Adaptive Binarization for Weakly Supervised Affordance Segmentation

Johann Sawatzky
University of Bonn
sawatzky@iai.uni-bonn.de

Juergen Gall
University of Bonn
gall@iai.uni-bonn.de

Abstract

The concept of affordance is important to understand the relevance of object parts for a certain functional interaction. Affordance types generalize across object categories and are not mutually exclusive. This makes the segmentation of affordance regions of objects in images a difficult task. In this work, we build on an iterative approach that learns a convolutional neural network for affordance segmentation from sparse keypoints. During this process, the predictions of the network need to be binarized. To this end, we propose an adaptive approach for binarization and estimate the parameters for initialization by approximated cross validation. We evaluate our approach on two affordance datasets where our approach outperforms the state-of-the-art for weakly supervised affordance segmentation.

1. Introduction

Affordances are properties of regions of scenes or objects which indicate their relevance for a certain functional interaction. Examples are *holdable* for the external part of a mug or *drivable* for a road. Localizing affordances is therefore an important task for autonomous systems that interact with the environment [17] as well as assistive systems that support visually impaired people [11]. Segmenting affordance regions, however, is a more difficult task than classical semantic image segmentation, which focuses on objects or categories that summarize regions of similar appearance like sky or grass.

Affordances are not only much more fine-grained than object categories, they represent a more abstract concept that generalizes across object categories. This requires that an affordance segmentation approach recognizes affordance for a previously unseen object class. For instance, it should generalize *cuttable* from the blades of scissors or knives to the blade of a saw. Furthermore, affordance segmentation is a multi-label segmentation problem since affordance regions spatially overlap. This is in contrast to classical semantic image segmentation where the categories are mutually exclusive. This is in particular for weakly supervised learning, as it is addressed in this work, a big challenge.

Since acquiring pixelwise segmentation masks for training is very time consuming, methods for weakly supervised learning have been proposed that learn to segment object categories either from image labels [25, 16] or keypoint annotations [1]. Our work builds on [27] where an approach for affordance segmentation has been proposed that uses only keypoint annotations as weak supervision for training. The approach employs an iterative approach alternating between updating the parameters of a convolutional neural network and estimating the unknown segmentation masks of the training images. During this process, the predictions of the network need to be binarized. Since thresholding at the 50% decision boundary, as it is done in a fully supervised setting, does not work for weakly supervised learning, an additional binary segmentation step is used in [27].

In this work, we propose an adaptive approach that determines the threshold for binarization for each training image and affordance class. Our approach not only avoids the additional segmentation step used in [27] but also increases the affordance segmentation accuracy substantially. Since the initialization of the affordance segments based on the keypoints has a high impact on the accuracy, we show further how the parameters for initialization can be determined by cross validation using an approximation of the Jaccard index based on the given keypoints. We evaluate our approach on the CAD 120 affordance dataset [27] and the UMD part affordance dataset [22] using two different network architectures. In all settings, our approach outperforms [27]. On the CAD 120 affordance dataset, the mean accuracy is increased by up to 17 percentage points compared to [27].

2. Related Work

Our work is related to affordance modeling as well as weakly supervised semantic segmentation methods. An affordance is an attribute of an object part that implies the possible usage of this object. Assigning affordances to object parts is not trivial, while [27] and [22] simply let a human annotator decide, others use more sophisticated statis-

tics like mining of word co-occurrences [3] or object attribute graph structures [28].

Modeling affordances can be the final goal or an intermediate step. In [2], affordances of an object are defined in terms of hand poses during interaction. These affordances are used along with object appearance features for object classification. [15] apply implicit affordance modeling for simultaneous hand action and object detection. While [29] combine object affordances with physical observables and human pose to obtain a generative model for object tasks, [19] use object affordance, object appearance and human poses for action detection.

Since recognizing affordances is crucial for the constructive manipulation of objects by robots, several approaches that require full supervision have been proposed. While some use geometric information like orientation of object surfaces [13], 3d point clouds [14], or normal and curvature features [22], others rely only on appearance. [9] predict attributes from appearance and affordances from attributes, [6] measure similarity between query and training image by the location of object parts, and [27] train a deep model on RGB data. RGB-D data is exploited by [21] who propose a two stage cascade to model graspable regions and [26] who train a CNN to predict depth information and affordances from RGB data simultaneously. CNNs were also used in [23] for a pixelwise affordance segmentation in RGB-D data and in [20] to predict grasps. [8, 12, 18] exploit human poses to localize object affordances.

Weakly supervised semantic image segmentation faced rapid progress in recent time. [25] use an expectation-maximization (EM) approach with area constraints to train a CNN. While [1] use keypoint annotations and incorporate objectness into their loss function, [16] exploit localization cues from an image level classifier, area constrains and CRFs. [10] rely on superpixels and [24] learn a model from image labels and saliency predictions. In [7], an approach based on pooling of classwise heat maps along with image labels was proposed. A weakly supervised affordance segmentation approach based on EM similar to [25] was proposed in [27]. Our approach builds on this work.

3. Weakly Supervised Affordance Segmentation

Our approach for weakly supervised affordance segmentation extends the approach [27] by adaptive binarization and approximated cross validation for estimating hyperparameters. We therefore briefly describe [27] first and then describe in Section 3.2 the adaptive binarization and in Section 3.3 approximated cross validation.

3.1. Method

The approach [27] extends fully convolutional neural networks like [4] or [5] for the task of affordance segmentation. In contrast to semantic image segmentation, where only one label per pixel needs to be predicted, affordance segmentation requires to predict a set of labels per pixel since an object region might contain multiple affordance types. The approach predicts $P(Y|I;\theta)$ where I denotes the input image, θ denotes the parameters of the model, i.e. the weights of the neural network, and $Y = \{y_{i,l}\}$ with $y_{i,l} \in \{0,1\}$ is the pixelwise segmentation. If $y_{i,l} = 1$ the affordance type l is predicted for pixel i. Due to the multi-label problem, the network uses a sigmoid layer instead of a softmax layer [4, 5]:

$$P(y_{i,l} = 1|I;\theta) = \frac{1}{1 + \exp\left(-g_{i,l}\left(y_{i,l}|I;\theta\right)\right)}, \quad (1)$$

where $g_{i,l}$ is the value of the previous layer of the neural network. For segmentation, the predicted probabilities $P(y_{i,l}|I;\theta)$ need to be binarized. In [27], this is achieved by the standard 50% threshold:

$$\hat{y}_{i,l} = \begin{cases} 1 & \text{if } P(y_{i,l} = 1|I;\theta) \geq 0.5 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The model parameters θ are determined during training. In the strongly supervised setting, training means optimizing the log-likelihood:

$$J(\theta) = \log P(Y|I;\theta) = \sum_{i=1}^{n} \sum_{l \in \mathcal{L}} \log P(y_{i,l}|I;\theta). \quad (3)$$

In the weakly supervised setting, the log-likelihood can not be calculated since Y is not given during training. In [27], it was proposed to train the model only from a set of keypoints $Z = \{(l_k, i_k)\}$, which denote the presence of the affordance l_k at pixel i_k, using expectation-maximization (EM). During training, both Y and θ need to be estimated from Z. The approach starts with an initial estimate $\hat{Y}$, which is derived from the keypoints Z by labeling all pixels within a radius of σ around a keypoint:

$$\hat{y}_{i,l} = \begin{cases} 1 & \text{if } |\{(l_k, i_k) \in Z : l_k = l \wedge |i_k - i| \leq \sigma\}| > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

In contrast to [27] that uses fixed values for initialization, we discuss in Section 3.3 how σ can be estimated by approximated cross validation.

After $\hat{Y}$ is estimated, the weights of the network θ are updated by optimizing $J(\theta) = \log P(\hat{Y}|I;\theta)$. Given the new weights θ, the CNN predicts $P(y_{i,l}|I;\theta)$ for each training image and $\hat{Y}$ is refined by binarization of the CNN predictions. The 50% threshold used in (2), however, is only valid for the fully supervised setting. While in [27] an additional GrabCut step is used to address this issue, we propose an adaptive approach that determines the threshold for

binarization for each training image and affordance class. This not only increases the accuracy, but it also reduces the training time since an additional GrabCut step is not needed anymore by our approach. The approach for adaptive binarization is discussed in Section 3.2. Our weakly supervised approach for affordance segmentation is illustrated in Figure 1.

To reduce overfitting and perform approximated cross validation as described in Section 3.3, we split the training set into three equally sized subsets A, B, and C. During the M-step, we train the convolutional network on each of the tuples (A,B), (B,C), and (C,A). During the E-step, each network predicts $P(y_{i,l}|I;\theta)$ for the set that was not used for training. As in [27], we use two EM iterations to obtain $\hat{Y}$ for all training images. The final CNN model is then obtained by optimizing $J(\theta) = \log P(\hat{Y}|I;\theta)$ on the entire training set.

3.2. Adaptive Binarization

We first want to explain why the binarization as described in Equation 2 is not optimal for the weakly supervised case. Let us first consider an optimal classifier that separates two classes perfectly in the training data. In this case, $P(y_{i,l} = 1|I;\theta) \geq 0.5$ if a pixel is annotated by $y_{i,l} = 1$ and $P(y_{i,l} = 1|I;\theta) < 0.5$ if it is annotated by $y_{i,l} = 0$. Hence, using 50% as threshold for binarization is optimal. For weakly supervised learning, $\hat{Y}$ is in particular after the initialization only a poor estimation of the unknown ground truth segmentation masks Y of the training data such that $\hat{y}_{i,l} \neq y_{i,l}$ for many pixels. This means that the optimal threshold is unknown. However, we can use the keypoints Z to obtain an estimate of the threshold:

$$\hat{y}_{i,l} = \begin{cases} 1 & \text{if } P(y_{i,l} = 1|I;\theta) \geq t \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where

$$t = \min\left\{0.5, f\left(\{P(y_{i_k,l} = 1|I;\theta)\}_{l_k=l}\right)\right\}. \tag{6}$$

$\{P(y_{i_k,l} = 1|I;\theta)\}_{l_k=l}$ are the predictions of the classifier for all keypoints in the training image I with label l and f computes either the mean or median of $\{P(y_{i_k,l} = 1|I;\theta)\}_{l_k=l}$. In our default experimental setting, we will have only one keypoint for each affordance occurring in an image. In general, one can expect that the threshold is below 0.5 since the ratio $\frac{|\{i:y_{i,l}=0 \wedge \hat{y}_{i,l}=1\}|}{|\{i:y_{i,l}=0\}|}$ is usually lower than $\frac{|\{i:y_{i,l}=1 \wedge \hat{y}_{i,l}=0\}|}{|\{i:y_{i,l}=1\}|}$. As soon as the threshold reaches 0.5, we can replace the adaptive threshold by 0.5. We therefore limit the threshold by 0.5.

3.3. Approximated Cross Validation

In the fully supervised setup, hyperparameters can be optimized by cross-validation on the training set using the same measure that is also used for evaluation. Since the ground truth masks Y, however, are unknown in the weakly supervised setup, exact cross validation is not possible. We therefore propose to approximate the Jaccard index, which measures the intersection over union between the ground-truth Y and the prediction $\hat{Y}$, on the validation set. Since the Jaccard index is computed per affordance class l and then averaged over all classes, we discuss only the binary case with $y_i \in \{0,1\}$. Let $P(y_i = 1) = \frac{|\{i:y_i=1\}|}{|\{i\}|}$ be the unknown percentage of pixels with $y_i = 1$ and $P(\hat{y}_i = 1) = \frac{|\{i:\hat{y}_i=1\}|}{|\{i\}|}$ the known percentage of pixels that have been classified with $\hat{y}_i = 1$. We can approximate $P(\hat{y}_i = 1|y_i = 1)$ by measuring how often a keypoint annotated by the affordance class has been correctly classified. Similarly, $P(\hat{y}_i = 1|y_i = 0)$ is given by the percentage of keypoints that have been misclassified. This gives the relation

$$\begin{aligned} P(\hat{y}_i = 1) =& P(\hat{y}_i = 1|y_i = 1)P(y_i = 1) \\ &+ P(\hat{y}_i = 1|y_i = 0)(1 - P(y_i = 1)) \end{aligned} \tag{7}$$

and thus

$$P(y_i = 1) = \frac{P(\hat{y}_i = 1) - P(\hat{y}_i = 1|y_i = 0)}{P(\hat{y}_i = 1|y_i = 1) - P(\hat{y}_i = 1|y_i = 0)}. \tag{8}$$

The Jaccard index which is

$$J = \frac{|\{i : y_i = 1 \wedge \hat{y}_i = 1\}|}{|\{i : y_i = 1\}| + |\{i : y_i = 0 \wedge \hat{y}_i = 1\}|} \tag{9}$$

can then be approximated by

$$J_{approx} = \frac{P(\hat{y}_i = 1|y_i = 1)P(y_i = 1)}{P(y_i = 1) + P(\hat{y}_i = 1|y_i = 0)(1 - P(y_i = 1))}. \tag{10}$$

As mentioned in Section 3.1, we split the training set into three subsets for approximate cross-validation.

4. Experiments

For evaluation, we use the CAD 120 affordance dataset [27] and the UMD part affordance dataset [22]. We use the splits separating the object classes (novel split on UMD and object split on CAD) and the splits which do not separate the object classes (category split on UMD and actor split on CAD). As measure, we use the Jaccard index. We report the results using the VGG architecture [4] and the ResNet architecture [5] as underlying convolutional network. First, we conduct ablation experiments to show the impact of our two key components, adaptive binarization and approximated cross validation. Second, we compare our approach with other weakly supervised segmentation

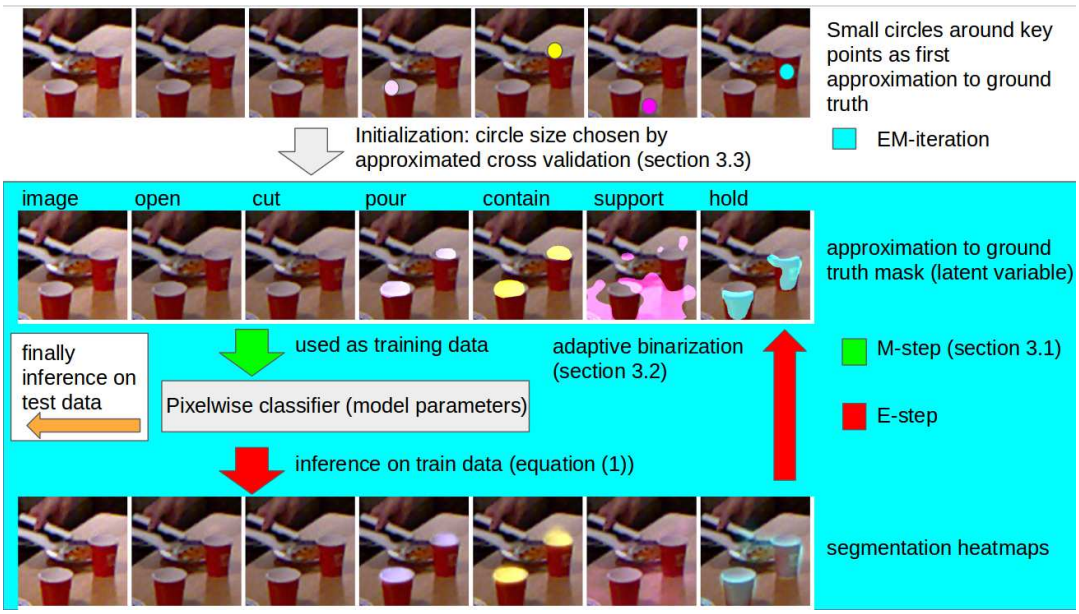

Figure 1: Illustration of our approach for affordance segmentation using keypoints as weak supervision. The CNN is trained by iteratively updating the segmentation masks for the training images (E-step) and the parameters of the network (M-step).

CAD 120	Background	Open	Cut	Contain	Pour	Support	Hold	Mean
non-adaptive (VGG)	0.62	0.09	0.20	0.41	0.35	0.11	0.40	0.31
adaptive (VGG)	0.68	0.10	0.23	0.44	0.36	0.50	0.47	0.40

UMD	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
non-adaptive (VGG)	0.32	0.12	0.48	0.46	0.08	0.33	0.69	0.36
adaptive (VGG)	0.31	0.18	0.56	0.49	0.08	0.41	0.66	0.38

Table 1: Comparison of adaptive binarization with non-adaptive binarization. The Jaccard index is reported for the object split of CAD 120 affordance dataset and the novel split of the UMD part affordance dataset.

approaches. If not otherwise specified, we use our approach based on the VGG architecture with adaptive binarization and approximate cross validation to determine σ (4). As in [27], we use one keypoint per affordance class and training image. In Section 4.3, we also evaluate the impact of the number of keypoints.

4.1. Adaptive Binarization

First we evaluate the impact of adapting the binarization to each training image and affordance class in comparison to using a constant threshold for each affordance class. To this end, instead of using $\min\{0.5, f(\{P(y_{i_k,l}=1|I;\theta)\}_{l_k=l})\}$ as an individual threshold for each image I, we take the average of these thresholds over all images in the training set labeled with the affordance class l. Note that $f(\{P(y_{i_k,l}=1|I;\theta)\}_{l_k=l}) = P(y_{i_k,l}=1|I;\theta)$ in this experiment since we use only one keypoint per affordance l and image I.

The results for the object split of the CAD 120 affordance dataset and the novel split of the UMD part affordance dataset are shown in Table 1. Compared to the proposed adaptive binarization approach, the accuracy decreases for all affordance classes and the background, which are regions annotated without any affordance class. In average, the accuracy decreases by -9%. On UMD, the decrease is smaller but still -2%. The effect on CAD is larger since the size of the affordance regions varies more across the training images in comparison to UMD.

As discussed in Section 3.2, we limit the adaptive threshold by 0.5, which is the optimal threshold for a fully supervised trained model. Table 2 shows the results when the threshold is not limited, i.e., the adaptive threshold can even get close to one. As expected, the accuracy drops for both datasets by -9% since a threshold above 0.5 would produce even in the fully supervised case too small affordance

CAD 120	Background	Open	Cut	Contain	Pour	Support	Hold	Mean
Max thres. 1.0 (VGG)	0.62	0.08	0.21	0.34	0.33	0.39	0.19	0.31
Max thres. 0.5 (VGG)	0.68	0.10	0.23	0.44	0.36	0.50	0.47	0.40

UMD	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
Max thres. 1.0 (VGG)	0.32	0.04	0.36	0.42	0.05	0.23	0.64	0.29
Max thres. 0.5 (VGG)	0.31	0.18	0.56	0.49	0.08	0.41	0.66	0.38

Table 2: Impact of limiting the adaptive threshold (5) by 0.5. The Jaccard index is reported for the object split of the CAD 120 affordance dataset and the novel split of the UMD part affordance dataset.

segments.

4.2. Approximated Cross Validation

The initialization depends on the value σ, which determines the initial affordance segments around the keypoints (4). This is shown in the last column of Table 3 where we report the mean Jaccard index for three values of σ. Note that σ is set proportional to the image width w. The results show that the accuracy strongly depends on the initialization. The strongest variation can be observed for the category split of the UMD part affordance dataset where the accuracy varies between 0.44 to 0.61. The approximated Jaccard index computed from the keypoints in the training set, which is reported in the second column of Table 3, however, correlates with the Jaccard index on the test set. This shows that using approximated cross validation to determine σ works very well in practice. Note that the values between the Jaccard index and its approximation differ since the first measure is computed over the test set and the second over the training set. In all experiments except of Table 3, we have determined σ by approximated cross validation.

4.3. Varying Number of Keypoints

Our approach also works with multiple keypoints per affordance class in an image. In this case, we compare two functions for $f\left(\{P(y_{i_k,l}=1|I;\theta)\}_{l_k=l}\right)$ (5), namely taking the average or the median of $\{P(y_{i_k,l}=1|I;\theta)\}_{l_k=l}$. The results are reported in Figure 2. For the object split of the CAD 120 affordance dataset, average and median perform similar and the accuracy increases only slightly after three keypoints. A similar behavior can be observed for the novel split of the UMD part affordance dataset, but here average performs better than the median.

4.4. Comparison to the State-of-the-art

We finally compare our approach with other weakly supervised semantic segmentation approaches [16, 1, 25, 27]. The results for both splits on the CAD 120 affordance dataset are reported in Table 4, while the results for the UMD part affordance dataset are reported in Table 5. The methods [16, 25] use only image labels and therefore weaker supervision. It is therefore expected that methods that use more supervision in form of keypoints achieve a higher accuracy. For the methods [1, 27] and our approach, we use one keypoint for each affordance class in an image. The parameter σ has been determined by approximated cross validation. We also report the results as in [27] for the VGG architecture and the ResNet architecture. Our approach outperforms [27] and the other methods on both datasets. While our approach achieves with the ResNet architecture on all datasets and splits a better mean accuracy than VGG, this is not the case for [27] where VGG is sometimes better. For the actor split of the CAD 120 affordance dataset, the mean accuracy is improved by $+17\%$ compared to [27]. This shows the benefit of adaptive binarization for weakly supervised affordance segmentation. Qualitative results are shown in Figure 3.

5. Conclusion

In this work, we have proposed an approach for affordance segmentation that requires only weak supervision in the form of sparse keypoints. Our approach builds on the method [27], but it does not require an additional graph cut segmentation step. This has been achieved by an adaptive approach for binarizing the predictions of a convolutional neural network during training. By approximating the Jaccard index based on the keypoints, we are also able to optimize parameters for the initialization. This approach could also be used to optimize other hyperparameters. We evaluated our approach on the CAD 120 affordance and the UMD part affordance dataset. Our approach outperforms the state-of-the-art for weakly supervised affordance segmentation. On the CAD 120 affordance dataset, the mean accuracy is increased by up to 17 percentage points compared to [27].

Acknowledgments. The work has been financially supported by the DFG projects GA 1927/5-1 (DFG Research Unit FOR 2535 Anticipating Human Behavior) and GA 1927/2-2 (DFG Research Unit FOR 1505 Mapping on Demand).

	approx. Jaccard train			Jaccard test		
σ relative to image width	$0.03w$	$0.06w$	$0.12w$	$0.03w$	$0.06w$	$0.12w$
CAD actor split	0.38	**0.40**	0.30	0.41	**0.42**	0.37
CAD object split	0.48	**0.50**	0.39	0.38	**0.40**	0.35
UMD category split	0.57	**0.58**	0.44	**0.61**	0.59	0.44
UMD novel split	**0.66**	0.62	0.44	**0.38**	**0.38**	0.35

Table 3: Impact of σ (4). The second column contains the approximated Jaccard index (10) computed on the training data for three values of σ. The approximated Jaccard index is used to determine σ. The third column contains the Jaccard index computed on the test data for three values of σ.

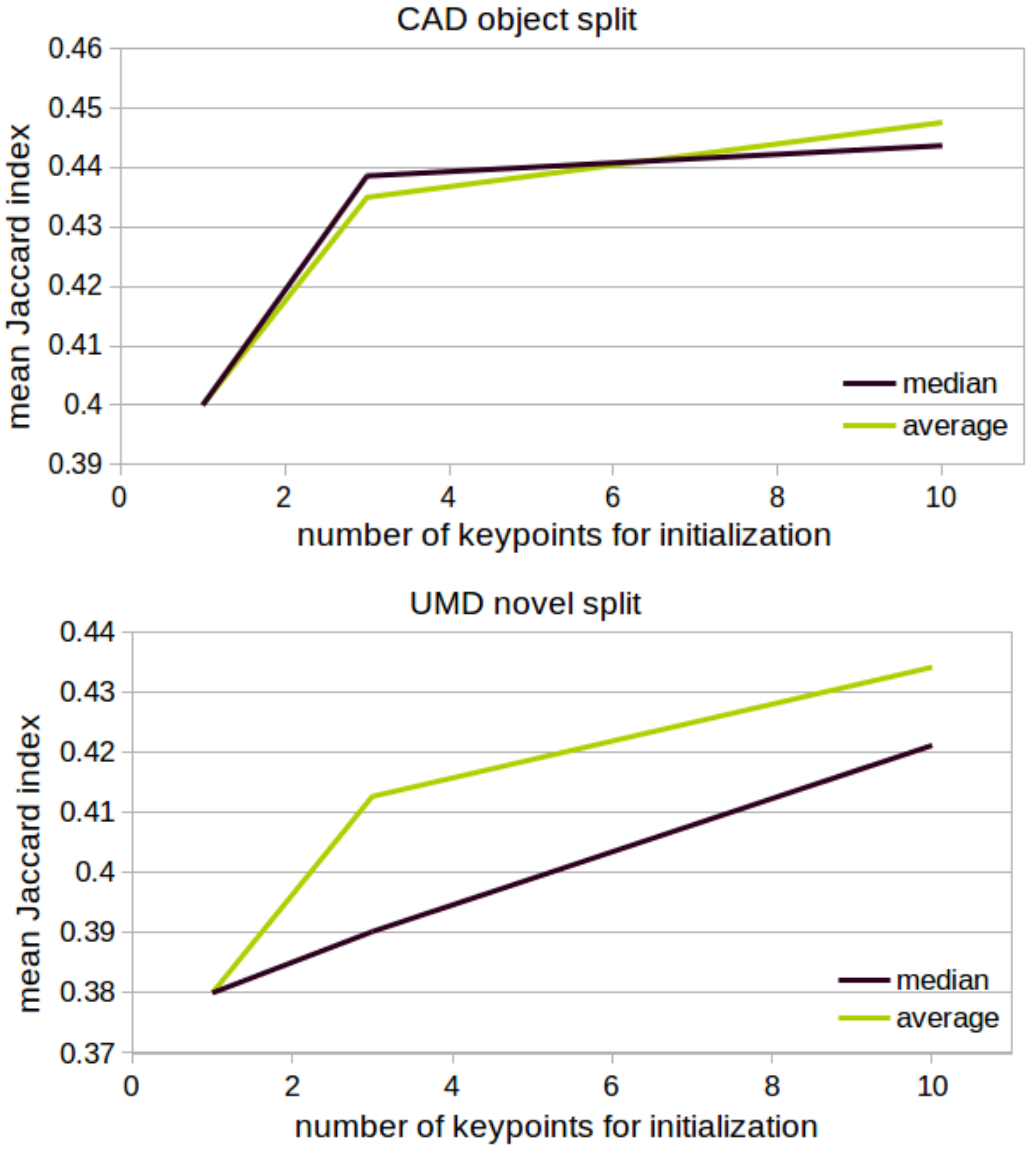

Figure 2: Affordance segmentation with more than one keypoint per image and affordance. For the function f (5), we compare average and median. The mean Jaccard index is plotted over the number of keypoints.

CAD 120	Background	Open	Cut	Contain	Pour	Support	Hold	Mean
image label supervision - actor split								
Area constraints [25]	0.53	0.11	0.02	0.09	0.09	0.07	0.15	0.15
SEC [16]	0.53	0.43	0.00	0.25	0.09	0.02	0.20	0.22
keypoint supervision - actor split								
WTP [1]	0.53	0.13	0.00	0.10	0.08	0.11	0.22	0.17
[27] (VGG)	0.61	0.33	0.0	0.35	0.30	0.22	0.43	0.32
Proposed (VGG)	0.71	0.47	0.0	0.36	0.37	0.56	0.49	0.42
[27] (ResNet)	0.60	0.25	0.00	0.35	0.30	0.17	0.42	0.30
Proposed (ResNet)	0.77	0.50	0.00	0.43	0.39	0.64	0.56	0.47
image label supervision - object split								
Area constraints [25]	0.59	0.03	0.03	0.01	0.02	0.02	0.28	0.14
SEC [16]	0.54	0.04	0.09	0.13	0.09	0.08	0.13	0.16
keypoint supervision - object split								
WTP [1]	0.57	0.01	0.00	0.02	0.09	0.03	0.19	0.13
[27] (VGG)	0.62	0.08	0.08	0.24	0.22	0.20	0.46	0.27
Proposed (VGG)	0.68	0.10	0.23	0.44	0.36	0.50	0.47	0.40
[27] (ResNet)	0.69	0.11	0.09	0.28	0.21	0.36	0.56	0.33
Proposed (ResNet)	0.74	0.15	0.21	0.45	0.37	0.61	0.54	0.44

Table 4: Comparison of our method to the state-of-the-art on the CAD 120 affordance dataset. The Jaccard index is reported.

UMD	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
image label supervision - category split								
Area constraints [25]	0.06	0.04	0.10	0.14	0.22	0.04	0.37	0.14
SEC [16]	0.39	0.16	0.27	0.13	0.35	0.19	0.07	0.22
keypoint supervision - category split								
WTP [1]	0.16	0.14	0.20	0.20	0.01	0.07	0.13	0.13
[27] (VGG)	0.46	0.48	0.72	0.78	0.44	0.53	0.65	0.58
Proposed (VGG)	0.55	0.48	0.72	0.76	0.49	0.48	0.67	0.59
[27] (ResNet)	0.42	0.35	0.67	0.70	0.44	0.44	0.77	0.54
Proposed (ResNet)	0.57	0.54	0.71	0.70	0.43	0.54	0.69	0.60
image label supervision - novel split								
Area constraints [25]	0.05	0.00	0.04	0.16	0.00	0.01	0.32	0.09
SEC [16]	0.12	0.03	0.06	0.23	0.07	0.12	0.25	0.13
keypoint supervision - novel split								
WTP [1]	0.11	0.03	0.18	0.11	0.00	0.02	0.23	0.10
[27] (VGG)	0.27	0.14	0.55	0.58	0.02	0.37	0.67	0.37
Proposed (VGG)	0.31	0.18	0.56	0.49	0.08	0.41	0.66	0.38
[27] (ResNet)	0.25	0.21	0.62	0.50	0.08	0.43	0.67	0.40
Proposed (ResNet)	0.34	0.34	0.58	0.40	0.07	0.42	0.77	0.42

Table 5: Comparison of our method to the state-of-the-art on the UMD part affordance dataset. The Jaccard index is reported.

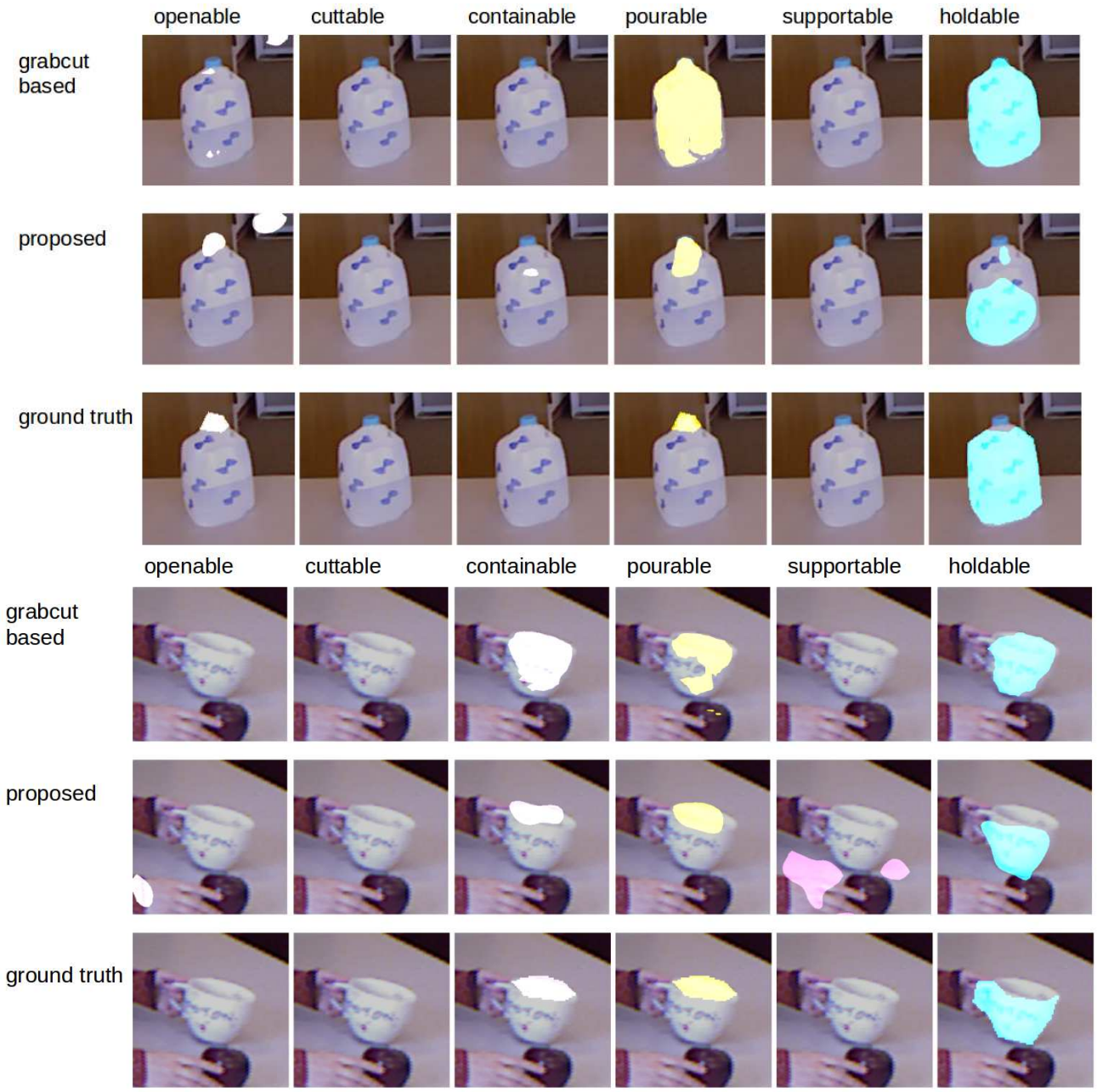

Figure 3: Qualitative comparison of our approach (second and fifth row) with [27] (first and fourth row). Our approach localizes even small affordance parts while the GrabCut step in [27] merges the cap with the entire object.

References

[1] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the Point: Semantic Segmentation with Point Supervision. *ECCV*, 2016.

[2] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and B. Caputo. Using object affordances to improve object recognition. *Autonomous Mental Development*, 3(3):207–215, 2011.

[3] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng. Mining semantic affordances of visual object categories. In *CVPR*, pages 4259–4267, 2015.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy,

and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *arXiv:1506.02106v5*, 2016.

[6] C. Desai and D. Ramanan. Predicting functional regions on objects. In *CVPR Workshops*, pages 968–975, 2013.

[7] T. Durand, T. Mordan, N. Thome, and M. Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. *CVPR*, 2017.

[8] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *CVPR*, pages 1529–1536, 2011.

[9] T. Hermans, J. M. Rehg, and A. Bobick. Affordance prediction via learned object attributes. In *ICRA: Workshop on Semantic Perception, Mapping, and Exploration*, 2011.

[10] Q. Hou, P. K. Dokania, D. Massiceti, Y. Wei, M.-M. Cheng, and P. H. S. Torr. Mining pixels: Weakly supervised semantic segmentation using image labels. *arXiv:1612.02101v2*, 2016.

[11] R. Jafri, S. Ali, H. Arabnia, and F. Shameem. Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. *The Visual Computer*, pages 1197–1222, 2014.

[12] Y. Jiang, H. Koppula, and A. Saxena. Hallucinated humans as the hidden context for labeling 3d scenes. In *CVPR*, pages 2993–3000, 2013.

[13] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz. Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. *Autonomous Robots*, 37(4):369–382, 2014.

[14] D. I. Kim and G. Sukhatme. Semantic labeling of 3d point clouds with object affordance for robot manipulation. In *ICRA*, pages 5578–5584, 2014.

[15] H. Kjellström, J. Romero, and D. Kragić. Visual object-action recognition: Inferring object affordances from human demonstration. *CVIU*, 115(1):81–90, 2011.

[16] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, pages 695–711, 2016.

[17] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. *IJRR*, 32(8):951–970, 2013.

[18] H. S. Koppula and A. Saxena. Physically grounded spatio-temporal object affordances. In *ECCV*, pages 831–847. 2014.

[19] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *PAMI*, 38(1):14–29, 2016.

[20] S. Kumra and C. Kanan. Robotic grasp detection using deep convolutional neural networks. *arXiv:1611.08036v3*, 2017.

[21] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *IJRR*, 34(4-5):705–724, 2015.

[22] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *ICRA*, pages 1374–1381, 2015.

[23] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. Tsagarakis. Detecting Object Affordances with Convolutional Neural Networks. *IROS*, 2016.

[24] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele. Exploiting saliency for object segmentation from image level labels. *arXiv:1701.08261v1*, 2016.

[25] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, pages 1742–1750, 2015.

[26] A. Roy and S. Todorovic. A multi-scale CNN for affordance segmentation in RGB images. In *ECCV*, pages 186–201, 2016.

[27] J. Sawatzky, A. Srikantha, and J. Gall. Weakly supervised affordance detection. *CVPR*, 2017.

[28] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *ECCV*, pages 408–424. 2014.

[29] Y. Zhu, Y. Zhao, and S. Chun Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *CVPR*, pages 2855–2864, 2015.