Deep Learning-based Fast Grasp Planning for Robotic Bin-picking by Small Data Set without GPU

S. Arai^{a*}, Z. Feng^a, F. Tokuda^a, A.S.Z. Purnomo^a, Y. Xu^a and K. Kosuge^a

^a*Graduate School of Engineering, Tohoku University, Aramaki Aza Aoba 6-6-01, Aoba-Ku, Sendai 980-8579, Japan;*

This paper proposes a deep learning-based fast grasp detection method with a small dataset for robotic bin-picking. We consider the problem of grasping stacked up mechanical parts on a planar workspace using a parallel gripper. In this paper, we use a deep neural network to solve the problem with a single depth image. To reduce the computation time, we propose an edge-based algorithm to generate potential grasps. Then, a convolutional neural network (CNN) is applied to evaluate the robustness of all potential grasps for bin-picking. Finally, the proposed method ranks them and the object is grasped by using the grasp with the highest score. In bin-picking experiments, we evaluate the proposed method with a 7-DOF manipulator using textureless mechanical parts with complex shapes. The success ratio of grasping is 97%, and the average computation time of CNN inference is less than 0.23[s] on a laptop PC without a GPU array. In addition, we also confirm that the proposed method can be applied to unseen objects which are not included in the training dataset.

Keywords: Grasp detection; robotic bin-picking; deep learning; small dataset; potential grasp.

1. Introduction

Industrial robots have been used in various fields and applications, such as cooperative tasks [1, 2], visual servoing [3–5], robotic bin-picking [6, 7], and assembly line [8, 9]. Among these applications and tasks, grasp detection is essentially necessary and important. This paper especially focuses on the grasp detection for robotic bin-picking, which remains a difficult challenge for robotics research.

Analytical grasp metrics for physical characteristics of the grasping, such as force-closure [10], and grasp wrench space (GWS) [11] have been proposed to evaluate the robustness of a grasp. These methods require 3D models of target objects and 6D pose estimation of the objects for robotic bin-picking. With the development of technology of three-dimensional measurements [12–15], a lot of techniques for pose estimation algorithms with 3D point cloud have been proposed, such as 3D feature [7, 16–19], 3D keypoint detection [20, 21], segmentation [22, 23], and Iterative Closest Point (ICP) [24], have been proposed. With the perfect knowledge of the object’s 6D pose, the robot can grasp the object with pre-designed grasp configurations. However, the estimated pose generally includes some errors due to the measurement noise. Meanwhile, recent works show that data-driven methods can be used to detect grasps directly from RGBD image or point cloud with hand-designed features [25] or auto-learned features [26].

For robotic bin-picking, fast execution of grasp detection is required, and it needs to be applicable to heavy-cluttered scenes. With rapid advances of deep learning technology, the deep

*Corresponding author. Email: arai@tohoku.ac.jp

Figure 1.: Structure of a deep learning network for computing grasp scores. The input and output are a potential grasp and the corresponding grasp score, respectively. Grasp scores stand for the robustness of the grasp, higher is better. The network is trained with binary scores.

learning-based grasp detection has recently gained attention. However, most deep learning-based techniques need a large scale of the training dataset and heavy computational resources giving off heats, such as GPU. These requirements do not especially meet industrial scene, such as factory and Warehouse, because computers with cooling fans are not preferred for factory automation. Computers without fans and GPU are more robust to dusty and high temperature industrial environment [27] than ones with fans and have lower risks of failure due to few moving parts.

In this paper, we propose a fast and robust deep learning-based grasp detection method for robotic bin-picking without a large amount of dataset and heavy computational resources. To reduce the computation time, we use a single depth image instead of RGBD data or point cloud, and we propose a simple CNN architecture that works well to detect robust grasp candidates. Even though the depth image only contains 2.5D information, the proposed approach is proven to show 95% success rate of actual bin-picking.

The proposed method adopts *grasp rectangle* as the representation of grasp candidates, which has been shown to be effective in previous works [26, 28]. The grasp rectangle-based approach assumes a robotic hand as a parallel gripper[29, 30] which are most commonly used hand in the industrial field compared to the other hands[31, 32].

To generate potential grasp rectangles, we propose an edge-based grasps sampling algorithm which has low computational complexity. We also propose a novel data preprocessing method for grasp rectangles, that makes the proposed method adaptable well to novel objects which the network has not been trained on. Firstly, the proposed method extracts the edge of a depth image and search possible contact points of the gripper, and then potential grasps are generated based on these contact points. Then, a simple convolutional neural network computes *Grasp Scores* for each grasp candidate to evaluate the robustness of potential grasps, and the grasp candidate with the highest grasp score is selected. Finally, by combining the depth image and grasping rectangle, the configuration of a grasp is obtained.

2. Related Work

In general, most of the grasp detection methods can be categorized into two groups: *Analytic-based* and *Learning-based*.

Analytical methods exploit the physical and geometrical properties of target objects and utilize analytical metrics such as force closure [33] and wrench space [34]. Even though analytical methods can be applied to any kind of object, these methods tend to not transfer well to the real-world environment since the methods require 3D modeling of the object and precise 6D pose estimation.

For 6D pose estimation in a cluttered scene, local features, such as gradient, surface normals, distances, and more complicated ones are used [35, 36] for first step estimation. As a second

step, the iterative closest point (ICP) with a point-to-point metric is widely utilized for more precise estimation. There have also been many improved algorithms based on ICP using point-to-line metric [37] or point-to-plane metric [38, 39]. However, the performance of these algorithms depends in large on the initial estimation of the first step, and tend to converge to a local minima [40]. In the bin-picking problem, a lot of objects are stacked up on top of each other, and most of them are partially visible. As a result, pose estimation becomes significantly more complex. In addition, ICP based algorithms are computationally expensive.

Learning-based methods utilize various techniques of machine learning and have recently become popular in the robotics field due to its efficacy in solving complex robotics tasks. While a lot of pose estimation methods[41, 42] and related ones[43, 44] with deep learning have also been proposed, end-to-end deep learning-based approaches of the grasp detection attract much more attention.

Learning-based grasp detection methods are mainly classified into two types: non-deep-learning and deep-learning based methods. Non-deep-learning grasping methods use contact points [45, 46], geometry characteristics [47], or the similarities among graspable parts of objects [49]. In [48], a dictionary learning and sparse representation framework for the grasp detection has been proposed. The proposed framework can be trained with small amount of data and outperformed its previous neural network-based approaches.

Most deep learning-based grasp detection share the same pipeline: taking RGBD images or point cloud from the scene as the input and predicting the most robust grasp candidate from the image. Some methods require semantic segmentation to distinguish objects in a cluttered scene [51], while others do not require the object identities[52–55]. Various deep learning-based methods have been explored to tackle the bin-picking problem such as the use of sparse auto-encoder (SAE) [26], a fully convolutional neural network (FCN) that outputs pixel-wise prediction [52–54] and deep reinforcement learning [55]. However, deep learning-based methods tend to work well only when the large training dataset is available. The training dataset can be gathered from real-world experiments [54, 56], or obtained from synthetic data provided by a physics simulation engine [57–59]. These approaches are time-consuming and computationally expensive. Without large datasets, these methods are not able to generalize to novel objects which are not included in the training datasets. Besides, most deep learning-based grasp detection method cannot be applied to robotic bin-picking in industrial situation. Almost all types of factory automation require a lot of processes of automatic parts supplying which are expected to be replaced by the robotic bin-picking. In this process, a lot of industrial parts with the same type are stacked together in one bin.

To solve these problems, we propose the grasp detection method aimed at industrial bin-picking where a lot of objects with the same class are stacked together in a bin. The proposed method requires a small dataset and does not need a graphic processing unit (GPU) to train the network while obtaining the same performance as previous methods. The scale of the required training dataset is significantly decreased by using a simple and effective network, and a novel processing method for grasp rectangles. In validation experiments, the proposed method only used 30 depth images as the training data, and achieve 97% success ratio of bin-picking with the average inference time of network is 0.23[s] using a laptop without a GPU array, which is a significant advantage over other deep learning methods.

3. Proposed grasp planning

This section proposes deep learning-based fast grasp detection with a small dataset. The method consists of the following five steps:

- (1) A depth image of the bin-picking scene is obtained by three-dimensional measurement
- (2) The proposed method generates potential grasps
- (3) The proposed method creates images of potential grasps as the input of CNN

Figure 2.: An example of a grasping rectangle consisting of blue and red lines. Left: An image of a target object. Right: A depth image of the target object. The blue lines represent the place of fingers of the fingers of the parallel gripper at the time of the grasping. The length of red lines indicates the open width of the parallel gripper. The grasping pose is computed as follows: The orientation around the optical axis and position on the plane which is parallel to the image plane can be computed from the depth image, and the position in the optical axis (height above the work surface) is obtained from pixel values in the depth image. To execute the grasp, we compute the three-dimensional positions of two fingers.

- (4) CNN outputs the grasp score showing the robustness of the grasp
- (5) The grasp with the highest grasp score is selected

3.1 Grasp rectangle

This section describes grasp rectangles which are used for representing potential grasps. We use the idea of Jiang’s work [28] to represent the grasp of a parallel gripper, as shown in Fig. 2. It is assumed that contact length of the finger of the parallel gripper is constant in the depth image, while the open width is changeable. The grasp rectangles are then rotated to horizontal and processed to the same size. The details of the processing is shown in the next section.

3.2 Generation of Potential Grasps

A naive way to generate potential grasps is randomly generating grasp rectangles around the object. This method is an effective way for a single or several objects [26]. However, random generation of potential grasp will result in an ultra-large amount of potential grasps, since the bin-picking scene has more than 20~30 objects that can be grasped. This is not applicable for robotic bin-picking which is one of the real-time applications due to unacceptable computation time. Thus we need a fast method for generating potential grasps, which decreases the total amount of potential grasps and includes feasible grasps at the same time. For considering the method satisfying these requirements, we first define the following properties of the good grasp for the parallel gripper:

- Two contact points are located on the planar surfaces of the target object.

- Two fingers of the gripper are approximately parallel to the edges near the contact areas.

By generating grasps satisfying the two properties, we can filter most of the invalid grasps. As a result, the number of total potential grasps can be limited to an acceptable scale.

The details of the fast method of generating the potential grasps in the depth image are shown below. The proposed method first uses *Canny* edge algorithm to extract edge pixels in the depth image as the contact points of the gripper. After that, by setting the minimum distance of these edge points, the total number of contact points is decreased. For each contact point (an example is shown as Point 1 in Fig. 4), the proposed method crops a small area near each point and find two endpoints of edge within the area, and then the normal is computed from these two endpoints. This simple but effective method is not only applicable to the scene with a lot of objects. The intersection of normal and the other edge can be regarded as another contact point (an example is shown as Point 2 in Fig. 4). Finally, the grasp rectangles are generated based on two contact points and the finger size.

3.3 Conversion from grasp rectangle to EGR

The proposed method does not directly use the grasp rectangles generated by the previous section as the input of CNN. To compute reasonable the grasping scores by CNN, we here propose a data processing method convert the grasp rectangles to appropriate grasping rectangle for CNN named “Efficient Grasp Rectangle (EGR).”

Most existing methods for grasp detection by CNN simply resize the image of the grasping rectangle to a target size. In contrast, we consider the area near the contact point has more important than the other area. Thus we give more weight to the contact areas. The grasping rectangle can be divided into three areas, as shown in Fig. 3. Area 1 and 3 are areas near the contact points and we keep the size of these areas. Area 2 is an area between contact points, we resize it to make it smaller than Area 1 and 3. Finally, we assemble all these areas and resize them to the target size for CNN and generate EGR. The bottom figure in Fig. 3 illustrates two grasp rectangles; Grasp A and B. It is obvious that Grasp A can provide a better grasp than Grasp B. However, the simply resizing approach gives the higher grasp score to Grasp B compared to Grasp A. In contrast, the proposed method with EGR rates Grasp A higher than Grasp B because, through the above processing, we increase the weights of contact areas. We can conclude that the proposed method with EGR improves the performance of rating of potential grasps compared with conventional approaches without EGR.

4. Network Modeling, Training, and Inference

This section presents a simple but effective convolutional neural network (CNN) to compute grasp scores of potential grasps represented by EGR. The input and output of the network are the cropped depth image in each EGR and the corresponding grasping score, respectively. The network includes two convolutional layers, two pooling layers, and two fully connected layers, as shown in Fig. 1.

Let $G \in \mathbb{N}_0^{m \times n}$ denote the depth image of GR-CN. Firstly, we rescale G to $\bar{G}$ so that all the elements are in a range of $[0, 1]$. By using the function of the network represented by f and the parameters (weights and bias) of the network denoted by Θ , the output of the network, that is, the grasp score is given by

$$y = f(\bar{G}, \Theta). \quad (1)$$

During the training, we use the following output p :

$$p = \sigma(y), \quad (2)$$

Figure 3.: Converting process from grasp rectangle to the proposed grasp rectangles named “Efficient Grasp Rectangle (EGR).” A generating process of EGR is shown in the upper figure. The bottom figure shows two examples of the grasp rectangles. It is obvious that Grasp A is more stable than Grasp B. However, the right table shows that the grasp score of B computed by CNN is higher than A when EGR is not applied. In contrast, the proposed EGR gives a reasonable result: the grasp score of A is higher than B. Because the proposed EGR gives higher weights near the area of two contact points, as shown in the upper figure.

where σ is a sigmoid function. We manually labeled typical good and bad grasps in the depth images as the training data. For good and bad grasps, $\hat{p} = 1$ and $\hat{p} = 0$ are labeled, respectively. A goal here for the training is to find Θ^* which satisfies:

$$\Theta^* = \arg \min_{\Theta} L(p, \hat{p}; \Theta), \quad (3)$$

where L is the cross entropy loss function.

During the inference, the grasp scores of all potential grasps are computed and ranked. The grasp score for EGR $\tilde{G}$ is computed by (1).

Figure 4.: The figure on the top shows the procedure of potential grasps generation. For the given gray-scale depth image, the edges are extracted. Then candidate contact points of fingers of the gripper are sampled with a user-defined distance shown as the red crosses. For each point, the normal vectors are computed (see blue-colored line denoted by “Normal”), and the other contact point is found at the intersection between the normal and edge (see Point 2). Every two contact points in the same point pair are regarded as the location of the fingers. Finally, grasp rectangles representing potential grasps are generated considering the size of the finger and open width. The figure on the bottom shows a method of how we obtain the normal of the edge near the contact point, instead of using Sobel kernel or Laplace kernel. The normal is directly computed from the edge image based on the following process. A small segment of the edge near the contact point is cropped, and the normal should be perpendicular to this edge segment. The gradient of normal can be computed from the endpoints of the edge segment. This method works very well for objects shown in the bottom right figure.

Figure 5.: Left figure shows experimental environment. EPSON C3 is used as 7-DOF manipulator. The middle figure shows three target objects which are parts of an actual air conditioner and bin-picking for these objects is manually performed in real factories. The parts A and B are firstly utilized for both training and grasp detection. The success ratios of parts A and B are 100% and 95%, respectively. Then we test the generalization of the proposed scheme by using the network trained by the dataset including object A and B, not C, and obtain 95% success ratio.

5. Validation Experiment

This section validates the proposed grasp detection by the bin-picking experiment. The experimental situation is shown in Fig. 5. For a 7-DOF manipulator, C3 made by SEIKO EPSON CORPORATION is utilized. A parallel gripper is attached to the manipulator. A bin containing a lot of objects is set in front of the manipulator. Three types of objects shown in the middle of Fig. 5 are used in this experiment and these objects are actual parts of an air conditioner. ENSENSE N30 with 1280×1024 resolution made by IDS Imaging Development Systems GmbH is installed above the bin and can capture the depth images for the piled-up objects. A laptop with a Core i5 6200U@2.3GHz CPU and 4 GB RAM is used for the training and inference. The laptop has no GPU array.

5.1 Training

The training data is gathered for Part A and B, not C. We labeled grasps from 30 depth images, and every dataset includes only 1,000 good and bad grasps, respectively. Because of the well-designed structure of the network and EGR, we do not need a large scale dataset.

Training data 1 shown in Fig. 5 comes from depth images of A and is used for training the network. Then the network is applied to detect grasps for part A. We perform the same procedure for part B. To detect grasps for part C, we combine training data 1 and 2, and use the combined training data to train a new network. In summary, we train the network separately for parts A and B, then apply the combined training data to part C to test the generalization of the proposed method.

The resolution of all grasp rectangles is 20×20 pixels. Our network is implemented in the Python and Tensorflow CPU. We use RMSProp as the optimizer, the min-batch size is 32, and the dropout ratio is 50% to avoid overfitting. It takes about two minutes to train the network for 100 epochs. We split 20% of the total dataset as validation data. The accuracies on the training and validation data are about 98% and 95%, respectively.

5.2 Bin-picking experiment

A robot system executed 20 grasps for each object and succeed 58 times over total of 60 trials. For all failed grasps, the system succeeds at the second trial. The success ratio of part A, B, and C are 100%, 95%, and 95%, respectively, as shown in table 1. The number of success grasping trials of object A, B, and C are 20, 19, and 19, respectively. Thus the average success rate is computed by $(20 + 19 + 19)/60 \approx 0.97$.

The average number of potential grasps is less than 1,000. As a result, the computation time of the inference is only 0.227[s] by the laptop without GPU. The proposed method shows a good prospect for real-time applications including industrial one.

The network shows the ability in learning the similarities among good grasps. For the same mechanical part in the environment, we can grasp it from different orientations according to its layout.

Part B and C can be grasped from many orientations in the experiment. The proposed method can detect different grasp configurations according to the layout of the object in the environment, as shown in Fig. 6. Besides, the proposed method still shows a high success ratio for part C which never appears in the training data. These results show that the proposed method has the ability in learning the similarities among good grasps.

Table 1.: Results of bin-picking experiment for three objects

Object ID	The number of potential grasps	Total Trails	Success Ratio	Inference Time
A	311	20	100%	0.205[s]
B	375	20	95.0%	0.235[s]
C	718	20	95.0%	0.240[s]

Figure 6.: Examples of grasps with highest score in bin-picking experiment.

6. Conclusion and Future Work

This paper proposes a deep learning-based fast grasp detection for robotic bin-picking. The proposed grasp detection adopts grasp rectangles as the representation of grasp configurations, and uses a novel edge-based algorithm to generate potential grasps. A convolutional neural network is applied to evaluate the robustness of the potential grasps and rank them. Experimental results show about 97% success ratio of bin-picking including never seen objects and high speed of the method (0.227[s] per grasp).

References

- [1] L. Rozo, S. Calinon, D. G. Caldwell, P. Jiménez, and C. Torras, “Learning physical collaborative robot behaviors from human demonstrations,” *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.
- [2] S. Arai, A. L. Pettersson, and K. Hashimoto, “Fast Prediction of a Worker’s Reaching Motion without a Skeleton Model (F-PREMO),” *IEEE Access*, vol. 8, no. 1, pp. 90340–90350, 2020.
- [3] F. Chaumette and S. Hutchinson, “Visual servo control. I. Basic approaches,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
- [4] C. Kingkan, S. Ito, S. Arai, T. Nammoto and K. Hashimoto, “Model-based virtual visual servoing with point cloud data,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5549–5555, 2016.
- [5] F. Tokuda, S. Arai, and K. Kosuge, “Neural Network based Visual Servoing for Eye-to-Hand Manipulator,” *TechRxiv Preprint*, <https://doi.org/10.36227/techrxiv.13154075.v1>, 2020.
- [6] M. Fujita, Y. Domae, A. Noda, G. A. Garcia Ricardez, T. Nagatani, A. Zeng, S. Song, A. Rodriguez, A. Causo, I. M. Chen, and T. Ogasawara, “What are the important technologies for bin picking? Technology analysis of robots in competitions based on a set of performance metrics,” *Advanced Robotics*, DOI:10.1080/01691864.2019.1698463, 2019.
- [7] D. Liu, S. Arai, Z. Feng, J. Miao, Y. Xu, J. Kinugawa, and K. Kosuge, “2D object localization based point pair feature for pose estimation,” *IEEE Int’l. Conf. on Robotics and Biomimetics*, pp. 1119–1124, 2018.
- [8] Y. Yokokohji, Y. Kawai, M. Shibata, Y. Aiyama, S. Kotosaka, W. Uemura, A. Noda, H. Dobashi, T. Sakaguchi, and K. Yokoi, “Assembly challenge: a robot competition of the Industrial Robotics Category, World Robot Summit - summary of the pre-competition in 2018,” *Advanced Robotics*, vol. 33, no. 17, pp. 876–899, DOI: 10.1080/01691864.2019.1663609, 2019.
- [9] K. Konada, J. Kinugawa, S. Arai and K. Kosuge, “B-PaDY: Robot co-worker in a bumper assembly line: System integration of the prototype,” *IEEE International Conference on Mechatronics and Automation*, pp. 722–727, 2017.
- [10] B. Faverjon and J. Ponce, “On computing two-finger force-closure grasps of curved 2D objects,” *IEEE International Conference on Robotics and Automation*, pp. 424–429, 1991.
- [11] D. Prattichizzo and J. C. Trinkle, “Grasping,” *Springer Handbook of Robotics*, pp. 671–700, 2008.
- [12] N. Chiba, S. Arai, K. Hashimoto, “Feedback projection for 3D measurements under complex lighting conditions,” *American Control Conference*, pp. 4649–4656, 2017.
- [13] Z. Li, K. Zhong, Y. F. Li, X. Zhou, and Y. Shi, “Multiview phase shifting: a full-resolution and high-speed 3D measurement framework for arbitrary shape dynamic objects,” *Optics letters*, vol. 38, pp. 1389–1391, 2013.
- [14] S. Arai, Y. Iwatani, K. Hashimoto, “Fast sensor scheduling with communication costs for sensor networks,” *Proceedings of the American Control Conference*, pp. 295–300, 2010.
- [15] S. Arai, Y. Iwatani, K. Hashimoto, “Fast sensor scheduling for spatially distributed sensors,” *IEEE transactions on automatic control*, vol. 56, no. 8, pp. 1900–1905, 2011.
- [16] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, “CAD-model recognition and 6DOF pose estimation using 3D cues,” *IEEE International Conference on Computer Vision Workshops*, pp. 585–592, 2011.
- [17] S. Salti, F. Tombari, L. D. Stefano, “SHOT: Unique Signatures of Histograms for Surface and Texture description,” *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.
- [18] D. Liu, S. Arai, F. Tokuda, Y. Xu, J. Kinugawa, and K. Kosuge, “Deep-learning based robust edge detection for point pair feature-based pose estimation with multiple edge appearance models,” *IEEE International Conference on Robotics and Biomimetics*, pp. 2920–2925.
- [19] A. Aldoma, F. Tombari, R.B. Rusu, M. Vincze, “OUR-CVFH - Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation,” *Pattern Recognition*, pp. 113–122, 2012.
- [20] S. Azimi, B. Lall, and T. K. Gandhi, “Performance evaluation of 3D keypoint detectors and descriptors for plants health classification,” *International Conference on Machine Vision Applications*, pp. 1–6, doi: 10.23919/MVA.2019.8758002, 2019.
- [21] S. Arai, N. Fukuchi and K. Hashimoto, “FASt Detection Algorithm for 3D Keypoints (FADA-3K),” *IEEE Access*, 10.1109/ACCESS.2020.3025534, 2020.
- [22] A. Nguyen and B. Le, “3D point cloud segmentation: A survey,” *IEEE Conference on Robotics*,

- Automation and Mechatronics,” pp. 225–230, 2013.
- [23] Y. Xu, S. Arai, F. Tokuda, and K. Kosuge, “A Convolutional neural network for point cloud instance segmentation in cluttered scene trained by synthetic data without color,” *IEEE Access*, Vol. 8, pp. 70262–70269, doi: 10.1109/ACCESS.2020.2978506, 2020.
 - [24] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239–256, 1992.
 - [25] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, “Learning to grasp objects with multiple contact points,” *IEEE International Conference on Robotics and Automation*, pp. 5062–5069, 2010.
 - [26] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.
 - [27] T. A. Schroeder, I. Knauer, and G. Scholl, “Planning Tool for Implementing Radio Networks in Industrial Environments,” *German Microwave Conference*, pp. 1–4, 2014.
 - [28] Y. Jiang, S. Moseson, and A. Saxena, “Efficient grasping from RGBD images: Learning using a new rectangle representation,” *IEEE International Conference on Robotics and Automation*, pp. 3304–3311, 2011.
 - [29] Y. Lu, Z. Xie, J. Wang, H. Yue, M. Wu, and Y. Liu, “A novel design of a parallel gripper actuated by a large-stroke shape memory alloy actuator,” *International Journal of Mechanical Sciences*, vol. 159, pp. 74–80, <https://doi.org/10.1016/j.ijmecsci.2019.05.041>, 2019.
 - [30] A. Kobayashi, J. Kinugawa, S. Arai and K. Kosuge, “Design and Development of Compactly Folding Parallel Open-Close Gripper with Wide Stroke,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2408–2414, doi: 10.1109/IROS40897.2019.8967838, 2019.
 - [31] K. Telegenov, Y. Tlegenov and A. Shintemirov, “A Low-Cost Open-Source 3-D-Printed Three-Finger Gripper Platform for Research and Educational Purposes,” *IEEE Access*, vol. 3, pp. 638–647, doi: 10.1109/ACCESS.2015.2433937, 2015.
 - [32] A. Kobayashi, K. Yamaguchi, J. Kinugawa, S. Arai, Y. Hirata and K. Kosuge, “Analysis of precision grip force for uGRIPP (underactuated gripper for power and precision grasp),” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1937–1942, 2017.
 - [33] V. D. Nguyen, “Constructing stable force-closure grasps,” *IEEE International Conference on Robotics and Automation*, pp. 1368–1373, 1986.
 - [34] C. Ferrari and J. Canny, “Planning optimal grasps,” *IEEE International Conference on Robotics and Automation*, pp. 2290–2295, 1992.
 - [35] R. Cabrera and T. Tuytelaars, “Discriminatively trained templates for 3d object detection: A real time scalable approach,” *IEEE International Conference on Computer Vision*, pp. 2048–2055, 2013.
 - [36] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, “Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes,” *Asian Conference on Computer Vision*, pp. 548–562, 2012.
 - [37] A. Censi, “An ICP variant using a point-to-line metric,” *International Conference on Robotics and Automation*, pp. 19–25, 2008.
 - [38] Y. Chen and G. Medioni, “Object modeling by registration of multiple range images,” *IEEE International Conference on Robotics and Automation*, pp. 2724–2729, 1991.
 - [39] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, “Towards a general multi-view registration technique,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 5, pp. 540–547, 1996.
 - [40] T. Schmidt, R. Newcombe, and D. Fox, “DART: dense articulated real-time tracking with consumer depth cameras,” *Robotics: Science and Systems, Autonomous Robots*, vol. 39, pp. 239–258, 2015.
 - [41] T. Hodan, M. Sundermeyer, B. Drost, Y. Labbe, E. Brachmann, F. Michel, C. Rother, J. Matas, “BOP Challenge 2020 on 6D Object Localization,” *arXiv:2009.07378*, 2020.
 - [42] D. Liu, S. Arai, Y. Xu, F. Tokuda and K. Kosuge, “6D Pose Estimation of Occlusion-Free Objects for Robotic Bin-Picking using PPF-MEAM with 2D Images (Occlusion-Free PPF-MEAM),” *IEEE Access*, vol. 9, pp. 50857–50871, 2021.
 - [43] W. Wang, R. Yu, Q. Huang, and U. Neumann, “SGPN: Similarity group proposal network for 3d point cloud instance segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2569–2578, 2018.
 - [44] Y. Xu, S. Arai, D. Liu, F. Lin. and K. Kosuge, “FPCC: Fast Point Cloud Clustering for Instance Segmentation,” *arXiv:2012.14618*, 2020.
 - [45] A. Saxena, J. Driemeyer, and A. Y. Ng, “Robotic grasping of novel objects using vision,” *The International Journal of Robotics Research*, Vol. 27, No. 2, pp. 157–173, 2008.

- [46] A. Saxena, L. Wong and A. Y. Ng, “Learning grasp strategies with partial shape information,” AAAI Conference on Artificial Intelligence, Vol. 3, pp. 1491–1494, 2008.
- [47] J. Glover, D. Rus, and N. Roy, “Probabilistic models of object geometry for grasp planning,” The International Journal of Robotics Research, Vol. 28, no. 8, pp. 999–1019, 2009.
- [48] T. Ludovic, P. Giguère, and B. Chaib-draa, “Dictionary learning for robotic grasp recognition and detection,” arXiv:1606.00538, 2016.
- [49] R. Detry, C. Ek, M. Madry, and D. Kragic, “Learning a dictionary of prototypical grasp-predicting parts from grasping experience,” IEEE International Conference on Robotics and Automation, pp. 601–608, 2013.
- [50] D. Morrison, A.W. Tow, M. McTaggart, R. Smith, N. Kelly-Boxall, S. Wade-McCue, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, K. Vijay, Z. Zhuang, C. Lehnert, I. Reid, P. Corke, and J. Leitner, “Cartman: The Low-Cost Cartesian Manipulator That Won the Amazon Robotics Challenge,” Intelligent Autonomous Systems 15 Advances in Intelligent Systems and Computing, pp. 786–798, 2018.
- [51] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, “Segmenting unknown 3D objects from real depth images using mask R-CNN trained on synthetic data,” arXiv: 1809.05825, 2019.
- [52] A. Zeng, S. Song, K. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Daffe, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, “Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching,” IEEE International Conference on Robotics and Automation, pp. 3750–3757, 2018.
- [53] V. Satish, J. Mahler, K. Goldberg, “On-Policy Dataset Synthesis for Learning Robot Grasping Policies Using Fully Convolutional Deep Networks,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1357–1364, 2020.
- [54] D. Morrison, P. Corke and J. Leitner, “Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach,” Robotics: Science and Systems, 2018.
- [55] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, K. Goldberg, “Learning Ambidextrous Robot Grasping Policies,” Science Robotics, vol 4, no.26, 2019.
- [56] 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, Vol. 37, No. 4-5, pp. 421–436, 2018.
- [57] J. Mahler, J. Liang, S. Niyaz, M. Laskey et al., “Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics,” arXiv:1703.09312, 2017.
- [58] R. Matsumura, K. Harada, Y. Domae, and W. Wan, “Learning based industrial bin-picking trained with approximate physics simulator,” arXiv:1805.08936, 2018.
- [59] U. Viereck, A. Pas, K. Saenko, R. Platt, “Learning a visuomotor controller for real world robotic grasping using simulated depth images,” arXiv:1706.04652, 2017.
- [60] D. G. Lowe, “Local feature view clustering for 3D object recognition,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, doi: 10.1109/CVPR.2001.990541, 2001.