Proceedings of the Joint Austrian Computer Vision and Robotics Workshop 2020 DOI: 10.3217/978-3-85125-752-6-29

Grasping Point Prediction in Cluttered Environment using Automatically Labeled Data

Stefan Ainetter, Friedrich Fraundorfer
Graz University of Technology
{stefan.ainetter,fraundorfer}icg.tugraz.at

Abstract. *We propose a method to automatically generate high quality ground truth annotations for grasping point prediction and show the usefulness of these annotations by training a deep neural network to predict grasping candidates for objects in a cluttered environment. First, we acquire sequences of RGBD images of a real world picking scenario and leverage the sequential depth information to extract labels for grasping point prediction. Afterwards, we train a deep neural network to predict grasping points, establishing a fully automatic pipeline from acquiring data to a trained network without the need of human annotators. We show in our experiments that our network trained with automatically generated labels delivers high quality results for predicting grasping candidates, on par with a trained network which uses human annotated data. This work lowers the cost/complexity of creating specific datasets for grasping and makes it easy to expand the existing dataset without additional effort.*

1. Introduction

Automated grasping is a very active field of research in robotics. The process of having a robot manipulator successfully grasp objects in a cluttered environment is still a challenging problem. Recent state-of-the-art for grasping position computation often use deep learning techniques and supervised learning. However, these methods usually need to be trained on a large amount of labeled data. Therefore, it is of high interest to find techniques to automatically label data for robotic grasping. Previous work [17, 19] focused on using raw RGBD data for automatic object segmentation by leveraging sequential depth information from the scene. However, the segmentation mask is not sufficient as annotation for grasping point prediction because many state-of-the art approaches define the grasping proposal using a bounding box representation.

We propose a fully automatic pipeline from raw RGBD data to a system that predicts grasping point candidates using our automatically labeled data for training. Figure 1 shows our workflow. As practical example, we captured RGBD data from log ordering in the wood industry. We will demonstrate the usefulness of our approach by training a deep neural network to predict grasping points using our automatically generated labels as ground truth. The main contributions of this work are:

1. A fully automatic annotation pipeline for grasping point prediction using sequential RGBD data.

2. An automatic annotation method that allows dense labeling of grasping points for graspable objects. Additionally, the annotations contain implicit information about the order of object removal due to the usage of sequential input data. These labels can be directly used for training a supervised learning approach.

3. A deep neural network which is able to predict grasping points in a cluttered environment, solely trained with a small number of automatically labeled images.

2. Related Work

Grasping point detection. The conventional method for grasping point detection uses information about object geometry, physics models and force analytics [1]. With the rise of deep learning, data-driven methods [2] became more common. Methods like [13, 9, 7, 20] use deep neural networks and supervised learning to predict multiple grasping points for a single object. Chu et al. [4] were able

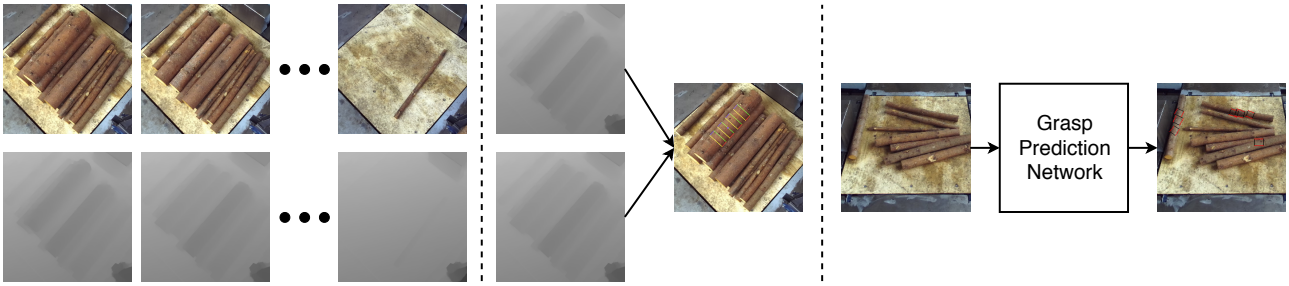

Figure 1. Overall workflow of our method containing data acquisition, automatic grasping point annotation using depth images and training a deep network for grasping point prediction. (**Left**) Our dataset is constructed by recording sequences of RGBD images while a human expert removes wooden logs from the scene. (**Middle**) The sequence of captured depth images is used to automatically annotate grasping points in every corresponding RGB image. (**Right**) This automatically annotated data are then used to train a deep neural network to predict grasping points.

to predict multiple grasping points for multiple objects in an image. Zeng et al. [18] showed that they are able to grasp unseen objects with their winning contribution for the Amazon Robotics Challenge in 2017. Other approaches [12, 10] use Reinforcement Learning (RL) on a real or simulated robot to perform thousands of grasp attempts and use the feedback to improve the grasping point predictions. RL has the advantage that no labeled data are necessary for training, but it is on the other hand very time and hardware consuming.

Representations of grasping points in 2D. Saxena et al. [16] described a grasping point as $g = \{x, y\}$, where x and y define the center of the grasping point proposal. This representation lacks information about the opening width of the gripper. Redmon and Angelova [13] overcame this limitation by using a rectangular representation for the grasping point. This is very similar to the bounding box representation of objects in the field of object detection, with the addition of a rotation angle θ which describes the orientation of the bounding box. An overview about other common representations can be found in [3].

Automatic label generation. Datasets used for deep learning are often hand annotated, which is time consuming and can be error prone due to the involvement of human annotators. In the domain of object segmentation, modern tools like DeepExtremeCut [11] or GrapCut [15] significantly reduce the amount of work for labeling RGB data to a small number of clicks. However, they are not fully automatic and are not able to work with depth data. Zeng et al. [19] showed that they are able to use background subtraction to generate segmentation masks of new objects in the scene. Suchi et al. [17], most similar to our approach, use sequences of depth images to predict segmentation masks of the objects in the scene. However, the difference of our method compared to all previously mentioned approaches is that we do not only calculate the segmentation mask, but directly infer grasping proposals. Furthermore, segmentation masks do not give any information in which order the objects should be removed, which can be crucial for grasp success in cluttered environment.

3. Data Acquisition and Automatic Annotation

This section describes our simple strategy to automatically label grasping points for scenes with objects in a cluttered environment.

3.1. Data Acquisition Protocol

The process requires a statically mounted RGBD camera which records color and depth information from the scene. We then ask human experts to remove one object after the other from the scene. After each successful grasp, we capture depth and color images. Figure 2 shows a sequence of recorded RGB images. This method provides us not only with consecutive RGBD images of the picking procedure, but also gives implicit information about the optimal order of object removal according to a human expert. This information is highly important because not all objects are equally easy to grasp due to their random placement (e.g. objects on top of one another).

3.2. Automatic Label Generation

As illustrated in Figure 3, we perform automatic grasping point annotation through an 3-stage pipeline. Our algorithm takes two consecutive depth images from the scene as input and calculates grasp proposals for the object which was removed. A grasp

Figure 2. Sequence of recorded RGB images. The sequence starts in the top left with the full stack of objects and we record an RGB image after each object removal. We also record the corresponding depth image for every RGB frame.

proposal $\mathbf{g}$ is defined as

$$\mathbf{g} = \{x, y, \theta, w, h\}, \tag{1}$$

where x and y describe the center of the grasp proposal, θ describes the angle of the rotated bounding box, and w and h describe the width and height of the predicted box.

Initial depth segmentation. The main focus of our algorithm is to detect depth changes in the scene after a successful grasp was performed by a human expert. Therefore, we calculate the depth difference I^* of two consecutive depth images as

$$I^* = |I_1 - I_2|, \tag{2}$$

where I_1 and I_2 are the depth images previously normalized between 0 to 255. The output I^* is a rough estimate of the segmentation mask of the removed object.

Segmentation mask refinement. The intermediate segmentation is coarse and contains noise mainly due to inaccurate sensor values and small movements of the objects. Therefore, further refinement of the segmentation mask is needed. We apply binary image morphology to remove the majority of noise and smooth the mask edges. A Gaussian filter is then applied for further noise reduction and to create the refined mask which is used for further processing. The Gaussian filter g_{filter} is defined as

$$g_{filter}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3}$$

where x and y are the spatial dimensions of the intermediate mask I^*, and σ is defined as the standard deviation for the Gaussian kernel. In our experiments, we set $\sigma = 1$, which means that it is equal for both axes.

Automatic grasping point annotation. The refined segmentation mask is then used to calculate geometric features of the object. The skeleton of the object mask is calculated by using [8] to remove boarder pixels as long as the connectivity does not break. The resulting skeleton of the object is approximated with a line segment, which makes it more robust to outliers. Each point on this line segment can then be used as a possible center of a grasp proposal. The height h and the rotation angle θ of a grasp proposal is determined by calculating the intersection between a line, which is normal to the skeleton and passes through the center of a grasp proposal, and the edges of the mask. The bounding box width w is directly dependent on the used gripper and we set this parameter manually to suit our robotic gripper. All this information are then combined and used to generate the final grasping proposals. The proposals have certain characteristics:

1. The center of a bounding box is located at the spine of the object.
2. The height of the bounding boxes are bounded to the edges of the object mask.
3. The width of the bounding boxes can be set manually, because this parameter highly depends on the gripper characteristics.
4. The majority of the grasp proposals are generated near the center of mass, which is based on the assumption that these points more likely lead to an successful grasp.

Results. Our automatic annotation pipeline allows us to generate a high number of grasping labels without any supervision of human annotators. Furthermore, due to the fact that the data is recorded while an expert did the grasping, we implicitly have supervision about which object should be removed from

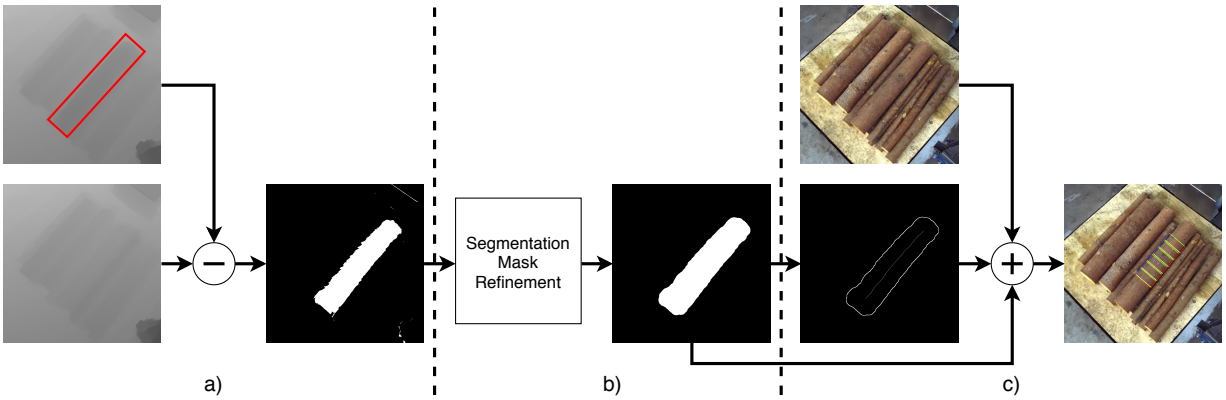

Figure 3. Our automatic annotation pipeline. a) Two consecutive depth images with one object removed (marked in red). Calculating the difference of the depth images gives a rough segmentation mask of the removed object. b) Refinement of the mask using morphological operations and Gaussian filtering. c) Geometric features (object edges, skeleton, center of mass) are calculated using the refined segmentation mask and are used afterwards to calculate the final position of the grasping point proposals. The last step transfers the proposed bounding boxes to the corresponding RGB image.

the scene, without any additional costs. The only time humans are involved is, when checking all the predicted labels via manual inspection to find images which contain erroneous labels. In this process we roughly drop 10% of the images to avoid inaccurate labeled training data. Figure 4 shows results of our automatically labeled dataset.

3.3. Human-based Data Annotation

Additionally to our automatic labeling approach, we also labeled the whole dataset manually. The idea is to train a grasp prediction network on both types of labels independently, and then compare the performance of both approaches. All hand labeled data were checked by human experts with domain knowledge to verify the correctness of the annotations.

4. Grasping Point Prediction in a Cluttered Environment

Chu et al. [4] proposed a deep neural network to predict multiple grasping points for multiple objects in the scene. We adapted their approach and retrained the network with our specific dataset.

4.1. Network Architecture and Loss Function

The network architecture is based on the Faster R-CNN object detection framework [14] using a ResNet-50 [6] as backbone. It takes a three channel RGB image as input and predicts a number of grasping point candidates, whereas one candidate $\mathbf{g}$ is defined as described in Equation 1. Note that the rotation angle θ is quantized into $R = 19$ intervals, which makes the prediction of this parameter a classification problem. All other parameters (see Equation 1) are predicted using regression. During training, the composite loss function $\mathcal{L}_{total}$ is defined as

$$\mathcal{L}_{total} = \mathcal{L}_{gpn} + \mathcal{L}_{gcr}, \tag{4}$$

where $\mathcal{L}_{gpn}$ describes the loss according to the grasp proposal net and $\mathcal{L}_{gcr}$ is the grasp configuration prediction loss. The loss term $\mathcal{L}_{gpn}$ is used to define initial rectangular bounding box proposals without orientation ($\{x, y, w, h\}$), whereas $\mathcal{L}_{gcr}$ is used to define the orientation and the refined bounding box prediction $\{x, y, \theta, w, h\}$. Figure 5 shows the structure of the prediction network and indicates how the loss parts $\mathcal{L}_{gpn}$ and $\mathcal{L}_{gcr}$ are calculated. Further information about the network architecture and the loss function can be found in [4].

4.2. Data Preprocessing and Augmentation

Our dataset for training the prediction network consists of only 52 images. Therefore, data augmentation is used to increase the size of the training data by the factor of 100. Figure 6 shows examples of the augmented data. This increases the variation in the training data and decreases the possibility of overfitting during training. After augmentation, each image was resized to $227 \times 227px$ to fit the input dimension of the network.

4.3. Training Schedule

Pre-trained ImageNet [5] weights are used as initialization for the ResNet-50 backbone to avoid overfitting and ease the training process. All other layers beyond ResNet-50 are trained from scratch. The whole structure of the network can be seen in Figure 5. We used the Adam Optimizer and trained our

Figure 4. Visualization of automatically generated labels. Each edge of one grasping point proposal is visualized with a different color to show the orientation of the box. Our method allows dense labeling of the object but only four grasping point proposals are visualized in each image to guarantee the clarity of the visualization. Note that only one object per image is labeled which implicitly adds expert knowledge about the optimal order of object removal.

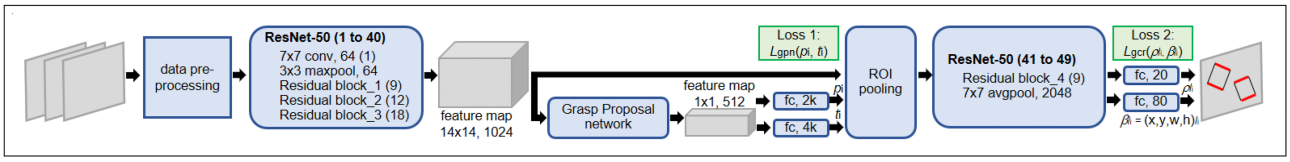

Figure 5. Architecture of the grasping point prediction network. The network takes RGB images as input, and predicts multiple grasping candidates. The grasping candidates are defined as an oriented rectangular bounding box. The output bounding boxes are drawn with different colors, whereas the red edges denote the parallel plates of the gripper and the black lines indicate the opening width of the gripper. Figure was taken from [4].

Figure 6. Data Augmentation. (**Left**) RGB input image, (**others**) randomly shifted and rotated input image.

network for 50000 iterations with a initial learning rate $\alpha = 0.0001$. The anchor sizes for the bounding box proposals are chosen according to the size of the objects in our dataset using $[8, 16, 24, 28]px$, with anchor ratios of $[0.5, 1, 2]$. All other hyperparameters were taken from [4]. Note that the goal of these experiments was to show the practical benefit of our method for automatic label generation, rather than to compete for the best possible performance for grasping point prediction. We believe that a more careful selection of hyperparameters, combined with an optimized training schedule could further boost the results.

5. Experiments and Evaluation

We trained the previously described prediction network two times separately, once with automatically annotated data and once with the same data labeled by hand. Both networks were evaluated using a test set containing 22 images which are independent from the training data (different camera position, random placement of objects) to verify the generalization capabilities of our network. We used the same training schedule for both methods, as well as the same parameters for non-maximum suppression for both experiments to ensure a fair comparison. The evaluation of our predicted grasping candidates is divided into two parts:

1. Quantitative evaluation of the predicted grasping points by calculating the ratio of graspable / non-graspable candidates.

2. Qualitative evaluation by visualizing the predicted grasping candidates.

5.1. Quantitative Evaluation

For quantitative evaluation we decided to calculate the relative number of predicted grasp candidates that are non-graspable for both networks trained with manually/automatically labeled data. We define a non-graspable prediction as 1) the size of the predicted bounding box is unsuitable (either too big or too small) or 2) grasping is not feasible due to partial occlusion of the object. Figure 8 shows examples of non-graspable candidates. Table 1 shows the quantitative results indicating that a deep network trained with automatically labeled data can achieve similar performance compared to the same network trained with manually labeled data.

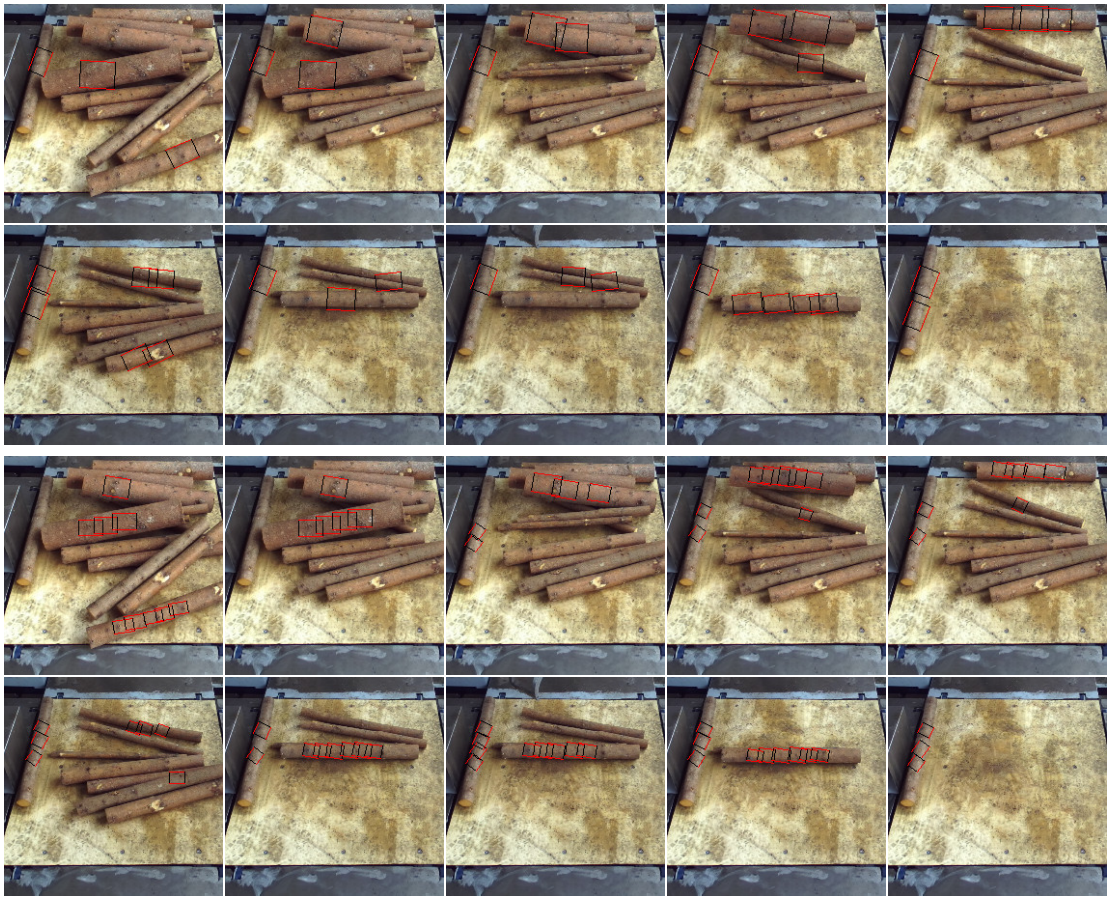

Figure 7. Comparison of predicted grasping candidates for both networks trained on automatically labeled data (**top two rows**) and manually labeled data (**bottom two rows**). We apply non-maximum suppression to reduce the number of visualized boxes and to ensure the clarity of the visualization.

Method	Valid grasping candidates in %
Auto-Label	81.17
Man-Label	83.43

Table 1. Relative number of valid grasping candidates for both approaches. The network trained with automatically labeled data is named **Auto-Label**, whereas the network trained with manually labeled data is named **Man-Label**. Both networks show similar performance which emphasizes the usefulness of our automatically labeled data.

Figure 8. Examples for non-graspable predictions. (**Left**) predicted bounding box not graspable because another object is on top; (**middle**) box too big; (**right**) box too small.

5.2. Qualitative Results

Qualitative results of our grasping point predictions are shown in Figure 7 for the networks trained with the manually annotated data and the automatically generated labels respectively.

6. Conclusion

We have proposed an automatic annotation method for easily generating grasp proposals for robotic manipulations using only one RGBD camera. Our annotation method requires minimal human interaction and is highly cost effective. With the proposed method, we generated ground truth data and successfully trained a deep neural network to predict grasping candidates. To underline the usefulness of our approach, we trained our grasp prediction network with hand annotated and automatically annotated data separately, and our experiments showed similar performance for both attempts. This leads to the conclusion that our automatically generated labels are highly accurate.

We believe that the best strategy to train a deep network for grasping point predictions is to initially train with a large number of automatically annotated frames using our method, and afterwards fine-tune it with a small number of frames annotated by human experts. This strategy can lead to highly accurate results with minimal human interaction.

References

[1] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *2000 International Conference on Robotics and Automation (ICRA)*, volume 1, pages 348–353. IEEE, 2000.

[2] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2013.

[3] S. Caldera, A. Rassau, and D. Chai. Review of deep learning methods in robotic grasp detection. *Multimodal Technologies and Interaction*, 2(3):57, 2018.

[4] F.-J. Chu, R. Xu, and P. A. Vela. Real-world multiobject, multigrasp detection. *IEEE Robotics and Automation Letters*, 3(4):3355–3362, 2018.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] S. Kumra and C. Kanan. Robotic grasp detection using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 769–776, 2017.

[8] T.-C. Lee, R. L. Kashyap, and C.-N. Chu. Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.

[9] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[10] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[11] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 616–625, 2018.

[12] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 International Conference on Robotics and Automation (ICRA)*, pages 3406–3413. IEEE, 2016.

[13] J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. In *2015 International Conference on Robotics and Automation (ICRA)*, pages 1316–1322. IEEE, 2015.

[14] 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*, pages 91–99, 2015.

[15] C. Rother, V. Kolmogorov, and A. Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[16] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

[17] M. Suchi, T. Patten, D. Fischinger, and M. Vincze. Easylabel: A semi-automatic pixel-wise object annotation tool for creating robotic rgb-d datasets. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6678–6684. IEEE, 2019.

[18] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[19] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 International Conference on Robotics and Automation (ICRA)*, pages 1386–1383. IEEE, 2017.

[20] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng. Fully convolutional grasp detection network with oriented anchor box. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7223–7230, 2018.