A Novel Generative Convolutional Neural Network for Robot Grasp Detection on Gaussian Guidance

Yuanhao Li\textsuperscript{a}, Graduate Student Member, IEEE, Yu Liu\textsuperscript{b}, Zhiqiang Ma\textsuperscript{c}, Member, IEEE, and Panfeng Huang\textsuperscript{d}, Senior Member, IEEE

Abstract—The vision-based grasp detection method is an important research direction in the field of robotics. However, due to the rectangle metric of the grasp detection rectangle's limitation, a false-positive grasp occurs, resulting in the failure of the real-world robot grasp task. In this article, we propose a novel generative convolutional neural network model to improve the accuracy and robustness of robot grasp detection in real-world scenes. First, a Gaussian-based guided training method is used to encode the quality of the grasp point and grasp angle in the grasp pose, highlighting the highest-quality grasp point position and grasp angle and reducing the generation of false-positive grasps. Simultaneously, deformable convolution is used to obtain the shape features of the object in order to guide the subsequent network to the position. Furthermore, a global–local feature fusion method is introduced in order to efficiently obtain finer features during the feature reconstruction stage, allowing the network to focus on the features of the grasped objects. On the Cornell Grasping datasets and Jacquard datasets, our method achieves an excellent performance of 99.0% and 95.9% detection accuracy, respectively. Finally, the proposed method is put to the test in a real-world robot grasping scenario.

Index Terms—Gaussian-based guided training (GGT), global–local feature fusion (GLFF), robotic grasp detection.

I. INTRODUCTION

W

ITH the development of information technology and control technology, robots are becoming more important in industrial manufacturing [1], underwater detection [2], social service [3], and defect detection [4]. Grasping objects is the most widely used robot task and one of the most challenging technologies in robot operation. Grasping an object is very basic behavior for humans, but, for robots, it involves a series of visual detection, instrument and measurement, and control systems. To complete the grasping task, the robot must first perceive the object, just as a human does with his eyes, in order to determine important information, such as its position, direction, and grasping position. As a result, the grasp detection system serves as a starting point from which to plan future grasping paths and carry out the whole grasping action.

The key to the success of the grasping operation is obtaining an accurate grasping position. The early research on grasp detection is based on the analysis method [5], which is based on the analysis of geometric and physical characteristics, as well as manual design features to achieve the best grasp point selection. Although grasping accuracy is improved for specific objects, the manually designed features are cumbersome and time-consuming, with poor generalization ability and weak universality. Deep learning development provides a new way to solve generalization problems and has made significant progress in many fields, such as object detection [6], [7], [8], demonstrating a strong ability for feature extraction. Object detection methods, on the other hand, typically only output a unique ground truth to express the detection position and category of objects. However, this does not apply to the grasped object because an object does not usually have only one grasping position. As a result, the grasping representation of objects is designed to effectively deal with the robot grasping task in an unstructured environment. In 2014, Lenz et al. [9] proposed a rectangle box description of the grasping position with reference to Jiang et al. [10], as shown in Fig. 1, to transform grasping detection into a problem similar to object detection. They achieved 75% accuracy on the Cornell Grasping datasets. Most of the subsequent grasp
In order to solve the above problems, a Gaussian-based guided training (GGT) method is used to encode the quality of the grasp points and angles in a grasp pose that directs the network’s learning to focus on improving the quality of grasp points and angles for points on the object’s center line and for points that are perpendicular to the object’s center line. On this basis, a generative convolutional neural network model for grasping detection of real-scene robots is proposed. Simultaneously, a deformable convolution is introduced to obtain the object’s shape features, which will be used to guide the subsequent network back to its original position. In addition, a global–local feature fusion (GLFF) method is introduced to efficiently obtain finer features in the feature reconstruction stage, allowing the network to focus on the features of grasped objects. Finally, we propose a multiscale loss function for grasp detection that predicts grasp positions at various scales in order to adapt to grasping objects of various scales in the real world.

In summary, the main contributions of this article are given as follows.

1) We use a GGT method to improve the existing grasping representation, which standardizes the position and angle information of the grasping rectangle to the maximum extent possible and significantly improves the grasping success rate in real-world tests.

2) To obtain refined object features, an attention network for GLFF is proposed, which divides feature enhancement into global and local branches.

3) On the Cornell Grasp dataset and the Jacquard dataset, our method achieves excellent results with 99.0% and 95.9% detection accuracies, respectively. Simultaneously, the superiority of our method is demonstrated on a real robot grasp system.

II. RELATED WORK

The research on object grasping position detection began in the 1980s, and most of the early studies were mainly focused on the detection of grasping points, which was using heuristic algorithms to grasp and detect objects with specific shapes [13], [14]. These methods could only achieve good results for objects with shape characteristics, they were not able to provide a definite description of the grasping method, and the generalization performance was poor. Jiang et al. [10] proposed a rectangle box description of grasping position, which transformed the problem of object grasping position detection into the problem of finding a rectangle in image space. However, this model required to design visual features manually for specific objects, instead of extracting the features of the grasped region in a self-learning method.

The deep learning method has been shown to be effective for a wide range of perceptual problems [15], [16], which allows the perceptual system to learn mappings from some feature sets to various visual features. More researchers are gradually introducing deep learning into grasp detection, applying learning methods to vision, and introducing learning methods to score for grasp quality. Lenz et al. [9] first found all possible grasp rectangles by using a shallow convolutional neural network and then a deep convolutional neural network to find a grasp rectangle that is in line with the rectangle metric and, thus, a good grasp rectangle. The rectangle metric is given as follows: 1) the angle difference between the predicted and ground-truth grasp angles is less than 30° and 2) the Jaccard index of the predicted and ground-truth grasps is greater than...
25%; the Jaccard index is defined as
\[
Jacc = \frac{\text{area}(A \cap B)}{\text{area}(A \cup B)}
\] (1)
where \( A \) and \( B \) represent the predicted and ground truth, respectively. All subsequent studies are verified on the rectangle metric.

Redmon and Angelova [17] abandoned the framework based on the sliding window box and used the powerful feature extraction capability of the AlexNet to transform the prediction problem of the grasp region parameters into a regression problem, but this method could only predict a single grasp region for the input image, and its mapping mechanism often led to the average effect of grasping prediction results. In [18], a shared convolutional neural network model was proposed to simultaneously complete the detection and classification tasks of the grasped region, and the results showed that the performance of the parameter shared network was due to the single detection network. Chu et al. [19] used the “Grasp Region Proposal Network” to predict the undirected grasp candidate regions and then delineated the rotation angle corresponding to the grasp candidate region from the perspective of classification, which could predict the grasp candidate box of multiple objects at the same time. Lan et al. [20] based on a fully convolutional grasp part detection network on the directional anchor box, which would be more suitable for the detection of multilayer grasping parts by adding angle information to the preset anchor box. On the basis of Lenz et al., Morrison et al. [21] directly generated pixel-level representations of grasping parameters through the grasping generative convolutional neural network and proposed a new type of representation method for grasp detection, thereby avoiding the sampling process of grasp candidate regions, which improved the detection efficiency significantly, and achieved a detection accuracy of 73% in the Cornell dataset with only depth data as input. Kumra et al. [22] took RGB-D as input and proposed a novel generative residual convolutional neural network (GR-ConvNet) model based on [21], which achieved a detection accuracy of 97.7% on the Cornell dataset. Liu et al. [23] used a robot grasping pose detecting method based on a cascade neural network, which can be applied to unseen objects in a real environment. It has achieved excellent performance in both public datasets and real cluttered scenes. Chen et al. [12] proposed a two-stage grasp detection method, which only takes RGB images as input and utilizes low-level features and grasping criteria to select a small number of grasp candidates, and introduced a lightweight CNN model to evaluate grasp quality. Cheng et al. [24] proposed a novel end-to-end grasp generation model based on the key-point detection strategy to restore the grasp rectangles densely. They achieved a detection accuracy of 95.4% and 91.8% on the Cornell dataset and the Jaccard dataset, respectively.

Although the above methods have achieved good performance in public datasets, they lack detailed analysis of the overall object and grasp region, and the limitations of the rectangle metric will lead to excessive false-positive grasps, resulting in actual grasping failures. In the meantime, the focus of grasping is on the object to be grasped rather than the background information of the image or other targets. Therefore, we need to accurately determine the grasp region to reduce the area of grasp detection and improve the efficiency and robustness of grasp detection.

III. PROBLEM FORMULATION

A. Grasp Representation

Since Jiang et al. [10] proposed rotating rectangle boxes to represent the grasping pose, many researchers have built a grasp detection network based on the object detection network that can output multiple grasp rectangle boxes [9], [17], [24], [25]. The observation demonstrates that the gripper’s width \( h \) is a fixed parameter. Furthermore, the types of grippers chosen in each literature are distinct. As a result, we use a simplified representation of robot grasping similar to Morrison et al. [21] and define the representation of robot grasping as
\[
G = (p, \psi, w, q)
\] (2)
where \( p = (x, y, z) \) is the central position of the robot gripper in Cartesian coordinates, \( \psi \) is the gripper’s rotation angle around the \( z \)-axis, \( w \) is the opening width of the gripper, and \( q \) is grasp confidence. Compared with the 5-D grasp representation, (2) can measure the probability of a successful grasp and select the grasp with the highest quality value without evaluating multiple grasp candidates. Assuming that the camera intrinsics parameters and calibration results are known, the robot derives the grasp pose in the plane from the depth image \( I \) of size \( H' \times W' \)
\[
\hat{g} = (\hat{p}, \hat{\psi}, \hat{w}, \hat{q})
\] (3)
where \( \hat{p} = (\hat{x}, \hat{y}) \) is the pixel coordinate of the grasp center point, \( \hat{\psi} \) is the rotation angle of the camera reference coordinate system around the \( z \)-axis, \( \hat{w} \) is the opening width of the grasp gripper, and \( \hat{q} \) is the grasp confidence. In order to perform grasping in image space on the robot, we convert \( \hat{g} \) into grasp pose \( g \) in the real world by the following formula:
\[
g = t_{ic}(t_{oc}(\hat{g}))
\] (4)
where \( t_{ic} \) is the transformation matrix from the image plane coordinate system to the camera coordinate system and \( t_{oc} \) is the transformation matrix from the camera coordinate system to the robot (world) coordinate system. We can do multiple grasps of images in the grasp dataset in image space, which can be denoted as
\[
G = \{\Phi, W, Q\} \in \mathbb{R}^{3 \times W \times H}
\] (5)
where \( \Phi, W, Q \) and \( Q \) are respective each \( 1 \times W \times H \) and contain the \( \phi, \tilde{w}, \) and \( \tilde{q} \) values in each pixel.

B. Gaussian-Based Guided Training Method

Since the discrete rectangle box cannot cover all grasp positions on the object, Morrison et al. [21] mark all pixels in the center 1/3 area of the grasp rectangle as grasp points with a grasp quality of 1, the grasp point in the same area has the same grasp angle and width as the grasp rectangle, and then, the pixel-level grasp pose is output by predicting the
obtain the category corresponding to the labeled grasp angle as a radian value in the interval $[0, \pi]$. We predict the grasp angle by classification. We define the grasp angle $\theta$ at the $i$th position represents the grasping quality of the point closest to the grasping angle $\theta$ has the highest quality. Given that angles close to the grasping angle $\theta$ still have high grasp quality, we use a Gaussian distribution with mean $\theta$ to determine the value of $\Theta$ in the vector. For the value $\Theta_i$ at the $i$th position of vector $\Theta$, it is defined as follows:

$$
\Theta_i = \exp\left(-\frac{(i - k)^2}{2\sigma_i^2}\right), \quad |i - k| \leq th
$$

where $th$ is the error range of the grasp angle. Considering that the object can still be grasped when the closing angle of the manipulator is $30^\circ$ different from the marked grasp angle, we set $th$ to 3. When $|i - k| > th$, $\Theta_i = 0$ as shown in Fig. 5.

IV. PROPOSED METHOD

In this section, we propose a novel network architecture for grasp detection, as shown in Fig. 6. To obtain more information about the image, the input is first passed through a convolutional layer for feature extraction and downsampling. To learn the shape information of different objects, we use deformable convolution to replace the last convolutional layer in the feature extractor. Simultaneously, we propose a global–local attention network for feature fusion in the feature enhancement stage after feature reconstruction. The convolution layer is used to obtain global feature information.
through the global feature aggregation block (GFAB), and global average pooling and global feature aggregation are used to obtain global feature information. The final enhanced feature map is obtained by using convolutional layers and dense connections in the local feature enhancement block (LFEB). In the output prediction stage, we predict the grasp values of the feature maps of three different scales, obtain the quality, angle, and width under different scales, and calculate the multiscale loss with the ground truth to improve the ability of the network to grasp detection objects under different scales. Each module is described in detail in the following.

A. Network Architecture

Our proposed network is composed mainly of four parts: feature extraction, feature reconstruction, feature enhancement, and output prediction. In the feature extraction module, we use two basic blocks consisting of $3 \times 3$ convolutional layers with stride $= 2$ to get the downsampled feature maps, whose height and width become 1/4 of the input images. After the downsampling, five residual blocks are used to further extract features. Generally, the shape of the object is not fixed, and effective modeling of the object improves grasping performance. As a result, in the last stage of feature extraction, deformable convolution is used to obtain objects’ feature information to reduce the interference of background features, which can improve the modeling ability of irregular objects. A 2-D convolution process with a convolution kernel size of $3 \times 3$ is often divided into two parts. Each point on the feature map $F_i$ is first sampled in a grid $R$, and then, these sampling points are weighted by the weight $w$. $R$ is the convolution kernel’s receptive field, and $R$ of $3 \times 3$ convolution kernel is represented as follows:

$$R = \{(−1, −1), (−1, 0), \ldots, (0, 1), (1, 1)\}. \quad (10)$$

For point $P_0$ on the feature map $F_i$ extracted from each grasped image, the equation can be represented as follows:

$$F_o(p_0) = \sum_{p_n \in R} w(p_n) \cdot F_i(p_0 + p_n + \Delta p_n). \quad (11)$$

where $p_n$ is sampled position. In deformable convolution, the grid $R$ is accompanied by different offsets $\{\Delta p_n \mid n = 1, \ldots, N\}$, where $N$ is the number of points in $R$. Thus, (11) can be defined as

$$F_o(p_0) = \sum_{p_n \in R} w(p_n) \cdot F_i(p_0 + p_n + \Delta p_n). \quad (12)$$

We perform bilinear interpolation sampling on each sampling point to obtain the final output feature map $F_o$ to learn the shape features of the object.

In the feature enhancement stage, we perform GLFF on the feature map to obtain finer object information and attenuate unwanted noise. The module consists of GFAB and LFEB, the details are described in Section IV-B. Each feature enhancement module is followed by a feature reconstruction module that uses transposed convolution to restore the image’s size. Finally, we make an output prediction and calculate the loss of multiple scales using different scales.

B. Global–Local Feature Fusion Module

Following the feature extraction stage, the image’s rich feature information frequently contains some redundant noise. A lot of attention mechanism models have been proposed in the field of computer vision to solve the problem of feature redundancy, and let the model focus on more interesting parts of the image. To ensure that the model is focused on the object rather than on the background noise during grasping detection, we design a GLFF module during the feature enhancement stage to extract more useful information from the grasped image. It is composed primarily of the GFAB and the LFEB.

1) Global Feature Aggregation Block: The GFAB module is divided into two branches: global channel weighting and local feature extraction. For the input feature map $F_{in}^{g}$, after passing through the convolutional layer through the channel weighting branch and Relu activation layer, the global average pooling and fully connected layer are used to aggregate the information of global features to obtain the weight $\alpha$ on the channel; through the $1 \times 1$ convolution layer and Relu activation layer of the local feature extraction branch, feature map $F_{in}^{l}$ obtains a local feature map $F_{lifo}^{in}$ and then products with $\alpha$ on the channel to obtain the output $F^{out}_{GFAB}$ of GFAB via the final
has three components of quality, angle, and width. In each scale, we calculate smooth L1 loss with the label of article, we use a multiscale loss to predict the corresponding output as $\hat{y}_s = \{\hat{y}_{s1}, \ldots, \hat{y}_{sn}\}$, respectively.

### C. Loss Function

For the input image $I$, the grasp label can be represented as $Y = \{y_1, \ldots, y_n\}$. After the grasp network is proposed, we can get the corresponding output as $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_n\}$. In this article, we use a multiscale loss to predict the corresponding output $\hat{Y}^s$ at three stages of feature reconstruction, where $s = 1, 2, 3$, corresponding to four times downsampling, two times downsampling, and original size, respectively. In each scale, we calculate smooth L1 loss with the label of the corresponding scale. For a certain scale $s$, the loss function can be defined as

$$L^s(\hat{Y}^s, Y^s) = \sum_{i=1}^{n} l_1(\hat{y}^s_i - y^s_i)$$

(15)

where $n$ is the number of grasp positions, and each prediction has three components of quality, angle, and width. $l_1$ represents smooth L1 loss; it can be defined as

$$l_1(x) = \begin{cases} \frac{(\sigma x)^2}{2}, & \text{if } |x| < 1 \\ |x| - 0.5/\sigma^2, & \text{otherwise.} \end{cases}$$

(16)

In the smooth L1 loss, $\sigma$ represents the hyperparameter, which is used to adjust the smooth index. The total loss $L$ can be formulated as

$$L = \frac{1}{s} \sum_{i=1}^{s} L^i$$

(17)

where $s = 3$, which represents three different scales.

## V. EXPERIMENTS AND RESULT

### A. Datasets

The Cornell Grasping dataset and the Jacquard dataset are widely used as verification standards for robotic grasp detection. As a result, we train our method on these two datasets and compare it to other algorithms.

1) Cornell Grasp Dataset: The Cornell Grasp dataset contains 240 different objects, a total of 885 images, and point cloud data in the global coordinate system. The image and point cloud data are aligned. Each image is labeled with multiple good ground truths of grasp points, with a total of 5110 positive and 2909 negative grasp rectangles. As with many previous studies [9], [12], [18], the dataset is split into the training set and the test set into two distinct categories. The training set contains 708 images, and the test set contains 177 images.

1) Imagewise Split: The datasets are randomly divided into training sets and test sets. This is mainly to test the adaptability of the network model in detecting the same object at different positions and angles.

2) Objectwise Split: The networks are split according to object instances, and the generalization of the model to unseen objects is tested by using data that did not appear in the training set before.

In addition, compared with other deep learning datasets, the Cornell dataset is very small. Therefore, in order to avoid overfitting the model, we expand the dataset by cropping, random flipping, and translation before training the network.

2) Jacquard Dataset: The Jacquard dataset is a simulation grasp dataset nearly 50 times larger than the Cornell dataset, including 54485 images of 11619 objects. Simultaneously, it has multiple data types, including rgb-d and mask, which does not need data enhancement, such as the Cornell dataset. The dataset is split into training and test sets in a 5:1 ratio.

### B. Implementation Details

Our method is implemented in Pytorch 1.5.0, and the experimental platform is based on a single NVIDIA GeForce RTX 2080Ti (Pascal architecture with 12-GB memory) and the Ubuntu 16.04 operating system. During the training phase, we used the Adam optimizer to propagate back with a learning rate of 0.001 at the start and total training epochs of 60. When the loss is no longer reduced in ten consecutive batches, the initial learning rate is reduced to a tenth of its original value. However, we notice that the deformable convolutions increase the computational cost when applied to our method. As a result, we use gradient accumulation (GA) to reduce training’s computational cost. By reducing the batch size to 1/2 of the original, the parameters are optimized every two batches of
training to achieve the purpose of reducing the GPU memory usage. The results of the tests are shown in Table I.

To conduct an objective evaluation of our work, we use the same evaluation index as many previous studies [see (1)]. Due to the limitations of the rectangle metric and in order to reduce the false-positive grasps, we will improve the standard, that is, increase the Jaccard threshold to 0.30, 0.35, and 0.4 and reduce the angle difference between predicted and ground truth, and then test the performance on two public datasets. In this article, the Jaccard index is only used to evaluate the model, not to the matching strategy.

C. Result and Analyses

To verify the performance of our proposed method, we compare it with the previous algorithms on the Cornell Grasp dataset, and the results are shown in Table II. Our method achieves 99.0% and 98.3% detection accuracy in imagewise and objectwise, respectively, and achieves state-of-the-art performance. In addition, the inference time of our method is only 10 ms, which indicates real-time performance. Similarly, we also carry out experimental verification on the Jacquard dataset, as shown in Table III. In addition, our method also achieves excellent performance, reaching 95.9% detection accuracy.

To further verify the suppression of false-positive grasps by our method, we increase the Jaccard index in the rectangle metric to 0.3, 0.35, and 0.4, and the angle difference between the prediction and the ground truth decreases in order. Take Cornell as an example, and compare it with several excellent grasping methods, as shown in Fig. 9. It is worth noticing that, while the rectangle metric has increased, the performance of other ways has deteriorated to varied degrees, but our method still retains a 93.9% accuracy at Jaccard > 0.4, which is far better than all other methods, and the benefit becomes more and more obvious with the increase in the Jaccard value. Similarly, we gradually reduce the angle difference between the predicted and ground-truth grasp angles to 10°.
Fig. 10. Experimental results on the Cornell Grasp dataset under different angles with various methods.

Fig. 11. Effect of different modules on network performance.

Fig. 12. Examples of detection results on Cornell. The columns represent the quality, width, and angle output, respectively. The last column’s green grasp rectangle is ours, while red is [12].

TABLE IV

| Component          | Operating range | Weight | Other specifications |
|--------------------|-----------------|--------|----------------------|
| Kinect V2          | 0.5-4.5m        | 1.5kg  | 1920 × 1080, 30fps  |
| KUKA LBR iiwa r820 | 0.820mm         | 29.9kg | 7 DOF, 14kg load    |
| Shadow Hand Lite   | 16 joints       | 2.4kg  | 13 DOF, 4kg load    |

D. Real-World Robotic Grasping

To verify whether the method proposed in this article can improve the success rate of real robot grasping, we build a reasonable experimental system for verification, which is composed of KUKA (LBR iiwa 14 r820), a shadow hand, and a platform. A Kinect v2 real-time camera was used to acquire image data. Details of the specifications of different components in the experimental system are shown in Table IV. To obtain a more precise measurement without encroaching on the robot’s space, we mounted the camera on the robot’s left front and slightly above its line of sight to ensure global information throughout the experiment. The robot experimental system is shown in Fig. 13.
VI. Conclusion

In this article, we present a novel generative convolutional neural network model to increase the accuracy and robustness of real-world robot grasping detection tasks. To begin with, a Gaussian distribution is employed, which standardizes the position and angle information of the grasping rectangle to the maximum extent. On this basis, a deformable convolution and a global local feature fusion method are presented to guide the network’s attention to the grasped object’s features. Our method outperforms other methods on the Cornell and Jacquard datasets. Finally, we perform a real-world scenario experiment to prove the efficacy of the proposed method.

REFERENCES

[1] M. Catelani, L. Ciani, D. Galar, and G. Patrizi, “Optimizing maintenance policies for a yaw system using reliability-centered maintenance and data-driven condition monitoring,” IEEE Trans. Instrum. Meas., vol. 69, no. 9, pp. 6241–6249, Sep. 2020.
[2] Y. Wang et al., “Real-time underwater onboard vision sensing system for robotic gripping,” IEEE Trans. Instrum. Meas., vol. 70, pp. 1–11, 2021.
[3] B. Cheng, W. Wu, D. Tao, S. Mei, T. Mao, and J. Cheng, “Random cropping ensemble neural network for image classification in a robotic arm grasping system,” IEEE Trans. Instrum. Meas., vol. 69, no. 9, pp. 6795–6806, Feb. 2020.
[4] B. Fang, X. Long, F. Sun, H. Liu, S. Zhang, and C. Fang, “Tactile-based fabric defect detection using convolutional neural network with attention mechanism,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–9, 2022.
[5] J. Bohg, A. Morales, T. Asfour, and D. Kragic, “Data-driven grasp synthesis—A survey,” IEEE Trans. Robot., vol. 30, no. 2, pp. 289–309, Apr. 2014.
[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 779–788.
[7] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 1–9.
[8] S. Hare, A. Saffari, S. Golodetz, V. Vineet, M. Cheng, and P. Torr, “Struck: Structured output tracking with kernels,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 10, pp. 2096–2109, Oct. 2015.
[9] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” Int. J. Robot. Res., vol. 34, no. 4–5, pp. 705–724, 2015.
[10] Y. Jiang, S. Moseson, and A. Saxena, “Efficient grasping from RGBD images: Learning using a new rectangle representation,” in Proc. IEEE Int. Conf. Robot. Autom., Shanghai, China, May 2011, pp. 3304–3311.
[11] Y. Li, P. Huang, Z. Ma, and L. Chen, “A context-free method for robust grasp detection: Learning to overcome contextual bias,” IEEE Trans. Ind. Electron., vol. 69, no. 12, pp. 13121–13130, Dec. 2022.
[12] L. Chen, P. Huang, Y. Li, and Z. Meng, “Edge-dependent efficient grasp rectangle search in robotic grasp detection,” IEEE/ASME Trans. Mechatronics, vol. 26, no. 6, pp. 2922–2931, Dec. 2021.
[13] K. B. Shimoga, “Robot grasp synthesis algorithms: A survey,” Int. J. Robot. Res., vol. 15, no. 3, pp. 230–266, 1996.
[14] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sadasang, and A. Ng, “Grasping novel objects with depth segmentation,” in Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., Taiwan, Oct. 2010, pp. 2578–2585.
[15] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” Commun. ACM, vol. 64, no. 3, pp. 107–115, Mar. 2021.
[16] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” ACM Comput. Surv., vol. 52, no. 1, pp. 1–38, 2019.
[17] J. Redmon and A. Angelova, “Real-time grasp detection using convolutional neural networks,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), Seattle, WA, USA, May 2015, pp. 1316–1322.
[18] D. Guo, T. Kong, F. Sun, and H. Liu, “Object discovery and grasp detection with a shared convolutional neural network,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), Stockholm, Sweden, May 2016, pp. 2938–2943.
[19] F.-J. Chu, R. Xu, and P. A. Vela, “Real-world multi-object, multi-grasp detection,” 2018, arXiv:1802.00520.
[20] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, “Fully convolutional grasp detection network with oriented anchor box,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Madrid, Spain, Oct. 2018, pp. 7223–7230.

[21] D. Morrison, P. Corke, and J. Leitner, “Learning robust, real-time, reactive robotic grasping,” Int. J. Robot. Res., vol. 39, nos. 2–3, pp. 183–201, Mar. 2020.

[22] S. Kunura, S. Joshi, and F. Sahin, “Antipodal robotic grasping using generative residual convolutional neural network,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Las Vegas, NV, USA, Oct. 2020, pp. 9626–9633.

[23] D. Liu, X. Tao, L. Yuan, Y. Du, and M. Cong, “Robotic objects detection and grasping in clutter based on cascaded deep convolutional neural network,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–10, 2022.

[24] H. Cheng, Y. Wang, and M. Q.-H. Meng, “A robot grasping system with single-stage anchor-free deep grasp detector,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–12, 2022.

[25] W. Shang, F. Song, Z. Zhao, H. Gao, S. Cong, and Z. Li, “Deep learning method for grasping novel objects using dexterous hands,” IEEE Trans. Cybern., vol. 52, no. 5, pp. 2750–2762, May 2022.

[26] H. Cao, G. Chen, Z. Li, J. Lin, and A. Knoll, “Lightweight convolutional neural network with Gaussian-based grasping representation for robotic grasping detection,” 2021, arXiv:2101.10226.

[27] S. Wang, X. Jiang, J. Zhao, X. Wang, W. Zhou, and Y. Liu, “Efficient fully convolution neural network for generating pixel wise robotic grasps with high resolution images,” in Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO), Dali, China, Dec. 2019, pp. 474–480.

[28] Y. Song, L. Gao, X. Li, and W. Shen, “A novel robotic grasp detection method based on region proposal networks,” Robot. Comput.-Integr. Manuf., vol. 65, no. 10, 2020, Art. no. 101963.

Yuanhao Li (Graduate Student Member, IEEE) received the M.S. degree in measurement and control technology and instrumentation from Northwestern Polytechnical University, Xi’an, China. He is currently pursuing the Ph.D. degree with the School of Astronautics, Northwestern Polytechnical University, and the National Key Laboratory of Aerospace Flight Dynamics, Xi’an.

His research interests include machine learning, computer vision, object detection, and robotics.

Yu Liu received the B.S. degree in instrumental science and the master’s degree in control science from Tsinghua University, Beijing, China, in 2005 and 2008, respectively.

He is currently a Research Assistant with the Beijing Institute of Tracking and Telecommunication Technology, Beijing, China. His research interests include robot dynamics, spatial operation, and satellite ground applications.

Zhiqiang Ma (Member, IEEE) received the B.S. and M.S. degrees in control engineering from Northwestern Polytechnical University, Xi’an, China, in 2011 and 2014, respectively, and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2018.

He is currently an Associate Professor with the Research Center for Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University. His current research interests include nonlinear dynamics and control, tethered space systems, tele-robotics systems, and mechatronics.

Panfeng Huang (Senior Member, IEEE) received the B.S. and M.S. degrees from Northwestern Polytechnical University, Xi’an, China, in 1998 and 2001, respectively, and the Ph.D. degree in automation and robotics from The Chinese University of Hong Kong, Hong Kong, in 2005.

He is currently a Professor with the Research Center for Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University. His research interests include tethered space robotics, intelligent control, machine vision, and space teleoperation.