Detecting Object Affordances with Convolutional Neural Networks

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

***Abstract*— We present a novel and real-time method to detect object affordances from RGB-D images. Our method trains a deep Convolutional Neural Network (CNN) to learn deep features from the input data in an end-to-end manner. The CNN has an encoder-decoder architecture in order to obtain smooth label predictions. The input data are represented as multiple modalities to let the network learn the features more effectively. Our method sets a new benchmark on detecting object affordances, improving the accuracy by 20% in comparison with the state-of-the-art methods that use hand-designed geometric features. Furthermore, we apply our detection method on a full-size humanoid robot (WALK-MAN) to demonstrate that the robot is able to perform grasps after efficiently detecting the object affordances.**

I. INTRODUCTION

Humans have an astonishing capability to detect object affordances using vision [1], as well as to using this information to complete daily tasks such as picking up objects. This capability has been studied by multiple disciplines such as neuroscience and cognitive robotics. For instance, there is neuroscientific evidence [2] which suggests that humans can easily determine, from a priori experience, which is the best way of grasping by selecting the appropriate grasp surface. On the other hand, cognitive robotics such as imitation learning [3] focuses on developing an architecture that allows a robot to learn and reason about affordances and generate complex intelligent behaviors.

In robotics, detecting object affordances is an essential capability that allows a robot to understand and autonomously interact with objects in the environment. Most of the prior works on affordances detection have focused on grasp detection using RGB-D images [4] or point cloud [5] data. While these methods can lead to successful grasping actions, their failures in detecting other types of object affordances prevents robots from completing real world human-like tasks, such as using a tool. Man-made objects usually have many parts, where each one has its own functionality. Thus, an object may have more than one affordance (e.g. a knife usually has two affordances, one for cutting and another for grasping). Therefore, to achieve a human-like object manipulation, the robot should be able to detect and localize all the affordances in order to choose the right action for a real world scenario.

From the visual perception point of view, however predicting affordances from an image is not a trivial task, because of variations in the shape, orientation, and appearance of the objects in the environment. The problem becomes even harder for cluttered scenes due to occlusions. However, an efficient affordance detection method could enable the robot to interact with an extensive variety objects, including novel unseen ones, given that man-made objects often have a similar set of affordances.

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. 1. Affordance detection and its application. **From left to right:** Our system uses an RGB-D image to detect object affordances. An example of detection results for the `grasp` affordance. A grasp is defined as a rectangular box based on the detected `grasp` affordance.

This paper addresses the problem of learning visual features for affordance detection in RGB-D images as shown in Fig. 1. Our goal, which is similar to the recent state-of-the-art work in [6], is to detect affordances for object parts. However, unlike [6] where hand-designed features are used, we treat this problem as a *pixel-wise labeling* task and use Convolutional Neural Networks (CNN) to learn deep features from RGB-D images. We show that a large CNN can be trained to detect object affordances from rich deep features. Since the detection stage of our method runs in real-time, we apply it to a real robotic grasping application using a full-size humanoid robot (WALK-MAN), and show that by extracting object affordances the robot can successfully perform grasp actions in the environment.

The rest of the paper is organized as follows. We start with a review of the related work in Section II, followed by the description of our methodology in Section III. In Section IV we present our experimental results on an affordance dataset. Then, we describe a grasping method based on the detected affordances and apply it on a real full-size humanoid robot (WALK-MAN) in Section V. Finally, we present the future work and conclude the paper in Section VI.

II. Related Work

The affordance detection problem has been extensively studied in robotics and computer vision over the last few years. Many works have focused on localizing grasp location on objects using vision [7] [8]. In [9] the authors proposed a method to detect grasp affordances by learning a mapping from local visual descriptors to grasp parameters. In [10] a set of the so-called 0-ordered affordances is detected from the full 3D object mesh models. The authors in [11] proposed a method to learn tool affordances by clustering the effects of robot's actions and applied it to a humanoid platform. The work in [12] proposed a method to identify color, shape, material, and name attributes of objects selected in a bounding box from RGB-D data, while in [13] the authors introduced the concept of relational affordances to search for objects in occluded environments.

Deep learning methods have shown impressive results in computer vision. The authors in [14] applied a CNN for image classification, while in [15] a deep CNN and fully connected CRF were combined to segment images. Similarly to [15], the authors in [16] proposed a deep convolutional encoder-decoder architecture for semantic image segmentation. The success of deep learning methods in computer vision, has led recently to an interest in understanding their feature learning capabilities for the grasp detection problem. For example, deep learning has been used in [4] to detect grasp affordances, while a rectangle-based grasp technique was applied for real robotic applications. A similar grasping concept has been also used in [17] for object reorientation.

In [6], the authors used a traditional machine learning approach to detect the affordances of tool parts from RGB-D images. The extracted features were geometrically meaningful and were learned using the classifiers. Additionally, the authors released the RGB-D Affordance dataset, which we subsequently use in this work. The main challenge with this approach is to decide which visual cues should be used as features. Designing features, however, is not a trivial task and all hand-designed ones can only capture low-level information from the data [18]. More recently, the work in [19] used human pose as the context to weekly supervised learn the affordances using a deep CNN. Both approaches in [6] [19] were visually tested over the dataset, but were never applied in a real-world robotic application.

Unlike the work in [6], which focuses on choosing hand-designed features to detect affordances and then generalize this knowledge for novel objects, we focus on detecting affordances from deep features and in this way aim to understand the relationship between each part of the object. Similar to [19], our method uses a deep CNN to automatically learn depth features from the training data, however we use the novel encoder-decoder architecture and remove the fully connected layer in our network to enable real-time inference. Based on the detected affordances, we develop a grasping application to be tested with a full-size humanoid robot, showing that object affordances can be used in real-world robotic applications such as grasping.

III. Affordance Detection

Inspired by the results from the computer vision community, we train a large CNN on RGB-D images to generate rich features for affordance detection. To allow the network to effectively learn the features from the input data, we represent them as multiple modalities. Next, we explain in details the introduced data representation and architecture, as well as the way to train the network.

A. Data Representation

Recently, many works in computer vision and machine learning have investigated the effectiveness of using multiple modalities as inputs to a deep network, such as video and audio [20] or RGB-D data [21]. However, the problem of picking the best combination of these modalities for a new task is still an open problem. Ideally, they should represent important properties of the data so that the network can effectively learn deep features from them.

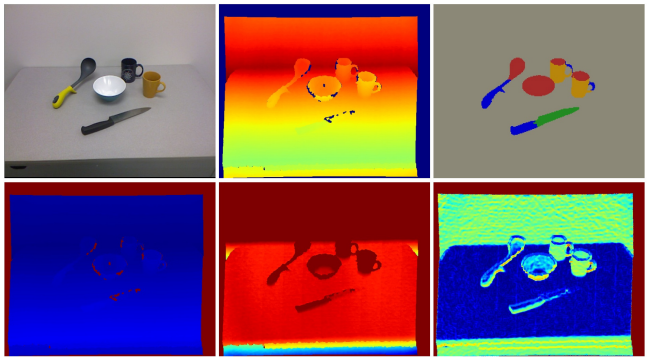

Fig. 2. Data representation. **Top row:** The original RGB image, its depth image, and the ground-truth affordances, respectively. **Bottom row:** The HHA representation of a depth image.

In this paper, we focus on detecting affordances from RGB-D images. Intuitively, we can either use only RGB images or combine both RGB and their associated depth images as the input to our network. In this work we also investigate other ways of data representation that may improve further the performance. In [21], the authors showed that when the training data is limited (which is true in our case since the affordance dataset [6] that we use for training has only 30,000 images, compared to other ones that are deep learning oriented with million of images [14]), it is unlikely that the CNN would automatically learn important depth properties. To deal with this problem a new method [21] was proposed to encode the depth images into three channels at each pixel: the horizontal disparity, the height above the ground, and the angle between each pixel's surface normal and direction of inferred gravity (denote as HHA). The HHA encoder is calculated based on an assumption that the direction of gravity would impose important information about the environment structure. We adapted this representation since the experimental results in [21] have shown that the features can be learned more effectively for object recognition tasks in indoor scenes. We show an example of different data representations for our network in Fig. 2.

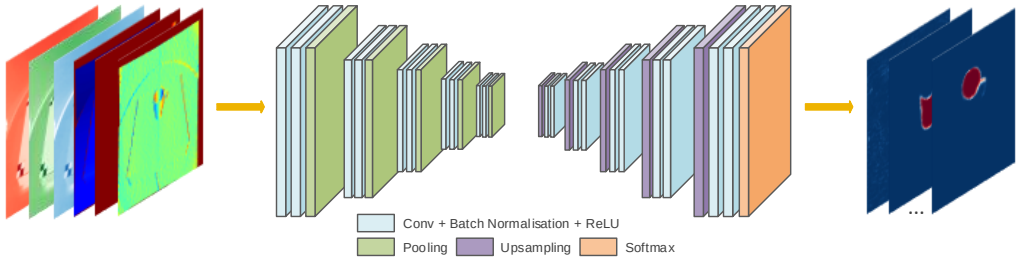

Fig. 3. An illustration of our affordance detection method. **From left to right:** The input data are represented as multiple modalities and learned by a CNN with an encoder-decoder architecture. The CNN produces a k channel image of probabilities, where k is the number of affordance classes. Each channel is visualized as an image in this figure.

B. Architecture

In 2012, the authors of [14] used CNN for classifying RGB images and showed substantially higher accuracy over the state-of-the-art. Many works have applied CNN to different vision problems since then [22] [23]. Nonetheless, the design of a CNN for image segmentation still remains challenging. More recently, the work of [24] proposed an encoder-decoder architecture for pixel-wise image labeling. However, the encoder of this work includes the fully connected layers that make the training very difficult due to a huge amount of parameters (approximately 134M), and also significantly increases the inference time. The authors in [16] pursued the same idea but they discarded the fully connected layers to reduce the number of parameters. They showed that the encoder-decoder architecture without fully connected layers can still be trained end-to-end effectively without sacrificing the performance and enabling real-time reference.

In this paper, we use the state-of-the-art deep convolutional network described in [16]. In particular, the network contains two basic components: the encoder and the decoder network. The encoder network has 13 convolutional layers that were originally designed in the VGG16 network [22] for object classification. Each encoder has one or more convolutional layers that perform batch normalization, ReLU non-linearity, followed by a non-overlapping max-pooling with a 2×2 window to produce a dense feature map. Each decoder layer is associated with an encoder one, ending up in a 13 layers decoder network. In each one, the input feature map is upsampled using the memorized pooled indices and convoled with a trainable filter bank. The final decoder layer produces the high dimensional features that are fed to a multi-class soft-max layer, which classifies each pixel independently. The output of the softmax layer is a k channel image of probabilities, where k is the number of classes.

We adapt the above architecture to detect object affordances at pixel level. Fig. 3 shows an overview of our approach. The data layer is modified to handle multiple modalities as input, while each image in the training set is center cropped on all channels to 240×320 size from its original 480×640 size. In testing step, we don't crop the images but use the sliding window technique to move the detected window over the test images. The final predicted result corresponds to the class with the maximum probability at each pixel over all the sliding windows. Finally, since the dataset that we use has a large variation in the number of pixels for each class in the training set, we weigh the loss differently based on this number.

C. Training

For the training, we generally follow the procedure described in [16] using the Caffe library [25]. Given that the gradient instability in the deep network can stall the learning, the initialization of the network weights is very important. In particular, we initialized the network using the technique described in [23]. The network is end-to-end trained using stochastic gradient descent with a fixed 0.1 learning rate and 0.9 momentum. The cross-entropy loss [26] is used as the objective function for the network. The batch size was set to 10 while the learning rate was initialized to 0.001, and decreased by a factor of 10 every $50,000$ iterations. The network is trained from scratch until convergence with no further reduction in training loss. The training time is approximately 3 days on an NVIDIA Titan X GPU.

IV. Experiments

A. Dataset and Baseline

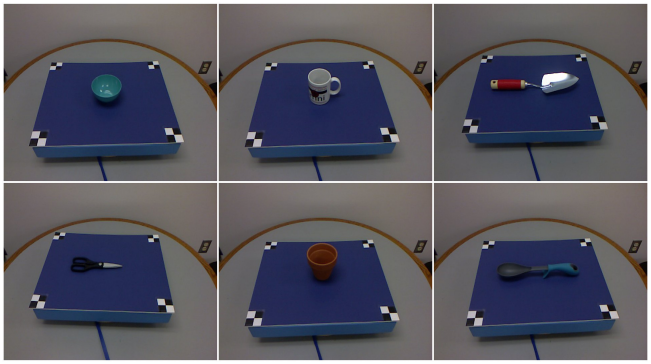

Fig. 4. Example images from the UMD dataset [6].

We used the UMD dataset that was recently introduced in [6] for our experiments. This dataset contains around 30,000 RGB-D image pairs of 105 kitchen, workshop, and garden tools. The tools were collected from 17 different categories, while the ground-truth images are annotated with 7 affordance labels: `contain`, `support`, `cut`, `w-grasp`, `scoop`, `grasp`, and `pound`. Fig. 4 shows some example images from this dataset.

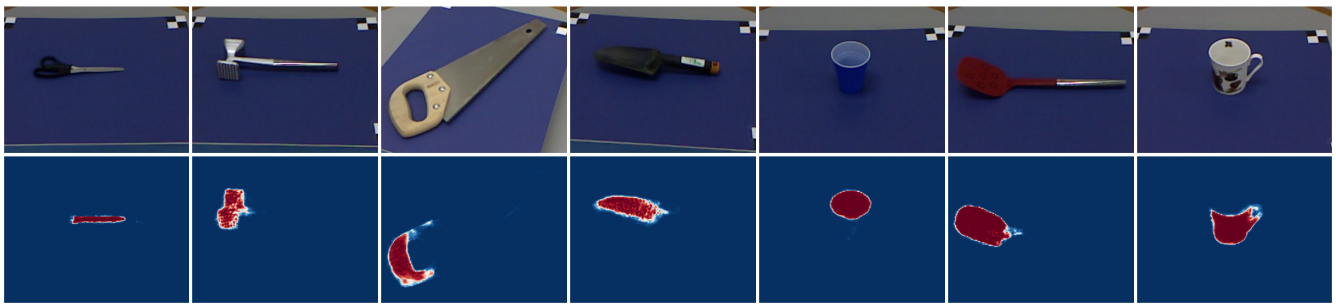

Fig. 5. Detection results on UMD dataset using our CNN-RGBD method. **Top row:** The original input image. **Bottom row:** The prediction results of an object affordance.

TABLE I
PERFORMANCE OVER UMD DATASET

	HMP	SRF	DeepLab	CNN-RGB	CNN-RGBD	CNN-RGBHHA
grasp	0.367	0.314	0.620	**0.719**	0.714	0.673
w-grasp	0.373	0.285	0.730	**0.769**	0.767	0.652
cut	0.415	0.412	0.600	**0.737**	0.723	0.685
contain	0.810	0.635	**0.900**	0.817	0.819	0.716
support	0.643	0.429	0.600	0.780	**0.803**	0.663
scoop	0.524	0.481	**0.800**	0.744	0.757	0.635
pound	0.767	0.666	**0.880**	0.794	0.806	0.701
Average	0.557	0.460	0.733	0.766	**0.770**	0.675

As a comparison, we baseline our approach with the hand-designed features approach combined with Hierarchical Matching Pursuit (HMP) and Structured Random Forests (SRF) classifiers as described in [6]. We followed the training and testing procedures described in this work for a fair comparison. To train our network three different kinds of input data are used. First, we use only the RGB images (CNN-RGB), then we use both the RGB and their corresponding depth images (CNN-RGBD), and last we encode the depth images into the HHA representation and use them with their RGB ones (CNN-RGBHHA) as input to our deep network. To compare with other deep learning methods, we benchmark our method with DeepLab [15]. This method was recently applied by [19] for the affordance detection problem.

B. Evaluation Metric

We evaluate our experimental results using the F_β^w metric. This metric was recently introduced in [27] to extend the well-known F_β^1 measure. The novelty of this measure is that it weighs 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.

C. Results

Table I summarizes the F_β^w results on the UMD dataset for single objects. We notice that the detection results are significantly improved using the deep learning approach compared to the baseline. In particular, our CNN-RGBD achieves the highest average detection accuracy, outperforming the HMP and SRF method by 21.3% and 31.0%, respectively. It demonstrates that our deep network is able to learn deep features from the data and therefore boost the performance significantly over the baseline methods that used hand-designed features. Moreover, we notice that another limitation of the hand-designed features method is that it only performs well with some specific classes, while in others it fails to capture important properties of the data. For instance, the HMP method showed good results for the `contain`, `pound`, and `support` classes ($F_\beta^w = 0.810, 0.767, 0.643$, respectively), while its accuracy was dramatically dropped for the `grasp`, `w-grasp`, and `cut` classes ($F_\beta^w = 0.367, 0.373, 0.415$, respectively). This limitation does not occur in our approach since the deep network learns the features of all classes through its layers independently, and hence there is not much fluctuation in our results. We also achieve the same improvement in cluttered scenes.

Within deep learning methods, while our CNN-RGBD gives the highest accuracy on average, DeepLab also achieves better results in 3 classes. We notice that even though DeepLab combined a fully connected CRF at the final layer of the network, its results seem to be more fluctuated than ours. Surprisingly the CNN-RGB performance is close to that for the CNN-RGBD and outperforms the CNN-RGBHHA. We also notice that even though the CNN-RGBHHA gives reasonable results, it turns out that encoding the depth image to HHA representation doesn't improve the accuracy compared to the original depth one. This is because the HHA encoding process mainly depends on the step that estimates the gravity direction from a single depth image. Due to the nature of the UMD dataset, where all the objects lie on a tabletop, it appears that the introduced algorithm [21] is unable to estimate the gravity direction using only the depth image in many scenes. Therefore, the HHA representation fails to capture important properties from the depth image. Fig. 5 shows some detection results on UMD dataset using our CNN-RGBD method.

To conclude, our approach outperforms the baseline methods and the results show that integrating multimodal information improves the resultant accuracy. The depth information is very useful in challenging scenarios, but at the same time its representation plays a significant role in the performance. Our method is suitable for real-time robotic application since the testing time for an input image is approximately 90 milliseconds on an NVIDIA Titan X GPU.

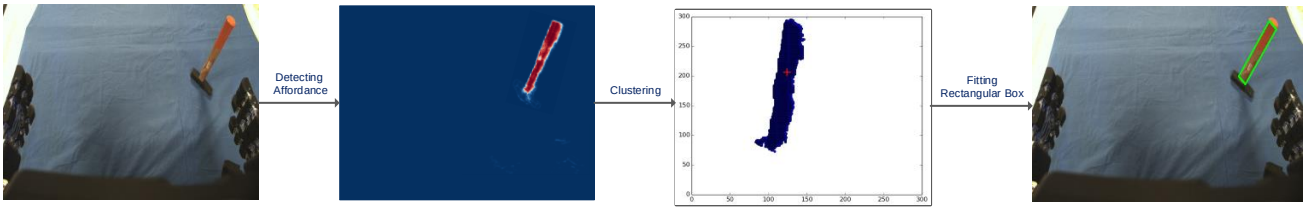

Fig. 6. A pipeline of our grasping method. **From left to right:** The RGB-D image is captured by the vision system of the robot and a CNN-RGBD is used to detect the affordances in the image (e.g. the *grasp* affordance as in the second image). All the points of the detected affordances are then grouped into clusters using the Mean Shift algorithm to eliminate the noisy points. Finally, a grasp is represented as a rectangular box [4] by fitting a minimum rectangle to each cluster.

V. Robotics Application

In order to evaluate the performance of our affordance detection method in real world scenarios, we train our CNN-RGBD network on both the RGB-D Affordance dataset and our additional data, and then run an extensive series of experiments on WALK-MAN, a full-size humanoid robot. We show that object affordances can be used by the robot to perform manipulation tasks such as grasping.

A. Hardware

WALK-MAN [28] is an $1.85m$ high full-size humanoid robot. It has two underactuated hands with five fingers [29] driven by a single motor. The arm has 7DoF and the vision sensing is equipped with a MultiSense-SL camera that can capture point cloud and stereo vision data. The stereo vision system returns 1024×1024 RGB-D images. The YARP middleware framework [30] is used to communicate with the robot while the OpenSoT [31] library is used to plan the whole-body motion. The robot is controlled by a computer with a Core i5 3.2GHz x 4 processor and 12GB RAM.

B. Grasping Objects using Affordances

Since the affordances from the dataset that we used to train our network are manually annotated by human, it provides meaningful information about the functionality of each object part. For example, the `grasp` and `w-grasp` ones indicate the region on an object that usually be grasped by human. Based on the detected affordances, we develop a method that allow the robot to grasp different objects. Fig. 6 shows the details of our framework.

In particular, we use the Mean Shift algorithm [32] to group all the points of the detected map into separated clusters. Mean Shift is a centroid based clustering algorithm that works by updating candidates for centroids to be the mean of the points within a given region and can automatically determine the number of clusters from the input. Given that noisy points may exist in the detected map, we only consider a cluster to be valid if it has more than 100 points. For each cluster, we find its convex hull which is the smallest polygon that encloses all the points where all internal angles are less than $180°$, and fit a minimum rectangular bounding box around the cluster based on this convex hull [33]. From this rectangle, we use the rectangle-based grasp strategy introduced in [4] to find the grasp frame on the object for the end-effector.

Fig. 7. Objects used in our robotics experiments.

TABLE II
Grasp Success Rate (in %)

	Accuracy	Affordance
Bottle	100	`w-grasp`
Comb	95	`grasp`
Cup	75	`contain`
Hammer	100	`grasp`
Headphone	80	`grasp`
Ruler	90	`grasp`
Scissors	100	`grasp`
Saw	90	`grasp`
Turner	100	`support`
Average	**92.2**	

C. Grasping Results

While the detected affordance provides information about the functionality of an object part, its rectangle bounding box provides all the details about the grasp location, orientation, and the physical size of the grasping region on the object. From this information, the robot can easily determine if an object is graspable from its affordances. For the experiments, we selected 9 different objects as shown in Fig. 7. For each one, we perform 20 trials and a grasp is considered successful if the robot can grasp, raise, and hold the object in the air for 15 seconds. Table II summarizes the success rate and the detected affordance that the robot used to grasp for each object. From the results, we can see that affordances usually lead to successful grasps, but with some cases of failure. We notice that the grasping success depends on the physical size of the detected affordances with respect to the robotic hand as well as the geometry of the hand-closing region. For

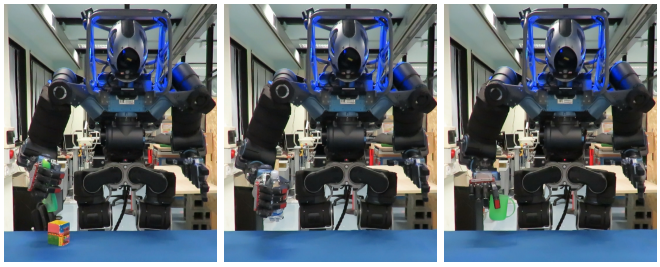

Fig. 8. Example of successful grasps based on the detected affordances.

instance, if all the detected affordances are too big compared with robotic hand limitation, the robot will unable to grasp the object. For each object, the total execution time is around 45 seconds, and the time that is needed to detect the object affordances and fit the rectangular box is approximately 1 second. Fig. 8 shows an example of successful grasps based on the detected affordances. The experimental video with all objects can be found in the following link:

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

VI. Conclusions and Future Work

In this paper, we present a novel method to detect object affordances using a deep convolutional neuron network. We have demonstrated that a large deep network can significantly improve the detection results compared to the state-of-the-art methods. Moreover, we have tested our method on grasping experiments with a full-size humanoid robot. Using our method, the inference procedure is real-time and the robot is able to perform grasping tasks using the detected affordances.

Currently, our grasping method based on the object affordances is limited to surfaces that fit in the robotic hand. We aim to develop a more general approach to overcome this limitation. Another interesting problem is to study the semantic relationship between object affordances that enables the completion of more types of tasks. Finally, we plan to release our new affordance dataset that has more challenging scenes and covers more types of 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. Gibson, *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin, 1979.

[2] E. Oztop, M. Kawato, and M. Arbib, "Mirror Neurons and Imitation: A Computationally Guided Review," *Neural Networks*, vol. 19, 2006.

[3] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning Object Affordances: From Sensory–Motor Coordination to Imitation," *IEEE Transactions on Robotics*, 2008.

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

[5] I. Gori, U. Pattacini, V. Tikhanoff, and G. Metta, "Ranking the Good Points: A Comprehensive Method for Humanoid Robots to Grasp Unknown Objects," in *ICAR*, 2013.

[6] 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.

[7] J. Bohg and D. Kragic, "Grasping Familiar Objects using Shape Context," in *Int. Conf. on Advanced Robotics (ICAR)*, 2009.

[8] A. ten Pas and R. Platt, "Using Geometry to Detect Grasp Poses in 3D Point Clouds," in *Int. Symp. on Robotics Research (ISRR)*, 2015.

[9] L. Montesano and M. Lopes, "Learning Grasping Affordances from Local Visual Descriptors," in *IEEE 8th International Conference on Development and Learning*, June 2009.

[10] A. Aldoma, F. Tombari, and M. Vincze, "Supervised Learning of Hidden and Non-Hidden 0-Order Affordances and Detection in Real Scenes," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[11] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, "Self-supervised Learning of Grasp Dependent Tool Affordances on the iCub Humanoid Robot," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[12] Y. Sun, L. Bo, and D. Fox, "Attribute Based Object Identification," in *IEEE ICRA*, May 2013, pp. 2096–2103.

[13] B. Moldovan and L. De Raedt, "Occluded Object Search by Relational Affordances," in *IEEE ICRA*, May 2014, pp. 169–174.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, 2012.

[15] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," *CoRR*, vol. abs/1412.7062, 2014.

[16] 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.

[17] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Preparatory Object Reorientation for Task-Oriented Grasping," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[18] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning Hierarchical Features for Scene Labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, 2013.

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

[20] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal Deep Learning," in *ICML*, 2011.

[21] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik, "Learning Rich Features from RGB-D Images for Object Detection and Segmentation," in *European Conference on Computer Vision (ECCV)*, 2014.

[22] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *CoRR*, vol. abs/1502.01852, 2015.

[24] H. Noh, S. Hong, and B. Han, "Learning Deconvolution Network for Semantic Segmentation," *arXiv preprint arXiv:1505.04366*, 2015.

[25] Y. Jia *et al.*, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *22nd ACM Int. Conf. on Multimedia*, 2014.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *CoRR*, vol. abs/1411.4038, 2014.

[27] 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.

[28] N. G. Tsagarakis *et al.*, "WALK-MAN: A High Performance Humanoid Platform for Realistic Environments," *JFR*, 2016.

[29] M. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, and A. Bicchi, "Adaptive Synergies for a Humanoid Robot Hand," in *International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 7–14.

[30] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: Yet Another Robot Platform," *International Journal on Advanced Robotics Systems*, 2006.

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

[32] D. Comaniciu and P. Meer, "Mean Shift: a Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 24, no. 5, pp. 603–619, May 2002.

[33] H. Freeman and R. Shapira, "Determining the Minimum-area Encasing Rectangle for an Arbitrary Closed Curve," *Commun. ACM*, vol. 18, no. 7, pp. 409–413, July 1975.