *CVPR*, 2010. 3282, 3283, 3286
- [17] M. Ryoo and J. Aggarwal. Hierarchical recognition of human activities interacting with objects. In *CVPR*, 2007. 3282
- [18] B. Schiele, N. Oliver, T. Jebara, and A. Pentland. An interactive computer vision system - dypers: dynamic personal enhanced reality system. In *ICVS*, 1999. 3282
- [19] J. Sivic, B. Russell, A. A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. In *ICCV*, 2005. 3282
- [20] E. H. Spriggs, F. De La Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *Egovision Workshop*, 2009. 3282
- [21] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, volume 2, pages 246–252, 1999. 3282
- [22] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. In *IJCV*, 2005. 3283
- [23] S. Vijayanarasimhan and K. Grauman. Keywords to visual categories: multiple-instance learning for weakly supervised object categorization. In *CVPR*, 2008. 3282
- [24] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *CVPR*, 2007. 3281, 3282, 3286
- [25] J. Yan and M. Pollefeys. A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006. 3282
- [26] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003. 3285