2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Object-Based Affordances Detection with Convolutional Neural Networks and Dense Conditional Random Fields

Anh Nguyen, Dimitrios Kanoulas, Darwin G. Caldwell, and Nikos G. Tsagarakis

***Abstract*— We present a new method to detect object affordances in real-world scenes using deep Convolutional Neural Networks (CNN), an object detector and dense Conditional Random Fields (CRF). Our system first trains an object detector to generate bounding box candidates from the images. A deep CNN is then used to learn the depth features from these bounding boxes. Finally, these feature maps are post-processed with dense CRF to improve the prediction along class boundaries. The experimental results on our new challenging dataset show that the proposed approach outperforms recent state-of-the-art methods by a substantial margin. Furthermore, from the detected affordances we introduce a grasping method that is robust to noisy data. We demonstrate the effectiveness of our framework on the full-size humanoid robot WALK-MAN using different objects in real-world scenarios.**

I. INTRODUCTION

Despite the variations in shape and appearance of real-world objects, humans can effortlessly recognize them and their functionalities within a fraction of a second [1]. The concept of understanding functional aspects of objects or *affordances* was originally studied in [2]. In many robotic applications, this concept plays an important role since it describes the potential actions that the robot can perform on the objects. Consider a robotic task such as pouring water from a bottle to a cup. To plan the actions, the robot will first need to understand the environment, i.e. to detect and localize which objects are present in the scene. Furthermore, it must also be able to identify object affordances (e.g. `contain` or `grasp`) to plan the grasp and complete the task.

From the biological point of view, there is evidence which shows that human brain solves the object recognition problem using a coarse-to-fine strategy [1]. Although neuro-scientists do not yet fully understand the mechanism behind this ability, it seems to play an important role in enabling the central visual field of human brain to rapidly recognize objects and their functionalities. This strategy was applied successfully in computer vision as a unified optimization problem [3] to segment objects and achieved competitive results. Inspired by these works, we have developed a framework which assigns affordance labels to particular objects by using the prior information from their locations.

Recently, there is a growing interest in detecting object affordances [4] [5] [6] [7]. In this paper, we target the same goal, but address the problem in a coarse-to-fine manner, in order to detect the affordances more robustly, while also including semantic information on the scene to assist the robot. Unlike the previous works that were tested only on simple [6] or synthetic [8] datasets, we introduce a new affordance dataset that contains real-world images. Our dataset creates a more challenging problem and is better suited to real-world robotic applications. We show that in complicated scenarios such as in our dataset, it is necessary to first localize the objects in the scene before performing the affordance detection.

Fig. 1. Object affordances and their grasp frames. **From left to right:** The original images, the affordances associated with each object in the images, and the grasp frames generated from the affordances.

Over the past few years, remarkable progress has been made in computer vision with the rise of deep learning [9]. It is, therefore, not surprising that many grasping methods have been proposed based on deep networks [10] [11]. Although these methods can provide robust grasps for novel objects, they only focus on finding optimal solutions, while discarding other semantic information such as the objects identities or affordances. Consequently, the robot can grasp the objects but has no understanding of its surrounding environment. We propose a grasping method from the detected affordances that can deal with noisy data and test it on a full-size humanoid robot. Using our approach, the robot can successfully perform grasps while being fully aware of the manipulated objects. Fig. 1 shows some examples of grasp localization from object affordances.

Next we review the related work in Section II and describe our approach in Section III. In Section IV, we introduce our new affordance dataset and present the results. We describe our grasping method and experiments on the WALK-MAN robot in Section V. Finally, we conclude the paper and discuss the future work in Section VI.

The authors are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163, Genova, Italy. {Anh.Nguyen, Dimitrios.Kanoulas, Darwin.Caldwell, Nikos.Tsagarakis}@iit.it

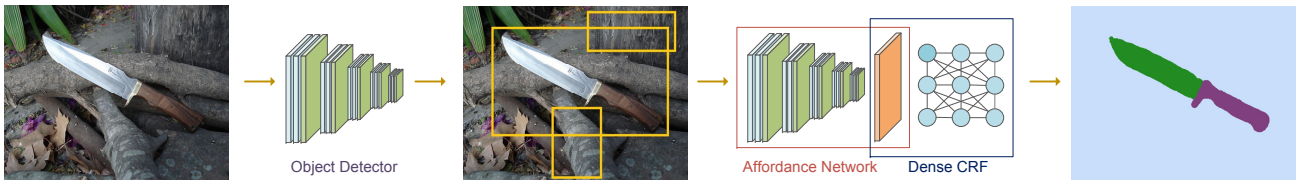


Fig. 2. An overview of our affordance detection method. **From left to right:** A deep network is first used as an object detector to generate the object bounding boxes that narrow down the region of interest. A second network is then used to produce feature maps from these bounding boxes. Finally, these maps are post-processed with dense CRF to improve the prediction along the class boundaries.

II. Related Work

The concept of affordances has been applied extensively in many robotic applications. One of the most popular is robotic grasping [12] [13]. Levine et al. [11] collected a large amount of grasping data and applied a deep network to learn successful grasps from monocular images. Similarly, the work in [10] proposed a method to detect grasp affordance using two deep networks. Besides the traditional grasping problem, many works have investigated different problems such as using tools from detected affordances [14], exploring actions and effects when robot interacts with objects [15], or reorienting objects for task-oriented grasping [16].

In computer vision, following the work in [9] on image classification, CNN has become very popular and been applied in many other problems. Ren et al. [17] proposed a method to combine a deep CNN with region proposal methods for detecting objects. Similarly, the work in [18] developed a method that integrated region proposals with the very deep network ResNets [19] for object detection. Badrinarayanan et al. [20] used a deep CNN with an encoder-decoder architecture for real-time semantic pixel-wise labeling. Recently, the work in [21] [22] combined CNN with dense CRF to further improve the segmentation results.

In [4], Myers et al. followed a traditional approach to detect object affordances using hand-designed geometric features. Recently, the work in [6] used a CNN with an encoder-decoder architecture to detect object affordances from depth features. This work also proposed to use the detected affordances as visual cues to grasp objects. This grasping method was tested on a humanoid robot. Srikantha and Gall [5] used a CNN within an expectation maximization framework when human pose is used as the context. The work in [8] focused on the same goal, however they used 3D point clouds as input, and tested their method on a synthetic dataset. Recently, multi-scale CNN was introduced in [7] to detect environment affordances in RGB images.

Our approach follows the same concept by using a CNN to learn rich depth features from the data. However, unlike the work in [6] that uses the CNN on the whole input image, we employ an object detector to narrow down the region of interest. In [5], Srikantha and Gall made the assumption that the human pose is known and can be integrated into the detection process. In this paper, we show that the detection results can be improved by integrating an object detector and refining the result with CRF. Furthermore, we introduce a grasping method based on the concept of principle component analysis and run extensive grasping experiments on the full-size humanoid robot WALK-MAN.

III. Affordances Detection

Inspired by [3] [21] [22], we propose a new framework to detect object affordances based on depth features. We first employ an object detector to predict the objects bounding boxes from the image, then a deep CNN is used to create a dense feature map in each bounding box, finally these feature maps are post-processed with dense CRF to further improve the result. Fig. 2 shows an overview of our approach.

A. Problem Formulation

We let an object be defined by two basic information features: its position in the image and its labels in the affordance classes. The position of an object is defined as a rectangle on the image, while an affordance label is defined as a region of pixels that share the same functionality. Each object may have many affordances, while we assume that a pixel in the image has only one affordance. Formally, let $\mathcal{L} = \{l_1, ..., l_k\}$ be the set of k affordance classes, and $\mathbf{P}$ be the set of pixels in the image. The affordance detection task is to find the assignment $\mathbf{x}: \mathbf{P} \rightarrow \mathcal{L}$ that maps each pixel $\mathbf{p} \in \mathbf{P}$ in the image with its most probable label in $\mathcal{L}$.

B. Object-based Affordances Detection

1) Object Detector: Detecting objects in RGB images is a well-known problem in computer vision. Recently, rapid advancements have been made in this field with the rise of deep learning [17]. Object detection approaches can be divided into region-based [17] [18] and non region-based methods [23]. Although non region-based methods can achieve real-time performance, they are still outperformed by region-based systems on public benchmarks [18].

In this paper, we use the method that was proposed recently in [18] (R-FCN) as our object detector since it achieves state-of-the-art results on public benchmarks, and has fast inference time. Unlike other object detection methods that modified the VGG-16 [24] network in their main architecture, R-FCN is designed to naturally adopt the state-of-the-art image classification network, such as ResNets [19], hence letting us train a very deep network in a fully convolutional manner. R-FCN contains two main steps: generating region proposals and classifying them. The candidate regions are first generated by a deep network that shares features with R-FCN, then the training process classifies each region into object categories and background. During testing, a threshold t is chosen to decide whenever a region proposal belongs to an object category or not. We refer readers to [18] for the full description of the R-FCN architecture.

2) Affordances Detection: Similar to [5] [6] [7], we cast the affordance detection problem as a pixel-wise labeling task. We modify the state-of-the-art VGG-16 network to produce dense feature map at each object region provided by R-FCN. Since the original VGG-16 network is designed for the image classification problem, its final layer is a classifier and can't produce dense heat maps to predict the affordance label for each pixel. Therefore, we replace this layer with an 1×1 convolution layer of 10 dimensions to predict scores for each class in our dataset (we use a dataset with 9 affordance classes and 1 class for the background). We then convert all the fully-connected layers of VGG-16 into convolutional ones. Furthermore, we use atrous convolution technique [21] to increase the field-of-view of the convolution layers without increasing the number of network parameters. This technique also helps us to balance the trade-off between small field-of-view for accurate localization and large field-of-view for incorporating context information.

To deal with arbitrary resolutions from the input images, we apply a multi-scale strategy introduced in [25]. During the training and testing, we re-scale the original image into three different versions and feed them to three parallel networks that share the same parameters. The final feature map is created by bilinearly interpolating the feature map from each network to the original image resolution, and taking the maximum value across the three scales at each pixel.

C. Post-processing with CRF

We adopt dense CRF to post-process the output from the deep network since it showed substantial performance gains in traditional the image segmentation task [21] [22]. The energy function of dense CRF is given by:

$$E(\mathbf{x}|\mathbf{P}) = \sum_{\mathbf{p}} \theta_{\mathbf{p}}(x_{\mathbf{p}}) + \sum_{\mathbf{p},\mathbf{q}} \psi_{\mathbf{p},\mathbf{q}}(x_{\mathbf{p}}, x_{\mathbf{q}}) \qquad (1)$$

In particular, the unary term $\theta_{\mathbf{p}}(x_{\mathbf{p}})$ indicates the cost of assigning label $x_{\mathbf{p}}$ to pixel $\mathbf{p}$. This term can be considered as the output of the last layer from the affordance network since this layer produces a probability map for each affordance class [21]. The pairwise term $\psi_{\mathbf{p},\mathbf{q}}(x_{\mathbf{p}}, x_{\mathbf{q}})$ models the relationship among neighborhood pixels and penalizes inconsistent labeling. The pairwise potential can be defined as weighted Gaussians [26]:

$$\psi_{\mathbf{p},\mathbf{q}}(x_{\mathbf{p}}, x_{\mathbf{q}}) = \mu(x_{\mathbf{p}}, x_{\mathbf{q}}) \sum_{m=1}^{M} w^m \kappa^m(\mathbf{f}_{\mathbf{p}}, \mathbf{f}_{\mathbf{q}}) \qquad (2)$$

where each κ^m for $m = 1, ..., M$, is a Gaussian kernel based on the features $\mathbf{f}$ of the associated pixels, and has the weights w^m. The term $\mu(x_{\mathbf{p}}, x_{\mathbf{q}})$ represents label compatibility and is 1 if $x_{\mathbf{p}} \neq x_{\mathbf{q}}$, otherwise 0. As in [26], we use the following kernel in the pairwise potential:

$$\begin{aligned} \kappa(\mathbf{f}_{\mathbf{p}}, \mathbf{f}_{\mathbf{q}}) = w_1 \exp\left(-\frac{|p_{\mathbf{p}} - p_{\mathbf{q}}|^2}{2\sigma_\alpha^2} - \frac{|I_{\mathbf{p}} - I_{\mathbf{q}}|^2}{2\sigma_\beta^2}\right) \\ + w_2 \exp\left(-\frac{|p_{\mathbf{p}} - p_{\mathbf{q}}|^2}{2\sigma_\gamma^2}\right) \end{aligned} \qquad (3)$$

where the first term depends both on pixel positions (denoted as p) and its color (denoted as I), and the second term only depends on pixel positions. The parameter σ controls the scale of the Gaussian kernel.

Our goal is to minimize the CRF energy $E(\mathbf{x}|\mathbf{P})$, which yields the most probable label for each pixel. Since the dense CRF has billion edges and the exact minimization is intractable, we use the mean-field algorithm [26] to efficiently approximate the energy function.

D. Training

We generally follow the procedure described in [18] to train our object detector. The network is trained using gradient descent with 0.9 momentum, 0.0005 weight decay. The input images are resized to 600×600 pixels resolution. The learning rate is first set to 0.001, then we decrease it by a factor of 10 every 20000 iterations. During training, we use 128 regions of interest as bounding box candidates for backpropagation. The network is trained using the lost function combined from cross-entropy loss and box regression loss. The training time is approximately 1 day on a NVIDIA Titan X GPU.

After training the object detector network, we do the inference to generate object bounding boxes in order to feed these boxes to the affordance network. The affordance network is trained using stochastic gradient descent with cross-entropy loss, 0.9 momentum and 0.0005 weight decay. The learning rate is initialized to 0.001 and decreased by a factor of 10 every 2000 iterations. Similar to [6], we weigh the loss differently based on the statistics of each class to deal with the large variation in the number of pixels in the training set. The network is trained until convergence with no further reduction in training loss. It takes approximately 1 day to train our affordance network on a Titan X GPU.

IV. EXPERIMENTS

A. IIT-AFF Dataset

The work in [4] proposed the first affordance dataset with pixel-wise labels. The data were collected using a Kinect sensor, which records RGB-D images at a 480×640 resolution. Although this dataset contains a large amount of annotated images, most of them were captured on a turntable table in a cluttered-free setup. Consequently, the use of this dataset may not be sufficient for robotic applications in real-world cluttered scenes.

Data collection Since CNN requires a large amount of data for training, we introduce a new affordance dataset to fulfill this purpose. In general, we want to create a large-scale dataset that enables the robot to infer properly in real-world scenes after the training step. In order to do this, we first choose a subset of object categories from the ImageNet dataset [27]. In addition, we also collect RGB-D images from various cluttered scene setups using an Asus Xtion sensor and a MultiSense-SL camera. The images from the Asus Xtion and MultiSense camera were collected at 480×640 and 1024×1024 resolutions, respectively.

Data annotation Our dataset provides both bounding box annotations for object detection and pixel-wise labels for affordance detection. We reuse the bounding boxes that come with the images from the ImageNet dataset, while all images are manually annotated with the affordance labels at pixel-level. Since the images from the ImageNet dataset don't have the associated depth maps, we use the state-of-the-art method in [28] to generate the relative depth maps for these images, which can be used by algorithms that need them.

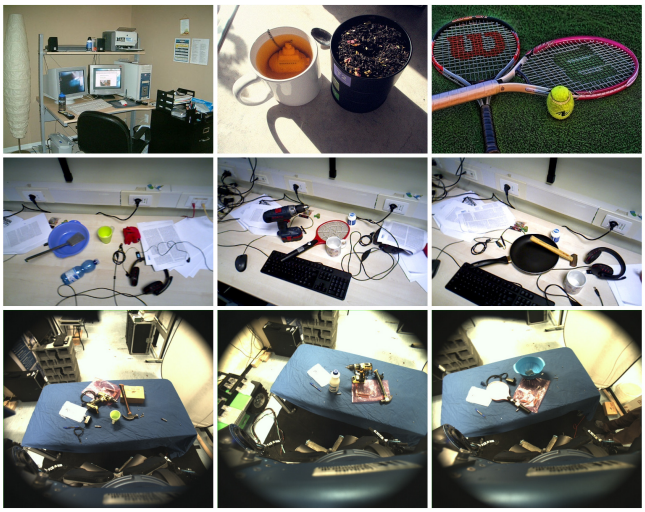
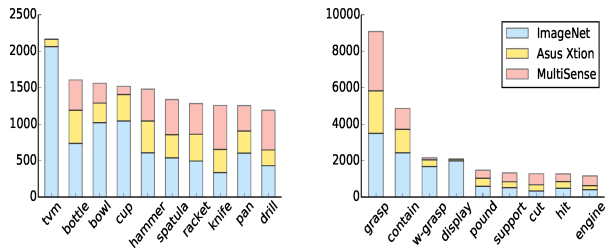


Fig. 3. Example images from our IIT-AFF dataset. **Top row:** Images from the ImageNet dataset. **Middle row:** Images from the Asus Xtion camera. **Bottom row:** Images from the MultiSense-SL camera.



(a) Object distribution (b) Affordance distribution

Fig. 4. The statistics of our IIT-AFF dataset. (a) Object distribution as the number of bounding boxes in each object category. (b) Affordance distribution as the number of regions in each affordance class.

Dataset statistic In particular, our dataset has 10 object categories (*bottle*, *bowl*, *cup*, *drill*, *hammer*, *knife*, *monitor*, *pan*, *racket*, *spatula*) and 9 affordance classes (`contain`, `cut`, `display`, `engine`, `grasp`, `hit`, `pound`, `support`, `w-grasp`), which are common human tools with their related manipulation capabilities (Table I). The dataset has $8,835$ images, containing $14,642$ bounding box annotated objects (in which $7,866$ bounding boxes come from the ImageNet dataset) and $24,677$ affordance parts of the objects. We use 50% of the dataset for training, 20% for validation, and the rest 30% for inference. Fig. 3 and 4 show some example images and the statistics of our dataset.

TABLE I
DESCRIPTION OF OBJECT AFFORDANCE LABELS

Affordances	Function
`contain`	Storing/holding liquid/objects (e.g. the inside part of bowls)
`cut`	Chopping objects (e.g. the knife blade)
`display`	Showing information (e.g. the monitor screen)
`engine`	Covering engine part of tools (e.g. the drill's engine)
`grasp`	Enclosing by hand for manipulation (e.g. handles of tools)
`hit`	Striking other objects with refection (e.g. the racket head)
`pound`	Striking other objects with solid part (e.g. the hammer head)
`support`	Holding other objects with flat surface (e.g. turners)
`w-grasp`	Wrapping by hand for holding (e.g. the outside of a cup)

B. Evaluation Metric, Baseline, and Implementation

1) Evaluation metric: We compare our affordance results with other methods using the F_β^w metric [29]. This metric extends the F_β measure and is calculated as follows:

$$F_\beta^w = (1+\beta^2)\frac{Precision^w \cdot Recall^w}{\beta^2 \cdot Precision^w + Recall^w} \quad (4)$$

where $\beta = 1$, $Precision^w$, and $Recall^w$ are the weighted versions of the standard $Precision$ and $Recall$ measures. The novelty of this measure as explained in [29] is to weigh the errors of the pixels by taking into account their location and neighborhood information to overcome three flawed assumptions: interpolation, dependency and equal importance of the prediction map.

2) Baseline: We evaluate our proposed method (denoted as BB-CNN and BB-CNN-CRF) and compare the results with following state-of-the-art approaches: CNN with encoder-decoder architecture [6] on RGB and RGB-D images (denoted as ED-RGB and ED-RGBD), DeepLab [21] with and without CRF (denoted as DeepLab and DeepLab-CRF). For all compared methods, we generally follow the training process described in the associated papers and use the versions that adapt the VGG-16 network as the main architecture for a fair comparison. We note that except ED-RGBD that uses the depth images, all other methods only use the RGB images.

3) Implementation: Similar to [26], we first empirically set the parameters $w_2 = 3$ and $\sigma_\gamma = 3$, while other parameters w_1, σ_α, σ_β are set using cross-validation on the validation set. The final values of w_1, σ_α, and σ_β are 5, 20, and 3, respectively. The max iteration of mean-field algorithm is empirically set to 10. We also perform cross-validation to search for the best threshold t of R-FCN from the set $\{0.3, 0.4, 0.5, 0.6\}$. The value of t is finally set to 0.5. Here, we notice that we do not choose t based on the object detection result but on the affordance segmentation result, since the input of our affordance network comes from the object detector. During testing, the bounding boxes provided by R-FCN may have overlap regions. In that case, the final affordance label is decided by taking the maximum value across all feature maps of these bounding boxes. The areas that are not covered by the bounding boxes will be considered as background.

TABLE II
OBJECT DETECTION RESULTS

	Faster R-CNN-ZF	Faster R-CNN-VGG-16	R-FCN-ResNet-50	R-FCN-ResNet-101
bowl	87.89	86.19	**88.55**	86.65
tvm	78.98	**83.58**	77.50	75.16
pan	87.69	89.92	**91.31**	90.24
hammer	78.21	82.93	80.48	**86.14**
knife	78.95	75.30	82.97	**84.23**
cup	81.03	**86.82**	81.05	81.46
drill	86.30	91.31	90.91	**92.08**
racket	87.35	91.96	**94.75**	94.71
spatula	73.05	77.92	76.50	**83.09**
bottle	87.80	**90.42**	89.29	90.30
mAP	82.73	85.64	85.33	**86.41**

TABLE III
AFFORDANCE DETECTION RESULTS

	ED-RGB [6]	ED-RGBD [6]	DeepLab [21]	DeepLab-CRF [21]	BB-CNN (ours)	BB-CNN-CRF (ours)
`contain`	66.38	66.00	68.84	69.68	75.60	**75.84**
`cut`	60.66	60.20	55.23	56.39	69.87	**71.95**
`display`	55.38	55.11	61.00	62.63	72.04	**73.68**
`engine`	56.29	56.04	63.05	65.11	72.84	**74.36**
`grasp`	58.96	58.59	54.31	56.24	63.72	**64.26**
`hit`	60.81	60.47	58.43	60.17	66.56	**67.07**
`pound`	54.26	54.01	54.25	55.45	64.11	**64.86**
`support`	55.38	55.08	54.28	55.62	65.01	**66.12**
`w-grasp`	50.66	50.42	56.01	57.47	67.34	**68.41**
Average	57.64	57.32	58.38	59.86	68.57	**69.62**

C. Results

1) Object Detection: Since our dataset provides bounding boxes for traditional object detection task, we briefly benchmark recent state-of-the-art methods Faster R-CNN [17] and R-FCN [18] with different popular networks: ZF, VGG-16, ResNet-51 and ResNet-101. Table II shows object detection results using the mean Average Precision (mAP) measure. Overall, R-FCN-ResNet-101 achieves the highest accuracy in average, while Faster R-CNN-VGG-16 also shows competitive results. The testing time of R-FCN-ResNet-101 is also faster than Faster R-CNN-VGG-16 (130ms vs 200ms per image). Since R-FCN gives higher accuracy in average and is also faster than Faster R-CNN, while we need a fast and accurate object detector to generate candidates for our affordance network, R-FCN is more suitable for our purpose than Faster R-CNN. In practice, we use R-FCN-ResNet-101 as our object detector.

2) Affordance Detection: Table III summarizes the F_β^w results on our IIT-AFF dataset. From this table, we notice that the results are significantly improved when the object detector is combined with our affordance network to detect object affordances. Overall, we achieve approximately 10% improvement in comparison with the recent state-of-the-art methods [6] [21]. In particular, our affordance network BB-CNN yields 68.57% accuracy while post-processing with CRF brings extra 1.05% improvement. Our approach also achieves the highest detection accuracy for all 9 affordance classes. It demonstrates that when we have a good object detector to generate bounding box candidates for the affordance network, the overall performance can be significantly improved. We also notice that DeepLab gives slightly higher accuracy than the ED-RGB, while the use of depth images in ED-RGBD is unable to improve the results.

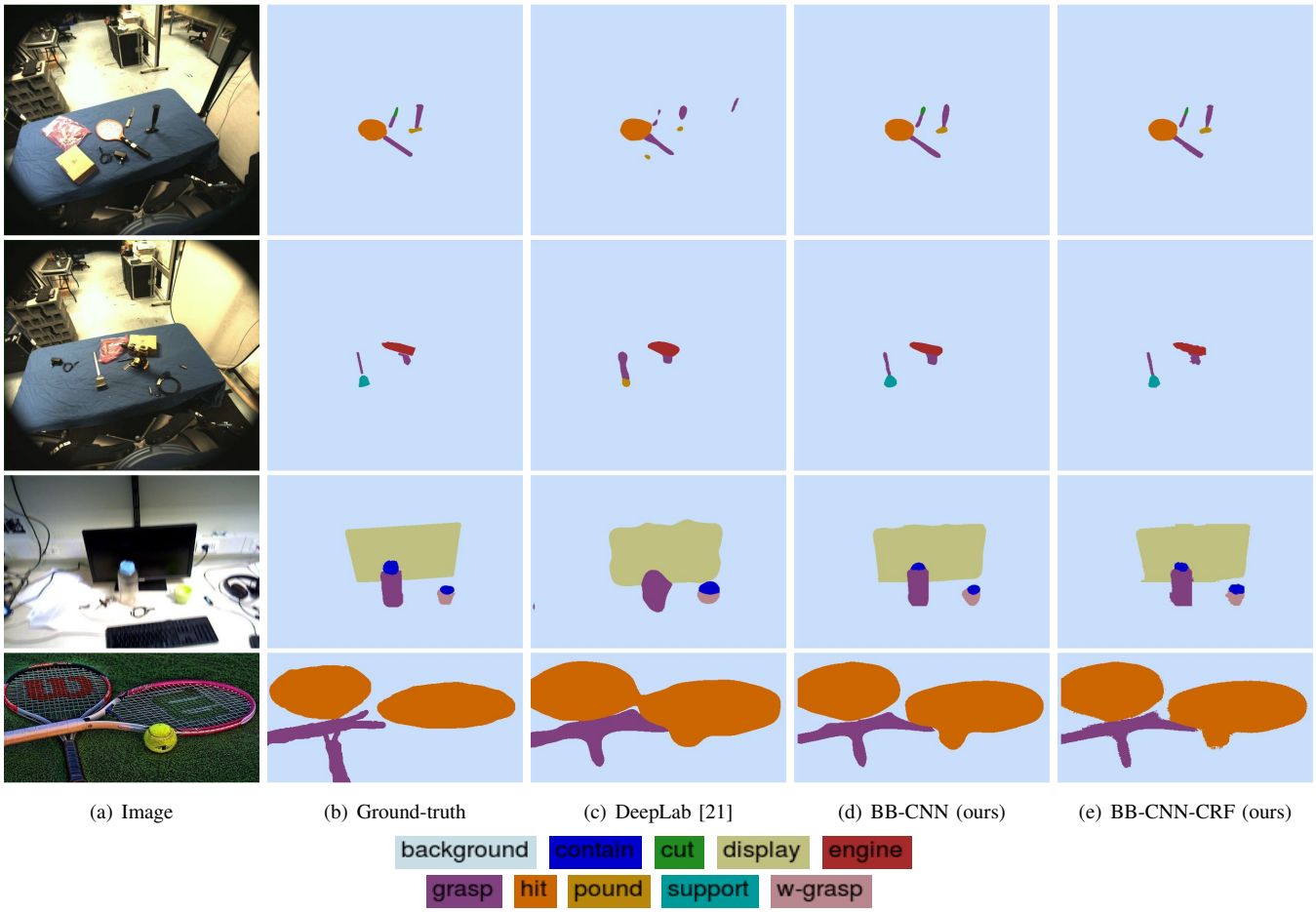
Unlike we expected, BB-CNN-CRF only outperforms BB-CNN by a small margin. We have observed similar result between DeepLab-CRF and DeepLab in our dataset. Although the improvement is not significant, the visualization shows that CRF successfully retains the boundaries between affordance classes and makes the final results look more natural. Fig. 5 shows some affordance detection results using different methods with our dataset.

D. Discussion

Architecture Since we use two separate networks and refine the results with CRF, our framework can not be end-to-end trained as a single network. However, it produces the semantic understanding of the scene. In many robotics applications, this understanding plays an important role since the robot may need to recognize both object affordances and its identity in order to complete a task. Uniform architectures such as [30] integrated both object bounding boxes and pixel labels in a jointly trained CNN, but it can only give the segmentation result as the output. Within the deep frameworks, our approach is inspired by the recent success in [21] [22]. While the authors in [22] focused on integrating CNN and CRF into an end-to-end trainable system, and [21] proposed spatial pyramid pooling method to handle multi-scale problem in traditional image segmentation task, we use multi-scale strategy [25] with the input from the object detector to improve the affordance detection results.

Generalization One important advantage of our framework is that we can detect object affordances on new object categories that don't exist in the training set. This is particularly helpful since the use of CNN requires a large amount of data and manually label data is an expensive task, while many large datasets with object bounding boxes are already available [27]. This is achieved since many man-made objects usually have parts that share the same functionality. Using the proposed framework, we can train the object detector on the dataset that has only object bounding boxes, and then detect affordances without requiring any more training data with affordances labels.

Failure cases Since our affordance network receives the input from the object detector, a typical failure case in our framework is when the object detector is unable to detect the object, or miss-recognize regions that should not be considered as objects. Although object detection is considered as a mature field in computer vision [18], in practice it is crucial to select the right threshold t for the best performance of the affordance network. Fig. 6 shows some qualitative results when we use our method to detect the affordances of an unseen object, and a failure case when the object detector miss-recognizes the object.



(a) Image (b) Ground-truth (c) DeepLab [21] (d) BB-CNN (ours) (e) BB-CNN-CRF (ours)

Fig. 5. Examples of affordance detection results on the IIT-AFF dataset.

V. Robotics Application

In this section, we propose a new method to grasp objects from their affordances. We run a series of experiments to evaluate our method on the full-size humanoid WALK-MAN with underactuated hands. The experimental video and our new affordance dataset can be found at the following link:

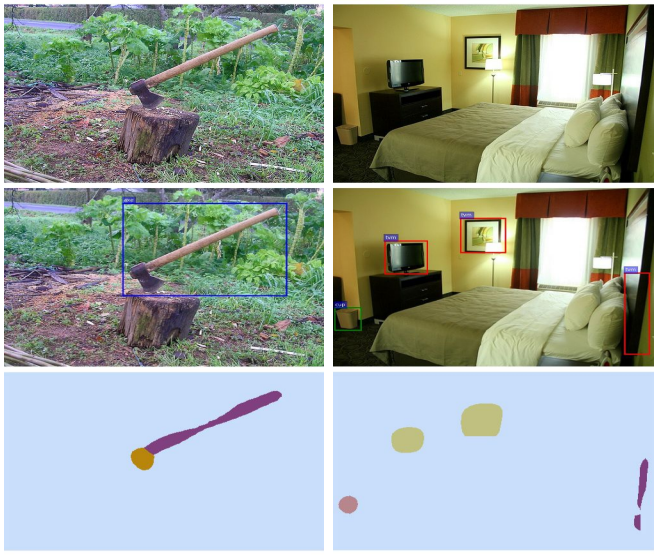
https://sites.google.com/site/ocnncrf/

A. Hardware

WALK-MAN [31] is a 31 DoF full-size humanoid robot with two underactuated hands. Each hand is controlled by one single motor, generating the open-close grasping motion of the five fingers simultaneously. The vision system is equipped with a MultiSense-SL camera. We use the OpenSoT [32] library to plan whole-body motions for the robot. A *control pc* with a Core i5 3.2GHz x 4 processor and 12GB RAM is used to control the robot, while the vision part runs on a *vision pc* with a NVIDIA Titan X GPU.

B. Generating Grasp Frames

Similar to [6] [10], we first detect the grasping frame in 2D, and then transform it into 3D in order to use it in robotics applications. In particular, to grasp an object, the robot needs to know the location (ℓ_u, ℓ_v, ℓ_w) and orientation $(\phi_\alpha, \phi_\beta, \phi_\gamma)$ of a 6DoF grasp frame G_f. The 3D grasp point



(a) Generalization (b) Failure case

Fig. 6. Qualitative results. (a) Our method successfully detects affordances for an *axe* which doesn't exist in our dataset. (b) A failure case when the object detector gives wrong input for the affordance network.

(ℓ_u, ℓ_v, ℓ_w) can be obtained by projecting a 2D pixel point into 3D space using a calibrated 3D range sensor. Similarly, the 3D orientation can be calculated by first finding two

TABLE IV
GRASP SUCCESS RATE (IN %)

	Rectangle-based [6]	Ours	Affordance
Bottle	60	80	`grasp`
Bowl	80	80	`contain`
Hammer	100	100	`grasp`
Knife	60	80	`grasp`
Pan	80	100	`grasp`
Racket	80	100	`grasp`
Average	**76.7**	**90.0**	

vectors $(\phi_\alpha, \phi_\beta)$ from the detected affordance, then the third component (ϕ_γ) is the cross product of these two.

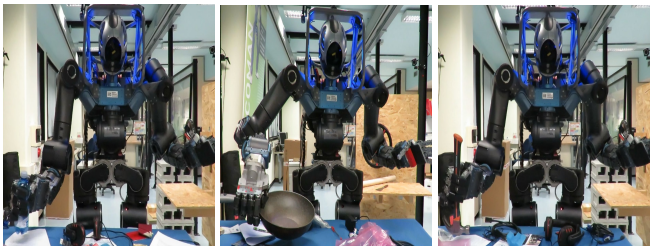
In particular, we represent the grasping frame of each affordance map by a fitted ellipse. Intuitively, the center of this ellipse represents our grasping point while the orientation of its two axes corresponds to the orientation $(\phi_\alpha, \phi_\beta)$ of the grasping frame we want to find. To estimate the orientation of the ellipse, we use the fact that the eigenvectors of the covariance matrix represent the dominant spreading directions of the data, while their associated eigenvalues define how large these spreads are. Therefore, the orientation $(\phi_\alpha, \phi_\beta)$ can be achieved by extracting two eigenvectors associated with the largest and second-largest eigenvalues. From the dominant eigenvector $\mathbf{v}$ (i.e. associated with the largest eigenvalue), the orientation angle δ of the ellipse with respect to the root frame of the image can be calculated as follow:

$$\delta = \arctan\left(\frac{\mathbf{v}_y}{\mathbf{v}_x}\right) \tag{5}$$

C. Grasping Experiments

For this experiment, we use objects from 6 categories in our dataset. The robot is positioned in front of a table, while the target objects are placed at a right-hand reachable position together with other random objects. Although the experiments run in cluttered scene setups, we assume that the robot can grasp the object with a collision free trajectory. For each object, we perform 10 trials and a grasp is considered successful if the robot can grasp, raise, and hold the object in the air for 10 seconds.

Table IV summarizes the grasp success rate and the detected affordances that were used by the robot to grasp the objects. To evaluate our new grasping method, we baseline it against the rectangle-based strategy in [6]. The BB-CNN network is used to detect object affordances as the input to the grasping process for both methods. It can be seen that the new method performs more robustly in comparison with the method in [6]. This is because the process to create the grasp frame in [6] depends on the rectangle generated from the convex hull of the detected affordance. This step is very sensitive to the outliers in cluttered scenes. This limitation does not occur in our method since we generate the grasp frame based on the principle axes of the data. Fig. 7 shows some successful grasps using WALK-MAN.



Bottle *Pan* *Hammer*

Fig. 7. Example of successful grasps using WALK-MAN.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new method to detect object affordances with CNN. We have demonstrated that the affordance detection results can be improved by using an object detector and dense CRF. Moreover, we introduced a challenging dataset that is suitable for real-world robotic applications. From the detected affordances, we presented a grasping method that is robust to noisy data. The effectiveness of our approach was demonstrated by performing different grasping experiments in cluttered scenes on the full-size humanoid robot WALK-MAN.

Since our framework is designed as separated modules, the affordance detection accuracy can be improved by upgrading each module. In future work, we aim to test more object detectors which may give better affordance detection results. Currently, in the robotic experiment, we assume that the affordances area can be fitted in the robot hand and treat all affordances as graspable. We plan to develop a more robust grasping method and use the actual affordance functionalities to complete more complicated manipulation tasks such as pick and place, pouring, or cutting objects.

ACKNOWLEDGMENT

This work is supported by the European Union Seventh Framework Programme [FP7-ICT-2013-10] under grant agreement no 611832 (WALK-MAN). The authors would like to thank the anonymous reviewers for their useful comments.

REFERENCES

[1] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How Does the Brain Solve Visual Object Recognition?" *Neuron Perspective*, vol. 73, no. 3, pp. 415–434, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.neuron.2012.01.010

[2] J. J. Gibson, *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin, 1979.

[3] W. Xia, Z. Song, J. Feng, L.-F. Cheong, and S. Yan, "Segmentation over Detection by Coupled Global and Local Sparse Representations," in *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, 2012, pp. 662–675.

[4] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance Detection of Tool Parts from Geometric Features," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[5] A. Srikantha and J. Gall, ""Weakly Supervised Learning of Affordances," *CoRR*, vol. abs/1605.02964, 2016.

[6] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Detecting Object Affordances with Convolutional Neural Networks," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[7] A. Roy and S. Todorovic, "A Multi-Scale CNN for Affordance Segmentation in RGB Images," *European Conference on Computer Vision*, 2016.

[8] M. Schoeler and F. Wörgötter, "Bootstrapping the semantics of tools: Affordance analysis of real world objects on a per-part basis," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 2, pp. 84–98, June 2016.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[10] I. Lenz, H. Lee, and A. Saxena, "Deep Learning for Detecting Robotic Grasps," *International Journal of Robotics Research*, 2015.

[11] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection," *CoRR*, vol. abs/1603.02199, 2016. [Online]. Available: http://arxiv.org/abs/1603.02199

[12] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-Driven Grasp Synthesis A Survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, April 2014.

[13] M. Gualtieri, J. Kuczynski, A. M. Shultz, A. ten Pas, R. Platt, and H. A. Yanco, "Open-world assistive grasping using laser selection," *International Conference on Robotics and Automation (ICRA)*, 2017.

[14] A. Dehban, L. Jamone, A. R. Kampff, and J. Santos-Victor, "Denoising auto-encoders for learning of objects and tools affordances in continuous space," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[15] E. Ugur and J. Piater, "Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.

[16] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Preparatory Object Reorientation for Task-Oriented Grasping," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[18] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 379–387.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[20] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[21] L. 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," *CoRR*, vol. abs/1606.00915, 2016.

[22] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1529–1537.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2016.

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[25] I. Kokkinos, "Pushing the boundaries of boundary detection using deep learning," *Int.l Conf. on Learning Representations (ICLR)*, 2016.

[26] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., 2011, pp. 109–117.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[28] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015. [Online]. Available: http://arxiv.org/abs/1411.6387

[29] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to Evaluate Foreground Maps," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 248–255.

[30] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," *European Conference on Computer Vision*, pp. 297–312, 2014.

[31] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, L. Natale, E. Mingo Hoffman, H. Dallali, N. Kashiri, J. Malzahn, J. Lee, P. Kryczka, D. Kanoulas, M. Garabini, M. Catalano, M. Ferrati, V. Varricchio, L. Pallottino, C. Pavan, A. Bicchi, A. Settimi, A. Rocchi, and A. Ajoudani, "Walk-man: A high-performance humanoid platform for realistic environments," *Journal of Field Robotics*, 2017. [Online]. Available: http://dx.doi.org/10.1002/rob.21702

[32] A. Rocchi, E. Hoffman, D. Caldwell, and N. Tsagarakis, "OpenSoT: A Whole-Body Control Library for the Compliant Humanoid Robot COMAN," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6248–6253.