Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.028369

ARTICLE

Multitarget Flexible Grasping Detection Method for Robots in Unstructured Environments

Qingsong Fan, Qijie Rao and Haisong Huang*

Key Laboratory of Modern Manufacturing Technology, Ministry of Education, Guizhou University, Guiyang, 550025, China

*Corresponding Author: Haisong Huang. Email: hshuang@gzu.edu.cn

Received: 14 December 2022 Accepted: 29 January 2023 Published: 28 June 2023

ABSTRACT

In present-day industrial settings, where robot arms perform tasks in an unstructured environment, there may exist numerous objects of various shapes scattered in random positions, making it challenging for a robot arm to precisely attain the ideal pose to grasp the object. To solve this problem, a multistage robotic arm flexible grasp detection method based on deep learning is proposed. This method first improves the Faster RCNN target detection model, which significantly improves the detection ability of the model for multiscale grasped objects in unstructured scenes. Then, a Squeeze-and-Excitation module is introduced to design a multitarget grasping pose generation network based on a deep convolutional neural network to generate a variety of graspable poses for grasped objects. Finally, a multiobjective IOU mixed area attitude evaluation algorithm is constructed to screen out the optimal grasping area of the grasped object and obtain the optimal grasping posture of the robotic arm. The experimental results show that the accuracy of the target detection network improved by the method proposed in this paper reaches 96.6%, the grasping frame accuracy of the grasping pose generation network reaches 94% and the flexible grasping task of the robotic arm in an unstructured scene in a real environment can be efficiently and accurately implemented.

KEYWORDS

Unstructured scene; robot; target detection; grab pose detection; deep learning

1 Introduction

The grasping action of a robotic arm is one of the basic skills to implement the flexibility of robot controls [1,2]. With the application of deep learning [3–5] in the field of machine vision, the accuracy and reliability of target detection have been significantly improved, the scenes captured by robotic arms are also more complex, and multiple objects may be randomly placed in any posture [6–8]. This makes the traditional programming teaching method no longer applicable. With the emergence of consumer-level depth cameras, guided by visual information, it is possible for robots to complete more complex tasks [9–11]. Therefore, the study of flexible grasping tasks in unstructured scenes has become a hot research topic in the field of robotic grasping in recent years [12–14].

Grabbing detection based on traditional vision methods mostly uses point cloud registration to estimate the object poses. PointNet-grasp pos detection (PointNet-GPD) [15], the Linemod algorithm

based on template matching [16], the voting-based point pair [17], the feature algorithm (point pair feature, PPF) and its improvement [18]. These types of algorithms require a known three-dimensional model of the target object, and the three-dimensional model is registered with the target object in the actual scene to calculate the spatial position and posture of the target object. However, in practical applications, this type of method is easily occluded and truncated by the object. Because of other influencing factors, an increasing number of researchers have begun to study how to apply deep learning and swarm intelligence optimization to the robot grasping problem [19–21]. Asif et al. [22] predicted the capture area from different levels of the image, overcame the limitations of predicting the capture area of an image since a single level, and had high accuracy. Li et al. [23] proposed a MobileNet SSD-based detection method, using a single-shot multi-box detector (SSD) network as a meta-structure, which can be more accurate and faster than lightweight network methods and traditional machine learning methods to achieve automatic detection. Hernandez et al. [24] combined the target detection algorithm with other algorithms, trained a network based on a faster area convolutional neural network to detect objects, and then used Super4PCS to estimate the object pose. Zeng et al. [25] used a fully convolutional network to achieve object segmentation and used the iterative closest point (ICP) to estimate the object pose. The neural network model is usually a multitask model, and the weight loss of each task during training has a great impact on the performance of the model. Kendall et al. [26] proposed that task uncertainty represents the relative confidence between the different tasks, and the loss weights of the different tasks are set by the same variance uncertainty of each task. Sener et al. [27] proposed treating multitask learning as a multiobjective optimization problem, turning the overall goal into finding Pareto optimization, and using gradient-based multiobjective optimization for processing. Zhang et al. [28] proposed a grasping angle classification model based on self-supervised learning, and constructed an automatic labeling method for disordered grasping dataset. Song et al. [29] proposed a new trimodal image fusion strategy to achieve salient object detection for robotic visual perception. Tee et al. [30] developed a tool cognition framework that enables the robot to recognize a previously unseen object as a tool for a task and plan how to grasp and use it. Benefiting from the breakthrough progress made by deep learning in the field of image recognition, compared with the traditional template matching method to obtain object space grasping pose. The image-based target object grasp and recognition method is more suitable for the needs of complex application scenarios at this stage. How to use deep learning technology to improve the accuracy and efficiency of robot grasping and detection in unstructured scenarios is still an urgent problem to be solved [31–33].

This paper proposes a multistage robot flexible grasping detection method based on deep learning. First, the improved Faster RCNN network model is used to significantly improve the accuracy of the multiscale target detection; second, the network structure of literature is optimized, and a high-precision multitarget grasping pose generation network is designed; finally, aiming at the multiple graspable poses generated in the multitarget scene, by defining the mutual exclusion relationship between the target recognition regions, a multitarget IOU hybrid region pose evaluation algorithm is proposed, which is based on multiple grasps generated by the grasping pose generation network Pose. It filters out the optimal grasping posture of the robotic arm in multitarget situations, and improves the success rate and accuracy of the robotic arm when performing grasping tasks.

In this paper, the proposed grasping detection method is applied to the double-arm collaborative robot, Baxter, and actual verification is carried out in a real scene. The coordinate conversion is carried out through the hand-eye relationship calibrated by the RAC calibration method [34], which implements the flexible grasping task of the robot arm and verifies the accuracy of the above method.

The remaining work of this paper is as follows. Section 2 briefly overviews the technical framework for flexible grasp detection of robots. Sections 3 to 5 introduce the proposed unstructured scene object perception, robot grasping pose generation network and multiobject IOU hybrid region pose evaluation algorithm in detail, respectively. Then, the real scene experiments and a discussion of the results are provided in Section 6. Finally, the conclusion and future work are given in Section 7.

2 The General Framework of the Grab Detection Method

The overall framework of the robot's flexible grasping detection technology is shown in Fig. 1. First, the RGB-D camera Kinect x1 is used to capture the image information in the captured scene, and the image is preprocessed. Second, each target in the scene is identified through the target detection network architecture, and then the proposed multitarget capture gesture generation network is used to generate each grasping pose of the target. Finally, based on the object area in the target detection result and the pose information in the grasping detection result, the optimal grasping pose is screened, and finally, the robotic arm is controlled to complete the grasping task.

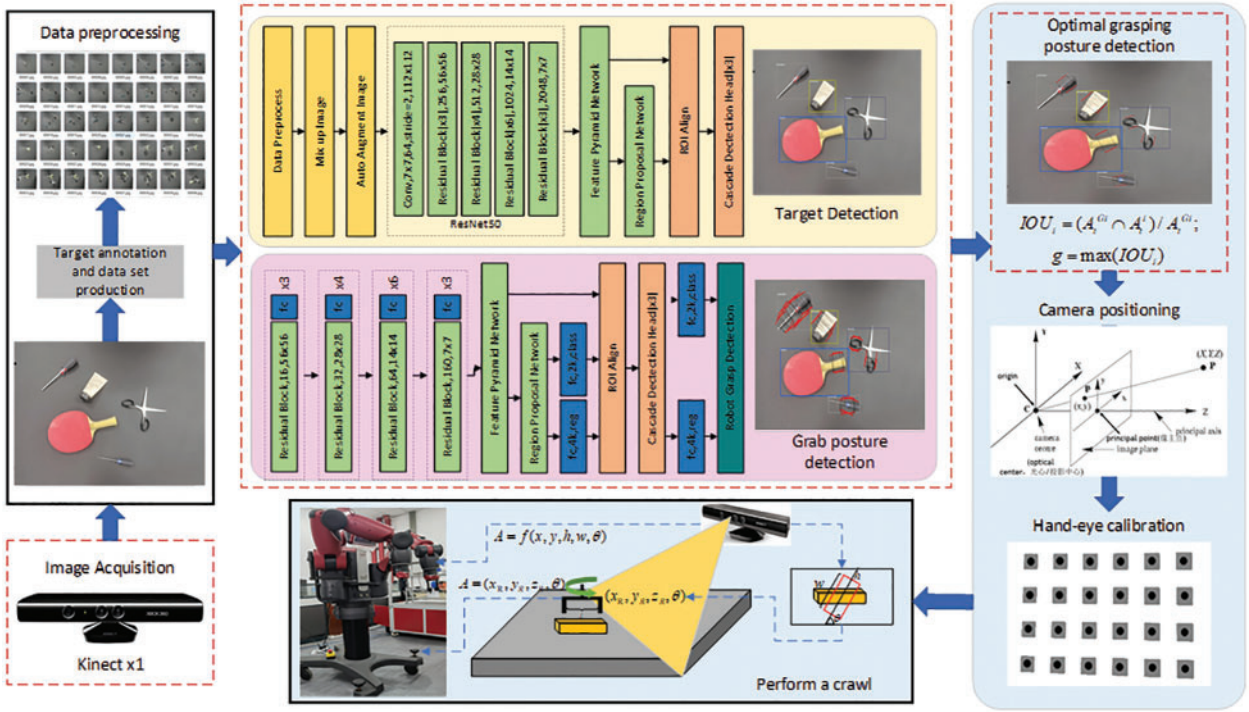


Figure 1: General framework of the robot grasping method

Different from a regression boundary task that needs to be predicted in the conventional target detection problem, the robot grasping detection problem not only needs to predict the regression boundary of the target to be grasped but also needs to filter the optimal grasping posture of the robotic arm [35]. To predict the optimal grasping posture of the grasping target in the image information, this paper represents the optimal grasping area in the input image and defines a 5D grasping posture representation method of the robotic arm.

$$A = f(x, y, h, w, \theta) \tag{1}$$

where (x, y) represents the coordinates of the upper left corner of the image capture rectangle, (h, w) represents the height and width of the capture rectangle, and θ represents the angle of the capture rectangle relative to the horizontal direction of the image. The three-dimensional imaging model is used to determine the optimal grasping posture of the robot corresponding to the rectangle.

3 Target Perception in Unstructured Scenes

3.1 Faster RCNN Target Detection Framework

The Faster RCNN network model is composed of a feature extraction network, a region suggestion network and an RCNN network. Fast RCNN is a two-stage end-to-end target detection method. The network is a milestone network structure in the field of target detection. It has the advantage of ensuring detection speed and high detection accuracy, which can completely meet the requirements of robot detection and grabbing [36]. The guiding principle is to integrate the RCNN network model with the regional suggestion network model and share the feature extraction network (such as: VGG16 [37], ResNet [38], ZFNET [39]) to extract the image features to form a complete target detection network architecture. The basic structure is shown in Fig. 2.

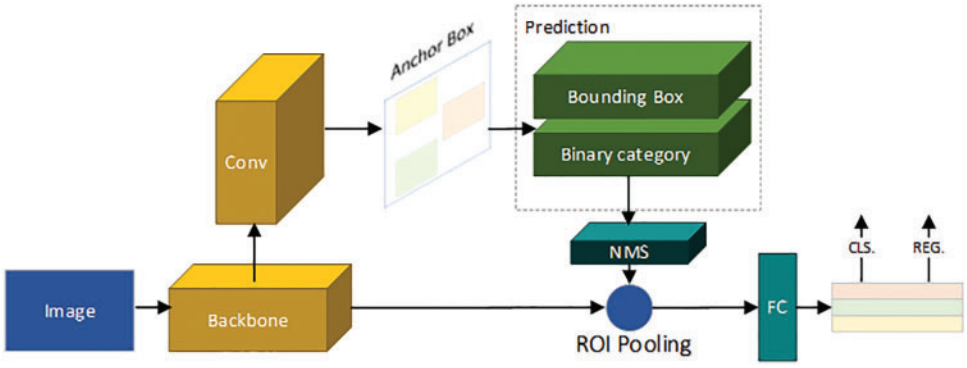


Figure 2: The network structure of Faster RCNN

The selecting search in the RCNN network structure recognizes and extracts the region of interest. To improve the recognition efficiency of the network structure and obtain high-precision and high-reliability feature suggestion regions, RPN uses features to extract the feature map output by the network, and adopts independent learning methods to identify and locate the regions of interest in the target image. Compared with the traditional RCNN network, Faster RCNN has a faster speed and a higher accuracy for target detection and positioning.

3.2 Improved Faster RCNN Target Detection

The traditional Faster RCNN network uses the VGG16 network architecture composed of 13 convolutional layers, 13 activation layers and 4 pooling layers as its feature extraction network. Since the pixels of the input image become 1/256 of the original image after multilayer pooling, the feature information of the small target object in the original image is lost in the process of multiple feature extraction, so Faster RCNN detects small target objects and the rate is poor. This paper makes corresponding improvements to the four parts of Faster RCNN to improve the accuracy of the network in detecting multiscale targets. The following is a detailed analysis of the improved Faster RCNN structure.

3.2.1 Data Enhancement

The quality of the dataset directly affects the quality of the feature extractions and the detection accuracy and generalization ability of the model, so this paper adds a data preprocessing module before the feature extraction network, through MixupImage, AutoAugmentImage and GirdMask, which are three data enhancement methods that improve the quality of the input dataset.

1) MixupImage is a data-enhanced pixel hybrid augmentation strategy. The strategy is based on the principle of empirical risk and proximity risk minimization, and multihot vector coding is obtained by weighting the single-hot vector coding of the traditional image labels. The specific operation is to add any two samples and the corresponding semantic labels through the weight parameters, and the formula is as follows:

$$\tilde{I} = \lambda I_i + (1 - \lambda) I_j \tag{2}$$

$$\tilde{Y} = \lambda y_i + (1 - \lambda) y_j \tag{3}$$

where I_i, I_j represent the pixel coordinates of any two images, and Y_i, Y_j represent the semantic information of the label. $\tilde{I}$ indicates the newly generated image, and $\tilde{Y}$ represents the label corresponding to the new image.

2) AutoAugmentImage is an image data augmentation method based on automatic machine learning. Its workflow is as follows: First, the image augmentation strategy is preset, set $S = \{S^i\}_{i=1}^{N}$, and generate a substrategy $S_i, S_i \in S$ from the augmentation strategy; set S through the search algorithm. Using the recurrent neural network as the controller, the model is obtained according to the training set of the strategy augmentation S_i; the model performance in the test set is used as feedback to update the search strategy.

3) GirdMaskImage is an image information deletion strategy that randomly discards an area on the image, which is equivalent to adding a regular term to the network, and can avoid network overfitting. GirdMask corresponds to $(r, d, \alpha_x, \alpha_y)$four parameters, and the specific settings are shown in Fig. 3.

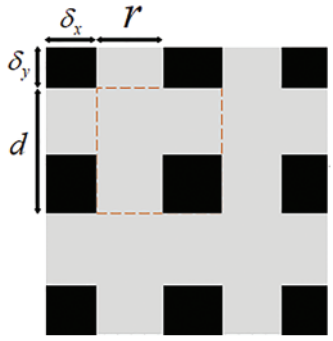


Figure 3: GirdMask

Where r represents the proportion of the original image information retained, d represents the size of a discarded area and α_x, α_y represents the distance between the first complete area and the image boundary.

3.2.2 Multiscale Feature Fusion

The traditional Faster RCNN network only uses the last layer feature map of the feature extraction network as the input of the RPN module. However, due to the rich semantic information of the deep feature map, a large number of detailed features are ignored and the ability to detect small targets is poor, so this paper uses a balanced semantic feature based on the same depth integration to identify the multilevel semantic features to improve the model's ability to detect small target objects. The specific structure of the model is shown in Fig. 4.

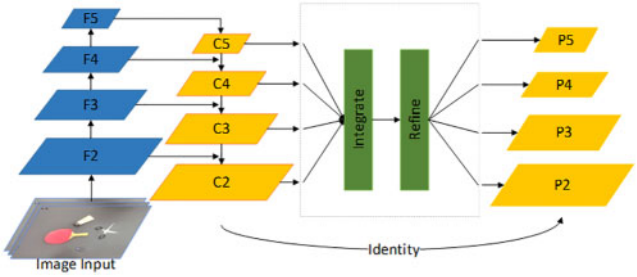


Figure 4: Pipeline visualization of balanced feature pyramid

The module consists of four steps, scale adjustment, integration, refinement and enhancement. The feature maps of different levels l in the feature extraction network structure are defined as F_l. The number of multilevel feature maps is defined as L, and the indices of the highest and lowest levels are defined as $l_{\max}$ and $l_{\min}$. The feature atlas $\{F_2, F_3, F_4, F_5\}$ can be obtained from the backbone network. The multiscale feature fusion process is as follows:

First, a 1×1 convolutional layer is input to reduce the number of feature maps and generate low-resolution feature map C_5. Then, C_5 performs 2 times the most recent upsampling and extracts the low-resolution feature map from the input P_5 to the 1×1 convolutional layer. Both have the same scale and add the fusion feature element-by-element to generate the required fusion feature map C_4. By analogy, a new multiscale fusion feature atlas $\{C_2, C_3, C_4, C_5\}$ is obtained.

In Fig. 4, C_2 is the highest-level feature map. To integrate multilevel features while retaining the original semantic features, we readjust the size of each level feature map$\{C_2, C_3, C_4, C_5\}$ to the same size as C_4 through bilinear interpolation and the maximum pooling method to compare each level feature map. The feature is scaled, and finally, a balanced semantic feature map is obtained through the following formula:

$$C = \frac{1}{L} \sum_{l=l_{\min}}^{l_{\max}} C_l \tag{4}$$

We obtain a further refined feature map by nonlocal manipulation of the obtained balanced feature map with an embedded Gaussian function, as follows:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f\left(x_i, x_j\right) g\left(x_j\right) \tag{5}$$

The resulting balanced semantic feature map is then rescaled using the same but opposite method to obtain a new feature set $\{P_2, P_3, P_4, P_5\}$ that enhances the original features. Through these processes, the feature maps at each level not only aggregate the features from the lower to the higher levels, but also obtain an equal amount of semantic information from the other levels.

Finally, the feature set $\{P_2, P_3, P_4, P_5\}$ is fed into the target detection network RCNN for category and location prediction, where the RCNN network structure selects feature maps P_k of different scales for the multiscale candidate regions as input to the ROI pooling layer, and the coefficients k are defined as

$$k = k_0 + \log_2 \frac{\sqrt{wh}}{224} \tag{6}$$

The parameter 224 indicates the size of the input data, the default k_0 is the feature map P_5, and w, h represents the length and width of the candidate regions, respectively.

The multiscale fusion feature map used in this paper contains different levels of semantic information and detailed features from the bottom to the top layer, with strong generalization. More shallow features are extracted while retaining the deep semantic layer, which helps in the recognition of small targets.

3.2.3 ROI Alignment

The traditional Faster RCNN uses ROI pooling to make the candidate frames generated by RPN share the feature map features and keeps the output size consistent. However, ROI pooling will perform approximate processing in the two steps of rounding the positions of candidate frames and rounding when extracting features, resulting in a mismatch between the detection information and the extracted features, and eventually affecting the detection of small targets, so this paper adopts the ROI align method, as Fig. 5 shows, to replace ROI pooling.

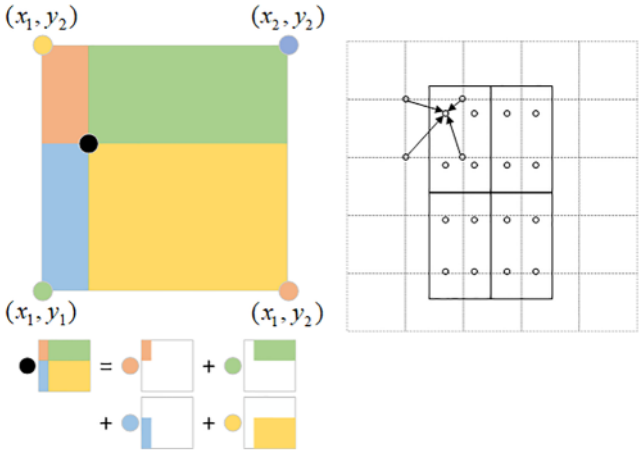


Figure 5: ROI alignment

ROI Align uses a regional feature aggregation approach, in which four points are evenly selected in the region N, the four closest points to each point are found on the feature map, the output values of the points are obtained by bilinear interpolation, and finally the output of the region is obtained by averaging the points N, as follows:

$$f = (1 - \Delta h)(1 - \Delta w)\frac{\partial L}{\partial y_{rj}} \tag{7}$$

$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j f * [d(i, i * (r, j)) < 1] \tag{8}$$

where $d(.)$ represents the distance between two points, Δh and Δw represent the difference with i and $i*(r,j)$ for the abscissa and ordinate, respectively.

3.2.4 Cascade Detection Head

Aiming at the problem that a single regression framework cannot effectively solve the ROI selection problem of the multiscale targets, this paper adopts a network framework of cascade regression to implement the dynamic selection of the IOU threshold. The specific formula is as follows:

$$f(x,b) = f_T \circ f_{T-1} \circ \cdots \circ f_1(x,b) \tag{9}$$

where T represents the total number of cascades used, f_T represents the result of each regression, and the initial distribution $\{b^1\}$ is optimized for each regression, and finally reaches the sample distribution of the corresponding stage $\{b^t\}$. In the target detection framework of this paper, we use three consecutive structures, as shown in Fig. 6.

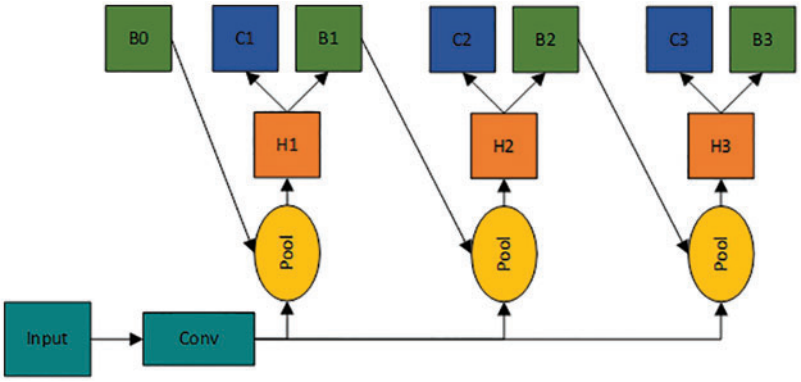


Figure 6: Cascade structure

Where B_0 represents the candidate area generated in the RPN network, and Conv represents the convolutional neural network. The specific process is as follows:

1) Input B_0 into ROI pooling to obtain the characteristic information of the region of interest.
2) Input the feature information obtained in Step 1 into the fully connected layer H_1, and then input the output features of H_1 into the classifier for classification and the frame regression function B_1 for fine-tuning and positioning.
3) Use the fine-tuned candidate box as the new input, and enter the next cascade structure.
4) Repeat Steps 1∼3 until the output result.
4) Through this method, the quality of the candidate frame is gradually improved, which can significantly improve the positioning accuracy of the bounding box.

4 Grab Gesture Generation Network

This paper divides the grabbing detection task into grabbing angle classification and grabbing frame regression. By using convolutional neural networks, the generalization ability of large-scale convolutions is used to perform global grasping predictions on the input graphics.

To improve the accuracy of the detection results, we design a more accurate grab detection model based on the grabbing detection network structure based on region extraction. First, ResNet50 is

selected as the backbone network for feature extraction, and the SE module is added to the residual structure. Through the two key operations of squeeze and excitation, the importance of each feature channel is automatically obtained by learning, and then according to this importance, the useful features are improved, and the features that are not useful for the current task are suppressed, as shown in Fig. 7.

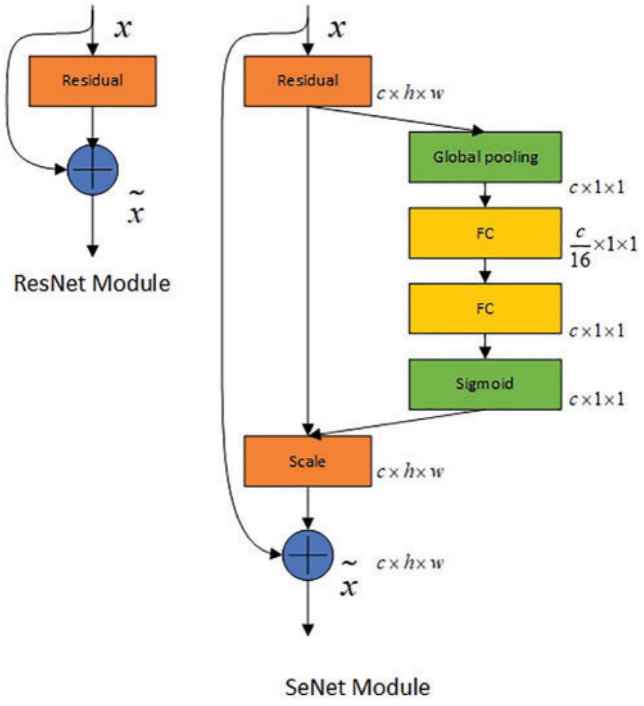


Figure 7: Structure comparison between ResNet and SeNet

The SE module mainly includes two operations, squeeze and excitation, which can be applied to any mapping: Assume that the convolution kernel is $V = [v_1, v_2, \cdots, v_C]$, where v_c represents the c convolution kernel. Define the output $U = [u_1, u_2, \cdots, u_c]$.

$$u_c = v_c * X = \sum_{s=1}^{C} v_c^s * x^s \tag{10}$$

where $*$ represents the convolution operation, and v_c^s represents the 2-D convolution kernel of the s channel. The SE module can extract the confounding caused by the mixture of the channel feature relationship and the spatial relationship learned by the convolution kernel, so that the model can directly learn the channel feature relationship.

Second, an FPN layer network is added after the feature extraction network to mix multiscale feature map information. RPN predicts the pose (anchor frame) of the selected region on the generated feature map and sends the generated feature vectors (anchor frame parameters) to two fully connected layers, namely, the classification layer and the regression layer. The cascade structure is added to obtain the evaluation score and regression coordinates of each anchor box.

The evaluation score and regression coordinates of the anchor box are represented by S.

$$S = (\alpha, \beta, x, y, w, h) \tag{11}$$

where (α, β) are the two scores used to determine whether the anchor frame is a grabbing area, (x, y, w, h) are the four physical values of the anchor frame's regression coordinates, x, y represent the center point coordinates of the anchor frame, and w, h represent the width and height of the anchor frame. In the prediction information that may be generated in the output image, the loss function is introduced as follows:

$$f(p) = \sum_{k} L_{gp\_cls}\left(p_k, p_k^*\right) \tag{12}$$

$$g(t, p) = \sum_{k} p_k^* L_{gp\_reg}\left(t_k, t_k^*\right) \tag{13}$$

$$L_{gpn}\left(\left\{(p_k, t_k)_{k=1}^{K}\right\}\right) = f + \lambda g \tag{14}$$

where L_{gp_cls} represents the cross entropy function, which is used to determine whether the capture area is included; L_{gp_reg} represents the regression loss function, which is used to predict the regression coordinates; λ represents the weight, k represents the index of the candidate area in a small batch of samples; $p_k^* = 1$ represents the anchor box containing the capture area, which is a positive sample; $p_k^* = 0$ indicates that the anchor frame does not contain the grabbing area and is a negative sample; t_k indicates the parameters of the anchor frame; and t_k^* indicates the coordinate vector k of the positive sample anchor frame mapped to the image. Input the obtained anchor frame and the feature map extracted by the ResNet structure into the ROI align layer of the above chapter, and perform feature normalization processing on the input features through the bilinear interpolation method.

The capture area prediction loss function defined by the formula is used to classify the angle of the capture frame and regress the coordinate position of the input anchor frame information.

$$L_{gcr}\left(\left\{(\rho_l, \beta_l)\right\}_{i=0}^{I}\right) = \sum_{c} L_{gcr\_cls}(\rho_l) + \lambda_2 \sum_{i} 1_{i \neq 0}(i)\, L_{gcr\_reg}\left(\beta_i, \beta_i^*\right), 0 \leq I \leq 20 \tag{15}$$

where I represents the number of angle categories, ρ_l represents the category probability of the anchor frame being the angle of the grabbing rectangle, β_l represents the corresponding grabbing bounding box ρ_l, L_{gcr_cls} represents the cross-entropy loss function used to predict the category of the grab angle, L_{gcr_reg} represents the anchor frame adjustment. The grab box of the coordinates returns to the loss function λ_2 represents the weight, used to balance the size of the two loss functions, and β_i^* represents the true value of the network candidate recommendation box. From this, the total loss function can be obtained as the formula.

$$L_{total} = L_{gpn} + L_{gcr} \tag{16}$$

The structure of the multitarget capture detection network is shown in Fig. 8. Drawing on the idea of the region extraction second-order target detection algorithm, we first determine whether each capture rectangle recommended by the RPN candidate region generation network contains objects that can be captured. Second, a capture is performed through prediction. The angle category to which the frame belongs determines the final capture angle and adjusts the boundary parameters of the predicted frame.

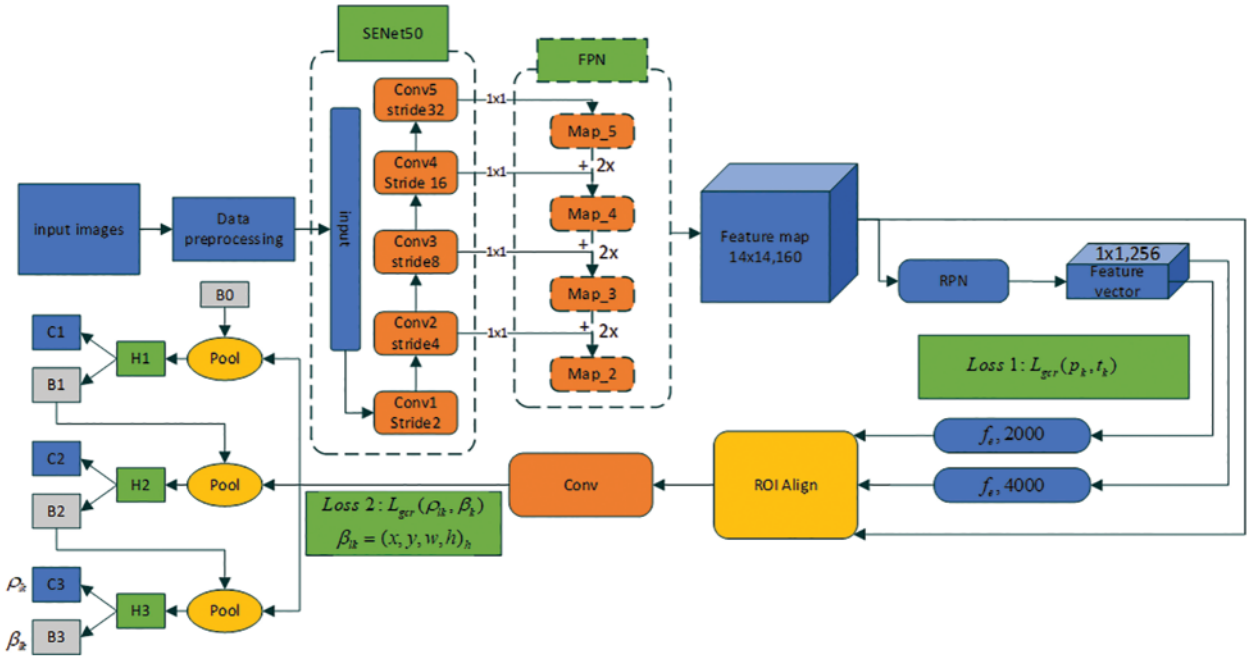


Figure 8: Multitarget crawling detection network structure

5 Multitarget IOU Hybrid Area Attitude Evaluation Algorithm

In the unstructured capture detection task, there is a situation where multiple capture targets interfere with each other. The captured target area is a subset of the target recognition background, and it is an inclusive relationship. Therefore, only the grasping area calculated with the target object as the background can finally determine the optimal grasping posture in the multitarget scene by analyzing the regional relationship between the target background and the grasping area. In summary, this paper proposes a multistage network architecture robot optimal grasping posture detection algorithm. The algorithm first identifies the target to be grasped through target detection and then generates a feasible grasping area for the target object through grasping detection. Finally, the multitarget IOU hybrid region pose evaluation algorithm is used to screen out the optimal grasping pose in the feasible grasping region of the target object.

The improved Faster RCNN is used to identify and locate the target object in the image and generate a positioning rectangle. The grabbing rectangle is generated by the grabbing detection model designed in this paper, and the grabbing area in the image is obtained. Using the detected bounding box of each target identified in the image as the background the following process is completed; calculate and generate the IOU of the grabbing rectangle and the target bounding box, filter out candidate grabbing areas, calculate the IOU of the candidate grabbing area and other target bounding boxes, and finally obtain the optimal grasping posture of the target object. The algorithm pseudocode is as follows.

Input: color map and data preprocessing
DATA: The t frame obtains the object position information T_t^i through the improved detection model.
DATA: Grab frame information T_t^{Gi} obtained by grabbing gesture generation network in frame t.
1: for each T_t^i in T_t do // Traverse all target detection frames on frame t.
2: Determine the capture target T_t^i in the input image information at frame t.
3: Define T_t^i as the capture area ROI in the image.
4: $IOU_i = \left(T_t^{Gi} \cap T_t^i\right) / T_t^{Gi}$
5: for each T_t^j in T_t do// Traverse all target detection frames on frame t.
6: $IOU_j = \left(T_t^{Gi} \cap T_t^j\right) / T_t^{Gi}$
7: if $(T_t^{Gi} \cap T_t^i \&\& IOU_j == 0)$
Then g$\leftarrow T_t^{Gi}$
8: else if $(IOU_i > 0.7 \&\& IOU_j < 0.1)$
Then g$\leftarrow \max\left(IOU_i - IOU_j\right)$
9: else go to 2:
10: end if
11: end if
12: end for
13: $d_c \leftarrow f\,(x_c, y_c)$ //$g = (x_c, y_c, w, h, \theta)$, g is the grab area parameter.
14: if $(d_c > threshold)$ // d_c is the maximum gripping distance.
15: then go to 2:
16: else $g \leftarrow (x_w, y_w, z_w, \theta)$
17: end if
18: end for

First, preprocess the input image, align the depth map to the color map, and use the coordinate conversion formula to calculate the three-dimensional coordinate value of the pixel in the robot coordinate system; second, input the processed image into the target detection module and the capture module. The detection module obtains the detection bounding box and the graspable area of the target to be grasped. Through the hybrid multitarget IOU hybrid region pose evaluation algorithm established in this paper, the detection boundary of each grasped target is used as the background to filter out the optimal grabbing posture, as shown in Fig. 9.

Calculate the IOU between all the grabbing areas of the target to be grabbed and the boundary of the grabbing target. When IOU > 0.7 or ${A_t}^{Gi} \subset {A^i}_t$, ${A_t}^{Gi}$ will be regarded as the candidate grabbing area of ${A_t}^i$. Calculate the IOU between the candidate grabbing area ${A_t}^{Gi}$ and the other target boundaries ${B_t}^i{C_t}^i$. When IOU < 0.1 or when IOU = 0, ${A_t}^{Gi}$ is set as the grabbing area, as shown in the grabbing rectangle ${A_t}^{Gi}$ in the figure. Taking the grabbing rectangle as a reference, using the center pixel value of the rectangle to calculate the 3D grab point of the robot end effector, and taking the angle of the grabbing rectangle relative to the axis of X in the image as the rotation angle of the robot end effector, the robot can establish the optimal grasping posture.

The algorithm proposed in this paper uses the detection bounding box of the target object as the benchmark and uses the mixed IOU of the graspable area and each target object as a factor to generate the optimal grasping posture of the target object. The algorithm in this paper is compared with the direct output of the target detection and capture detection, as shown in Fig. 10.

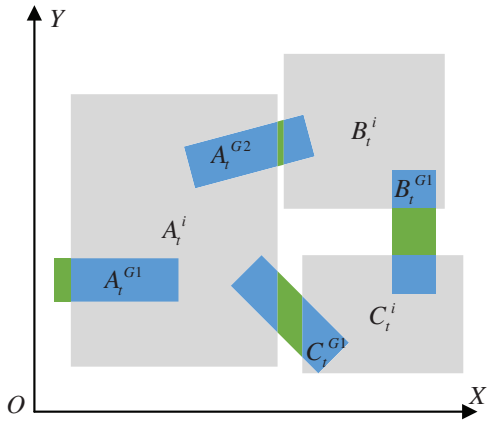
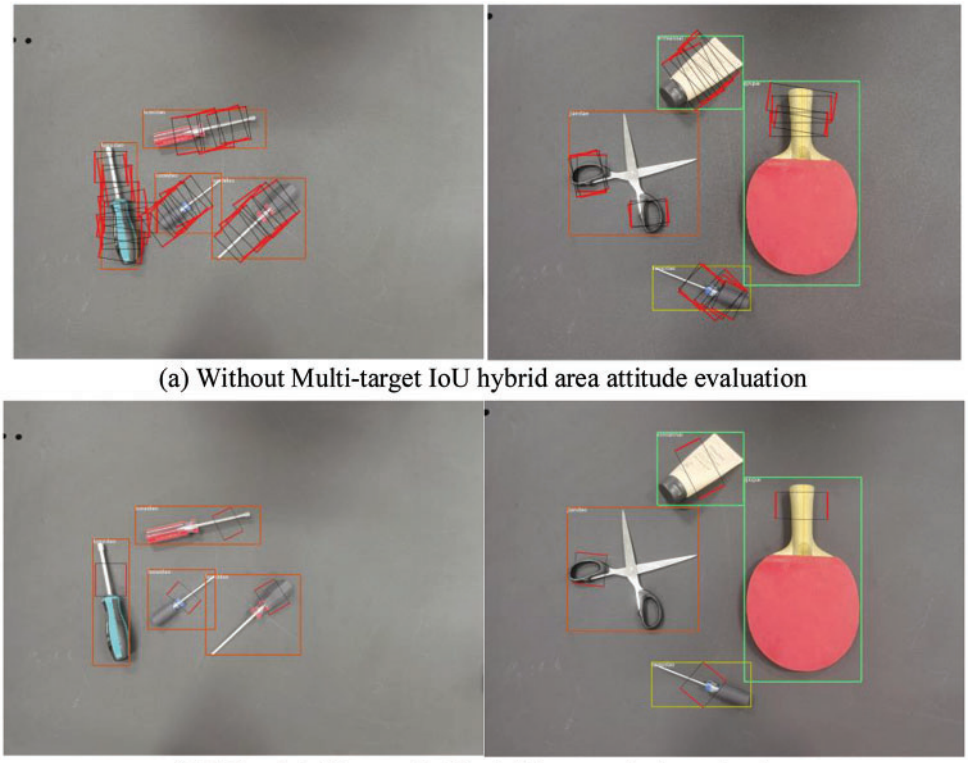


Figure 9: Multitarget IOU hybrid area attitude evaluation



(a) Without Multi-target IoU hybrid area attitude evaluation

(b) Using Multi-target IoU hybrid area attitude evaluation

Figure 10: Comparison of the different grasp detection method outputs

Fig. 10 shows the results of the grab detection and target detection of the multitarget IOU hybrid region evaluation algorithm. The grab detection cannot confirm the grab pose with the grab target, and grab detection will be interfered with by the target detection frame. Fig.10b is based on the results of the multitarget IOU hybrid region evaluation algorithm. Through comparison, the multitarget IOU hybrid region evaluation algorithm proposed in this paper can effectively attain the optimal grasping

posture generation in multitarget scenes, and can effectively avoid the interference of the background on the grasping detection, which is suitable for unstructured scenes.

6 Experimental Results and Analysis

6.1 Improved Faster RCNN Target Detection Experiment

In order to obtain better scene perception results, based on the Faster RCNN network, this paper introduces multi-scale balanced semantic feature fusion, cascade detection structure and various data enhancement methods to improve the detection ability of the model. To verify the improved model detection ability of this paper, this paper uses the COCO2017 data set to train and test the Faster RCNN, FPN, Cascade RCNN, and Libra R-CNN models, respectively. The default parameters of each network model are used as training parameters. The models of each network. The results are shown in table below.

As can be seen from Table 1, when ResNet 101 is used as the backbone network, the AP value of the traditional Faster RCNN network model in COCO tes-dev increases from 29.2% to 35.2% after using the FPN structure. On this basis, Cascade cascade structure and Balanced Semantic Feature Fusion (BFPN) are introduced respectively, and their AP indexes reach 42.8% and 41.1%, respectively. It can be concluded that the proposed algorithm has a significant improvement in detection accuracy compared with the traditional algorithm.

Table 1: Comparison of the detection results of each model on COCO tes-dev

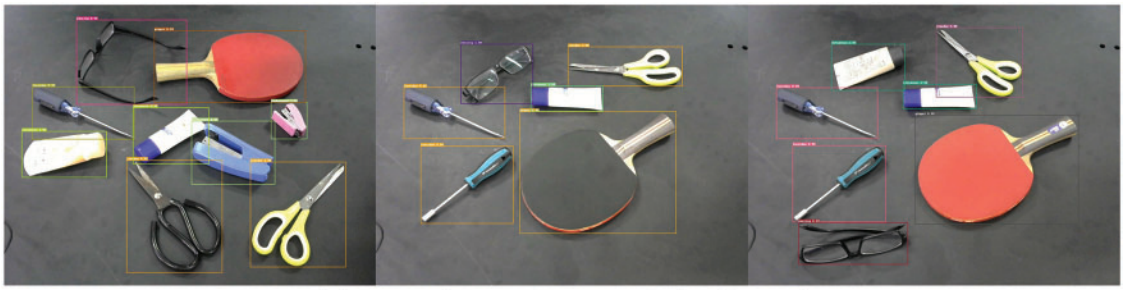
Method	Backbone	AP	AP50	AP75	APS	APM	APL
Faster RCNN	R-101	29.2	50.1	30.39	9.5	32	48.1
Faster RCNN + FPN	R-101	35.2	60.1	40.9	21.5	41.3	50.2
Cascade RCNN	R-101	42.8	62.1	46.3	23.7	45.5	55.2
Libra RCNN	R-101	41.1	62.1	44.7	23.4	43.7	52.5
Ours	R-101	**44.0**	**65.1**	**49.4**	**26.7**	**48.6**	**58.2**

For self-built datasets, when training the improved Faster RCNN network model, we set the momentum = 0.9, batch_size = 1, the number of iterations to max_iter = 10000, the initial learning rate base_lr = 0.00125, and the learning rate adopts the LinearWarmup update method. At the beginning of the training, the learning rate is reduced linearly from a very small value to a preset value, and then linearly decreases. In the experiment of this paper, the improved grab target detection network is trained on GTX2080ti, Window 10, Pytorch 1.7.0, and the Faster RCNN + FPN network structure (hereinafter referred to as FPN) is also trained in the same environment.

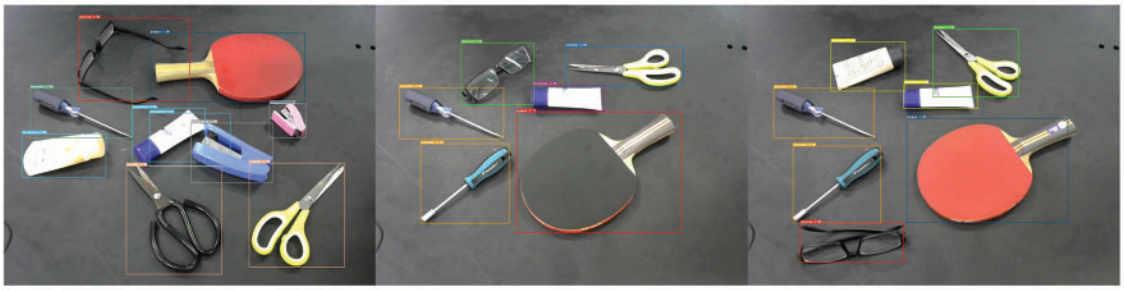
The experiment refers to the types of graspable objects that appear in the Cornell grasping dataset, and selects experimental samples for the object detection. Each image in the original Cornell grab dataset contains only one type of object. To resolve the multitarget object situation that often occurs in real scenes, this paper collects and creates a multitarget detection dataset by itself. This dataset can identify various samples in the dataset captured by Cornell. The dataset contains 309 RGB images of $1280 * 960$, and each image contains three to eight target objects. A comparison of some experimental results is shown in Fig. 11.

From the model detection results before and after improvement in Fig. 11, it can be seen that both the FPN network structure selected in this paper and the improved network model structure

can implement the position prediction and object classification of multitarget objects. However, in the detection results of the original model structure, it can be observed that there are some obvious detection problems, as shown in the red circle in Fig. 11. The type of problem mainly consists of four aspects: (1) The problem of multiple grabbing boxes for the same object. For example, there are two detection frames for the facial cleanser in the first picture of the comparison picture. (2) The position deviation problem of the detection box prediction, such as the prediction of the blue screwdriver in the second picture of the comparison picture and the glasses in the third picture. In the original model, there is a phenomenon in which some areas of the object are not in the detection frame. (3) The object classification error problem. (4) The missing detection problem. The reasons for these problems in the detection results of the original model structure are that the number of training set samples is too small, and the original model's ability to extract detailed features for various types of objects is not as good as the improved model structure.



（a）Improved Faster RCNN network detection results



（b）FPN network detection results

Figure 11: Comparison of different network detection results

As shown in Table 2, the improved network model structure in this paper achieved an average detection accuracy of 96.6% and a detection frame intersection ratio of 0.86 in the grab dataset created in this paper, and while maintaining a high detection accuracy, the improved Faster RCNN network model can achieve a detection speed of 17.5 FPS when detecting images.

Table 2: Detection network performance comparison

Network	P/%	$R_{(IOU)}$/%	FPS
Faster RCNN + FPN	85.2	0.78	12.7
Ours	**96.6**	**0.86**	**17.5**

6.2 Multitarget Grabbing Detection Experiment

The image of the dataset used for training and testing of the grasping pose generation network model is composed of the Cornell grasping dataset and the pictures in the self-built multitarget object dataset above. The new grab dataset was obtained by labeling the merged data samples with the rotation annotation tool, roLabelImg. It refers to the annotation method of the Cornell grab dataset, which is used to test the pose generation network model. Before training, the images of the Cornell dataset are randomly divided into a by of 5:1:1.

In this paper, the rectangle grabbing metric is used as the method to evaluate the network accuracy, and it is compared with the other grabbing detection models. The rectangle measurement uses a grabbing rectangle as a grab evaluation index. If the following two points are satisfied at the same time, the grabbing rectangle is considered for use to grab objects. First, the difference between the grabbing angle of the prediction box and the angle of the truth label must be less than 30°; second, the predicted Jaccard similarity coefficient must be greater than 25%. The Jaccard similarity coefficient predicts the similarity between the grabbed area and the true value label as:

$$J\left(G_p, G_t\right) = \frac{\left(G_p \cap G_t\right)}{\left(G_p \cup G_t\right)} \tag{17}$$

where G_p is the prediction to grab the rectangular area, and G_t is the true value of grabbing the rectangular area. The training parameters are batch_size = 128, lr = 0.0001, the attenuation coefficient is 0.1, the number of steps is 20000 and the total number of steps is 100000.

The improved model in this paper is compared with other capture detection models, 6 different types of objects are selected from the Cornell capture dataset and the real physical scene to evaluate the model, and the test results of the Cornell capture dataset and the real physical scene are shown in Table 3.

Table 3: Comparison experiment of the grab detection network

Method	Cornell data		Real scene	
	Time/s	Accuracy/%	Time/s	Accuracy/%
GD-CNN [40]	0.90	75	0.91	72
Faster R-CNN [41]	0.32	90	0.33	86
Ours method	**0.34**	**96**	**0.35**	**94**

The experimental results show that the grasping detection model designed in this paper can greatly improve the grasp accuracy and meet the robot's grasp accuracy requirements while ensuring the model's calculation time. Whether in Cornell data or real scene experiments, the proposed method has great advantages in accuracy, achieving 96% and 94%, respectively. Although it is 0.02 s slower than GD-Faster R-CNN, it can meet the actual situation of robot grasping.

The output result of the multitarget capture detection algorithm designed in this paper is shown in Fig. 12. The capture area predicted by the model performs well in the multitarget capture detection scene.

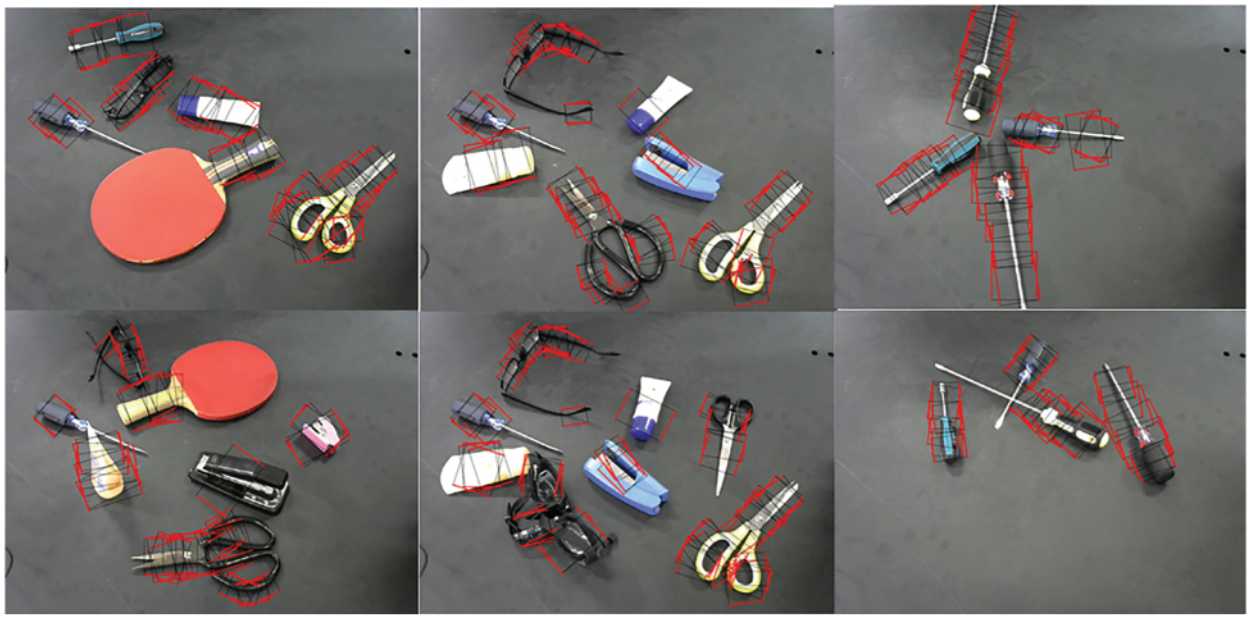


Figure 12: Multitarget grasp detection

6.3 Robot Optimal Grasping Experiment

As shown in Fig. 13, the hardware platform used in this experiment is mainly composed of three parts, the Baxter robot, Kinetic v1 camera and a mobile computer. The experiment uses the eye-to-hand system construction method, which uses a tripod to fix the camera in the opposite direction of the robot and makes the lens face the horizontal plane.

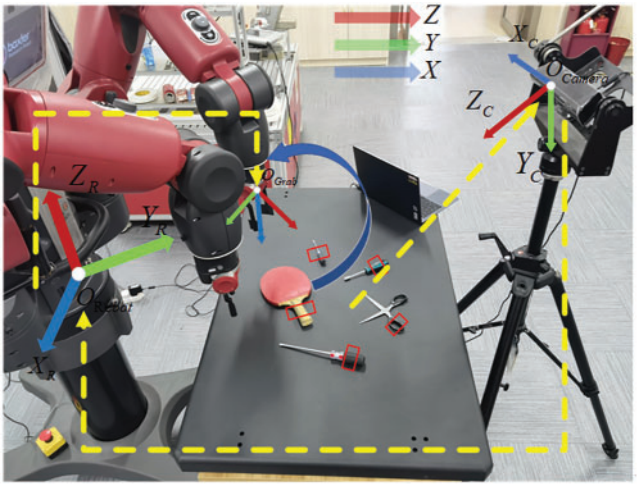


Figure 13: Grasp experiment platform

In this paper, four sets of grasping experiments are designed for grasping in different situations. In the experiment, two grasping objects, the tool category and the daily necessities category, are set up for grasping objects that may appear in application scenarios such as warehousing and logistics,

housekeeping services, and manufacturing. The object set includes the tool object set that has a vise, wrench, bearing, scissors, screwdriver and tape measure; the daily necessities object set includes a mouse, stapler, box, facial cleanser; screws and nuts are included as interference objects. The specific experimental content is as follows:

The first group of grasping Experiments A randomly selects a single target from the two grasping object sets as the grasping object in the scene. At the same time, there was only one object in the grasping scene, and the position of the grasping object was arbitrarily placed in the grasping workbench. The Baxter robot performs 20 grasping experiments and records the results. Part of the grabbing process is shown in Fig. 14.

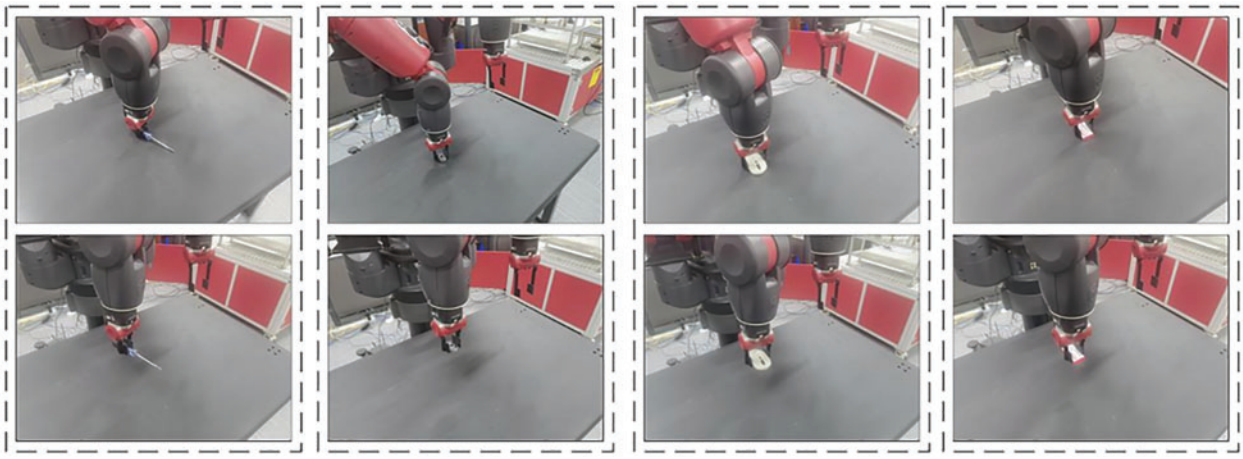


Figure 14: Experiment A-Example of the single-target grasping experiment process

The second group of grasping Experiments B selects multiple targets from two grasping object sets as grasping objects and simultaneously grasps multiple objects in the scene, including the objects to be grasped and the nongrasping objects for interference. The objects are scattered on the grabbing table, and they do not block or touch each other, simulating the needs of the grabbing task sequence planning, and controlling the Baxter robot to grab the objects in a preset order (bearing, wrench, screwdriver, tape measure, mouse), binding machine, box) for 40 grasping experiments and record the results. Part of the grabbing process is shown in Fig. 15.

The third group of grasping Experiments C randomly selects multiple targets from the two grasping object sets as grasping objects in the scene. At the same time, only objects are to be grasped in the grasping scene, and each object is placed in close proximity to the grasping table, and put them in contact with each other. The Baxter robot grabs 40 times and records the results. Part of the grabbing process is shown in Fig. 16.

The fourth group of grasping Experiments D randomly selects multiple targets from the two grasping object sets as grasping objects in the scene, and there are multiple objects in the grasping scene at the same time, including the objects to be grasped and the nongrasping objects used for interference. Each object is placed randomly on the grasping table, and there is a state of contact with each other. The Baxter robot is controlled to carry out 50 grasping experiments and records the results. Part of the grabbing process is shown in Fig. 17.

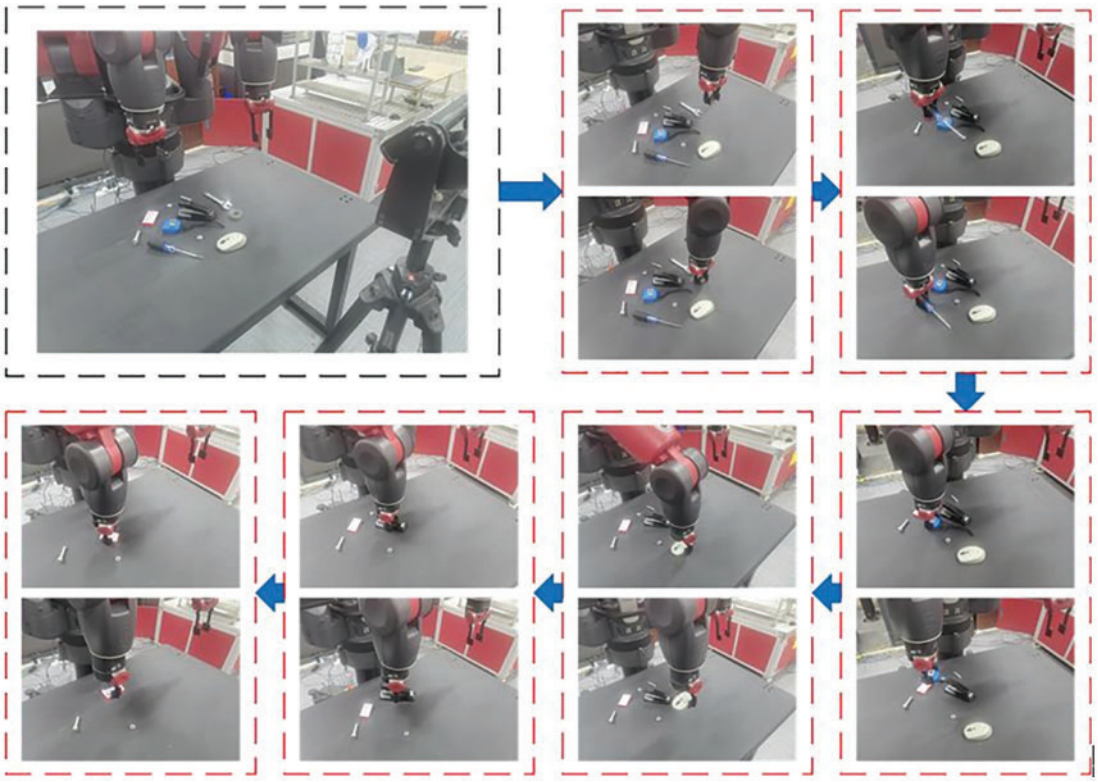


Figure 15: Experiment B-Example of the experimental process of multitarget, scattered and interfered grasping

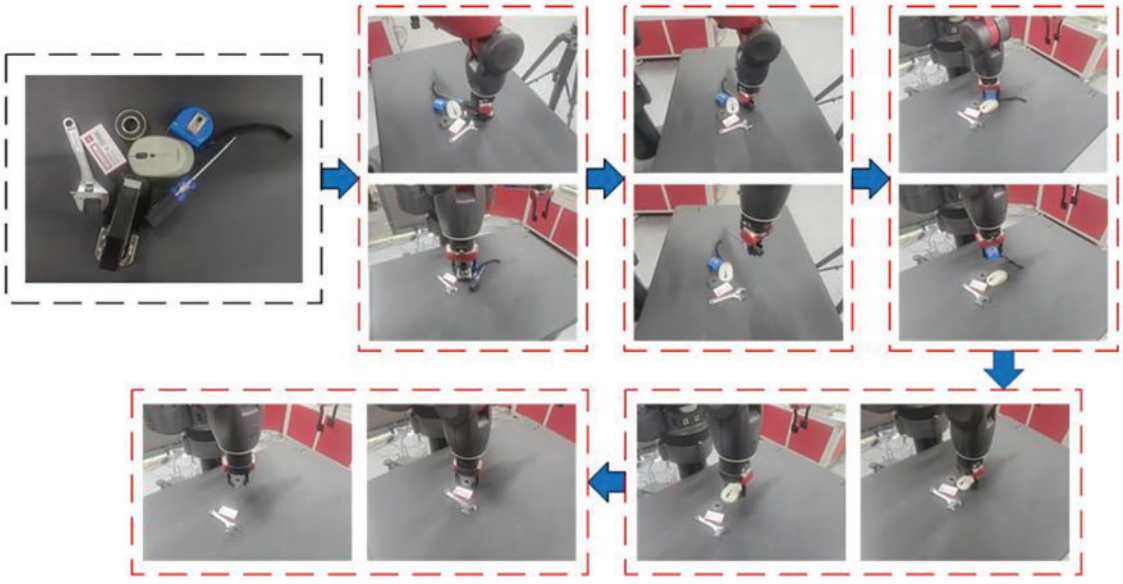


Figure 16: Experiment C-Example of the multitarget, contact, nondistraction grasping experimental process

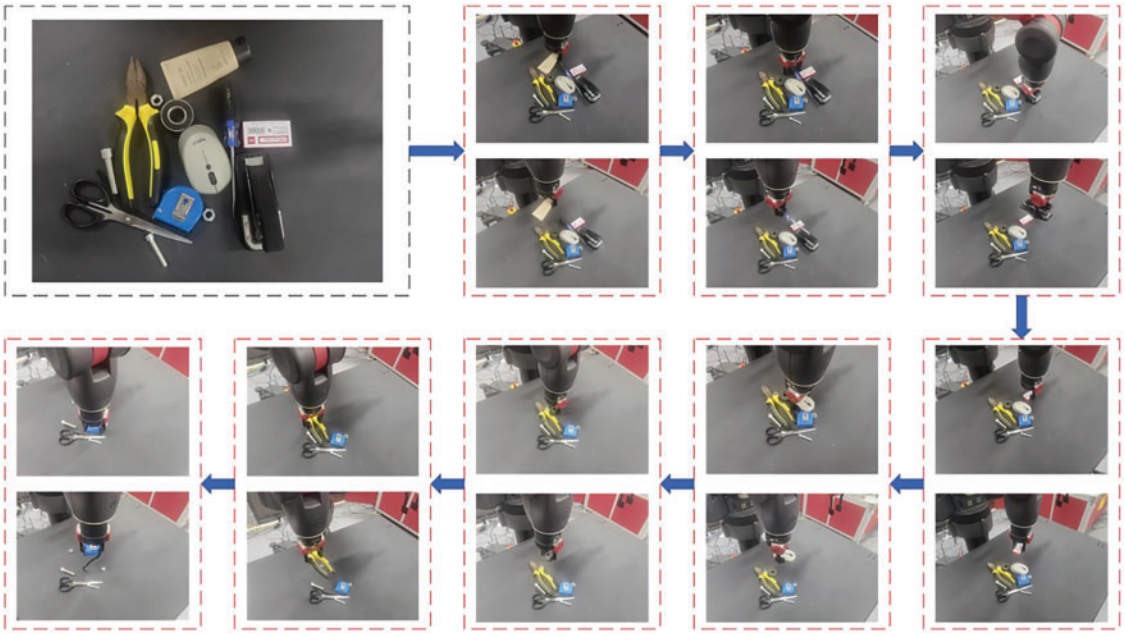


Figure 17: Experiment D-Examples of experimental procedures for multitarget, contact, and interference grasping

The results of the above robot grasping Experiments A, B, C and D are shown in the following table. The table includes the number of successful grasping object recognitions and the grasping success times of each group of experiments, and the overall grasping object recognition accuracy and grasping success rate are calculated.

From the experimental results of Table 4, we can see that the optimal grasping posture detection algorithm proposed in this paper has a high grasping success rate and accuracy, which can meet the actual needs of multitarget grasping tasks. The reason is that the grasping detection algorithm proposed in this paper combines the target detection results and pose generation results of multiple target objects in the image to generate the optimal grasping pose, which greatly avoids the possibility of capturing other targets when one of the objects is captured. The impact greatly reduces the probability of crawling failure.

Table 4: Experimental results and analysis

Experiments	Experiment times	Number of successful identifications	Crawl success times	Recognition accuracy	Grab accuracy	Number of successful identifications
A	20	20	20	100%	100%	20
B	40	39	37	97.5%	92.5%	39
C	40	38	36	95%	90%	38
D	50	45	43	90%	86%	45
Total	150	142	136	94.7%	90.7%	142

7 Conclusion

This paper presents a method for robot grasping detection. In an unstructured environment, this method uses multitarget object recognition boundary information as a reference to filter the multiple feasible capture regions generated in the image to obtain the optimal capture pose and to improve the optimal capture of the target object. The method proposed in this paper achieves an accuracy rate of 96.6% for target detection and recognition, surpassing the 85.2% accuracy rate of the traditional network, and the generation accuracy rate of the optimal grasping frame reaches 94%. The detection accuracy of the area avoids the robot arm interfering with other targets when grasping, and enhances the robustness and adaptability of the robot arm to grasp multitarget objects in unstructured scenes in the grasping task. In addition, the overall accuracy can reach 90.7% in the robot real grasp detection experiment.

In summary, for multitarget irregular objects in unstructured scenes, the algorithm proposed in this paper effectively completes the grasping task. However, the structure of the method proposed in this paper is slightly complex. In the future, we will take lessons from the actual implementation of multitask target detections and optimize the multinetwork architecture as a whole to achieve more efficient object classification and grab detection. In addition, to realize the application of robots in more complex scenes, and to monitor the whole process of robots' flexible grasping tasks in real time, it is possible to consider adding various sensors such as force, position, and EEG to improve the robot's ability to perceive the environment and interact with people.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China (No. 52165063), Guizhou Provincial Science and Technology Projects (Qiankehepingtai-GCC [2022] 006-1, Qiankehezhicheng [2021] 172, [2021] 397, [2021] 445, [2022] 008, [2022] 165), Natural Science Research Project of Guizhou Provincial Department of Education (Qianjiaoji [2022] No. 436) and Guizhou Province Graduate Research Fund (YJSCXJH [2021] 068).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Kiatos, M., Sarantopoulos, I., Koutras, L., Malassiotis, S., Doulgeri, Z. (2022). Learning push-grasping in dense clutter. *IEEE Robotics and Automation Letters, 7(4),* 8783–8790. https://doi.org/10.1109/LRA.2022.3188437
2. Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D. et al. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature, 485(7398),* 372–375. https://doi.org/10.1038/nature11076
3. Wen, L., Gao, L., Li, X., Zeng, B. (2021). Convolutional neural network with automatic learning rate scheduler for fault classification. *IEEE Transactions on Instrumentation and Measurement, 70,* 1–12. https://doi.org/10.1109/TIM.2020.3048792
4. Wen, L., Li, X., Gao, L. (2021). A new reinforcement learning based learning rate scheduler for convolutional neural network in fault classification. *IEEE Transactions on Industrial Electronics, 68(12),* 12890–12900. https://doi.org/10.1109/TIE.2020.3044808
5. Wen, L., Wang, Y., Li, X. (2022). A new cycle-consistent adversarial networks with attention mechanism for surface defect classification with small samples. *IEEE Transactions on Industrial Informatics, 18(12),* 8988–8998. https://doi.org/10.1109/TII.2022.3168432

6. Dhal, K., Karmokar, P., Chakravarthy, A., Beksi, W. J. (2022). Vision-based guidance for tracking multiple dynamic objects. *Journal of Intelligent & Robotic Systems, 105(3),* 1–23. https://doi.org/10.1007/s10846-022-01657-6
7. Pan, Z., Zeng, A., Li, Y., Yu, J., Hauser, K. (2022). Algorithms and systems for manipulating multiple objects. *IEEE Transactions on Robotics,* 1–19. https://doi.org/10.1109/TRO.2022.3197013
8. Wang, H., Li, H., Wen, X., Luo, G. (2021). Unified modeling for digital twin of a knowledge-based system design. *Robotics and Computer-Integrated Manufacturing, 68(3),* 102074. https://doi.org/10.1016/j.rcim.2020.102074
9. Xu, X., Hu, Z., Su, Q., Li, Y., Dai, J. (2020). Multivariable grey prediction evolution algorithm: A new metaheuristic. *Applied Soft Computing Journal, 89,* 106086. https://doi.org/10.1016/j.asoc.2020.106086
10. Wang, J., Zheng, P., Qin, W., Li, T., Zhang, J. (2019). A novel resilient scheduling paradigm integrating operation and design for manufacturing systems with uncertainties. *Enterprise Information Systems, 13(4),* 430–447. https://doi.org/10.1080/17517575.2018.1526322
11. Fan, Q., Huang, H., Li, Y., Han, Z., Hu, Y. et al. (2021). Beetle antenna strategy based grey wolf optimization. *Expert Systems with Applications, 165(5),* 113882. https://doi.org/10.1016/j.eswa.2020.113882
12. Li, S., Zheng, P., Fan, J., Wang, L. (2022). Toward proactive human-robot collaborative assembly: A multimodal transfer-learning-enabled action prediction approach. *IEEE Transactions on Industrial Electronics, 69(8),* 8579–8588. https://doi.org/10.1109/TIE.2021.3105977
13. Zhong, H., Li, X., Gao, L., Li, C. (2022). Toward safe human-robot interaction: A fast- response admittance control method for series elastic actuator. *IEEE Transactions on Automation Science and Engineering, 19(2),* 919–932. https://doi.org/10.1109/TASE.2021.3057883
14. Wang, H., Tao, J., Peng, T., Brintrup, A., Kosasih, E. E. et al. (2022). Dynamic inventory replenishment strategy for aerospace manufacturing supply chain: Combining reinforcement learning and multi-agent simulation. *International Journal of Production Research, 60(13),* 4117–4136. https://doi.org/10.1080/00207543.2021.2020927
15. Liang, H., Ma, X., Li, S., Görner, M., Tang, S. et al. (2019). PointNetGPD: Detecting grasp configurations from point sets. *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3629–3635. Montreal, QC, Canada, IEEE.
16. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G. et al. (2012). Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. *Asian Conference on Computer Vision*, pp. 548–562. Daejeon, Korea, Springer.
17. Drost, B., Ulrich, M., Navab, N., Ilic, S. (2010). Model globally, match locally: Efficient and robust 3D object recognition. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 998–1005. San Francisco, CA, USA, IEEE.
18. Hinterstoisser, S., Lepetit, V., Rajkumar, N., Konolige, K. (2016). Going further with point pair features. *European Conference on Computer Vision*, pp. 834–848. Amsterdam, Netherland, Springer.
19. Chu, F., Xu, R., Vela, P. A. (2018). Real-world multiobject, multigrasp detection. *IEEE Robotics and Automation Letters, 3(4),* 3355–3362. https://doi.org/10.1109/LRA.2018.2852777
20. Fan, Q., Huang, H., Chen, Q., Yao, L., Yang, K. et al. (2022). A modified self-adaptive marine predators algorithm: Framework and engineering applications. *Engineering with Computers, 38(4),* 3269–3294. https://doi.org/10.1007/s00366-021-01319-5
21. Fan, Q., Huang, H., Yang, K., Zhang, S., Yao, L. et al. (2021). A modified equilibrium optimizer using opposition-based learning and novel update rules. *Expert Systems with Applications, 170(9),* 114575. https://doi.org/10.1016/j.eswa.2021.114575
22. Asif, U., Tang, J., Harrer, S. (2019). Densely supervised grasp detector (DSGD). *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8085–8093. Honolulu, Hawaii, USA.

23. Li, Y., Huang, H., Xie, Q., Yao, L., Chen, Q. (2018). Research on a surface defect detection algorithm based on mobilenet-SSD. *Applied Sciences, 8(9),* 1678. https://doi.org/10.3390/app8091678
24. Hernandez, C., Bharatheesha, M., Ko, W., Gaiser, H., Tan, J., et al. (2017). Team delft's robot winner of the amazon picking challenge 2016. In: Behnke, S., Sheh, R., Sanel, S., Lee, D. (Eds.), *Lecture notes in computer science*, vol. 9776. Cham: Springer. https://doi.org/10.1007/978-3-319-68792-6_51
25. Zeng, A., Yu, K. T., Song, S., Suo, D., Walker, E. et al. (2017). Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1383–1386. Singapore, IEEE.
26. Kendall, A., Gal, Y., Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491. Salt Lake City, UT, USA.
27. Sener, O., Koltun, V. (2018). Multi-task learning as multiobjective optimization. *Advances in Neural Information Processing Systems, 31,* 525–536.
28. Zhang, T., Zhang, C., Hu, T. (2022). A robotic grasp detection method based on auto-annotated dataset in disordered manufacturing scenarios. *Robotics and Computer-Integrated Manufacturing, 76(6),* 102329. https://doi.org/10.1016/j.rcim.2022.102329
29. Song, K., Wang, J., Bao, Y., Huang, L., Yan, Y. (2022). A novel visible-depth-thermal image dataset of salient object detection for robotic visual perception. *IEEE/ASME Transactions on Mechatronics,* 1–12. https://doi.org/10.1109/TMECH.2022.3215909
30. Tee, K. P., Cheong, S., Li, J., Ganesh, G. (2022). A framework for tool cognition in robots without prior tool learning or observation. *Nature Machine Intelligence, 4(6),* 533–543. https://doi.org/10.1038/s42256-022-00500-9
31. Laili, Y., Chen, Z., Ren, L., Wang, X., Deen, M. J. (2023). Custom grasping: A region-based robotic grasping detection method in industrial cyber-physical systems. *IEEE Transactions on Automation Science and Engineering, 20(1),* 88–100. https://doi.org/10.1109/TASE.2021.3139610
32. Yin, Z., Li, Y., Cai, J., Lu, H. (2022). Robotic grasp detection for parallel grippers: A review. *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1184–1187. Los Alamitos, CA, USA.
33. Yin, Z., Li, Y. (2022). Overview of robotic grasp detection from 2D to 3D. *Cognitive Robotics, 2(3),* 73–82. https://doi.org/10.1016/j.cogr.2022.03.002
34. Scaramuzza, D., Martinelli, A., Siegwart, R. (2006). A flexible technique for accurate omnidirectional camera calibration and structure from motion. *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pp. 45. New York, USA, IEEE.
35. Kumra, S., Kanan, C. (2017). Robotic grasp detection using deep convolutional neural networks. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 769–776. Vancouver, BC, Canada, IEEE.
36. Liu, Y., Sun, P., Wergeles, N., Shang, Y. (2021). A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications, 172(4),* 114602. https://doi.org/10.1016/j.eswa.2021.114602
37. Sikha, O. K., Bharath, B. (2022). VGG16-random fourier hybrid model for masked face recognition. *Soft Computing, 26(22),* 12795–12810. https://doi.org/10.1007/s00500-022-07289-0
38. Qian, Y., Woodland, P. C. (2016). Very deep, convolutional neural networks for robust speech recognition. *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 481–488. San Diego, CA, USA.
39. Liu, B., Zhang, X., Gao, Z., Chen, L. (2017). Weld defect images classification with VGG16-based neural network. *International Forum on Digital TV and Wireless Multimedia Communications*, pp. 215–223. Singapore: Springer.

40. Redmon, J., Angelova, A. (2015). Real-time grasp detection using convolutional neural networks. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1316–1322. Seattle, WA, USA, IEEE.
41. Ren, S., He, K., Girshick, R., Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 39(6),* 1137–1149.