Single-Grasp Detection based on Rotational Region CNN

Shan Jiang[1*], Xi Zhao[1], Zhenhua Cai[1], Kui Xiang[1] and Zhaojie Ju[2]

[1] Wuhan University of Technology, Hubei, China
[2] University of Portsmouth, Portsmouth, U.K.
jswhut@163.com

Abstract. Object grasp detection is foundational to intelligent robotic manipulation. Different from typical object detection tasks, grasp detection tasks need to tackle the orientation of the graspable region in addition to localizing the region since the ground truth box of the grasp detection is arbitrary-oriented in the grasp datasets. This paper presents a novel method for single-grasp detection based on rotational region CNN (R^2CNN). This method applies a common Region Proposal Network (RPN) to predict inclined graspable region, including location, scale, orientation, and grasp/non-grasp score. The idea is to deal with the grasp detection as a multi-task problem that involves multiple predictions, including predict grasp/non-grasp score, the inclined box and its corresponding axis-align bounding box. The inclined non-maximum suppression (NMS) method is used to compute the final predicted grasp rectangle. Experimental results indicates that the presented method can achieve accuracies of 94.6% (image-wise splitting) and 95.6% (object-wise splitting) on the Cornel Grasp Dataset, respectively. This method outperforms state-of-the-art grasp detection models that only use color images.

Keywords: Robotic Grasp, Convolutional Neural Network, Region Proposal Network, Faster-RCNN, Rotational Region CNN.

1 Introduction

With the advance of robotics and relevant applications in industry and daily life, researchers pay more attention to robotic grasp which is foundational to robotic manipulation. When humans try to grasp an object for the first time, they can perceive, think, and probably figure out how to grasp it effectively. However, grasping a novel object is relatively challenging for robots as this task involves many subjects, including computer vision, robot kinematics, control science and path planning. Nowadays, most of robotic grasp schemas highly rely on the predefined programs, which simply depend on repeating a series of predetermined basic motions. Obviously, such schemas subject to a lack of generalization and robustness if objects or environments are varying. Therefore, practical robotic grasp manipulation requires more intelligent and robust strategies.

For the task of grasp detection, many studies [1-7], [17-19] focused on predicting grasp rectangles. Compared with single grasp point, a grasp rectangle contains more information including the graspable region as well as the orientation information. This orientation indicates a proper opening direction for parallel robotic gripper. Consequently, the existence of the orientation makes the problem of grasp detection distinguishable from other general object detection algorithms.

This paper aims to propose a new method to improve grasp rectangle detection using images for practical real-time robotic grasp. To achieve this goal, the authors employ convolutional neural network (CNN) and regional proposal network (RPN) to explore feasible models that can effectively determine inclined grasp region. By combining inspirations from some state-of-the-art methods [8-10], a deep learning method is presented to detect the grasp region of objects for accurate robotic manipulation. Experimental results based on the Cornel Grasp Dataset are discussed as well as future work to improve the presented method.

The major contributions of this paper are summarized as below:

(1) A new grasp detection method, which is specifically designed for inclined grasp region, is presented. This method considers the grasp detection problem as a triple-task problem by adding the axis-align box as well. It outperforms existing grasp detection methods that only use image information.

(2) The smaller anchor scales is added to cover the tiny objects and the inclined non-maximum suppression (inclined NMS) is adopted to select the optimal grasp rectangle.

2 Related Work

As an elemental manipulation, the task of robotic grasp has been extensively studied [11]. Most research can be divided into two categories: heuristic method and machine learning methods.

Ying Jiang et al. [1] adopted manually designed two-step color feature to achieve the detection result with 85.5% accuracy. Dogar and Goldfeder [12-13] used full physical simulation given 3D models to predict correct grasps. The results are not very satisfying whereas the process was very time-consuming. Traditional heuristic methods [1], [12-15] for feature extraction are not suitable enough for robotic grasp detection.

With the advance of computer vision and deep learning, some researchers studied the robotic grasp problem using those new technologies, which significantly improved both accuracy and computational speed. Ian Lenz et al. [2] proposed a two-stage cascaded system for detecting robot grasps, wherein a small deep network was used to generate some potential rectangles and then a larger deep network was used to select the top-ranked rectangle from these candidates. Joseph Redmon et al. [3] conducted the detection based on RGB-D images with a neural network inspired by the AlexNet to address the same problem as the former researchers. Sulabh Kumra et al. [4] used two 50-layer deep convolutional residual neural networks in parallel for RGB-D feature extraction, one for RGB feature and the other for depth information. Then these features were fused and fed into the detection network. Their research showed that the use of deep residual layers can extract better features from the input images than the ordinary convolutional layers. Di Guo et al. [5] associated each grid cell with several reference rectangles in different scales and ratios and then these reference rectangles were refined to their corresponding predicted rectangles. In follow-up study [6], they proposed a hybrid system that combined the vision and tactile information for robotic grasping. Furthermore, a new THU grasp dataset which contained the visual, tactile and grasp configuration information was collected, and the results showed that the tactile data can help improve the accuracy for grasp detection. Although the result was relatively outstanding, it was hard and too complex to conduct the grasp experiment as it required plenty of experiment instruments. Fu-Jen Chu et al. [7] presented a system that can be applied to both single-grasp and multi-grasp detection situation. In their research, they converted the problem of regression for grasp orientation into a problem of classification. The system quantitated the orientation into 19 categories and assigned the predicted rectangles to the corresponding classes. The prediction accuracy on RGB model was 94.4% and 95.5% on image-wise and object-wise splitting, respectively.

3 Rotated-RCNN for Single-Grasp Detection

3.1 Grasp Configuration

Similar to the previous work [1-7], [17-19], the five-dimensional grasp configuration is applied to this paper. The configuration consists of both position information and orientation information. The ground truth rectangle is defined as:

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

wherein, the (*x*, *y*) represents the coordinate of the center point, *w* and *h* mean the width and height of the rectangle respectively. The angle θ symbolizes the angle between the rectangle and the horizontal x-axis and the gripper can get close the object in this direction. Fig. 1 shows an example of the grasp configuration.

Different from normal object detection which the bounding box is axis-align, the ground-truth box of grasp detection is normally inclined. So how to tackle the orientation problem is the key for grasp detection.

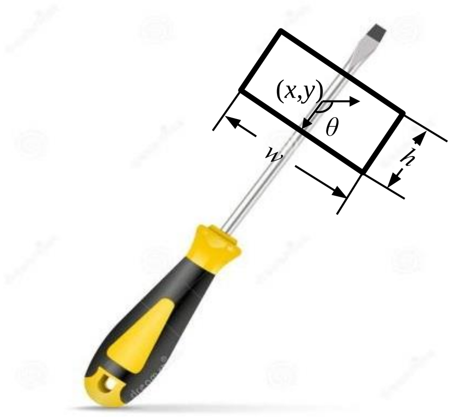

Fig. 1. Five-dimensional grasp configuration

3.2 Network Architecture

Different from the horizontal ground truth box, grasp rectangles are normally arbitrary-orientated since the object is placed on the platform randomly. Under the circumstances, the Rotated-RCNN is applied to tackle the orientation problem in the grasp detection problem. Fig. 2 shows the whole network architecture of the grasp detection system. The input images pass through multiple convolutional layers to produce the feature maps. The RPN is used to generate axis-align bounding boxes that encircle the graspable region. Considering about the diversity of the sizes and aspect ratios of the grasp rectangles, a smaller anchor scale is applied to the model. Experimental result shows the smaller anchor is effective in the grasp detection.

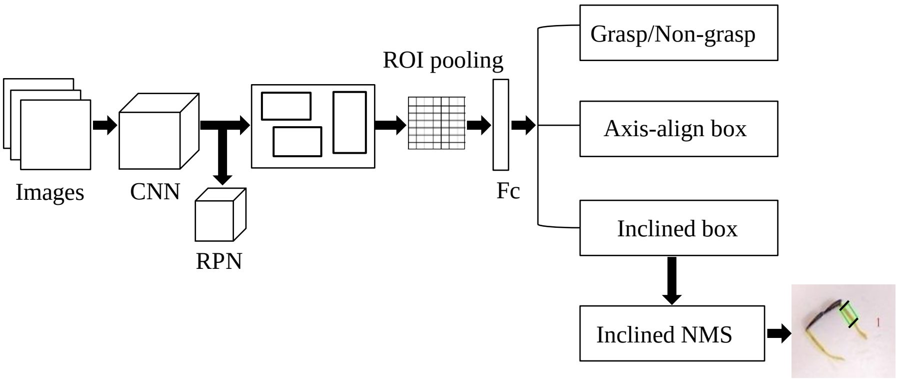

Fig. 2. The whole architecture of the grasp detection network

Two fully-connected layers are used for classification and regression in parallel. The region proposal produced by RPN is classified as grasp or non-grasp. The inclined bounding box and its associated axis-aligned bounding box get refined as well. The regression loss of the axis-align bounding box is considered in the whole loss function, the evaluation of [8] confirmed the effectiveness of this idea.

As the subject of the grasp detection is oriented rectangle, the inclined non-maximum suppression is utilized to post-process detection candidates as to obtain the output grasp rectangle.

3.3 Loss Function

The loss function in this paper contains two parts, the loss of the region proposal network L_{RPN} and the loss of grasp configuration detection L_{GCD}. The RPN generates axis-align region proposals that encircle the inclined graspable region. The loss function of the RPN consists of the classification loss and regression loss, defines as:

$$L_{RPN}\left(p_i, t_i\right) = \sum_i L_{RPN\_cls}\left(p_i, p_i^*\right) + \lambda_1 \sum_i p_i^* L_{RPN\_reg}\left(t_i, t_i^*\right) \tag{2}$$

wherein, L_{RPN_cls} defines the log loss of the proposal classification and L_{RPN_reg} is the smooth L1 loss of the proposal regression. p_i is the probability of the anchor belonging to the foreground. The ground truth label p_i^* is 1 if the anchor is positive and is 0 if the anchor is negative [9]. The regression loss is calculated only when the anchor is assigned to the foreground. t_i represents the four-dimensional coordinate vector of the predicted axis-align bounding box and t_i^* means that of the horizontal box rotated from the inclined ground truth box.

The loss function of the grasp configuration detection consists of three parts: the grasp/non-grasp classification loss, the loss of the axis-align box that encircles the graspable region and the loss of the inclined box. The loss function defines as:

$$\begin{aligned} L_{GCD}\left(\rho_i, \beta_i, \delta_i\right) = \sum_i L_{GCD\_cls}\left(\rho_i, \rho_i^*\right) + \lambda_2 \sum_i \rho_i^* L_{GCD\_regh}\left(\beta_i, \beta_i^*\right) \\ + \lambda_3 \sum_i \rho_i^* L_{GCD\_regi}\left(\delta_i, \delta_i^*\right) \end{aligned} \tag{3}$$

wherein, $L_{GCD\text{-}cls}$ means the log loss of the grasp classification. Grasps are labeled as 1 and others are assigned background. The parameter β=(β_x, β_y, β_w, β_h) means the predicted regression for axis-align bounding box for the graspable class and β^* means the true regression target. The parameter δ=(δ_x, δ_y, δ_w, δ_h, δ_θ) means the predicted regression vector for the inclined bounding box and δ^* is the corresponding ground truth grasp bounding box vector. Similarly, the regression loss is considered only when the proposal is assigned to the graspable class. The parameter λ_2 and λ_3 are used to balance these three kinds of loss. It should be noted that in all the experiments of this paper λ_2 and λ_3 are set 1.

The total loss for the end-to-end training of the grasp detection defines as:

$$L_{Total} = L_{RPN} + L_{GCD} \tag{4}$$

4 Experiment

4.1 Dataset

In order to evaluate the performance of the network compared with existing studies, the network is trained and tested on the standard Cornel Grasp Dataset. The Cornel Grasp Dataset applied to this research contains 885 images of 244 graspable objects and each object in these images is associated with several positive grasp rectangles and negative rectangles. In this research, the positive rectangles are defined as ground-truth box.

The five-fold cross-validation is conducted in the experiments and the dataset is divided in two different ways:

(1) Image-wise splitting divides the images into training set and validation set at random. This aims to test the generalization ability of the network to the new position and orientation of an object.

(2) Object-wise splitting divides the dataset at object instance level. The training and test dataset does not share the images of the same instance. This method aims to test the generalization ability of the network to the novel object.

In practice, both splitting methods give comparable performance. This may due to the similarity between different objects in the dataset [3].

4.2 Data Preprocessing

The training process for the deep neural network needs a large amount number of labeled data. Since the amount of the Cornel Grasp Dataset is insufficient, a process of data augmentation is required before the dataset is fed into the network. Several methods of data augmentation such as rotation, translation and crop have been adopted before the experiments. Firstly, a region of 320*320 pixels is center cropped from the original image. Secondly, the cropped region is padded with 50 pixels in both x and y directions. Then the padded image is translated with random pixel between -50 and 50 pixels in both x-axis and y-axis. Then the rotated image is rotated with a random angle between 0 ° to 360 °. Lastly, the image is resized to 512*512 pixels. Each original image extends to 25 images after the augmentation and these processed images will be sent to the input of the network. The label of the ground truth box is transformed as well.

4.3 Training

To improve the efficiency of the training process and avoid overfitting, the transfer learning is applied to this research. The network is initialized by the ResNet-50 pre-trained on the ImageNet.

The model is based on the GPU version of TensorFlow framework with cuda-8.0 and cudnn-5.1.0 package. The whole network is trained end-to-end for 200 epochs and the whole training and test process runs on a single NVIDIA GTX1080Ti. The initial learning rate is 0.001 with a weight decay of 0.0005 and the momentum of 0.9.

4.4 Evaluation Metric for Detection

The point metric and rectangle metric [3]-[4] are the most popular evaluation metrics in grasp detection on the Cornel Grasp Dataset. For the point metric, the distances between the center point of ground-truth grasps and center point of predicted grasp are considered. If any of these distances is less than the predefined threshold, the predicted grasp is regarded as a correct prediction.

Obviously, the point metric cannot comprehensively evaluate predicted grasp. This kind of metric does not evaluate the size and orientation of the predicted grasp and thus may overestimate the performance of the algorithm for grasp detection.

In this paper, the rectangle metric is chosen as the evaluation metric. In this metric, the predicted grasp is regarded to be correct if it satisfies both conditions:

(1) The angle difference between the predicted grasp and the ground-truth grasp is within 30 °.

(2) The Jaccard index of the ground-truth grasp and the predicted grasp is larger than 0.25.

The Jaccard index is defined as:

$$J\left(G,G^{*}\right)=\frac{Area\left(G\cap G^{*}\right)}{Area\left(G\cup G^{*}\right)} \tag{5}$$

The Jaccard index is similar to the Intersection over Union (IoU) threshold [7] for object detection. The *G* means the area of the top-ranked predicted grasp rectangle in this algorithm and G^{*} denotes the area of the ground-truth rectangle. $G \cap G^{*}$ is the intersection of these two rectangles and $G \cup G^{*}$ denotes the union of these two rectangles. Note that as the ground-truth grasp rectangles cannot be labeled exhaustively, the Jaccard index is 25 percent rather than 50 percent used in the normal object detection. A rectangle with the right orientation that only overlaps by 25 percent with one of ground truth boxes can still be considered as a reliable prediction. All the experiments are performed using this kind of rectangle metric.

5 Results and Discussion

Different from many methods adopted in the image augmentation for training dataset, the test dataset only uses the center crop. The model is evaluated by the metric mentioned above.

The result of self-comparison of the proposed algorithm with different parameters shows in Table Ⅰ and Table Ⅱ shows the comparison of this model and other previous works on the Cornel Grasp Dataset with the same evaluation metric. It should be noted that all of the results only consider about the single grasp of the object. In other words, the inclined bounding box with the highest confidence is set as the output grasp rectangle. It is clear that smaller anchor scale and inclined NMS can improve detection accuracy. In Table Ⅱ, the result shows the proposed model outperforms previous works with RGB images. On image-wise splitting, the accuracy is up to 94.8%, which is 0.4% higher than the up-to-date 94.4% accuracy [7] in grasp research. While on the object-wise splitting, the detection accuracy is 95.6%, which is 0.1% higher than the 95.5% accuracy of Chu's [7] work. Both Chu's work [7] and this approach are generated on the basis of the Faster-RCNN, wherein Chu's work converts the problem of the regression over the orientation to the problem of discretization orientation classification and this paper deals with this problem in a continuous manner. Moreover, this paper applies smaller anchor to improve the accuracy at the cost of little additional runtime.

Table 1. Result of network with different parameters

Anchor Scale	Inclined NMS	Prediction Accuracy(%)	
		Image-wise	Object-wise
(8,16,32)	No	92.5	94.2
	Yes	93.4	95.1
(4,8,16,32)	No	93.2	94.8
	Yes	94.8	95.6

Table 2. Single grasp evaluation

Approach	Prediction Accuracy(%)		speed
	Image-wise	Object-wise	fps
Jiang et al. [1]	60.5	58.3	0.02
Lenz et al. [2]	73.9	75.6	0.07
Redmon et al. [3]	88.0	87.1	3.31
Wang et al. [18]	81.8	N/A	7.10
Asif et al. [19]	88.2	87.5	-
Kumra et al. [4]	89.2	88.9	16.03
Mahler et al. [20]	93.0	N/A	~1.25
Guo et al. [5]	93.2	89.1	-
Chu et al.(Res50 RGB) [7]	**94.4**	**95.5**	8.33
Chu et al.(Res50 RGB-D) [7]	96.0	96.1	8.33
Ours(Res50 RGB)	**94.8**	**95.6**	7.25

As shown in Fig. 3, some positive rectangles were generated from the grasp detection system. The top row shows the ground truth grasp rectangles which are obtained from the Cornel Grasp Dataset. The red line in these pictures indicates the gripper's orientation. As shown in the picture, the number and the size of the grasp rectangles are varying and some of the rectangles are even small. Therefore, it is necessary to add smaller size to the anchor scale. The second row displays the top-ranked inclined grasp rectangle predicted by the detection system. The last row reveals all the inclined rectangles output from the detection system. The black line in the picture of the second and last rows means the gripper's orientation and the number in these pictures represents the number of the output rectangles. The results indicate that the detection network can accurately predict position and orientation of the grasp rectangles.

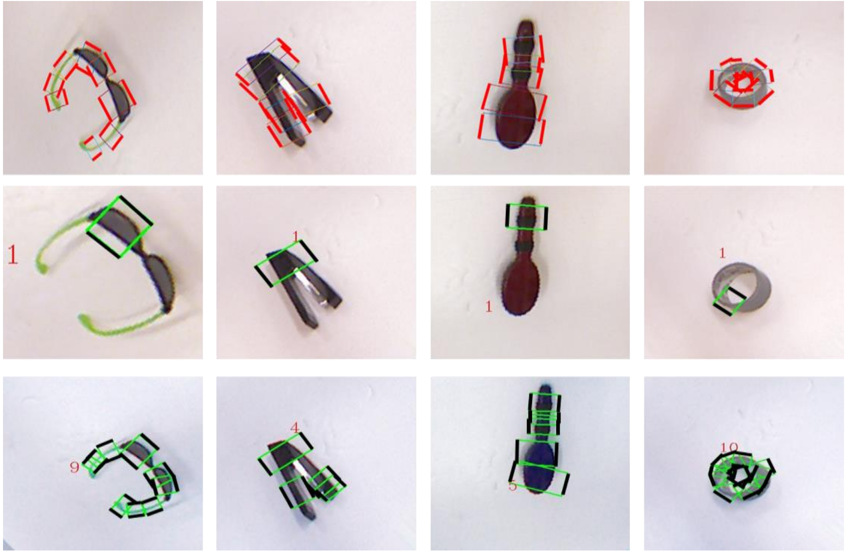

Fig. 3. Some positive examples from the network

Besides, some incorrect grasp rectangles are shows in Fig.4. The first row is the ground truth box and the other row shows the wrong results. It should be noted that the wrong results here means that the inclined rectangle does not meet the rectangle metric mentioned above. Although the left two pictures in the second row are assigned to be incorrect, as the grasp rectangle cannot be labeled completely, these outputs can be thought as proper prediction as well.

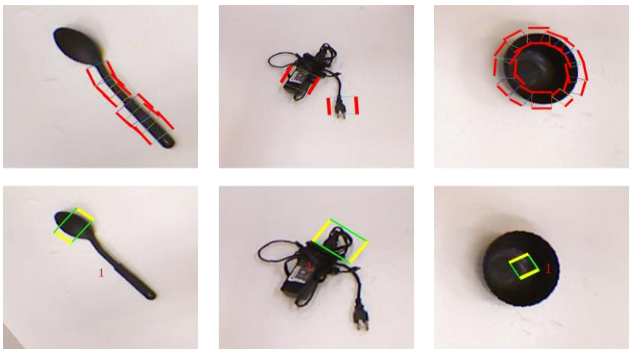

Fig. 4. Some incorrect prediction from the network

6 Conclusion

In this paper, a robust and accurate robotic grasp detection method based on CNN and RPN is presented. The architecture of the network is adapted from the R^2CNN which was originally designed to detect inclined scene text, which redefines the meaning of the network and shows the generalization of the network when it is comes to the grasp problem. Many modifications have been made to solve the grasp detection tasks, and the presented method is verified to effectively improve the accuracy of the grasp detection. The experimental results show that this novel network achieves an accuracy of 94.6% (image-wise splitting) and 95.6% (object-wise splitting), respectively. The network outperformed previous work with the same evaluation metric. Granted, the computational speed of the algorithm is not satisfactory enough and further work need to be conducted on the problem of shortening time as well as the practical grasp manipulation.

References

1. Jiang, Y., Moseson, S., Saxena, A.: Efficient Grasping from RGB-D images: Learning using a new rectangle representation. In 2011 IEEE International Conference on Robotics and Automation, pp. 3304-3311. IEEE(2011).
2. Lenz, I., Lee, H., Saxena, A.: Deep Learning for Detecting Robotic Grasps. The International Journal of Robotics Research, 34(4-5), 705-724(2015)
3. Redmon, J., Angelova, A.: Real-Time Grasp Detection Using Convolutional Neural Networks. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1316-1322. IEEE(2015).
4. Kumra, S., Kanan, C.: Robotic Grasp Detection using Deep Convolutional Neural Networks. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 769-776. IEEE(2017).
5. Guo, D., Sun, F., Kong, T., & Liu, H.: Deep Vision Networks for Real-time Robotic Grasp Detection. International Journal of Advanced Robotic Systems 14(1), (2016).
6. Guo, D., Sun, F., Liu, H., Kong, T., Fang, B., Xi, N.: A Hybrid Deep Architecture for Robotic Grasp Detection. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 1609-1614. IEEE(2017).
7. Chu, F. J., Xu, R., Vela, P. A.: Real-world Multi-object, Multi-grasp Detection. IEEE Robotics and Automation Letters, 3(4), 3355-3362. (2018).
8. Jiang, Y., Zhu, X., Wang, X., et al.: R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection. arXiv preprint arXiv:1706.09579. (2017).
9. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In Advances in neural information processing systems, pp. 91-99. (2015).
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. (2016).
11. Bohg, J., Morales, A., Asfour, T., Kragic, D.: Data-driven Grasp Synthesis—a survey. IEEE Transactions on Robotics 30(2), 289-309. (2013).
12. Dogar, M., Hsiao, K., Ciocarlie, M., Srinivasa, S.: Physics-based Grasp Planning through Clutter. (2012).

13. Goldfeder, C., Ciocarlie, M., Dang, H., Allen, P. K.: The Columbia Grasp Database. (2008).
14. Miller, A. T., Knoop, S., Christensen, H. I., Allen, P. K.: Automatic grasp planning using shape primitives. (2003).
15. Piater, J. H.: Learning visual features to predict hand orientations. (2002).
16. Girshick, R.: Fast R-CNN. In Proceedings of the IEEE international conference on computer vision, pp. 1440-1448. (2015).
17. Zhang, H., Zhou, X., Lan, X., Li, J., Tian, Z., Zheng, N.: A Real-time Robotic Grasp Approach with Oriented Anchor Box. arXiv preprint arXiv:1809.03873. (2018).
18. Pinto, L., Gupta, A.: Supersizing self-supervision: Learning to Grasp from 50k tries and 700 Robot hours. In 2016 IEEE international conference on robotics and automation (ICRA), pp. 3406-3413. IEEE. (2016).
19. Watson, J., Hughes, J., Iida, F.: Real-world, Real-time Robotic Grasping with Convolutional Neural Networks. In Annual Conference Towards Autonomous Robotic Systems, pp. 617-626. Springer, Cham (2017).
20. Wang, Z., Li, Z., Wang, B., Liu, H.: Robot Grasp Detection using Multimodal Deep Convolutional Neural Networks. Advances in Mechanical Engineering 8(9), (2016).
21. Asif, U., Bennamoun, M., Sohel, F. A.: RGB-D Object Recognition and Grasp Detection using Hierarchical Cascaded Forests. IEEE Transactions on Robotics 33(3), 547-564. (2017).
22. Mahler, J., Liang, J., Niyaz, S., et al.: Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. arXiv preprint arXiv:1703.09312. (2017).