Semi-automatic Training Data Generation for Semantic Segmentation using 6DoF Pose Estimation

Shuichi Akizuki[1,2] and Manabu Hashimoto[2]
[1]*Department of Electrical Engineering, Keio University, Hiyoshi, Yokohama, Japan*
[2]*Department of Engineering, Chukyo University, Nagoya, Aichi, Japan*

Keywords: Dataset Generation, Semantic Segmentation, Affordance.

Abstract: In this research, we propose a method using a low cost process to generate large volumes of real images as training data for semantic segmentation. The method first estimates the six-degree-of-freedom (6DoF) pose for objects in images obtained using an RGB-D sensor, and then maps labels that have been pre-assigned to 3D models onto the images. It also captures additional input images while the camera is moving, and is able to map labels to these other input images based on the relative motion of the viewpoint. This method has made it possible to obtain large volumes of ground truth data for real images. The proposed method has been used to create a new publicity available dataset for affordance segmentation, called the NEDO Part-Affordance Dataset v1, which has been used to benchmark some typical semantic segmentation algorithms.

1 INTRODUCTION

Achieving use of tools by robotic arms is one of important goal in the field of intelligent robotics. It requires: 1) Developing a gripper with which to grasp an object, 2) Generating grasping motions required to use the grasped object, and 3) Estimating of the type of actions that are supported by an object of that type (this is called affordance of the object). Of these sub-tasks, estimating affordance needs to be processed first, so it is particularly important, and has been an active research topic in the past several years.

Generally, affordance estimation is considered to be a semantic segmentation problem, of assigning pre-defined affordance labels to each pixel of an object in an image. However, different parts of everyday objects have different affordance types, so it has been extremely expensive to create correct data for training. For example, the handle of a hammer would have an affordance of Grasp because it can be held in the hand, while the head would have an affordance of Pound because it can be used to hit other objects. Further, more than 10k training images taken from multiple viewpoints are needed to create a high-performance classifier, so it is not practical to apply these labels manually.

In this research, we propose a method involving a low-cost procedure to generate large volumes of images that are suitable for training a classifier to solve semantic segmentation problems. Specifically, affordance labels are pre-assigned to 3D models of the objects to be recognized, and these labels are mapped onto images by estimating the 6DoF pose of these objects in images captured using an RGB-D sensor. We also capture additional input images by moving the camera, and map the label data onto these other images based on the relative motion of the viewpoints. The approach of the proposed method differs from the artificial operations referred to as data augmentation, in that it provides large volumes of training images with associated ground truth from real measurement data. The contributions of this research are as follows:

- We propose a method for generating training images annotated with pixel-wise labels, using a semi-automatic procedure with 6DoF pose estimation.
- We propose a new dataset for estimating the affordance of object parts, created using the proposed algorithm. The dataset can be downloaded from our website [1].
- We use this dataset to benchmark some prominent semantic segmentation algorithms.

In section 2, we discuss recent trends regarding affordance estimation and semantic segmentation methods. In section 3 we discuss a dataset for affordance

[1] http://isl.sist.chukyo-u.ac.jp/archives/nedopro/

Akizuki, S. and Hashimoto, M.
Semi-automatic Training Data Generation for Semantic Segmentation using 6DoF Pose Estimation.
DOI: 10.5220/0007568706070613
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 607-613
ISBN: 978-989-758-354-4

segmentation that we have created, called the NEDO Part Affordance Dataset v1. In section 4, we describe the low-cost ground-truth annotation procedure using 6DoF pose estimation, which we proposed for creating this dataset. In section 5, we describe benchmark tests done on de-facto standard semantic segmentation methods using the NEDO Part Affordance Dataset v1, and in section 6, we summarize the results of this research.

2 RELATED WORK

2.1 Affordance Estimation

Various earlier methods have been proposed for estimating affordance of objects appearing in a scene.

One approach is to model human poses for affordance, and to evaluate their consistency with real scenes. For example, one procedure for handling the sittable affordance is to prepare a human model in sitting poses, and then compare it with input scenes (Grabner et al., 2011). This approach is also closely related to Robotic grasping, and a method proposed for estimating graspability by comparing hand state with measurement data (Domae et al., 2014)

Recently, the main approach has been to train for the correspondence between affordance and local features extracted from the input scene to identify multiple affordance values. Affordance estimation is being solved using so-called multi-class classifiers. Myers et al. proposed a method that estimates seven affordances (Grasp, Cut, Scoop, Contain, Pound, Support, Wrap-grasp) for everyday objects at the pixel level. The method trains for so-called hand-crafted local features such as depth, color, and curvature, which are obtained from depth images of the everyday objects(Myers et al., 2015).

Prompted by efforts such as SegNet(Badrinarayanan et al., 2017), which have had success using Deep Learning (DL) for semantic segmentation, there has also been active research applying DL to estimation of affordances. Nguyen et al. proposed a method to estimate the affordance defined by Myers et al. using an encoder-decoder network, which demonstrated superiority for hand-crafted features(Nguyen et al., 2016). Another method that performs object-detection as a prior step, and then uses a network to estimate object class and affordance for each detected object region, has also been proposed(Do et al., 2018). Other methods have also been proposed, implementing affordance segmentation by taking RGB input images and using networks to estimate mid-level features, depth information, normal-vectors and object classes(Roy and Todorovic, 2016).

2.2 Semantic Segmentation

Semantic segmentation is the task of estimating the class to which each pixel of an input image belongs.

Fully Convolutional Networks(Long et al., 2015) implement the semantic segmentation task by using convolutional layers that output 2D heatmaps to replace the fully-connected layers of classification networks used in methods like VGG(Simonyan and Zisserman, 2014) and AlexNets(Krizhevsky et al., 2012). SegNet(Badrinarayanan et al., 2017) is an encoder-decoder network. The decoder is able to discriminate accurately by gradually increasing resolution. U-Net is able to reflect segmentation results on detailed shapes in the input image by concatenating the feature maps of each encoder with the decoder feature maps(Ronneberger et al., 2015).

3 PROPOSED DATASET

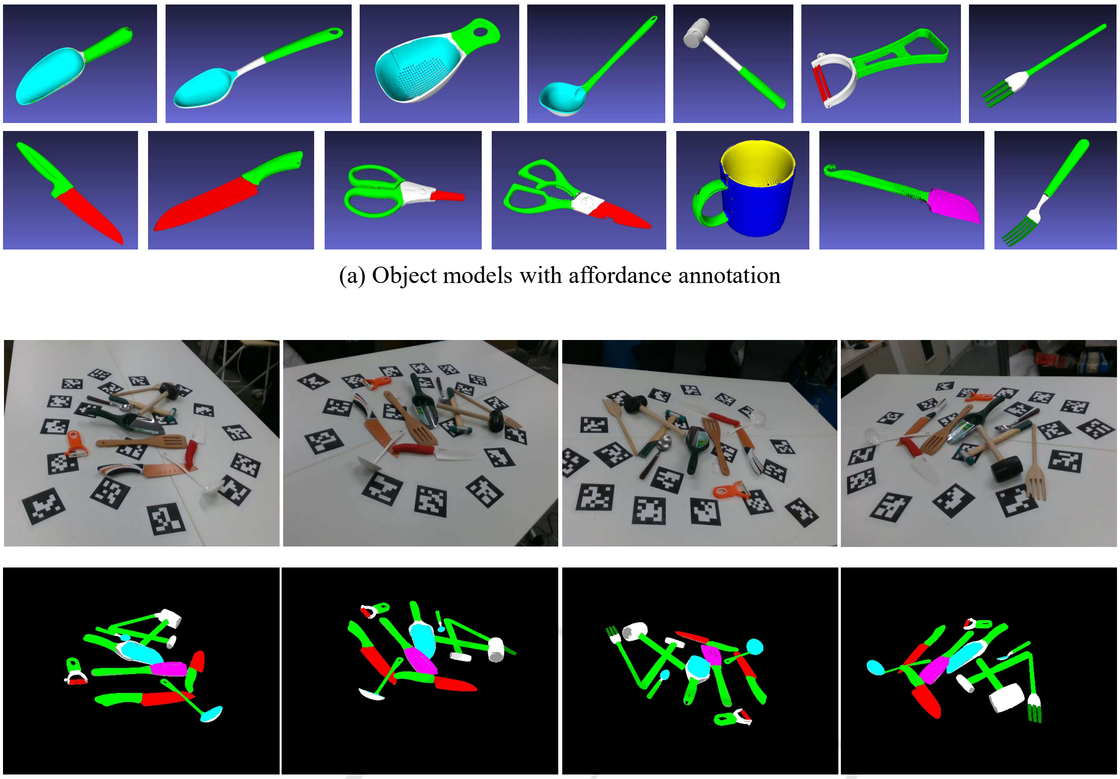
3.1 Dataset Overview

We now describe the NEDO Part Affordance Dataset v1 proposed for affordance segmentation in this research. The dataset is composed of the following data.

- 3D models created from measurements of real everyday objects with point-wise annotation.
- Multiple RGB-D images of these everyday objects.
- Ground-truth images with pixel-wise annotation, corresponding to the RGB-D images.

Examples of data in the NEDO Part Affordance Dataset v1 are shown in Figure 1. Examples of object models are shown in Figure 1(a). Object models are composed of meshes with affordance labels on each vertex. The colors on the 3D models indicate affordance labels as defined in Table 2. All objects in the dataset were measured using a precise 3D sensor, and modeled with actual-size dimensions. The affordance labels are defined in Table 2. These include labels defined by Myers et al. with the additional labels, Stick, None, and Background. A total of 74 3D models of 10 types of kitchen and DIY tools were measured. A breakdown of the objects is given in Table 1.

Figure 1(b) gives examples of input images and ground truth. A total of 10125 images were prepared, which were divided into 8706 training set and 1419



(a) Object models with affordance annotation

(b) Input scene and corresponding ground truth

Figure 1: Examples of object models in the dataset and scene images. (a) Object models. Colors indicate the affordance labels. (b) Scene images and corresponding ground truth.

Table 1: Types and number object models in the dataset.

Kitchen tools	Fork(9), Knife(8), Ladle(6), Mug(4), Peeler(10), Spatula(11), Spoon(8)
DIY tools	Hammer(4), Scissors(9), Shovel(5)

Table 2: Affordance labels and its label colors.

Affordance	Functionality	Color
Contain	With deep cavities to hold liquid.	(255,255,0)
Cut	Used for separating another object.	(255,0,0)
Grasp	Can be enclosed by a hand.	(0,255,0)
Pound	Used for striking other objects.	(160, 160, 160)
Scoop	A curved surface with a mouth for gathering soft material.	(0, 255, 255)
Stick	Sharp parts that can be pushed into something.	(0,150,0)
Support	Flat parts that can hold loose material.	(255,0,255)
Wrap-grasp	Can be held with the hand and palm.	(0,0,255)
None	Other region on objects	(255,255,255)
Background	Other region outside of objects.	(0,0,0)

testing set. Each pixel of the ground truth images indicates an affordance label. Each scene contains 5 to 11 (avg. 7.97) of the objects, selected and arranged at random.

3.2 Data Acquisition

The object models consist of mesh data measured from a full perimeter. A Solutionix C500 3D scanner from Medit Inc. was used for measurements, and mesh data comprised roughly 500,000 vertices for each object. For the scene data, objects were arranged on a table with several Augmented Reality (AR) markers, and for each arrangement, approximately 250 to 300 frames of RGB and depth images with resolution of 640x480 pixels were recorded. An Intel D415 3D sensor was used for these images. The AR markers were used for semi-automatic annotation, as described in the following section.

4 SEMI-AUTOMATIC ANNOTATION

4.1 Algorithm

We use a semi-automatic process to generate ground truth images with pixel-wise annotation for input images $I_i, i = \{1, \ldots n\}$, taken from multiple viewpoints. i represents the viewpoint index of the captured image. The algorithm consists of three steps, as shown in Figure 2.

1. 6DoF pose annotation

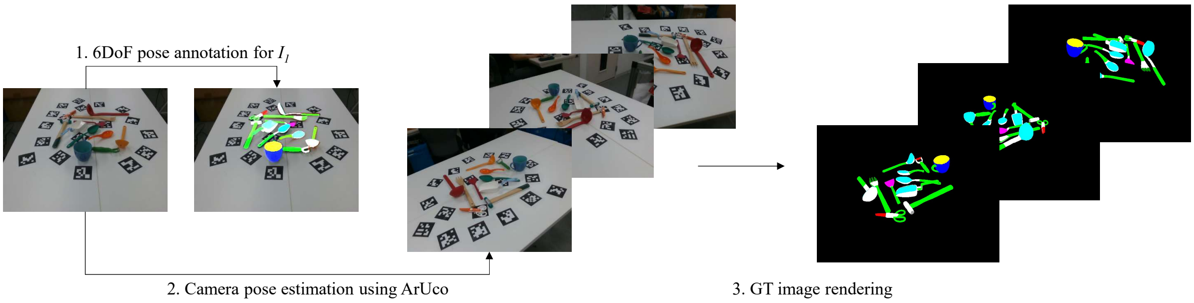

1. 6DoF pose annotation for I_1
2. Camera pose estimation using ArUco
3. GT image rendering

Figure 2: Ground truth data generation using semi-automatic process.

2. Camera pose estimation using ArUco(Garrido-Jurado et al., 2014)
3. Groud truth image rendering

In Step 1, the pose of each object in the first frame is computed using specially developed annotation software. In Step 2, the ArUco library, which is able to compute the pose of the AR markers accurately, is used to compute changes in the camera pose between the first frame and the other frames. In Step 3, the ground truth images are rendered based on the pose data computed in Steps 1 and 2. Details of each step are given below.

4.2 6DoF Pose Annotation

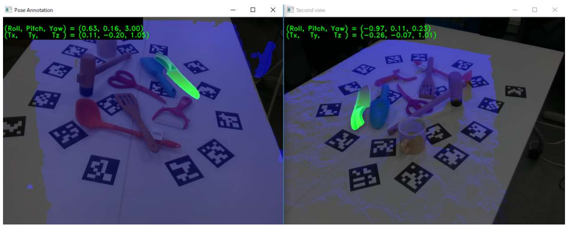
This module performs annotation of the 6DoF pose $\mathbf{T}_j^M = [\mathbf{R}, \mathbf{t}; \mathbf{0}, 1]$, for each object $M_j, j = \{1, ..., k\}$ in the first frame of input image I_1. $\mathbf{R}$ is a 3×3 rotation matrix, and $\mathbf{t}$ is a 3×1 translation vector.

We developed dedicated software for 6DoF pose annotation for this research. A screen shot of the software is shown in Figure 3. The annotation software allows object models to be transformed to a suitable pose using keyboard and mouse operations. The image on the left shows object model (Knife) overlaid on the image I_1 in green. Depth data for I_1 is also overlaid in blue. The image on the right is an image taken from a different viewpoint, also input when the software was started, with transformation applied to the object model synchronized with the image on the left. Accurate annotation can be done by checking the fit of the object model on both images. The software provides commands for the following operations.

Step Move: Provides fine adjustment of the pose in the x, y, and z directions and in the roll, pitch and yaw angles using keyboard input.

Direct Move: Moves the object to the position clicked on the image by translation.

ICP: Estimates the pose of the object model using the ICP algorithm (Besl and McKay, 1992), using the model's current position as the initial position.



```
(Roll, Pitch, Yaw) = (0.63, 0.16, 3.00)
(Tx,   Ty,    Tz ) = (0.11, −0.20, 1.05)
```

```
(Roll, Pitch, Yaw) = (−0.97, 0.11, 0.23)
(Tx,   Ty,    Tz ) = (−0.26, −0.07, 1.01)
```

Figure 3: Screen shot of the 6DoF pose annotator. The object model is rendered in green over images from two viewpoints. Accurate annotation can be done by checking over RGB images from two different viewpoints.

Direct move can be used to move the object model to near the position of the object in the scene, and then ICP can be used to compute the pose accurately. Occlusions can occur due to the placement of objects, so there are cases when the pose cannot be estimated using ICP. Step move is used for fine tuning in such cases. These operations are used to compute the 6DoF pose $\mathbf{T}_j^M, j = \{1, ..., k\}$ for each object in I_1.

4.3 Camera Pose Estimation

This module computes the transformation matrix $\mathbf{T}_i^C$ from the camera position for capturing I_1 to the camera position for capturing I_i. ArUco is first used to compute the 6DoF poses $\mathbf{T}_{i,j}$ of the AR markers in each image. j represents the marker ID. Then, the transformation matrix is computed as $\mathbf{T}_i^C = avg(\mathbf{T}_{1,j}\mathbf{T}_{i,j}^{-1})$. $avg()$ computes an average of the transformations. $\mathbf{T}_i^C$ is detected for several markers. These are converted to x, y, z, roll, pitch and yaw values, and the average of each is computed before converting back to a matrix. In this way, a highly accurate transformation matrix $\mathbf{T}_i^C$ is obtained.

It is also conceivable to find the object model 6DoF poses and annotate the first frame without using AR markers, by use tracking methods, but there would be a high risk of failure because occlusions and camera position can hinder object observations. As such, we decided to use AR markers, which enable transformations between camera poses to be computed accurately.

4.4 GT Image Rendering

The ground truth image G_i is obtained by transforming the object model M_j and mapping it onto image I_i with viewpoint i. If the 3D vertices of object model M_j are $\mathbf{x}$, then their position from viewpoint i is $\mathbf{x}^{'} = \mathbf{T}_i^C \mathbf{T}_j^M \mathbf{x}$. This computation is applied to all vertices of the models of all objects appearing in the scene. Then, each $\mathbf{x}^{'}$ is projected onto the image in order of decreasing z coordinate, which is from farthest to nearest to the camera. This produces an image that represents the front-to-back relationships between objects correctly. G_i is generated by setting the pixel values to the pre-assigned affordance labels on the objects.

5 EXPERIMENTS

5.1 Annotation Cost

This section discusses the annotation cost when using the proposed method. With the proposed method, the pose of each object in a scene is annotated using dedicated software for only the first frame in a set of RGB-D images captured from that scene. After this annotation is applied, pixel-wise annotation is applied automatically to all other frames.

6DoF pose estimation for a single scene using the software required approximately 15 to 20 minutes for each scene. This produced ground truth images for the multi-viewpoint images from more than 300 frames. In contrast, annotation work for general semantic segmentation requires roughly 10 minutes per image, so using the proposed method dramatically reduces the annotation cost relative to earlier methods.

5.2 Benchmarking of Semantic Segmentation Methods

We used the proposed NEDO Part Affordance Dataset v1 to benchmark semantic segmentation methods. The following algorithms were used for comparison.

1. Fully Convolutional Networks (FCN-8s) (Long et al., 2015)
2. SegNetBasic(Badrinarayanan et al., 2017)
3. U-net (Ronneberger et al., 2015)

The central 320 × 240 pixel part of each image was cropped and used for training. For FCN-8s, we used the VGG16 pre-trained model for the Encoder part, so we cropped to a size of 320 × 256 pixels to fit the pre-set input size for training.

Each network was trained using Adam gradient descent optimization(Kingma and Ba, 2014). The learning rate was changed between $10^{-2}, 10^{-3}, 10^{-4}$, and 10^{-5} to learn the optimal model parameters. As a result of our experiments, we achieved the good result of 10^{-4} for all models using our data set. With a batch size of 10, we repeated for 100 epochs.

To evaluate recognition performance, we used intersection over union (IoU), which is widely used as an index for segmentation tasks. The IoU for each affordance label is shown in Table 3, and examples of recognition results are shown in Figure 4. FCN-8s had the best average performance. Considering labels individually, the IoU was relatively low for None compared to other labels. This may be because the None label is assigned to parts that have no function as an object, so its shape varies widely, and the label gets assigned to shapes that are difficult to distinguish from RGB images, such as the back part and front part of the spoon. In fact, in the fourth image from the right in Figure 4, the green spoon is placed facing down, so the correct label for the visible surface is None, but all methods estimated it to be Scoop.

The right-most image in Figure 4 contains a scissors that has not provided ground truth image due to lack of 3D object model. Each of the methods were able to detect the scissors, so it seems that some generality has been achieved for variations in object shapes. This suggests that the proposed data set contains enough diversity for use in training for affordance of everyday objects.

6 CONCLUSION

This research has proposed a method for generating large volumes of training images for semantic segmentation using a low-cost procedure. The proposed method estimates the 6DoF pose of labeled 3D models and maps them onto input images to generate ground truth images. Then, by moving an RGB-D camera to capture many images and estimating the relative camera motion, the results of annotating the first frame are mapped onto all of the images. This procedure made it possible to obtain a large number of pixel-wise annotated real images, semi-automatically. We created the NEDO Part Affordance Dataset v1, containing 74 detailed 3D models with affordance labels and a set of more than 10,000 annotated images. This dataset was used to benchmark some prominent semantic segmentation algorithms. The pro-

Table 3: IoU score of each affordance label.

	Contain	Cut	Grasp	Pound	Scoop	Stick	Support	Wrap-grasp	None	BG	Mean
FCN-8s	**0.747**	**0.390**	**0.441**	**0.506**	**0.647**	**0.500**	**0.658**	**0.715**	**0.386**	**0.864**	**0.585**
SegNetBasic	0.676	0.318	0.404	0.338	0.558	0.373	0.501	0.622	0.306	0.802	0.490
U-Net	0.744	0.298	0.411	0.139	0.620	0.457	0.446	0.704	0.323	0.852	0.499

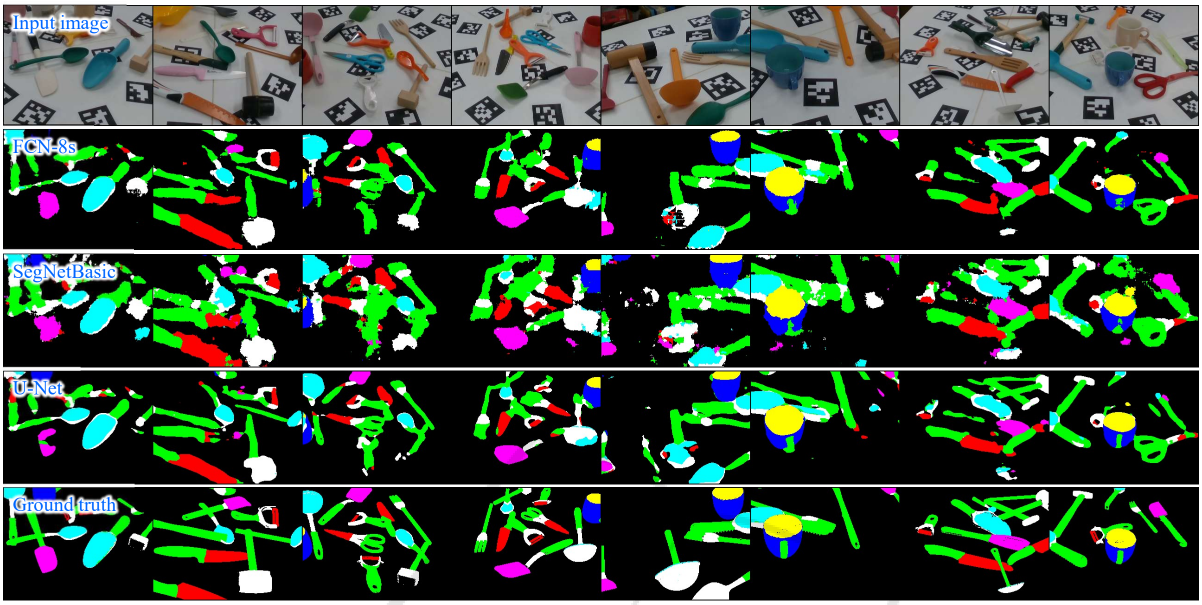


Figure 4: Examples of affordance segmentation for each method. From top to bottom, input images, FCN-8sSegNetBasicU-Netand Ground truth.

posed method can also be used to obtain 6DoF poses for each viewpoint and instance segmentation results. In the future, we plan to add to the annotation data provided by this dataset, so that it can be used to test other image recognition tasks.

ACKNOWLEDGEMENTS

This work was partially supported by the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Besl, P. and McKay, N. (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256.

Do, T.-T., Nguyen, A., and Reid, I. (2018). Affordancenet: An end-to-end deep learning approach for object affordance detection. In *International Conference on Robotics and Automation (ICRA)*.

Domae, Y., Okuda, H., Taguchi, Y., Sumi, K., and Hirai, T. (2014). Fast graspability evaluation on single depth maps for bin picking with general grippers. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1997–2004. IEEE.

Garrido-Jurado, S., Muñoz Salinas, R., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292.

Grabner, H., Gall, J., and Van Gool, L. (2011). What makes a chair a chair? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1529–1536. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

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

Nguyen, A., Kanoulas, D., Caldwell, D. G., and Tsagarakis, N. G. (2016). Detecting object affordances with convolutional neural networks. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2765–2770. IEEE.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Roy, A. and Todorovic, S. (2016). A multi-scale cnn for affordance segmentation in rgb images. In *European Conference on Computer Vision*, pages 186–201. Springer.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.