Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

A Grasping CNN with Image Segmentation for Mobile Manipulating Robot

Yingying Yu[1,2], Zhiqiang Cao[1,2], Shuang Liang[1,2], Zhicheng Liu[1,2], Junzhi Yu[1,2], Xuechao Chen[3]

1 State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
2 University of Chinese Academy of Sciences, Beijing 100049, China
3 Beijing Advanced Innovation Center for Intelligent Robots and Systems;
Intelligent Robotics Institute, BIT, Beijing 100081, China.

***Abstract* – This paper presents a grasping convolutional neural network with image segmentation for mobile manipulating robot. The proposed method is cascaded by a feature pyramid network FPN and a grasping network DrGNet. The FPN network combined with point cloud clustering is used to obtain the mask of the target object. Then, the grayscale map and the depth map corresponding to the target object are combined and sent to the DrGNet network for providing multi-scale images. On this basis, depthwise separable convolution is used for encoding. The results of encoders are refined according to the light-weight RefineNet as well as sSE, which can achieve a better grasp detection. The proposed method is verified by the experiments on mobile manipulating robot.**

Index Terms – Robotic grasping, grasping CNN, image segmentation, mobile manipulating robot

I. INTRODUCTION

With the development of robot technology, manipulating robots are expected more and more applications in our daily lives. In order to provide better services, the manipulating robots should possess the grasping ability, which has attracted much attentions [1]-[3].

For the implementation of robotic grasping, image segmentation provides a solution to segment the target class objects from the image captured by the vision sensor. On this basis, the pixel positions and the segmentation mask of the target class can be obtained. Compared with the traditional image segmentation methods such as the graph-based methods [4] and the active contours-based methods [5], the methods based on deep learning can obtain better segmentation results by using the convolutional neural network (CNN) where the features of the image can be automatically learned [6]-[10]. Long *et al*. [6] proposed fully convolutional networks (FCN) for semantic segmentation, which take input of arbitrary size and produce correspondingly-sized output for prediction of per-pixel. Chen *et al*. [7] proposed a DeepLab model for semantic segmentation, which applies the atrous convolution with upsampled filters for dense feature extraction. Atrous convolution can effectively enlarge the field of view of filters to incorporate larger context without increasing the number of parameters or the amount of computation. Zhao *et al*. [8] proposed pyramid scene parsing network to embed difficult scenery context features in a FCN-based prediction framework. Badrinarayanan *et al*. [9] proposed a Segnet architecture for semantic pixel-wise segmentation, which is efficient both in memory and computational time since it only stores the max-pooling indices of the feature maps and uses them in its decoder network. Seferbekov *et al*. [10] proposed an approach for automatic multi-class land segmentation based on a fully convolutional neural network of feature pyramid network (FPN) family.

After the segmentation mask of the target class is obtained, the optimal grasp for the manipulator can be determined according to the pixels of the target object. Traditional grasp detection methods include model-based methods [11], force-closure methods [12], etc. With the successful applications of deep learning, deep networks are also considered to solve the problem of grasp detection [13]-[17]. In [13], a huge number of candidate grasps are evaluated by a two-step cascaded system with two deep networks, where the top detections from the first network are re-evaluated by the second one. Redmon *et al*. [14] presented a grasp detection network with a single-stage regression of grasp rectangle. Kumra *et al.* [15] used ResNet [18] for feature extraction and introduced a multi-modal with RGB and depth images to predict the grasps. Morrison *et al.* [17] presented a fast object-independent grasp detection method, where a generative grasping network GG-CNN is adopted to predict the grasp at every pixel.

In this paper, we present a grasping CNN with image segmentation, which is used for the grasping task of mobile manipulating robot. The proposed network is cascaded by an existing FPN network and a grasping network named DrGNet. The segmentation mask of the target class can be obtained by FPN [10]. Specially, considering the case where several objects with the same class maybe overlap in segmentation, we combine point cloud clustering to determine the mask of the target object. Then, the grayscale and depth images related to the mask of the target object are sent to DrGNet for the multi-scale images. The encoder of the DrGNet integrates these images and adopts the depthwise separable convolution for feature extraction, and its decoder combines modules of light-weight RefineNet [19] with sSE [20] for optimal grasp detection.

The rest of the paper is organized as follows. Section II describes the grasping CNN with image segmentation. In Section III, the experimental results are presented. Finally, Section IV concludes the paper.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

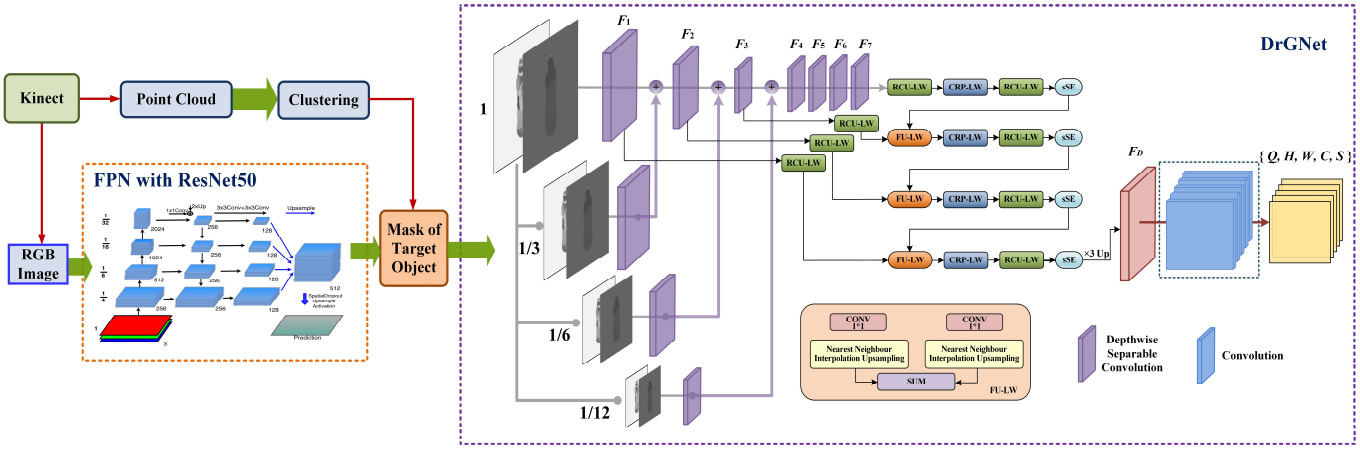


Fig. 1. The architecture of grasping CNN with image segmentation for mobile manipulating robot.

II. A Grasping CNN with Image Segmentation

The architecture of grasping CNN with image segmentation for mobile manipulating robot is shown in Fig. 1, which is composed of a FPN with ResNet50 and a grasping convolution network DrGNet. The RGB image is sent to FPN for image segmentation, and then the mask of the target object can be determined with the combination of point cloud clustering. The images corresponding to the mask of target object are sent to DrGNet, and one can obtain the optimized grasp, which will be transformed to the 6D pose of the manipulator for grasping.

A. Target Determination Based on Image Segmentation

In this section, FPN [10] is used to obtain the segmentation result, which adopts a pre-trained ResNet50 as the feature encoder. FPN includes bottom-up and top-down pathways, where the former deals with an RGB image with a decreasing size of the feature maps, whereas the latter upsamples feature maps, which are then concatenated. Finally, the feature maps' channel number decreases to the number of classes, and the size of prediction image also becomes the same as that of the original image.

With the segmentation result, the method of point cloud clustering such as K-means is used to deal with the situation that the segmentations of homogeneous objects maybe cover each other. The clustering results are beneficial to determine the mask of the target object.

B. The Grasping Network DrGNet

In order to prevent the optimal grasp of the target object in image from background interference, we set the values of pixels outside the mask of the target object in the grayscale map and the depth map to g_o and d_o, respectively. The input of DrGNet consists of multi-scale images, where the length and width of these images are 1, 1/3, 1/6 and 1/12 of the original scale, respectively. The encoder of DrGNet adopts depthwise separable convolution [21] for feature extraction. Depthwise separable convolution factorizes a standard convolution into a depthwise convolution and a 1×1 pointwise convolution, which can reduce the model's size and computation. The input images of different scales are added to the corresponding scale feature maps after convolution processing. The convolution kernels of feature maps F_i in the encoder are set to 5, 5, 5, 5, 5, 5, 3, the strides are set to 3, 2, 2, 1, 1, 1, 1 and the channel number of each feature map is 32, where i=1,…,7.

For the decoder of DrGNet, we adopt the modules including light-weight RCU, CRP and FUSION blocks in [19], as well as the channel squeeze and spatial excitation block in [20], which correspond to RCU-LW, CRP-LW, FU-LW, sSE in Fig. 1, respectively. RCU-LW is a residual block composed of Relu activation function and convolutions, CRP-LW is also a sequence of convolutional and pooling layers, while FU-LW combines feature maps together. Different from the FUSION block with bilinear interpolation for upsampling in [19], our FU-LW uses the nearest neighbour interpolation where the pixel value of each point in the new image is set to that of the nearest point in the image before interpolation, which is simpler and faster. Moreover, four sSE blocks are added to learn the importance of different pixel positions in their corresponding feature maps. We label $\mathbf{U}$ as the input feature map of sSE. 1×1 convolution is used to process $\mathbf{U}$ as a feature map with one channel. After sigmoid layer rescales activations to [0, 1], the weight of every pixel of $\mathbf{U}$ can be obtained, which corresponds to the relative importance of a spatial pixel position. The output of sSE is described as $\mathbf{U}_{sSE}=\mathbf{U}\odot Sigmoid(Conv(\mathbf{U}))$, where $Conv(\cdot)$ and $Sigmoid(\cdot)$ describe the processing of convolution and sigmoid layer, respectively.

The decoding process starts from the last layer of the encoder whose output passes through one RCU-LW block followed by one CRP-LW, another RCU-LW and sSE before being sent to FU-LW. Notice that the feature map F_3 is also sent to FU-LW. These two inputs of FU-LW are respectively processed by one 1×1 convolution with upsampling to the larger scale of these two inputs, and one acquires the summation result. Continue similar processing according to Fig. 1 until the FU-LW block with 1/3 scale is outputted, and this output is further processed and upsampled to the scale of original image, and we have the feature map F_D. Then five 3×3 convolutions are employed and the output $\boldsymbol{O}$ of DrGNet are generated with five maps Q, W, H, C and S, which represent the quality

evaluation, the width, the height, $\cos(2\theta_{x,y})$, $\sin(2\theta_{x,y})$ of the grasping rectangle for each pixel point, respectively. (x, y) refers to a pixel position and $\theta_{x,y}$ is the orientation of the grasping rectangle at (x, y). During the training of the DrGNet, the mean square error loss is used for the loss function.

And the grasp rectangle at the pixel point with the maximum value of Q is regarded as the best grasp $(x^*, y^*, \arctan(\frac{S_{x^*,y^*}}{C_{x^*,y^*}})/2, W_{x^*,y^*}, H_{x^*,y^*})$, where $(x^*, y^*) = \arg\max_{(x,y)} Q$.

The grasp detection process of the target object is described in Algorithm 1, where Mask_org is the segmentation mask of the target class K. Mask_tar is the mask that is related to the target object to be grasped according to task requirement, and C_f, D_f, g_f are the corresponding point cloud, depth map, and grayscale map, respectively. *minAreaRect*$(\cdot)$ is the function to obtain the minimum envelope rectangle, and *multi_scale*(D_f, g_f) outputs the images with four scales based on the concatenated result of D_f and g_f. With the aforementioned optimal grasp rectangle, for the manipulator, its desired 6D pose can be obtained with the combination of intrinsic and extrinsic matrixes of the camera. Then the manipulator executes grasping.

Algorithm 1. The grasp detection of the target object.

Input: RGB image I, point cloud I_p, and the target class K.
Output: The best grasp of the target object.

```
Mask_org←Null;
Mask_tar←Null;
Mask_org←FPN(I, K);
if (Mask_org≠Null) then
    bbox_org←minAreaRect(Mask_org);
    C1, C2, …←K-means(Ip, bbox_org);
    determine the target object;
    Mask_tar←Mask_org(Cf);
    obtain Df and gf according to Mask_tar;
    Q, W, H, C, S←DrGNet(multi_scale(Df, gf));
    calculate the best grasp;
end if
return
```

III. EXPERIMENTS

We conduct the experiments to verify the effectiveness of the proposed method. The experiments are carried out on a robot with a 6-DOF Kinova arm and a Kinect v2 sensor. This method is programmed under the framework of robot operating system (ROS). As shown in Fig. 2, the related nodes include: "/kinect2", "/ORB_SLAM_location", "/navigation_control", "/grasp_detection", "/grasp_motion" and "/m1n6s300_driver". The "/ORB_SLAM_location" node subscribes the image information published by the node "/kinect2" and provides the position of the mobile manipulating robot by ORB-SLAM2 [22]. The position information will be published on the topic "/current_position". The "/navigation_control" node receives the position of the robot, and publishes two topics "/cmd_vel" and "/stop_flag", which correspond to the motion command and the end flag of navigation, respectively. The "/grasp_detection" node also subscribes the image information published by the node "/kinect2", and it generates the desired 6D pose according to the optimal grasp determined by the DrGNet with image segmentation and publishes the pose information by a topic "/grasp_pose". The node "/grasp_motion" subscribes the topics "/grasp_pose" and "/stop_flag". Grasping planning will be carried out and published to the node "/m1n6s300_driver" after the robot navigates to the given position. The node "/m1n6s300_driver" drives the motion of the manipulator according to the subscribed planning information.

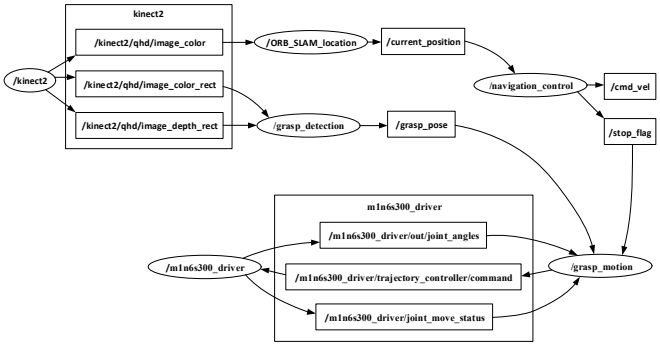


Fig. 2. The relationship of ROS nodes.

The experiment 1 considers five objects on a table and the mobile manipulating robot is in front of the table, as shown in Fig. 3(a). We appoint the bottle as the target class. The video snapshots of robotic grasping in experiment 1 are shown in Fig. 3. The experiment results show that the proposed method is effective. The experiment 2 considers a special case where several objects of the target class are close to each other. In this experiment, three different bottles (see Fig. 4(a)) are placed close together, and the object of the target class closest to the camera is selected as the target object. Other objects on the table include two cups, a banana, etc. As shown in Fig. 4(b), the segmentation mask of the target class is related to three bottles, which cannot be separated independently. By clustering the point cloud corresponding to the segmentation mask area, we can obtain the mask of the target object, as shown in the blue part of Fig. 4(c). The video snapshots of the experiment 2 are shown in Fig. 5.

The experiment 3 concerns a scenario where the mobile manipulating robot is away from the light-green target cup. And the robot has to first navigate to a location near the target object. The video snapshots of the experiment 3 is shown in Fig. 6. Fig. 7 demonstrates the variation curves of the joint angles for the manipulator during grasping. The results of experiment 3 shows that the proposed method can combine the existing navigation schemes to complete the task smoothly.

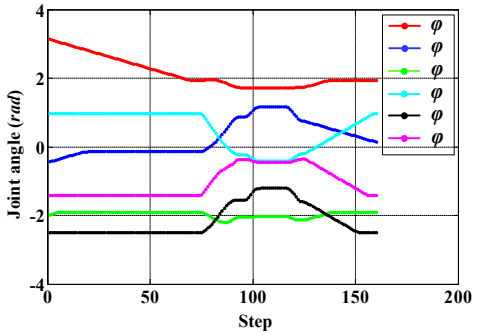


Fig. 7. The variation curves of joint angles for the manipulator in experiment 3.

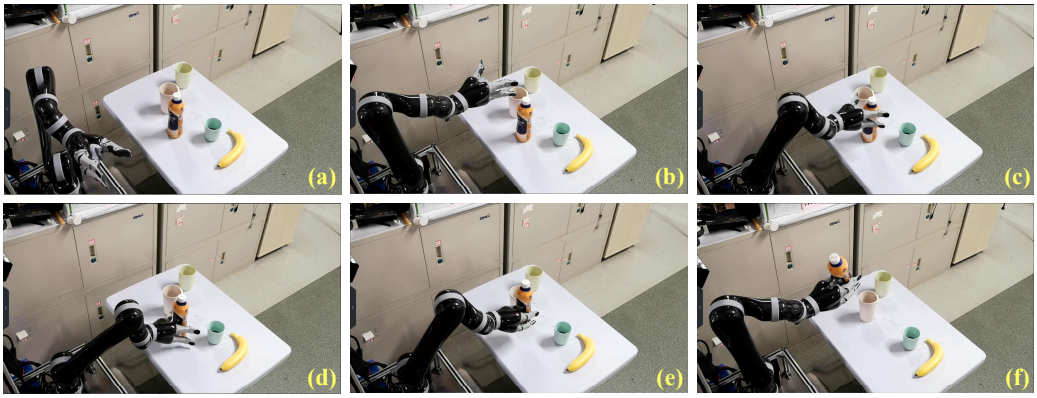
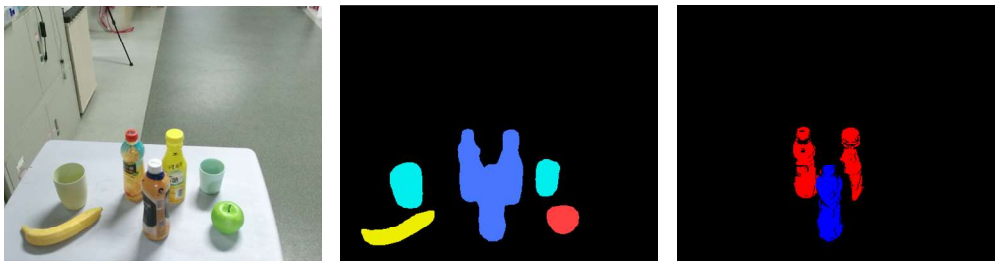


Fig. 3. The video snapshots of the experiment 1.



(a) (b) (c)

Fig. 4. The segmentation and grasp detection results of the experiment 2. (a) RGB image. (b) The segmentation result. (c) The result of point cloud clustering.

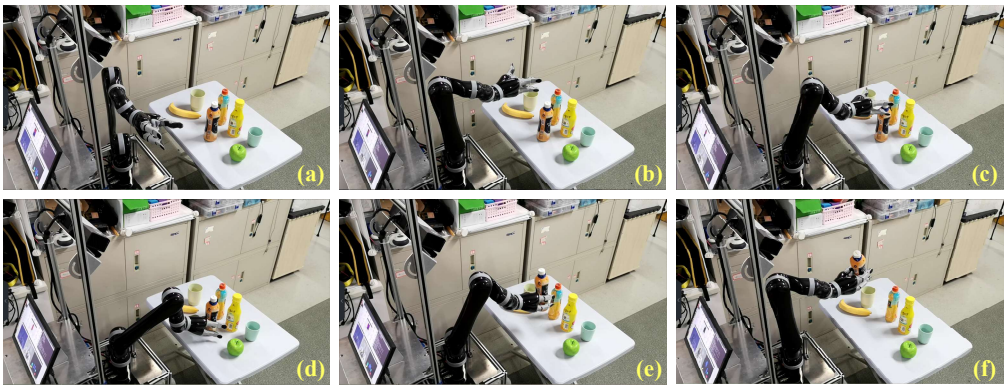


Fig. 5. The video snapshots of the experiment 2.

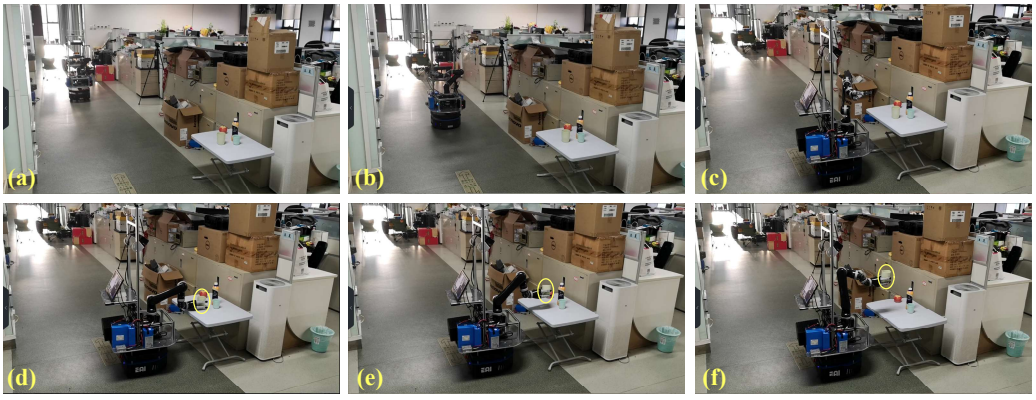


Fig. 6. The video snapshots of the experiment 3.

IV. CONCLUSION

In this paper, we present a grasping CNN with image segmentation for mobile manipulating robot, which is mainly composed of FPN and DrGNet. Combined with point cloud clustering, FPN is used to generate the mask of the target object. Then, the optimal grasp of the target object can be determined by DrGNet. The experiments have verified the effectiveness of the proposed method.

ACKNOWLEDGMENT

This work was supported in part by the Key Research and Development Program of Shandong Province under Grant 2017CXGC0925, in part by the National Natural Science Foundation of China under Grants 61633017, 61633020, in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant 2018IRS21, and in part by the Equipment Advance Research Foundation of China under Grant 61403120407.

REFERENCES

[1] N. Mavrakis, E. A. M. Ghalamzan, R. Stolkin. Safe robotic grasping: Minimum impact-force grasp selection. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017: 4034-4041.

[2] Y. Wang, R. Wang, S.Wang, M. Tan, and J. Z. Yu. Underwater bio-inspired propulsion: from inspection to manipulation. *IEEE Transactions on Industrial Electronics*, DOI: 10.1109/TIE.2019.2944082.

[3] C. Choi, W. Schwarting, J. DelPreto, et al. Learning object grasping for soft robot hands. *IEEE Robotics and Automation Letters*, 2018, 3(3): 2370-2377.

[4] P. F. Felzenszwalb, D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 2004, 59(2): 167-181.

[5] T. F. Chan, L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 2001, 10(2): 266-277.

[6] J. Long, E. Shelhamer, T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015: 3431-3440.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 40(4): 834-848.

[8] H. Zhao, J. Shi, X. Qi, et al. Pyramid scene parsing network. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017: 2881-2890.

[9] V. Badrinarayanan, A. Kendall, R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(12): 2481-2495.

[10] S. S. Seferbekov, V. Iglovikov, A. Buslaev, et al. Feature Pyramid Network for Multi-Class Land Segmentation. *CVPR Workshops*, 2018: 272-275.

[11] A. Boularias, O. Kroemer, J. Peters. Learning robot grasping from 3-d images with markov random fields. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011: 1548-1553.

[12] A. Morales, P. J. Sanz, A. P. Del Pobil, et al. Vision-based three-finger grasp synthesis constrained by hand geometry. *Robotics and Autonomous Systems*, 2006, 54(6): 496-512.

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

[14] J. Redmon, A. Angelova. Real-time grasp detection using convolutional neural networks. *IEEE International Conference on Robotics and Automation*, 2015: 1316-1322.

[15] S. Kumra, C. Kanan. Robotic grasp detection using deep convolutional neural networks. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017: 769-776.

[16] U. Asif, J. Tang, S. Harrer. GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices. *IJCAI*, 2018: 4875-4882.

[17] D. Morrison, P. Corke, J. Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *Robotics: Science and Systems*, 2018.

[18] K. He, X. Zhang, S. Ren, et al. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016: 770-778.

[19] V. Nekrasov, C. Shen, I. Reid. Light-weight refinenet for real-time semantic segmentation. arXiv:1810.03272, 2018.

[20] A. G. Roy, N. Navab, C. Wachinger. Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018: 421-429.

[21] A. G. Howard, M. Zhu, B. Chen, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, 2017.

[22] R. Mur-Artal, J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 2017, 33(5): 1255-1262.