Weakly Supervised Affordance Detection

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

Abhilash Srikantha*
Carl Zeiss AG
abhilash.srikantha@zeiss.com

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

Abstract

Localizing functional regions of objects or affordances is an important aspect of scene understanding and relevant for many robotics applications. In this work, we introduce a pixel-wise annotated affordance dataset of 3090 images containing 9916 object instances. Since parts of an object can have multiple affordances, we address this by a convolutional neural network for multilabel affordance segmentation. We also propose an approach to train the network from very few keypoint annotations. Our approach achieves a higher affordance detection accuracy than other weakly supervised methods that also rely on keypoint annotations or image annotations as weak supervision.

1. Introduction

The capability to perceive functional aspects of an environment is highly desired because it forms the essence of devices intended for collaborative use. These aspects can be categorized into abstract descriptive properties called *attributes* [28, 25, 30] or physically grounded regions called *affordances*. Affordances are important as they form the key representation to describe potential interactions. For instance, autonomous navigation depends heavily on understanding outdoor semantics to decide if the lane is *changeable* or if the way ahead is *drivable* [5]. Similarly, assistive robots must have the capability of anticipating indoor semantics like which regions of the kitchen are *openable* or *placeable* [20]. Further, because forms of interaction are given for virtually any object class, it is desirable to have recognition systems that are capable of localizing functionally meaningful regions or *affordances* alongside contemporary object recognition systems.

Existing methods [15, 17, 26, 13, 32] learn to infer pixel-wise affordance labels using supervised learning techniques. Since creating pixel wise annotated datasets is heavily labor intensive, recent works focus mainly on coarse affordance classes like *walkable* or *reachable* which are at a scene level but not at an object level [32]. An exception is the UMD part affordance dataset [26], which provides annotations for objects. In order to simplify the annotation process, a turntable setting has been used to capture the objects. This setup, however, simplifies the task since each image contains only one object that is easy to segment. We therefore propose a more challenging dataset containing images captured in a kitchen environment. The dataset consists of 3090 images containing 9916 object instances. As in [26], each pixel is annotated by none or several affordance classes. This is different to other semantic segmentation tasks where a pixel is usually labeled by only one semantic class. To address this, we extend a convolutional neural network (CNN) architecture for segmentation from singlelabel to multilabel classification.

Since CNNs require large amounts of annotated data, it is desirable to train them in a weakly supervised setting. In [2], supervision in form of keypoints has been proposed. Instead of providing segmentation masks, only a very small set of pixels in an image are annotated. We therefore propose an approach for affordance detection that can be learned by such keypoint annotations. In our experiments, we show that our approach outperforms [2] for affordance detection by a large margin. Our approach also achieves a higher affordance detection accuracy than other state-of-the-art methods that utilize weaker supervision at image level [27, 19].

2. Related Work

Properties of objects can be described at various levels of abstraction by a variety of attributes including visual properties [28, 16, 10, 23], *e.g.* object color, shape and object parts, physical properties [11, 41], *e.g.* weight, size and material characteristics, and categorical properties [1, 8]. Object affordances, which describe potential uses of an object, can also be considered as other attributes. For instance, [4] describes affordances by object-action pairs whose plausibility is determined either by mining word co-occurrences in textual data or by measuring visual consistency in images returned by an image search. [41] proposes to represent objects in a densely connected graph structure. While a node

*contributed equally

represents one of the various visual, categorical, physical or functional aspects of the object, an edge indicates the plausibility of both node entities to occur jointly. Upon querying the graph with observed information, *e.g.* {*round, red*}, the result is a set of most likely nodes, *e.g.* {*tomato, edible, 10-100gm, pizza*}.

Affordances have also been used as an intermediate representation for higher level tasks. In [3], object functionality is defined in terms of the poses of relevant hand-grasps during interaction. Object recognition is performed by combining individual classifiers based on object appearance and hand pose. [42] uses affordances as a part of a task oriented object modeling. They formulate a generative framework that encapsulates the underlying physics, functions and causality of objects being used as tools. Their representation combines extrinsic factors that include human pose sequences and physical forces such as velocity and pressure and intrinsic factors that represent object part affordances. [22] models action segments using CRFs which are described by human pose, object affordance and their appearances. Using a particle filter framework, future actions are anticipated by sampling from a pool of possible CRFs thereby performing a temporal segmentation of action labels and object affordances. [18] jointly models object appearance and hand pose during interactions. They demonstrate simultaneous hand action localization and object detection by implicitly modeling affordances.

Localizing object affordances based on supervised learning has been addressed in particular in the context of robotics applications. [15] performs robotic manipulations on objects based on affordances which are inferred from the orientations of object surfaces. [17] learns a discriminative model to perform affordance segmentation of point clouds based on surface geometry. [26] uses RGB-D data to learn pixelwise labeling of affordances for common household objects. They explore two different features: one based on a hierarchical matching pursuit and another based on normal and curvature features derived from RGB-D data. [13] learns to infer object level affordance labels based on attributes derived from appearance features. [24] proposes a two stage cascade approach based on RGB-D data to regress potential grasp locations of objects. In [9], pixelwise affordance labels of objects are obtained by warping the query image to the K-nearest training images based on part locations inferred using deformable part models. [35] combines top-down object pose based affordance labels with those obtained from bottom-up appearance based features to infer part-based object affordances. Top-down approaches for affordance labeling have been explored in [12, 14] where scene labeling is performed by observing possible interactions between scene geometry and hallucinated human poses. Localizing object affordances based on human context has been also studied in [21]. [32] uses CNNs to estimate a depth map and surface normals for a scene and a single-label CNN for semantic segmentation. The feature maps are then merged to predict affordances maps.

Figure 1. Example images from (top row) the UMD part affordance dataset and (bottom row) the CAD120 dataset.

Weakly-supervised learning for semantic image segmentation has been investigated in several works. In this context, training images are only annotated at the image-level and not at pixel-level. For instance, [36] formulates the weakly supervised segmentation task as a multiple instance and multitask learning problem. Further, [37, 38] incorporate latent correlations among superpixels that share the same labels but originate from different images. [39] simplifies the above formulation by a graphical model that simultaneously encodes semantic labels of superpixels and presence or absence of labels in images. [40] handles noisy labels from social images by using robust mid-level representations derived through topic modeling in a CRF framework. Recently, several weakly supervised approaches have been proposed for weakly supervised learning of image segmentations. An approach based on a CNN has been proposed in [27]. It uses an expectation-maximization framework to iteratively learn the latent pixel labels of the training data and the parameters of the CNN. A similar approach is followed by [29] where linear constraints derived from weak image labels are imposed on the label prediction distribution of the CNN. [19] proposes to use class wise regions of interest obtained by an image classification CNN and conditional random fields for segmentation. [33] follows a similar approach, but here only a class agnostic foreground mask is calculated. Closest to our work is the single-label approach [2], which uses an objectness prior, since it also utilizes keypoint annotations for weakly supervised learning.

3. Affordance Datasets

There are not many datasets with pixelwise affordance labels. Recently, the NYUv2 RGB-D dataset has been augmented with coarse affordance labels like *walkable* and *movable* for entire rooms instead of objects [32]. In

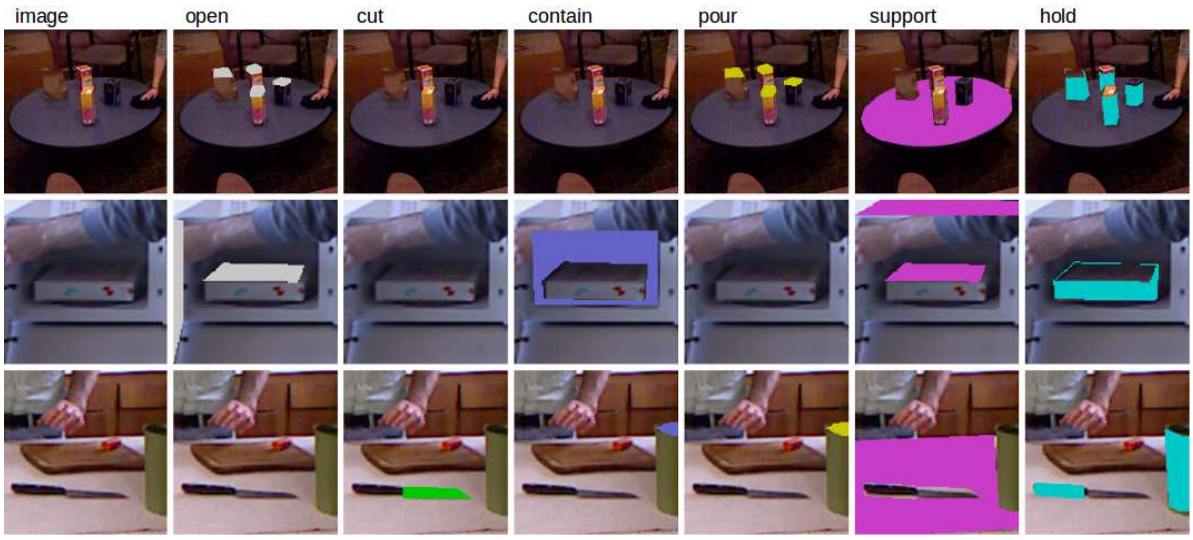


Figure 2. Example images with annotations from the proposed CAD120 affordance dataset. Pixels that do not belong to any affordance are considered as background. *Best viewed in color.*

contrast, the publicly available RGB-D dataset proposed by [26] focuses on part affordances of everyday tools. The dataset consisting of 28,074 images is collected using a Kinect sensor, which records RGB and depth images at a resolution of 640×480 pixels and provides 7-class pixel-wise affordance labels for objects from 17 categories. Each pixel may belong to multiple affordances at the same time. Each object is recorded on a revolving turntable to cover a full $360°$ range of views providing clutter-free images of the object as shown in Fig. 1. While such lab recordings provide images with high quality, they lack important contextual information such as human-interaction, other objects and typical background.

We therefore adopt a dataset that contains objects within the context of human-interactions in a more realistic environment. We found the CAD120 dataset [21] to be well tailored for our purpose. It consists of 215 videos in which 8 actors perform 14 different high-level activities. Each high-level activity is composed of sub-activities, which in turn involve one or more objects. In total, there are 32 different sub-activities and 35 object classes. A few images of the dataset are shown in Fig. 1. The dataset also provides frame wise annotation of the sub-activity, object bounding boxes and automatically extracted human pose.

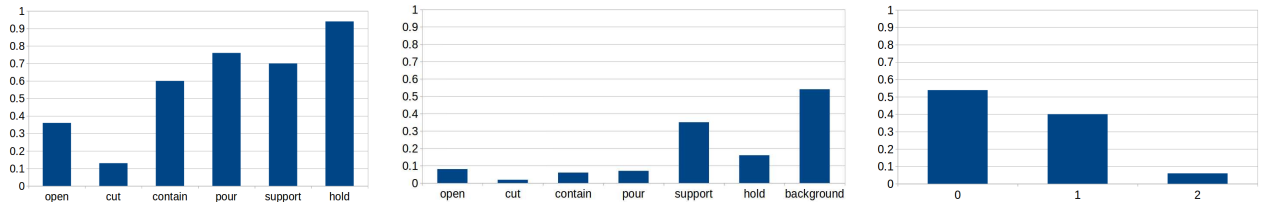
We annotate the affordance labels *openable, cuttable, containable, pourable, supportable, holdable* for every 10^{th} frame from sequences involving an active human-object interaction resulting in 3090 frames. Each frame contains between 1 and 12 object instances resulting in 9916 objects in total. We annotate all object instances with pixelwise affordance labels. Since the object bounding boxes in the dataset are annotated, we perform all experiments on cropped images after extending the bounding boxes by 30 pixels if possible in each direction. A few annotated cropped images from the dataset are shown in Fig. 2. As can be seen, the appearance of affordances can vary significantly, *e.g.* visually distinct object parts like the lid of a box or the door of a microwave have the affordance label *openable*. Similarly, the knife handle and the boxes are *holdable*.

We report some statistics regarding the annotations with respect to the cropped images in Fig. 3. Fig. 3(a) shows that the generic affordance classes *supportable* and *holdable* occur frequently. The classes *pourable* and *containable* also occur quite often due to the kitchen environment. The class *cuttable* occurs rarely. Except of the background class and *supportable*, the classes cover only a small portion of a cropped image when they are present as shown in Fig. 3(b). Fig. 3(c) shows that most of the pixels are labeled as background, *i.e.* they are not labeled by any affordance class, but there are also many pixels labeled by two classes. The dataset is well balanced in terms of the number of images contributed by each actor with a median of 382 and a range of 227–606 images per actor. The dataset is publicly available.[1]

4. Proposed Method

For semantic image segmentation, CNNs have shown very good results [6, 7]. For our experiments, we use the VGG-16 architecture as in [6] and the ResNet-101 as in [7]. In contrast to [6, 7], we do not use an additional CRF. Since the models [6, 7] do not handle the multilabel case, we have

[1] https://github.com/ykztawas/Weakly-Supervised-Affordance-Detection



(a) Probability of an affordance class to occur in an image

(b) Image area occupied by an affordance class given its presence in an image

(c) Percentage of pixels that are annotated by 0, 1 or 2 affordance classes

Figure 3. Statistics of the dataset

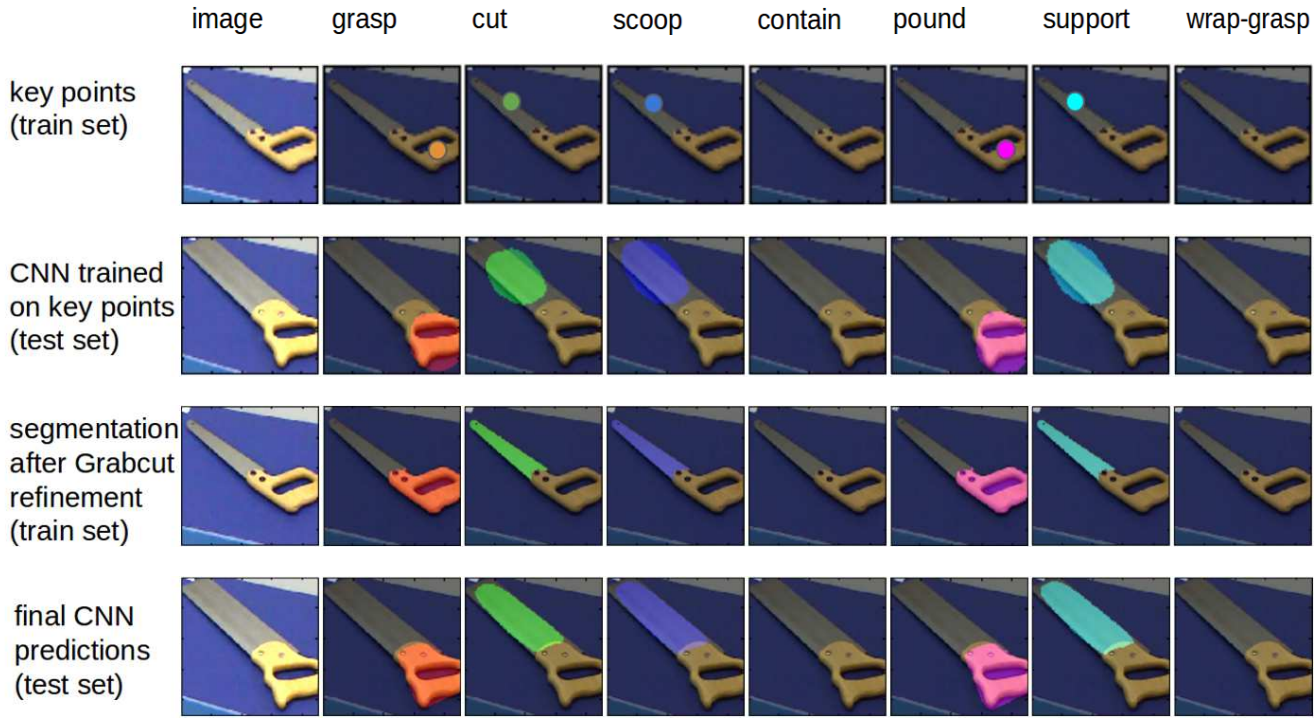


Figure 4. Illustration of our approach for weakly supervised affordance detection. The example images are taken from the UMD part affordance dataset [26]. The first row shows the weak annotations of the training images. The saw is annotated by five keypoints with affordance labels. The second row shows the prediction of the CNN on an image of the test set. If the CNN is only trained on the keypoint annotations, the predictions are not very precise. The third row shows the estimated annotation for the training image after the prediction of the CNN was refined by Grabcut for each affordance class. The last row shows the prediction of the CNN trained on the refined annotations of the training set. Compared to the second row, the affordances are precisely detected. *Best viewed in color.*

to modify the architecture. We first describe the learning procedure in the fully supervised setting and then discuss the weakly supervised setting.

4.1. Full Supervision

Given an image I with n pixels, we denote the image pixels as $X = \{x_1, x_2, \ldots, x_n\}$ and the corresponding labeling as $Y = \{y_{i,l}\}$ where $y_{i,l} \in \{0, 1\}$ indicates if pixel x_i is labeled by affordance class l. We denote the set of affordances as $\mathcal{L}$.

In the fully supervised case, we train the CNN by optimizing the log likelihood given by

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

where θ are the parameters of the CNN. A common loss function for semantic image segmentation is the cross-entropy based on the output of a final softmax layer. This does not work in the multilabel case and we define the loss based on the sigmoid function:

$$P(y_{i,l}|I;\theta) = \frac{1}{1 + \exp\left(-f_{i,l}\left(y_{i,l}|I;\theta\right)\right)}, \quad (2)$$

where $f_{i,l}\,(y_{i,l}|I;\theta)$ is the output of the CNN at pixel x_i and affordance l without the softmax layer.

4.2. Weak Supervision

While all pixels are labeled in the fully supervised setting, we will have only very few pixels annotated in the weakly supervised setting. In our setup, weak supervision is provided in terms of keypoints as illustrated in the first row of Fig. 4. In this case, the observed variables are image data X and keypoints $Z_x = \{(z_1, x_1), (z_2, x_2), \dots\}$, where x_l is an annotated keypoint with label z_l, but the pixel level segmentations Y are latent variables. The concept of weakly supervised learning consists of estimating Y for the training images while learning the parameters of the CNN.

We use the available pre-trained models on ImageNet for VGG-16 and ResNet-101 as initialization for the CNNs and initialize $\hat{Y}$ by

$$\hat{y}_{i,l} = \begin{cases} 1 & \text{if } |\{(z_l, x_l) \in Z_x : z_l = l \wedge |x_l - x_i| \leq \sigma\}| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where Z_x are the available keypoint annotations. We chose $\sigma = 40$ for the UMD dataset and $\sigma = 50$ for the CAD dataset. In order to learn the parameters θ of the CNN we maximize

$$\max_{\theta} \sum_{i=1}^{n} \sum_{l \in \mathcal{L}} \log P(\hat{y}_{i,l}|I;\theta). \tag{4}$$

The predictions of the learned CNN are reasonable but not very precise as illustrated in the second row of Fig. 4. We therefore add an additional training stage.

After updating the parameters of the CNN, we recompute $P(Y|I;\theta)$ for the training images and compute the probability for the latent variable Y by

$$P(Y|I, Z_x) = \sum_{l \in \mathcal{L}} P(Y, l|I, Z_x) \tag{5}$$

$$= \sum_{l \in \mathcal{L}} P(Y|l, I, Z_x) P(l|I, Z_x) \tag{6}$$

$$\approx \sum_{l \in \mathcal{L}} P(Y_l|I;\theta) P(l|Z_x). \tag{7}$$

Since we know from Z_x if an affordance label l is present, we have $P(l|I, Z_x) = P(l|Z_x)$ and

$$P(l|Z_x) = \begin{cases} 1 & \text{if } |\{(z_l, x_l) \in Z_x : z_l = l\}| > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

In (7), $P(Y_l|I;\theta)$ denotes the probabilities which have been predicted by the CNN for the affordance class l. In order to obtain the final annotation Y for the training images we binarize the predictions by setting

$$\hat{y}_{i,l} = \begin{cases} 1 & \text{if } P(y_{i,l}|I, Z_x) > 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

While this could be already considered as the final estimate $\hat{Y}$ to update the CNN as described in (4), we use Grabcut [31] to refine the labels for each affordance l independently. To model for each affordance l the color distribution of the affordance region and the background region, we use Gaussian mixture models with 6 components. The distribution for the affordance regions is initialized by the pixels with $\hat{y}_{i,l} = 1$ and distribution for the background by the pixels with $\hat{y}_{i,l} = 0$. The refinement by Grabcut is illustrated in the third row of Fig. 4. The final row shows the improved results of the CNN trained on the training images refined by Grabcut.

5. Experiments

We first evaluate the fully supervised approach (Section 4.1) and compare it with other fully supervised approaches for affordance detection. We then compare the discussed weakly supervised setting (Section 4.2) with the fully supervised baseline and state-of-the-art weakly supervised image segmentation methods. For the UMD part affordance dataset, we use the two defined train-test splits for evaluation. For the first split, which is denoted by *category split*, the object classes are shared among the training and test set. In the second split, which is denoted by *novel split*, the object classes in the test set are not present in the training set. The second protocol is more difficult and measures how well the methods generalize across object classes. For the CAD120 affordance dataset, we also propose two splits. For the first split, which we denote by *actor split*, we reserve images from actors $\{5, 9\}$ as test set and use the images from actors $\{1, 6, 3, 7, 4, 8\}$ as training data. For the second split, which we denote by *object split*, the training set contains the object classes *table*, *plate*, *thermal cup*, *medicine box*, *microwave*, and *bowl* while the test set contains all other object classes.

In [26] a ranked weighted F-measure was proposed for measuring the accuracy for affordance detection. The measure takes into account that a pixel can have multiple labels, but assumes that the labels can be ranked. Ranking the labels is often not very intuitive. We therefore also report the accuracy using per class intersection-over-union (IoU), which is also known as Jaccard index, for both datasets.

5.1. UMD Part Affordance Dataset

5.1.1 Supervised Setting

In [26], two approaches have been presented for learning affordances from local appearance and geometric features. The first approach is based on features derived from a superpixel based hierarchical matching pursuit (HMP) together with a linear SVM and the second approach is based on curvature and normal features derived from depth data used within a structured random forest (SRF). We compare two

UMD dataset (category split)	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
Fully Supervised Ranked F-Measure category split								
HMP + SVM [26]	0.15	0.04	0.05	0.17	**0.04**	0.03	0.10	0.08
DEP + SRF [26]	0.13	0.03	0.10	0.14	0.03	0.04	0.09	0.08
Proposed (VGG)	0.23	**0.08**	**0.18**	**0.21**	**0.04**	0.08	**0.11**	0.13
Proposed (ResNet)	**0.24**	**0.08**	**0.18**	**0.21**	**0.04**	**0.09**	**0.11**	**0.14**
Fully Supervised IoU category split								
HMP + SVM [26]	0.57	0.37	0.70	0.77	0.41	0.49	0.79	0.59
DEP + SRF [26]	0.35	0.15	0.38	0.65	0.18	0.26	0.80	0.40
Proposed (VGG)	0.66	0.77	0.85	0.84	0.64	**0.73**	0.82	**0.76**
Proposed (ResNet)	**0.71**	**0.79**	**0.86**	**0.86**	**0.72**	0.55	**0.84**	**0.76**
Weakly Supervised IoU category split								
Proposed (VGG) without Grabcut (Train)	0.30	0.21	0.46	0.48	0.26	0.32	0.50	0.36
Proposed (VGG)	0.46	0.48	0.72	**0.78**	0.44	0.53	0.65	0.58
Proposed (VGG) + Grabcut (Test)	**0.57**	**0.68**	**0.73**	0.73	**0.60**	**0.66**	0.76	**0.67**
Proposed (ResNet) without Grabcut (Train)	0.29	0.21	0.47	0.50	0.28	0.33	0.50	0.37
Proposed (ResNet)	0.42	0.35	0.67	0.70	0.44	0.44	**0.77**	0.54
Proposed (ResNet) + Grabcut (Test)	0.52	0.56	0.72	0.72	0.51	0.64	0.76	0.63
Image label [27]	0.06	0.19	0.04	0.22	0.12	0.02	0.08	0.10
Area constraints [27]	0.06	0.04	0.10	0.14	0.22	0.04	0.37	0.14
SEC [19]	0.39	0.16	0.27	0.13	0.35	0.19	0.07	0.22
WTP [2]	0.16	0.14	0.20	0.20	0.01	0.07	0.13	0.13

Table 1. Evaluation of fully and weakly supervised approaches for affordance detection on the UMD part affordance dataset (category split). Evaluation metrics are weighted F-measure and IoU.

UMD dataset (novel split)	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
Fully Supervised Ranked F-Measure novel split								
HMP + SVM [26]	0.16	0.02	0.15	0.18	0.02	0.05	0.10	0.10
DEP + SRF [26]	0.05	0.01	0.04	0.07	0.02	0.01	0.07	0.04
Proposed (VGG)	**0.18**	**0.05**	**0.18**	**0.20**	**0.03**	**0.07**	**0.11**	**0.12**
Proposed (ResNet)	0.16	**0.05**	**0.18**	0.19	0.02	0.06	**0.11**	0.11
Fully Supervised IoU novel split								
HMP + SVM [26]	0.29	0.10	0.61	**0.74**	0.03	0.24	0.63	0.38
DEP + SRF [26]	0.32	0.04	0.23	0.42	**0.16**	0.22	0.81	0.31
Proposed (VGG)	**0.37**	0.35	0.65	0.62	0.10	**0.52**	**0.85**	**0.50**
Proposed (ResNet)	0.33	**0.51**	**0.69**	0.52	0.09	0.51	**0.85**	**0.50**
Weakly Supervised IoU novel split								
Proposed (VGG) without Grabcut (Train)	0.16	0.14	0.43	0.45	0.02	0.37	0.40	0.28
Proposed (VGG)	0.27	0.14	0.55	0.58	0.02	0.37	0.67	0.37
Proposed (VGG) + Grabcut (Test)	**0.34**	0.34	0.65	**0.70**	0.08	0.54	**0.73**	0.48
Proposed (ResNet) without Grabcut (Train)	0.16	0.17	0.44	0.40	0.02	0.39	0.44	0.29
Proposed (ResNet)	0.25	0.21	0.62	0.50	0.08	0.43	0.67	0.40
Proposed (ResNet) + Grabcut (Test)	**0.34**	**0.70**	**0.78**	0.62	**0.09**	**0.72**	0.67	**0.56**
Image label [27]	0.04	0.00	0.09	0.16	0.01	0.02	0.32	0.09
Area constraints [27]	0.05	0.00	0.04	0.16	0.00	0.01	0.32	0.09
SEC [19]	0.12	0.03	0.06	0.23	0.07	0.12	0.25	0.13
WTP [2]	0.11	0.03	0.18	0.11	0.00	0.02	0.23	0.10

Table 2. Evaluation of fully and weakly supervised approaches for affordance detection on the UMD part affordance dataset (novel split). Evaluation metrics are weighted F-measure and IoU.

network architectures. The first one is based on the VGG-16 architecture [34]. For training, we use a mini-batch of 3 images and an initial learning rate of 0.001 (0.01 for the final classifier layer), multiplying the learning rate by 0.1 after every 2000 iterations. We use a momentum of 0.9, weight decay of 0.0005 and run for 6000 iterations. Additionally, we use the ResNet-101 architecture [7]. Here we maintained all the hyperparameters from the original paper. The performance comparison on both IoU and ranked weighted F-measure metrics are shown in Tables 1 and 2.

As can be observed, the trend in performance is similar irrespective of the evaluation metric. The `HMP+SVM` outperforms the `DEP+SRF` combination, indicating that learning features from data is more effective than learning complex classifiers on handcrafted features. Our approach based on the `VGG` architecture as well as the `ResNet` architecture in turn outperform `HMP+SVM` confirming the effectiveness of end-to-end learning. In average, both architectures achieve similar results for both protocols.

When we compare the results in Tables 1 and 2, which correspond to the protocols *category split* and *novel split*, we observe a lower accuracy for the second protocol that evaluates the generalization across object classes. For the supervised case, the accuracy drops from 0.76 to 0.50. The

affordance class *pound* has the largest drop. By looking at the data, we observe that only instances of the two object classes hammer and mallet are marked by *pound* in the training data. In the test data, the affordance appears for the object classes tenderizer, cup, and saw. While for hammer and mallet, the entire object is labeled by *pound*, the tenderizers are only partially labeled as *pound*. As a consequence, our approach tends to label also the entire tenderizer as *pound*. Our approach also does not label parts of a cup as *pound* since mugs, which are in the training set, are not labeled by *pound*. In general, the method needs to observe enough variation in the training data since it might otherwise overfit to an object class.

5.1.2 Weakly Supervised Setting

In case of weak supervision, we evaluate our approach for the VGG architecture and the ResNet architecture. For the VGG architecture we used the same hyperparameters as for supervised learning. For ResNet, we reduced the number of iterations from 20000 to 5000 to reduce the training time.

First, we evaluate the impact of the additional Grabcut step during training as discussed in Section 4.2. We denote the results without the Grabcut step by `VGG without Grabcut(Train)` and `ResNet without Grabcut(Train)`. The accuracy drops drastically compared to our proposed method independently of the network architecture as shown in Tables 1 and 2. When we compare the network architectures VGG and ResNet, we observe that they perform similarly. While `VGG` performs slightly better for the *category split*, `ResNet` is slightly better for the *novel split*. Since the Grabcut step is essential during training, we also evaluated if an additional refinement by Grabcut of the predictions of the CNN on the test images also improves the results. We denote this setting by `VGG+Grabcut(Test)` and `ResNet+Grabcut(Test)`. On the UMD dataset, this leads to a substantial improvement. For the *novel split*, the weakly supervised method `ResNet+Grabcut(Test)` even outperforms the ResNet trained with full supervision. However, we will see in the next section that this is not the case for the more challenging CAD120 affordance dataset.

We also compare our approach to other methods that have been proposed for weakly supervised image segmentation. The methods [19, 27] rely on weaker supervision and use annotations at an image level, *i.e*. instead of keypoints only the classes that are present in an image are given without any additional localization of the classes. The method [27] uses expectation-maximization to train a CNN. The image label based version rejects all classes proposed by the CNN during the E-step but not present in the training image. The area based version uses area priors for foreground and background. It also rejects classes not present in the image, but it also encourages that the background fills at least 40% of the image area and the foreground 20%, respectively. The approach did not always converge and oscillated instead. In these cases, we stopped after the 5th iteration. The SEC method [19] uses attention heat maps from classification CNNs and conditional random fields. It is currently the best weakly supervised method on the Pascal VOC dataset, although it only uses image level supervision.

The method [2] uses the same amount of supervision as our approach, namely keypoints. The method exploits an objectness prior to improve the accuracy. We observed that we obtained better results after removing the dropout layer and replacing the upconvolution layer by the upsampling as it is used in [27].

The results in terms of IoU are shown in Tables 1 and 2. SEC outperforms WTP despite of the weaker supervision. This is also consistent with the numbers reported in [19, 2]. Our approach outperforms the other methods for affordance detection by a margin. While our approach requires more supervision than SEC and [27], our approach also outperforms WTP, which also uses keypoints as annotations.

5.2. CAD120 Affordance Dataset

5.2.1 Supervised Setting

We first evaluate the fully supervised approaches on the proposed CAD120 affordance dataset, which is discussed in Section 3. The results are reported in Tables 3 and 4. If we compare the results for the *category split* for the UMD part affordance dataset, which is given in Table 1, with the *actor split* in Table 3, we observe that the accuracies on the proposed CAD120 affordance dataset are lower than the accuracies on UMD since the proposed CAD120 affordance dataset is more challenging, cf. Fig. 1. While on UMD both networks achieve 76% mean IoU, they achieve less than 60% on the proposed dataset. In contrast to UMD, the accuracy decreases only slightly when comparing the *actor split* and *object split* in Tables 3 and 4. This shows that the methods generalize very well across object classes for this dataset while on UMD the methods seem to overfit to the object categories of the training data due to the controlled recording setting. The larger drop in accuracy on UMD, however, can also be explained by annotation inconsistencies across object classes as it is discussed in Section 5.1.1.

5.2.2 Weakly Supervised Setting

We also evaluate our approach on the dataset in the weakly supervised setting. We perform the same experiments as on the UMD part affordance dataset. We first compare the results when Grabcut is removed from the training procedure, which is denoted by `VGG without Grabcut(Train)` and `ResNet without Grabcut(Train)`. The accuracy drops when Grabcut is omitted as shown in Ta-

CAD120 affordance dataset (actor split)	Bck	Open	Cut	Contain	Pour	Support	Hold	Mean
Fully Supervised IoU category split								
Proposed (VGG)	0.81	0.67	0.00	0.54	0.42	0.70	0.64	0.54
Proposed (ResNet)	**0.86**	**0.71**	0.00	**0.61**	**0.45**	**0.79**	**0.70**	**0.59**
Weakly Supervised IoU category split								
Proposed (VGG) without Grabcut (Train)	0.58	0.37	0.10	0.19	0.18	0.18	0.41	0.29
Proposed (VGG)	**0.61**	0.33	0.00	**0.35**	**0.30**	0.22	**0.43**	**0.32**
Proposed (VGG) + Grabcut (Test)	0.60	0.23	**0.14**	0.33	0.28	**0.24**	0.42	**0.32**
Proposed (ResNet) without Grabcut (Train)	0.60	0.37	0.08	0.20	0.17	0.22	0.41	0.29
Proposed (ResNet)	0.60	0.25	0.00	**0.35**	**0.30**	0.17	0.42	0.30
Proposed (ResNet) + Grabcut (Test)	0.58	0.22	0.0	0.29	0.22	0.20	0.32	0.26
SEC [19]	0.53	**0.43**	0.00	0.25	0.09	0.02	0.20	0.22
WTP [2]	0.53	0.13	0.00	0.10	0.08	0.11	0.22	0.17
Image label [27]	0.55	0.05	0.01	0.09	0.10	0.02	0.21	0.15
Area constraints [27]	0.53	0.11	0.02	0.09	0.09	0.07	0.15	0.15

Table 3. Evaluation of fully and weakly supervised approaches for affordance detection on the CAD120 affordance dataset (actor split). The evaluation metric used is IoU.

CAD120 affordance dataset (object split)	Bck	Open	Cut	Contain	Pour	Support	Hold	Mean
Fully Supervised IoU novel split								
Proposed (VGG)	0.76	0.10	0.27	0.60	0.45	0.66	**0.60**	0.49
Proposed (ResNet)	**0.80**	**0.22**	**0.50**	**0.62**	**0.48**	**0.75**	**0.60**	**0.57**
Weakly Supervised IoU novel split								
Proposed (VGG) without Grabcut (Train)	0.61	**0.13**	**0.15**	0.20	0.18	0.14	0.46	0.27
Proposed (VGG)	0.62	0.08	0.08	0.24	**0.22**	0.20	0.46	0.27
Proposed (VGG) + Grabcut (Test)	0.62	0.07	0.05	0.21	0.19	0.27	0.41	0.26
Proposed (ResNet) without Grabcut (Train)	0.60	0.10	0.10	0.16	0.16	0.18	0.38	0.24
Proposed (ResNet)	**0.69**	0.11	0.09	**0.28**	0.21	0.36	**0.56**	**0.33**
Proposed (ResNet) + Grabcut (Test)	**0.69**	0.09	0.04	0.20	0.18	**0.44**	0.48	0.30
SEC [19]	0.54	0.04	0.09	0.13	0.09	0.08	0.13	0.16
WTP [2]	0.57	0.01	0.00	0.02	0.09	0.03	0.19	0.13
Image label [27]	0.58	0.00	0.00	0.00	0.00	0.00	0.23	0.12
Area constraints [27]	0.59	0.03	0.03	0.01	0.02	0.02	0.28	0.14

Table 4. Evaluation of fully and weakly supervised approaches for affordance detection on the CAD120 affordance dataset (object split). The evaluation metric used is IoU.

bles 3 and 4. When we add Grabcut also for inference on the test images, denoted by `VGG+Grabcut(Test)` and `ResNet+Grabcut(Test)`, we observe that Grabcut does not improve the accuracy. This is in contrast to the UMD part affordance dataset where Grabcut during testing improved the results. The benefit of Grabcut during testing on UMD can be explained by the monotonous background as shown in Fig. 1, which simplifies the segmentation.

We also compare our approach to methods for weakly supervised image segmentation [2, 19, 27]. Among them, SEC [19] performs best, yielding 22% mean IoU for the *actor split* and 16% for the *object split*. As for UMD, our approach outperforms SEC and the other methods. While `ResNet` achieves 30% mean IoU for the *actor split* and 33% for the *object split*, `VGG` achieves 32% and 27%, respectively. This is consistent with the UMD dataset where `ResNet` also generalizes better across object categories in comparison to `VGG`.

6. Conclusion

In this work, we have addressed the problem of weakly supervised affordance detection. To this end, we proposed a convolutional network that can be trained from weak keypoint annotations. In contrast to object detection and segmentation, affordance detection is a more difficult task due to the higher abstraction level compared to objects and the fact that a part can be associated with multiple affordances. For evaluation, we introduced a pixel-wise annotated affordance dataset containing 3090 images and 9916 object instances with rich contextual information which can be used to further investigate the impact of context on affordance segmentation. To assess the quality of our method, we compared our approach to several state-of-the-art weakly supervised image segmentation methods on the proposed CAD120 affordance dataset and the UMD part affordance dataset [26]. On both datasets, our proposed method achieves state-of-the-art performance both in the fully supervised setting as well as in the weakly supervised setting.

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).

References

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015. 1

[2] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the Point: Semantic Segmentation with Point Supervision. *ECCV*, 2016. 1, 2, 6, 7, 8

[3] 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. 2

[4] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng. Mining semantic affordances of visual object categories. In *CVPR*, pages 4259–4267, 2015. 1

[5] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, pages 2722–2730, 2015. 1

[6] 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. 3

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and L. A. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *arXiv:1506.02106v5*, 2016. 3, 6

[8] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *CVPR*, pages 3450–3457, 2012. 1

[9] C. Desai and D. Ramanan. Predicting functional regions on objects. In *CVPR Workshops*, pages 968–975, 2013. 2

[10] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785, 2009. 1

[11] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, pages 433–440, 2007. 1

[12] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *CVPR*, pages 1529–1536, 2011. 2

[13] T. Hermans, J. M. Rehg, and A. Bobick. Affordance prediction via learned object attributes. In *ICRA: Workshop on Semantic Perception, Mapping, and Exploration*, 2011. 1, 2

[14] Y. Jiang, H. Koppula, and A. Saxena. Hallucinated humans as the hidden context for labeling 3d scenes. In *CVPR*, pages 2993–3000, 2013. 2

[15] 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. 1, 2

[16] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez. Color attributes for object detection. In *CVPR*, pages 3306–3313, 2012. 1

[17] D. I. Kim and G. Sukhatme. Semantic labeling of 3d point clouds with object affordance for robot manipulation. In *ICRA*, pages 5578–5584, 2014. 1, 2

[18] H. Kjellström, J. Romero, and D. Kragić. Visual object-action recognition: Inferring object affordances from human demonstration. *CVIU*, 115(1):81–90, 2011. 2

[19] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, pages 695–711, 2016. 1, 2, 6, 7, 8

[20] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. *IJRR*, 32(8):951–970, 2013. 1

[21] H. S. Koppula and A. Saxena. Physically grounded spatio-temporal object affordances. In *ECCV*, pages 831–847. 2014. 2, 3

[22] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *PAMI*, 38(1):14–29, 2016. 2

[23] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009. 1

[24] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *IJRR*, 34(4-5):705–724, 2015. 2

[25] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, pages 3337–3344, 2011. 1

[26] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *ICRA*, pages 1374–1381, 2015. 1, 2, 3, 4, 5, 6, 8

[27] 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. 1, 2, 6, 7, 8

[28] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, pages 503–510, 2011. 1

[29] D. Pathak, P. Krähenbühl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, pages 1796–1804, 2015. 2

[30] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, pages 2751–2758, 2012. 1

[31] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. 5

[32] A. Roy and S. Todorovic. A multi-scale CNN for affordance segmentation in RGB images. In *ECCV*, pages 186–201, 2016. 1, 2

[33] F. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *ECCV*, pages 413–432, 2016. 2

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[35] H. O. Song, M. Fritz, D. Goehring, and T. Darrell. Learning to detect visual grasp affordance. *IEEE Transactions on Automation Science and Engineering*, 13(2):798–809, 2016. 2

[36] A. Vezhnevets and J. M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *CVPR*, pages 3249–3256, 2010. 2

[37] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised semantic segmentation with a multi-image model. In *ICCV*, pages 643–650, 2011. 2

[38] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *CVPR*, pages 845–852, 2012. 2

[39] J. Xu, A. Schwing, and R. Urtasun. Tell me what you see and I will show you where it is. In *CVPR*, pages 3190–3197, 2014. 2

[40] W. Zhang, S. Zeng, D. Wang, and X. Xue. Weakly supervised semantic segmentation for social images. In *CVPR*, pages 2718–2726, 2015. 2

[41] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *ECCV*, pages 408–424. 2014. 1

[42] Y. Zhu, Y. Zhao, and S. Chun Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *CVPR*, pages 2855–2864, 2015. 2