Robotic Objects Detection and Grasping in Clutter Based on Cascaded Deep Convolutional Neural Network

Dong Liu, *Member, IEEE*, Xiantong Tao, Liheng Yuan, Yu Du, and Ming Cong

***Abstract*— The complex and changeable robotic operating environment will often cause the low success rate or failure of the robot grasping. This article proposes a grasp pose detection method based on the cascaded convolutional neural network, which can be applied to grasp unknown irregular objects under unstructured environment. The grasping feature and the grasp position candidate bounding-boxes of objects are extracted by Mask-RCNN. To guarantee the generalization and improve the detection rate, grasp angle estimation network Y-Net is proposed to accurately obtain the grasp angle. To solve the problem of insufficient accuracy of grasping position, grasping feasibility evaluation network Q-Net is proposed for acquiring the grasp quality distribution. Finally, the optimal grasp posture is obtained for the robotic object grasping task in cluttered scenes. Experiments are validated in Cornell datasets, Jacquard datasets, and real environments, respectively. The experimental results show that the proposed method can quickly calculate the robot posture for irregular objects with random poses and different shapes. Compared to the previous methods, it has considerable improvement in grasp accuracy and speed. The method can be applied to object grasping scenarios in cluttered scenes and has strong stability and robustness.**

***Index Terms*— Cascaded deep convolutional neural network, cluttered scenes, robotic grasping, unstructured environment.**

I. Introduction

ROBOTICS is widely used in industrial production and service fields, such as parts assembly, workpiece sorting, etc. Robotic grasping is an important way for robots to interact with the environment. However, the operating environment of the robot is complex and changeable, such as the irregular shape of objects or the stacked or truncated objects will often cause the low success rate or failure of the robot grasping. Therefore, how to achieve high-precision and rapid robotic grasping in cluttered scenes is still a big challenge.

Manuscript received August 13, 2021; revised October 26, 2021; accepted November 12, 2021. Date of publication November 30, 2021; date of current version March 2, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61873045, in part by the Central Government Guiding Local Science and Technology Development Project of Liaoning Province under Grant 2021JH6/10500144, and in part by Dalian Sci&Tech Innovation Foundation Program under Grant 2019J12GX043. The Associate Editor coordinating the review process was Dr. Damodar Reddy Edla. *(Corresponding author: Yu Du.)*

Dong Liu, Xiantong Tao, Liheng Yuan, and Ming Cong are with the School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: liud@dlut.edu.cn; taoxiantong@mail.dlut.edu.cn; yuanliheng@mail.dlut.edu.cn; congm@dlut.edu.cn).

Yu Du is with the School of Mechanical Engineering, Dalian Jiaotong University, Dalian 116028, China (e-mail: duyu@djtu.edu.cn).

Digital Object Identifier 10.1109/TIM.2021.3129875

The methods of robot grasping detection are mainly divided into analysis methods and data-driven methods. The analysis method uses manual design to extract object features [1] or obtains the optimal grasping position of the object according to the 3-D model of the object [2], [3]. But this method has poor generalization and is difficult to apply to unknown objects.

Robot grasping strategies based on data-driven methods [4] are mainly divided into sampling evaluation method [5] and end-to-end method. The sampling evaluation method generates multiple grasp positions according to the neural network and selects the optimal grasping position from them. Jiang *et al.* [6] used a vector grasp box to indicate the grasping position, and the feature extraction, classification, and decision-making are implemented based on an unsupervised hierarchical learning method. Lenz *et al.* [7] proposed a grasping detection system based on a cascade neural network to obtain the grasp object position, which uses a convolutional neural network to find all possible grasp rectangle, and then evaluates the optimal one through the cascaded convolutional neural network. On this basis, Pinto and Gupta [8] used a random sampling method to reduce the computation time of the grasping position. This method relies on the initial position of the sampling, which is likely to cause unstable grasping. The sampling estimation method needs to evaluate the candidate grasp positions one by one, which will lead to slow grasp detection speed.

The end-to-end method constructs a neural network to directly output the capture position by extracting information, such as image feature. Zeng *et al.* [9] and Morrison *et al.* [10] proposed an end-to-end neural network that outputs the grasp quality distribution of all pixels in the image, but the cumbersome way of determining the grasping angle leads to long grasping time. Yu *et al.* [11] proposed a robot grasping pose detection algorithm based on a three-step learning process. Although the detection speed of this method is improved compared with the sampling evaluation method, it is difficult to achieve high-precision grasping and has poor robustness for grasping scenes with a complex background color. Cheng *et al.* [12] proposed a random cropping ensemble neural network (RCE-NN), which solves the detection of similar overlapping objects, but it can only detect objects with similar textures. Xia *et al.* [13] proposed a two-stage robot grasping pose detection method, but it can only realize the grasping pose estimation of planar objects, and its applicability is restricted. Zhu *et al.* [14] proposed a grasping detection network that provides a prediction uncertainty esti-

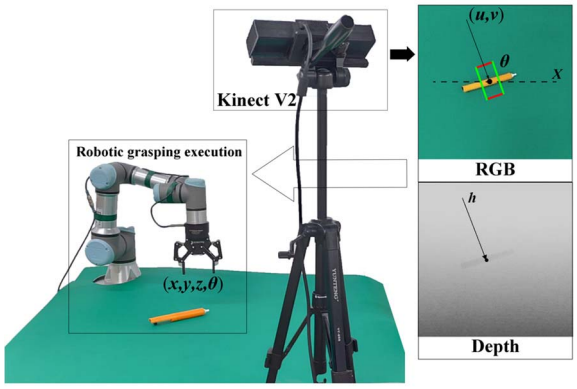

Fig. 1. Schematic of grasp position.

mation mechanism by leveraging on feature pyramid network. Wang *et al.* [15] proposed a PointNetRGPE model to solve the pose estimation problem. But the grasping pose estimation method based on 3-D neural network and point cloud [16] is difficult to complete autonomous accurate grasping due to the influence of object occlusion and truncation in cluttered scenes.

For solving the problems of robotic grasping in unstructured cluttered scenes, this article proposes an end-to-end robot grasping pose detection framework based on cascaded neural networks. The main contributions are as followed.

1) A two-stage progressive end-to-end robot grasp pose detection framework is proposed, which can accurately estimate the robot grasp angle and grasp position, and is suitable and robust for autonomous grasp of objects in cluttered scenes. The combination of different networks improves the accuracy of grasping pose estimation while ensuring generalization and detection speed. The method can reach a high grasping success rate and grasping speeds.
2) An end-to-end grasp angle estimation neural network (Y-Net) is built. Y-Net is a lightweight network and it is fast for angle recognition with a higher accuracy of 1°. It takes a local image as input and achieves fine estimation of angle, which can quickly predict the grasp angle of objects with higher accuracy in cluttered scenes.
3) A grasping feasibility evaluation neural network (Q-Net) is built, which is based on residual convolutional neural network. In Q-Net network, the mapping relationship between grasping quality label and the input scene is established, which acquires the grasp quality for each pixel in the image to evaluate the optimal position for stable grasping.

II. Problem Formulation

The task of robotic object grasping in cluttered scenes refers to inferring the optimal grasping pose for the hybrid and irregular objects with unknown shapes in cluttered scenes. The objects to be grasped are rigid objects that are common in daily life and soft objects that are not damaged by a certain limit of clamping force. To describe the grasping pose in cluttered scenes, a 5-D grasping pose representation method is adopted based on the commonly used grasping frame [7]. As shown in Fig. 1, the point pointed by the arrow is the grasping point, which is marked as (u, v) in the image coordinate, corresponding to the midpoint of the line of the two-finger gripper in the robot end-effector coordinate and the depth h of the point in the axis Z direction. θ is the angle between the clockwise direction of the solid line perpendicular to the grabbing frame and the positive direction of the X-axis in the image coordinate system, corresponding to the angle between the clockwise direction of the vertical line of the two-finger gripper center line in the robot end-effector coordinate and the positive direction of the X-axis. The grasping feasibility Q is the graspability of each pixel in the image and is the probability distribution between 0 and 1. Therefore, the above 5-D grasping pose representation method can be defined as

$$G = (u, v, h, \theta, Q). \tag{1}$$

Further, we can convert the 5-D grasping pose in the image coordinate system to the robot pose in the end-effector coordinate system through the robot kinematics, to realize the object grasping task of the robot in cluttered scenes.

III. Principles and Method

The fundamental purpose of the robotic grasping task is to establish the relationship between external sensor information and the robot's grasping pose. How to achieve this goal quickly and reliably is the key to completing the robot's grasping task. To solve the above problems, this article proposes a robot grasping pose estimation method based on a cascaded neural network. The method is divided into three parts: grasping position detection, grasping pose estimation, and grasping feasibility evaluation. The network architecture is shown in Fig. 2. The first stage mainly adopts an end-to-end idea and establishes a preliminary detection of grasping position with Mask-RCNN [17] as the core to obtain a preliminary grasping rectangle. The second stage extracts the grasping features, establishes a grasping angle evaluation network Y-Net, and builds a Q-Net neural network to evaluate the grasping feasibility of the grasping object, select the best point of grasping feasibility as the grasping point, and combine the grasping angle to finally obtain the grasping pose of the robot.

A. Grasping Position Detection Based on Mask-RCNN

The grasping position detection problem can be transformed into target detection or instance segmentation problem in essence. The end-to-end neural network method represented by Fast-RCNN, Faster-RCNN, etc. is one of the current main methods. Based on Fast-RCNN, Mask-RCNN combines with ROI align to furtherly improve the detection accuracy and adds mask branch to achieve instance segmentation, which can more accurately distinguish the background from the target object. This article implements a preliminary detection of grasping position based on Mask-RCNN, which has a high detection rate and accuracy. The grasping rectangle is generated by the smallest adjacent rectangle circumscribed by the Mask-RCNN mask.

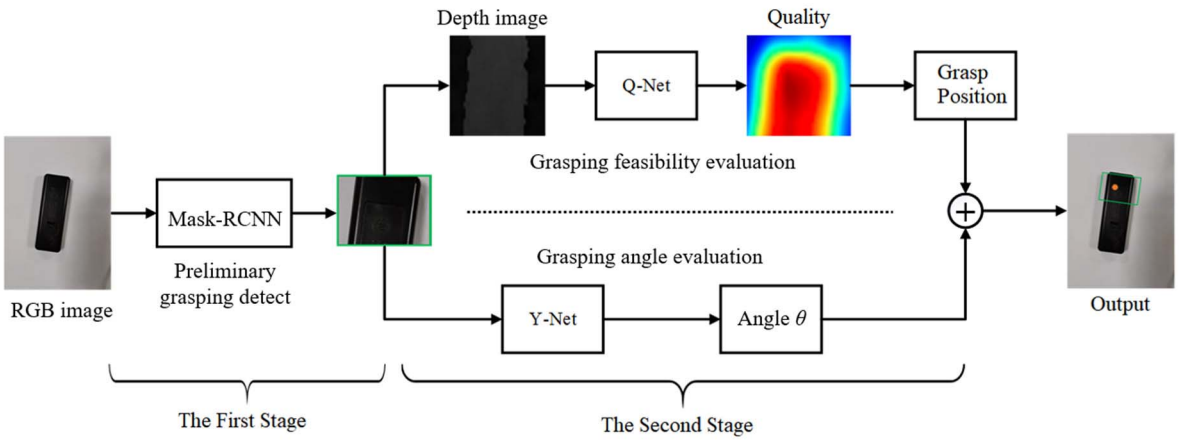

Fig. 2. Network architecture of robotic object grasping in a cluttered scene.

In this article, the grasping categories are set as background and grasping objects. The background is a flat desktop. The grasping objects are common rigid daily necessities or objects that are not easily deformed by a certain clamping force. The residual convolutional neural network (ResNet) is used as the backbone network for basic feature extraction. ResNet is composed of 50 convolutional layers and one fully connected layer. ResNet extracts the grasping feature map from the input RGB image. The region candidate network outputs multiple candidates grasping regions from the grasping feature map. After the grasping candidate boxes enter the Mask branch, it will generate a $k \times 28 \times 28$ dimensional output and output the mask according to the loss function. Finally, the smallest adjacency rectangle of the mask is used as the grasping rectangle and input to the second stage Y-Net.

B. Grasping Angle Evaluation Based on Y-Net

After obtaining the grasping position rectangle, it is necessary to further evaluate the grasping angle to adapt to the change of grasping object posture. The traditional sampling evaluation methods [6], [18] need to generate a large number of candidate images and select the optimal grasp angle from them when evaluating the grasp position. These methods often consume a large amount of time. In this article, a neural network (Y-Net) of grasping angle evaluation is built. The structure of the network is shown in Fig. 3. The Y-Net network structure is lightweight, so it is fast for angle recognition with a higher accuracy of 1°. It is essentially used for object classification, which means that only 180 object classifications are needed.

Take the local feature images in the grasp rectangle as the output of the Y-Net neural network. The network consists of four convolutional layers and one fully connected layer. The number of convolution kernels in the convolutional layer is 16, 32, 64, and 128, and the number of neurons in the fully connected layer is 4096. The convolutional layer uses the Relu function as the activation function, uses the loss function to evaluate the error of the relationship between the grasping prediction angle and the actual value of the grasping angle,

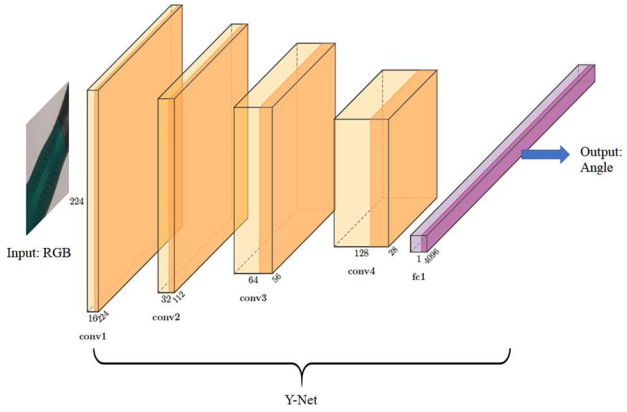

Fig. 3. Y-Net network structure.

and uses the batch normalize method for normalization. This article utilizes the L2 normalization as the loss function of Y-Net, which is defined as follows:

$$L_Y = \frac{1}{N}\left(\sqrt{|\theta^* - \theta_0|^2} + \sum_i^N \lambda \omega_i^2\right) \tag{2}$$

where θ^* is the predicted angle value, θ_0 is the expected angle value, λ is the regularization term, ω_i is the model weight parameter, and N is the number of parameters.

Finally, Y-Net outputs the grasp angle, which can be accurate to 1°. Compared with the sampling evaluation method, the accurate angle and detection speed have been greatly improved.

C. Grasping Feasibility Evaluation Based on Q-Net

Traditional robot grasping pose detection methods often take the center point of the grasping rectangle or the center of object gravity as the robotic grasping position, but the optimal grasping position of the object is not often in the above-mentioned position. To deal with cluttered scenes, this article constructs a grasping feasibility evaluation convolutional neural network Q-Net to evaluate the grasping quality of each pixel, and the point with the highest grasping quality score in

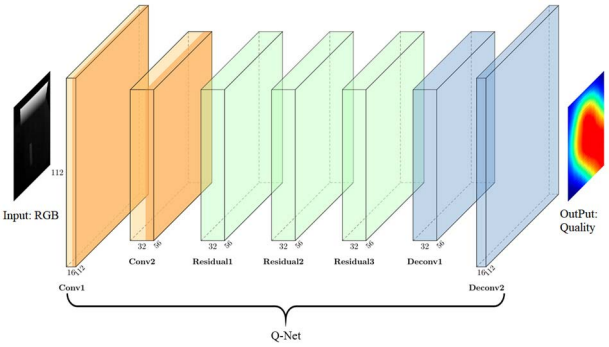

Fig. 4. Q-Net network structure.

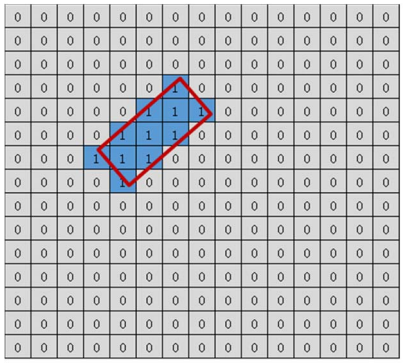

Fig. 5. Examples of the grasping quality label.

the grasp rectangle is used as the robotic grasp position. Q-Net is simpler and can evaluate grasping points without previous generation rectangles. The structure of Q-Net is shown in Fig. 4. It is constructed by the cascaded deep convolutional neural network, which has a deeper network and can extract deeper features from depth image.

First, the depth map corresponding to the image in the grasping rectangle is preprocessed and normalized. Second, the holes of the depth map are filled. Finally, the processed depth map is used as the input of the Q-Net neural network. The network consists of two convolutional layers, three residual layers, and two deconvolution layers. It uses the batch normalize method for normalization and uses the Relu function as the activation function. The number of convolution kernels in the convolution layer is 16 and 32, the number of convolution layers in the deconvolution layer is 32 and 16, and the number of convolution kernels in the residual layer is 32. The loss function of Q-Net uses the smooth L1 function. The output is the grasping quality of all pixels in the grasping rectangle, which is the same size as the original image and expressed in the form of a heat map. In each grasping process, the robot will first select the point with the highest grasping quality as the robotic grasping point. The highest grasping quality represents the grasping reliability of the current position that is the most suitable grasping position of the robot in the current scene. In the training, a large number of objects with various shapes and materials from datasets are used for training Q-Net.

The input scene is depth image of dataset $D = \{D^1, \ldots, D^n\}$, and the object $O = \{O_1, \ldots, O_m\}$ of grasping feasibility $Q_i = \{q_1^1, \ldots, q_m^1, \ldots, q_1^2, \ldots, q_m^n\}$. The grasping quality label is created, where the grasping quality is 1 in the grasping rectangle and others are 0, Fig. 5 shows the grasping quality label visually. The Q-Net minimizes the negative log-likelihood of grasping quality conditioned on the input depth image by the mapping function $Q_i = \gamma(D, O)$, and the minimizing [19] is given by:

$$-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\log\gamma\left(q_i^j/D^i\right). \tag{3}$$

The loss is defined by smooth L1 as

$$L_Q(Q_i, \dot{Q}_i) = \frac{1}{n}\sum_{0}^{k} Z_k \tag{4}$$

Fig. 6. Experiment platform.

where Q_i is the grasping feasibility generated by Q-Net and $\dot{Q}_i$ is the real truth grasping feasibility, z_k is defined as

$$Z_k = \begin{cases} \frac{1}{2}(Q_{ik}, \dot{Q}_{ik})^2, & \text{if } \left|Q_{ik}, \dot{Q}_{ik}\right| < 1 \\ \left|Q_{ik}, \dot{Q}_{ik}\right| - 0.5, & \text{otherwise} \end{cases} \tag{5}$$

where Q_{ik} is the grasping feasibility of each pixel in the input scene, which is generated by Q-Net. $\dot{Q}_{ik}$ is the real truth grasping feasibility of each pixel in the input scene.

IV. System Architecture and Model Training

A. Experiment Platform

The robot object grasping system is composed of UR3e, KinectV2 camera, PC, and Robotiq85 gripper. The robot is used to grasp the object to the target position. The KinectV2 camera is used to provide the RGB image and the depth image of the cluttered scene and transfer them to the PC. The camera is set up on a platform with the eye out of the hand. The CPU model is Intel CoreTM i9-9900K, the graphics card is NVIDIA GeForce RTX2080TI, and the operating system is Ubuntu 16.04, deep learning framework Tensorflow and Keras are used for network training and prediction. The grasping system platform is shown in Fig. 6.

B. System Calibration

The grasping of the target object needs a coordinate transformation in the real world. The target object data are obtained

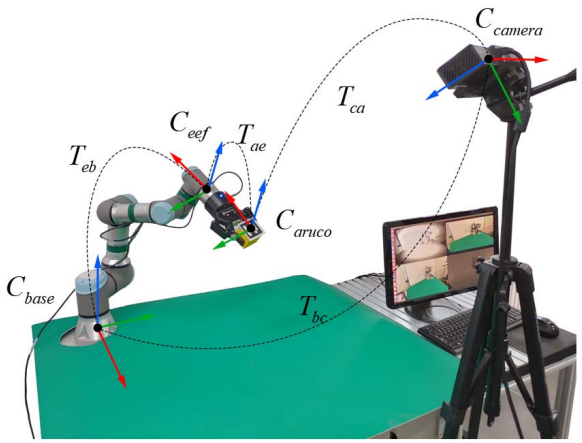

Fig. 7. Relationship of system calibration.

via Kinect v2, and the grasping pose is generated via deep learning, which is a pixel point (u, v), the grasping pose should be transformed to the robot coordinate in the real world. The system calibration includes UR3e calibration, Kinect V2 calibration, and robot hand–eye calibration.

UR3e calibration and Kinect V2 calibration calibrate the hardware itself. The grasping pose (x, y, z) in the robot coordinate system from the pixel point (u, v) in the image of Kinect v2 is obtained through robot hand–eye calibration [20]. The relationship of the system calibration is shown in Fig. 6. The mathematical model is expressed as

$$T_{\text{base}}^{\text{camera}} = T_{\text{base}}^{\text{eef}} \times T_{\text{eef}}^{\text{aruco}} \times T_{\text{aruco}}^{\text{camera}} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \tag{6}$$

where $T_{\text{base}}^{\text{camera}}$ is the transformation matrix from camera to robot base, R (rotation matrix) and T (translation matrix) are the basic parameters in the transformation matrix, $T_{\text{base}}^{\text{eef}}$ is the transformation matrix from robot base to the robot end-effector, $T_{\text{eef}}^{\text{aruco}}$ is the transformation matrix from the robot end-effector to ArUco marker, and $T_{\text{aruco}}^{\text{camera}}$ is the transformation matrix from the ArUco marker to the camera.

$T_{\text{base}}^{\text{camera}}$ and $T_{\text{eef}}^{\text{aruco}}$ are fixed as T_{bc} and T_{ae} shown in Fig. 7. So $T_{\text{base}}^{\text{camera}}$ can be obtained by two calculations at different positions. and the relationship can be expressed as

$$T_{ae} = T_{eb1} \times T_{bc} \times T_{ca1} = T_{eb2} \times T_{bc} \times T_{ca2}. \tag{7}$$

Equation (7) convert to $AX = XB$ problem [21], [22] as

$$T_{eb2}^{-1} \times T_{eb1} \times T_{bc} = T_{bc} \times T_{ca2} \times T_{ca1}^{-1}. \tag{8}$$

The $T_{\text{base}}^{\text{camera}}$ is T_{bc} shown in Fig. 6, and the grasping pose is expressed as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = T_{\text{base}}^{\text{camera}} \times \begin{bmatrix} x_{\text{camera}} \\ y_{\text{camera}} \\ z_{\text{camera}} \end{bmatrix} \tag{9}$$

where $(x, y, z)^T$ is the grasping pose of the robot in the real world, and $(x_{\text{camera}}, y_{\text{camera}}, z_{\text{camera}})^T$ is the pose of the object obtained by Kinect V2.

Fig. 8. Examples of grasp dataset.

C. Dataset and Processing

The dataset in this article is composed of RGB and depth images captured by Kinect V2 camera. With a flat desktop as background, it contains 24 common living objects, including pens, staplers, blackboard erasers, tape, scissors, cups, etc. living goods and tools, some examples of datasets are shown in Fig. 8.

To enrich the shape of the dataset, the rotation method is used to expand the dataset to simulate various forms of grasping objects. Through rotation, the dataset obtains objects with different inclination angles, which reduces the impact of the inclination angle of the image acquisition on the evaluation of grasp quality. First, 50 different objects of each type are obtained as the initial dataset, and they are rescaled to $112 \times 112 \times 3$ dimensional RGB images and $112 \times 112 \times 1$ dimensional Depth images. Each RGB image is labeled to ensure that the grasp target is located at the center of the image, and prevent the target beyond the edge of the image caused by subsequent rotation. Second, the affine transformation method is used to rotate the image, and transform the coordinates of the grasping rectangle. Rotating 360° with the center point of the image as the center of rotation every 5°, 86 400 new images with different angles are generated.

The image of the grasping rectangle is selected as the input of Y-Net for training. It is scaled to $112 \times 112 \times 3$ dimensional data and rotated 180° around the image center point every 1° to generate 432 000 images. The Robotiq gripper is a parallel two-finger gripper, so it only needs to rotate half a circle. The grasping angles are 1°, 2°, 3°, 4°, ..., 180°, and the grasping samples of each angle are generated as the format of Tfrecords for Y-Net training. The training set of Q-Net uses the Cornell grasp dataset [6] and Jacquard grasping dataset [23]. Cornell grasp dataset consists of 240 different real objects and 1035 RGB-D images. The Jacquard grasping dataset consists of 54k RGB-D images and 1.1M grasp examples. In these datasets, the ratio of the training dataset to the testing dataset is 0.8:0.2, which is randomly sampled from these datasets.

D. Model Training

The models mainly include Mask-RCNN, Y-Net, and Q-Net convolutional neural networks. The Mask-RCNN model uses

TABLE I
RESULTS COMPARISON OF DIFFERENT ALGORITHMS ON THE CORNELL DATASET

Algorithm	Speed (frame/s)	Accuracy (%)
Fast Search [6]	0.02	59.4
SAE [7]	0.07	74.8
FCN [9]	0.06	91.2
GG-CNN [10]	52	81
R-FCN + AngleNet [13]	17.5	92.5
AlexNet [26]	13	88.5
Dex-Net [27]	1.25	93
Our method	21	95.2

TABLE II
RESULTS COMPARISON OF DIFFERENT ALGORITHMS ON THE JACQUARD DATASET

Algorithm	Accuracy (%)
Jacquard [22]	59.4
FCGN, ResNet-101 [23]	91.2
GG-CNN2 [24]	74.8
Our method	92.1

the initial parameters of the ResNet model which is pretrained on the COCO dataset for migration learning and uses the self-built dataset to train the model. This method can quickly complete the model training and make up for the problem of small datasets. Y-Net and Q-Net are constructed based on Keras and Tensorflow, using the Xavier method [24] to initialize the model parameters, and using the Adam method [25] to optimize the parameters. The advantage is that the learning rate is adjusted adaptively. The initial learning rate is 0.001, and the batch size is set to 64. Both subnetworks use CUDA for accelerated training.

E. Evaluation Index

Grasping detection rate and grasping success rate are the two most important indices to measure the grasping method [14]. Fast and accurate grasping can facilitate the robot to be deployed in more scenes, such as grasping the moving objects and grasping at a fixed production cycle on the production line. Therefore, this article uses the following two indicators to evaluate the effectiveness of the grasping method.

1) Grasping detection rate: it starts from the visual system obtaining the image and ends with the calculation of the robotic grasping pose.
2) Grasping success rate S: In this experiment, the ratio of the number of times n_{finish} for robot success grasping to the total number of grasping n

$$S = \frac{n_{\text{finish}}}{n}. \tag{10}$$

V. EXPERIMENT RESULTS

A. Method Validation

On the Cornell and Jacquard datasets, this proposed algorithm is compared with other grasping detection algorithms on the testing dataset. The testing sets are randomly sampling split selected to test the algorithm. The testing set contains a variety of different types of objects from the Cornell dataset and Jacquard dataset. The grasping rectangle defined above is used to represent the grasping position and grasping angle, and the color of the heat map from blue to red indicates the graspability from low to high. To verify the accuracy of the grasping rectangle obtained by Mask-RCNN, the global heat map of the object to be grasped is represented, and the local heat map in the grasping image is shown in the upper right corner. The orange dot is the final grasping point, and the result is shown in Fig. 9. The grasping detection rate and grasping success rate of the proposed method on the Cornell dataset are compared with other popular grasping methods according to the above grasping indexes, and the results are shown in Table I. For the Jacquard dataset, Table II shows the comparison results of the proposed method and other methods [23], [26], [27] on the Jacquard dataset. The grasping detection rate of the proposed method can reach 21 frame/s, and the detection accuracy can reach 95.2% on the Cornell dataset, while the detection accuracy is 92.1% on the Jacquard dataset. Compared with the sliding window method [6] and random sampling method [7], the grasping detection rate and success rate have been significantly improved. Compared with the end-to-end methods of FCN [9], GG-CNN [10], R-FCN + Anglenet [13], and AlexNet [28], this article adds a parallel grasp feasibility evaluation method (Q-Net), which can adapt to more complex grasp scenarios and achieve a more accurate grasp. Moreover, since it is parallel in other networks, it has little influence on the grasp detection rate. Compared with the grasping method Dex-Net [29], the proposed method utilize Y-Net to avoid a large number of evaluations of candidate grasping positions, so the grasping detection rate is significantly increased, and the grasping success rate is improved to a certain extent through using Q-Net.

In addition, we evaluate the grasping stability of methods [10], [30] on the Cornell grasping dataset and the proposed method on the Cornell grasping dataset, Jacquard dataset, and real environment. The comparison results of different methods are shown in Fig. 10. The result shows that the proposed grasping method is superior to other methods.

To verify the effectiveness of Y-Net, this article compares Y-Net with the sampling evaluation method [7] on the open dataset. The single grasp angle detection time of the sampling evaluation method is 1350 ms, while the single grasp angle detection time of the proposed method is 47 ms, and its detection speed is about 29 times than that of the sampling evaluation method. In terms of detection accuracy, the detection accuracy of Mask-RCNN + Y-Net/Q-Net is 95.2%, while the detection accuracy of Mask-RCNN + Sampling

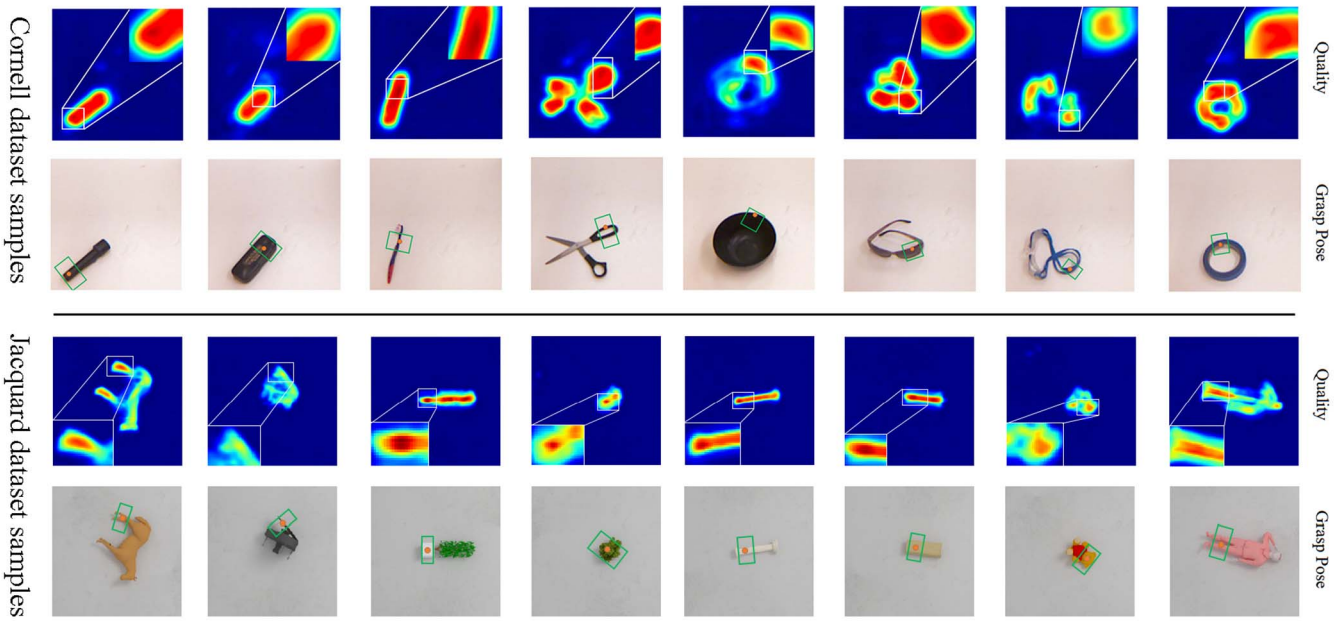

Fig. 9. Experimental results on the Cornell grasping dataset and Jacquard grasping dataset.

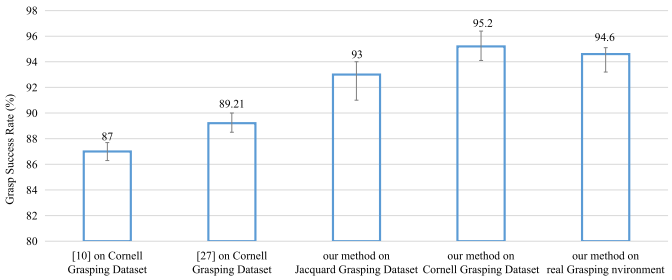

Fig. 10. Accuracy comparison of different methods.

Evaluation/Q-Net is 93%. It can be seen that Y-Net significantly improves the grasping detection speed, and also has a certain improvement in grasping accuracy.

Furthermore, to verify the effectiveness of Q-Net, this article sets the comparison method of selecting the center point of the grasping rectangle as the final grasping position [13], and the rest parts are the same as the method in this article. Fig. 11 shows the grasping comparison results of the two methods. It can be seen that in this grasping scene, the grasping position point determined by the proposed method is on the edge of the target object, which is consistent with the actual grasping position. However, the comparison method is at the bottom of the target object or not on the target object, which is difficult to complete the grasping task for the robot. From the visual gripper model in Fig. 10, if the center point of the grasping rectangle is the robotic grasping point, the grasping tool will impact the grasping object in some cases and cannot grasp correctly, such as the first and third object. Compared with the center point of the grasping rectangle as a grasping point, our method is easier to grasp successfully under the real robot, the feasibility estimation method based on Q-Net is more reliable.

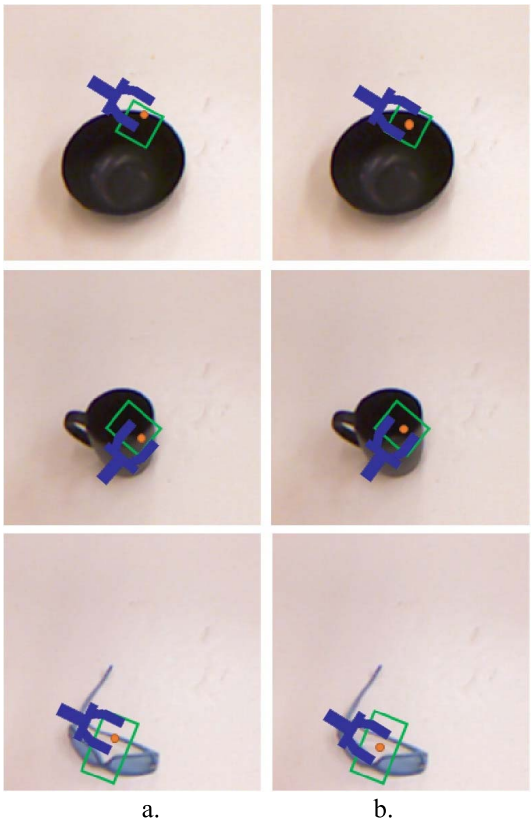

Fig. 11. Comparison results of grasping position. (a) Proposed method. (b) Comparison method.

B. Real Environment Grasping

To verify the effectiveness of the grasping algorithm in the real environment, four types of grasping scenes which include

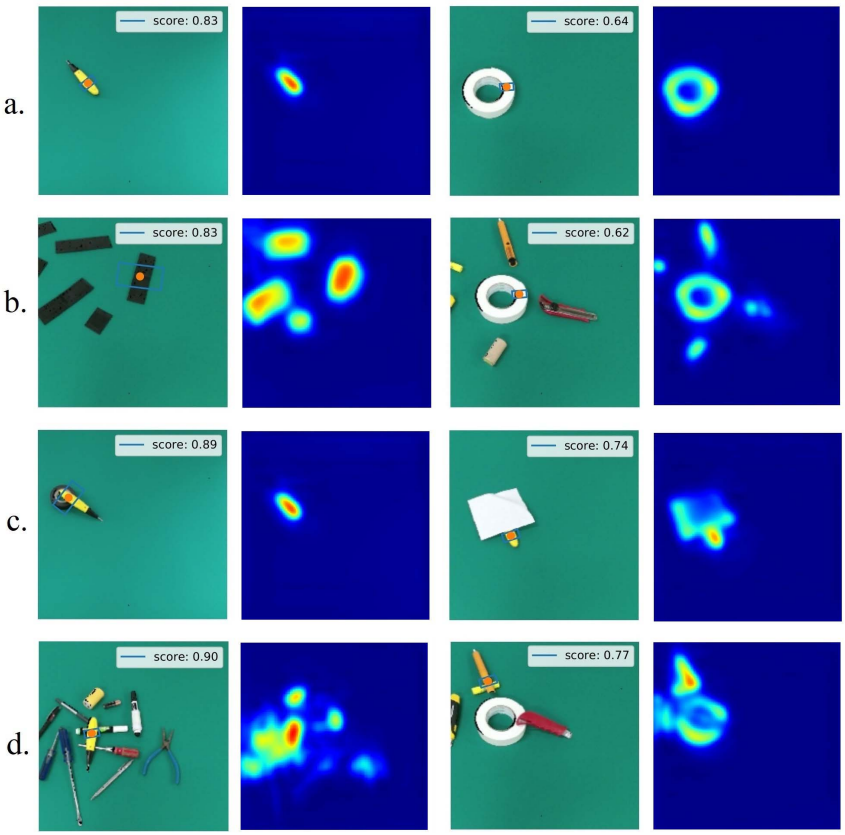

Fig. 12. Experimental results of online grasping detection on different scenes. (a) Single object scenes. (b) Multiple target object scenes. (c) Occluded objects scenes. (d) Clutter objects scenes.

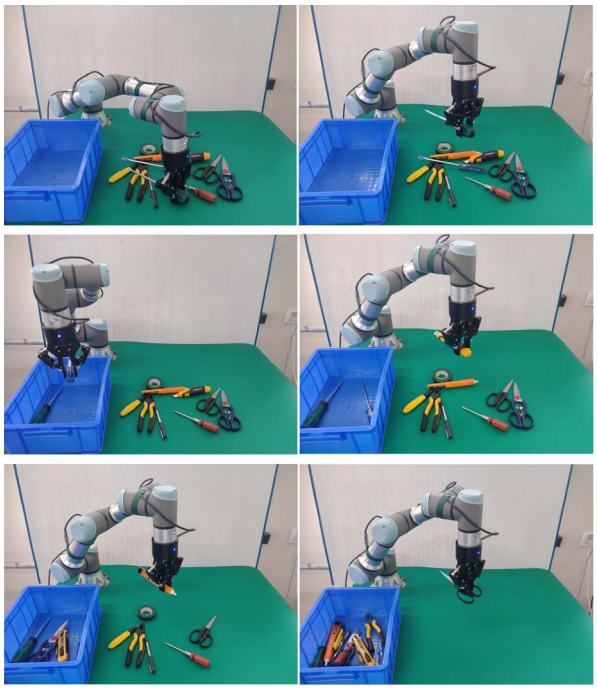

Fig. 13. Grasping effect of clutter environment.

TABLE III
GRASPING SUCCESS RATE IN DIFFERENT SCENES

scenes	number of successes	number of grasps	Grasp success rate/%
single object	88	93	94.6
multiple objects	92	98	93.9
occluded objects	92	100	92.0
cluttered objects scene	295	327	90.2

single objects, multiple target objects, occluded objects, and multiobjects in a cluttered scene are selected. The grasped objects are common rigid necessities for daily use. The single target object scene is an irregular object of arbitrary shape on the table. The multiple target object scene is to place several noninterfering objects on the table. The occluded object scene is that where there is a target object occluded by another object on a flat table. The cluttered objects scene is that multiple objects are stacked on the table in a jumbled manner with interference, occlusion, and stacking among them. The illumination of each grasp scene is different, and the objects to be grasped are randomly placed in the grasp detection area in each grasping experiment. The experimental goal is to grasp the objects in different scenes to a specified target area. The grasping rectangle is used to represent the grasping pose of

the robot for the object with arbitrary posture in the real environment. Fig. 12 shows some of the experimental results.

Fig. 13 shows the online objects grasping effect of the robot in a cluttered scene, which verifies the effectiveness of the proposed method in the actual scene. Table III shows the success rate of the robotic online grasping experiment under different scenes. It can be seen that the comprehensive grasping success rate of the proposed method can reach 90.2% in the cluttered grasping environment.

VI. Conclusion

Aiming at the problem of robot grasping in cluttered scenes, this article proposes a robot grasping pose detecting method based on a cascade neural network. The grasping rectangle is generated by Mask-RCNN, and the detailed features in the grabbing rectangle are further extracted through Y-Net to obtain the grasping angle of the two-finger gripper. Based on the obtained grasping depth map, and the grasping feasibility distribution of the robot is evaluated by Q-Net. The two-stage end-to-end neural network significantly improves the robot grasping accuracy and improves the detection speed to a certain extent. Experimental results show that the proposed method can effectively generate the grasping pose in experimental scenes, such as single objects, multiple objects, occluded objects, and multiobjects in the cluttered scene. The Cornell dataset verifies that the proposed method has a great improvement in grasping accuracy and detection rate compared with the other robot grasping methods. The grasping accuracy can reach 95.2%, and the grasping rate can reach 21 frames/s on the Cornell dataset, while the grasping accuracy is 92.1% on the Jacquard dataset. Online grasping experiments show that the success rate of the proposed method can reach 90.2% in clutter objects environment, and it is suitable for autonomous grasping scenes of clutter objects.

References

[1] M. R. Dogar, K. Hsiao, M. Ciocarlie, and S. S. Srinivasa, "Physics-based grasp planning through clutter," in *Robotics: Science and Systems VIII*. Cambridge, MA, USA: MIT Press, 2012, pp. 217–236.

[2] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2, Sep. 2003, pp. 1824–1829.

[3] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. Millennium Conf., IEEE Int. Conf. Robot. Automat., Symp. (ICRA)*, vol. 1, Apr. 2000, pp. 348–353.

[4] 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, doi: 10.1109/TRO.2013.2289018.

[5] I. Kamon, T. Flash, and S. Edelman, "Learning to grasp using visual information," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, Apr. 1996, pp. 2470–2476, doi: 10.1109/ROBOT.1996.506534.

[6] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 3304–3311.

[7] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 705–724, 2015.

[8] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50 K tries and 700 robot hours," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 3406–3413.

[9] A. Zeng *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 3750–3757.

[10] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," 2018, *arXiv:1804.05172*.

[11] Q. C. Yu, W. W. Shang, and C. Zhang, "Object grab detection based on three-level convolutional neural network," *Robot*, vol. 40, no. 5, pp. 762–768, 2018.

[12] 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.

[13] J. Xia, K. Qian, X. D. Ma, and H. Liu, "Fast detection of robot plane grab position based on concatenated convolution neural network," *Robot*, vol. 40, no. 6, pp. 794–802, 2018.

[14] H. Zhu *et al.*, "Grasping detection network with uncertainty estimation for confidence-driven semi-supervised domain adaptation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 9608–9613, doi: 10.1109/IROS45743.2020.9341056.

[15] Z. Wang, Y. Xu, Q. He, Z. Fang, G. Xu, and J. Fu, "Grasping pose estimation for SCARA robot based on deep learning of point cloud," *Int. J. Adv. Manuf. Technol.*, vol. 108, no. 4, pp. 1217–1231, May 2020, doi: 10.1007/s00170-020-05257-2.

[16] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[18] S. Y. Zhang, G. H. Tian, Y. Zhang, and X. L. Liu, "An autonomous grasping strategy based on two-stage progressive network guided by prior knowledge," *Robot*, vol. 42, no. 5, pp. 513–524, 2020.

[19] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 9626–9633, doi: 10.1109/IROS45743.2020.9340777.

[20] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[21] Y. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB," Dept. Elect. Comput. Eng., Tech. Rep., Dec. 1987, Paper 592. [Online]. Available: https://docs.lib.purdue.edu/ecetr/592

[22] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, Jun. 1995, doi: 10.1177/027836499501400301.

[23] A. Depierre, E. Dellandrea, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3511–3516, doi: 10.1109/IROS.2018.8593950.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Mar. 2010, pp. 249–256. Accessed: Oct. 26, 2021. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[26] 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)*, Oct. 2018, pp. 7223–7230, doi: 10.1109/IROS.2018.8594116.

[27] 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, doi: 10.1177/0278364919859066.

[28] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 1316–1322, doi: 10.1109/ICRA.2015.7139361.

[29] J. Mahler *et al.*, "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017, *arXiv:1703.09312*.

[30] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 769–776, doi: 10.1109/IROS.2017.8202237.

Dong Liu (Member, IEEE) received the Ph.D. degree in mechatronics engineering from Dalian University of Technology, Dalian, China, in 2014.

He was a Visiting Scholar in mechanical engineering with the University of British Columbia, Vancouver, BC, Canada, from 2011 to 2012 and a Research Fellow in electrical and computer engineering with the National University of Singapore, Singapore, from 2015 to 1016. He is currently an Associate Professor with the School of Mechanical Engineering, Dalian University of Technology. His research interests include intelligent robotics and system, machine learning, and cognitive control.

Xiantong Tao received the B.S. degree from the School of Mechanical Engineering, Dalian University of Technology, Dalian, China, in 2020, where he is currently pursuing the M.S. degree in mechanical engineering.

His research interests include intelligent robotics and computer vision.

Liheng Yuan received the B.S. degree from the School of Mechanical Engineering, Wuhan University of Technology, Wuhan, China, in 2018, and the M.S. degree in mechanical engineering from Dalian University of Technology, Dalian, China, in 2021.

His research interests include intelligent robotics and deep learning.

Yu Du received the Ph.D. degree in mechanical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2018.

She was the CEO of Dalian Dahuazhongtian Technology Company, Ltd., Dalian, China. She is currently an Associate Professor with the School of Mechanical Engineering, Dalian Jiaotong University, Dalian, China. Her main research interests include robotics and automation, and intelligent control.

Ming Cong received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 1995.

Since 2003, he has been a Professor with the School of Mechanical Engineering, Dalian University of Technology, Dalian, China. He was an Outstanding Expert enjoying special government allowances approved by the State Council, an advanced worker of intelligent robot theme in the field of automation by National High Technology Research and Development Program (863), and a member of the industrial robot expert group of the fifth intelligent robot theme for the 863 program. His research interests include robotics and automation, intelligent control, and biomimetic robots.