# A lightweight grasping pose estimation method for retail warehousing

**Qingni Yuan**
Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University

**Chen Wang** ( 944068633@qq.com )
Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University

**Jianyou Qi**
Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University

**Xiaoying Du**
Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University

**Research Article**

**Keywords:** Lightweight grasping, R-Resblock, RFB-SE, Dilated convolution, V-rep

**Posted Date:** October 28th, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-2189487/v1

# A lightweight grasping pose estimation method for retail warehousing

Qingni Yuan[1,2,3], Chen Wang[1*], Jianyou Qi[1] and Xiaoying Du[1]

[1]Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University, Guiyang, 550025, China.

[2]State Key Laboratory of Public Big Data, Guizhou University, Guiyang, 550025, China.

[3]School of Mechanical Engineering, Organization, Street, City, 610101, State, Country.

*Corresponding author(s). E-mail(s): 944068633@qq.com;
Contributing authors: qnyuan@gzu.edu.cn; 942793525@qq.com;
1831661907@qq.com;

**Abstract**

Robotic grasping has been widely used in various industries. How to meet the requirements of grasping accuracy and grasping speed at the same time is a challenging problem in real-time grasping tasks. In this paper, aiming at the real-time grasping task in retail warehousing, a lightweight grasping pose estimation model for retail warehousing is proposed. The model first uses the Focus module to perform lossless double downsampling, and learns each feature map of the upper layer through the dilated convolution block to expand the receptive field; then, the R-Resblock structure is improved to perform multi-scale feature fusion, and a lightweight RFB-SE module is designed to enrich feature information and reduce the number of parameters. Finally, after upsampling and restoring the image, the grasping quality, grasping angle, and grasping width of the target are regressed to obtain the optimal grasping pose of the target item. Experiments are carried out in the Cornell dataset, Jacquard dataset, and simulation environment respectively. The experimental results show that the method has a grasping accuracy of 97.8% and a grasping speed of 78FPS on the Cornell

dataset. The success rate is 91.5%, and the grasping task in a retail warehouse environment is simulated in grasping simulation experiments.

**Keywords:** Lightweight grasping,R-Resblock,RFB-SE,Dilated convolution,V-rep

# 1 Introduction

Robot grasping is an important way for robots to interact with the environment. At present, robots have been used in industrial production and service fields, such as parts assembly, workpiece sorting, and intelligent welding. With the rapid development of the retail industry, many companies have begun to develop robots for retail environments. High grasping success rate and grasping efficiency have always been two key indicators for robotic grasping tasks. However, grasping tasks in retail environments, such as real-time grasping and sorting of irregular-shaped items under complex backgrounds, are not the success rate and real-time performance of grasping has high requirements. Therefore, how to ensure the grasping accuracy and speed in the real-time grasping process is still a big challenge in the grasping field.

Robot grasp detection methods can be divided into two categories: analysis-based methods and data-driven methods. Analysis-based methods use manual design to extract object features or obtain the best grasping position of the object based on the 3-D model of the object [1, 2], but it requires the object to be known and is not suitable for use in an unstructured environment. Data-driven grasping methods[3] currently mainly use end-to-end methods.

The end-to-end method[4–6] extracts image feature information by constructing a neural network and directly obtains the grasping position information, which has good performance. Cheng et al. [7] proposed a Randomly Cropped Ensemble Neural Network (RCE-NN), which solved the detection of similar overlapping objects but could only detect objects with similar features. Park et al. [8] used a deep neural network (DNN) to fuse object detection and grasp detection, but ignored the shape information of the object and lacked a grasp of grasping angle, making it difficult to achieve effective grasping in complex environments. Zhu et al. [9] proposed a feature pyramid-based grasping prediction network to complete the uncertainty estimation of network grasping. Shang et al. [10] combined the grasping pose prediction network and grasping rectangle detection network to construct a multi-level convolutional neural network (ML-CNN), which effectively improved the grasping accuracy, but the number of parameters was too large. Zhang et al. [11] proposed a coarse-to-fine cascade Faster R-CNN based on multi-scale feature maps to achieve stacked fruit grasping. Liu et al. [12] developed a structure that combines interactive exploration with a composite robotic hand for robotic grasping in complex environments, but deep reinforcement learning methods have high hardware

requirements and are currently difficult to achieve in industrialized applications. Chiu et al. [13] fused object detection with image segmentation and used key points for grasping. Yu et al. [14] integrated the detection segmentation network with the optimal grasp pose selection network, which can optimize the grasp pose. The method of object detection and image segmentation can effectively improve the grasping accuracy, but at present, both object detection and image segmentation require a large number of parameter operations, which will lead to the problem of poor real-time detection and high hardware requirements for deployment in industrial applications. Chen et al. [15] proposed an edge-based grasp detection strategy, which fused low-level features and convolutional neural networks, but they did not consider depth image information, resulting in insufficient detection accuracy. Xu et al. [16] proposed a detection method by key points, which reduces the detection difficulty by grouping key points. Ruan et al. [17] proposed a novel surface contact model to parameterize the contact area, thereby evaluating the grasping quality and improving grasping accuracy.

At present, the research on grasping algorithms mainly focuses on improving the accuracy, and there are few types of research on lightweight development aimed at improving the speed. Most of the algorithms have high requirements on hardware system conditions, so it is difficult to realize industrial production and application. Aiming at grasping accuracy and grasping speed requirements in a retail environment, this paper proposes a lightweight grasping pose estimation method for retail warehousing. The main contributions are as follows:

1) A lightweight grasping detection algorithm (RS-ConvNet) is proposed, which can make full use of feature information to improve the detection accuracy, and at the same time, the number of parameters is lower, which can meet the requirements of robot grasping accuracy and speed at the same time.

2) A multi-scale fusion R-Resblock module is designed, and a multi-scale feature segmentation layer is added based on the residual structure, which can better utilize the features of different scales to optimize the algorithm performance.

3) A lightweight module RFB-SE is designed to enrich feature information while reducing the number of parameters. The attention mechanism SE is used to grasp the learning area and obtain richer and more effective image features.

The structure of this paper is summarized as follows: Section 2 describes the grasp representation method. In Section 3, a lightweight grasp pose estimation method RS-ConvNet is proposed. Then, the corresponding experimental verification is carried out in Section 4. Finally, Section 5 presents a summary and outlook for future work.

## 2 Problem Statement

In the retail warehousing environment, the robot needs to perform grasping and handling tasks in complex objects to ensure that the grasping task is

completed with the best grasping posture, and at the same time, the robot is required to meet the real-time operation requirements, which requires more position information to optimize the grasping position of the robotic arm. In common grasping frameworks, the 5-D grasping pose representation method is mostly used, and the grasping pose $G_r$ representation method in [18] is used in this paper. As shown in formula (1):

$$G_r = (P, \Theta_r, w_r, Q) \tag{1}$$

Where $P = (x, y, z)$ represents the three-dimensional coordinates of the grasping center point, $\Theta_r$ represents the rotation angle of the gripper around the positive direction of the x-axis, $W_r$ is the width of the gripper opening, and Q represents the probability of each pixel in the image, which is a probability distribution between 0 and 1.

As shown in figure 1, an RGB-D image of size h×w from the Kinect camera, the grasp pose $G_i$ of the image can be defined as:

$$G_i = (u, v, \theta_i, W_i, Q) \tag{2}$$

where (u,v) represents the grasping center in image coordinates, $\theta_i$ is the rotation angle in the image coordinate system,$W_i$ is the grasping width in image coordinates, and Q has the same meaning as in formula (2).

After obtaining the predicted value in the image coordinate system, the five-dimensional grasping pose in the image coordinate system can be converted into the robot pose in the end-effector coordinate system through the robot kinematics analysis, to realize the robot in the retail warehousing environment object grasping task.

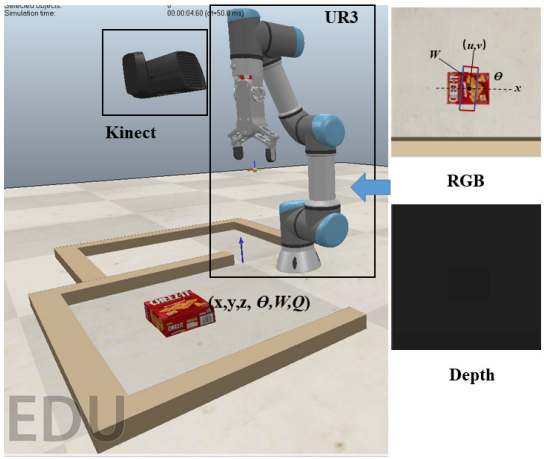


**Fig. 1**: Schematic diagram of grasping position

# 3 Principles and Methods

The robot grasping task needs to establish the relationship between the visual sensor information and the robot grasping pose. How to quickly and effectively obtain the best grasping pose is the key to completing the robot grasping task. Aiming at the above problems, this paper proposes a lightweight grasping pose estimation method for retail warehousing. As shown in Fig. 2, the image information obtained by the depth camera is first input into the Focus for lossless double downsampling, and the dilated convolution is used to expand the receptive field and learn the rich information in the feature map. After standard convolution downsampling, the improved R-Resblock module is used for multi-scale feature fusion, and the lightweight RFB-SE is used to better grasp the feature information of each scale. Finally, after three upsampling, the grasping quality evaluation, grasping angle regression, and grasping width detection are obtained. Through the three kinds of information, the grasping pose estimation of the robot is finally obtained.

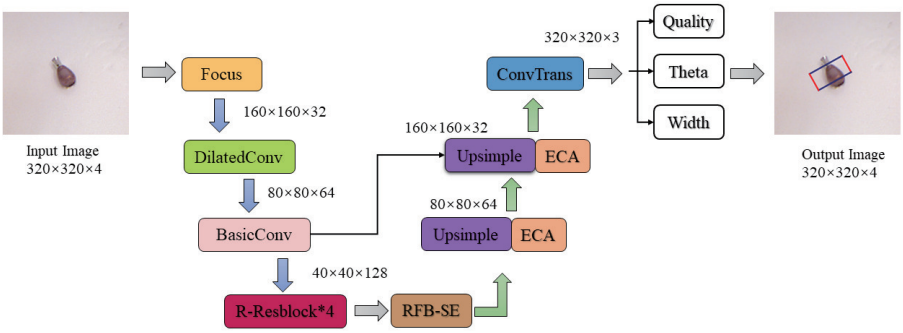


**Fig. 2**: Robotic grasping network architecture in retail warehousing scenario

## 3.1 Focus module

Inspired by YOLOv5, the Focus module can cut the input image, obtain a value for every other pixel, and finally generate four images, which can quadruple the channel expansion without information loss. If the input channel is an RGB-D four-channel, it will become 16 channels after passing through the Focus module. Finally, the obtained new image is subjected to convolution operation, the number of channels is further expanded to 32 by standard convolution, and the feature map of double downsampling without information loss is obtained. Fig. 3 shows the processing of the Focus module.

## 3.2 Dilated convolution and standard convolution

After downsampling without information loss, feature maps with rich information are obtained. To better learn more feature information without

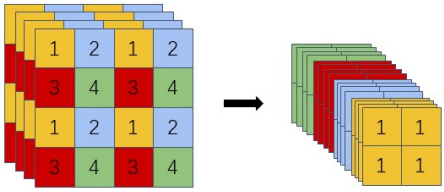


**Fig. 3**: Processing of the Focus module

introducing other parameters, this paper uses dilated convolution to expand the receptive field. Compared with the standard convolution operation, no additional parameters need to be introduced, and the output feature map size of the image can't be changed, which can better meet the needs of lightweight operations.

As shown in Fig. 4, the receptive field of the ordinary convolution in the left picture is 3, the receptive field of the dilated convolution with the dilation rate of 2 in the right picture is 5, and the dilated convolution can obtain more feature information. This paper sets the dilation rate of dilated convolution to 5.

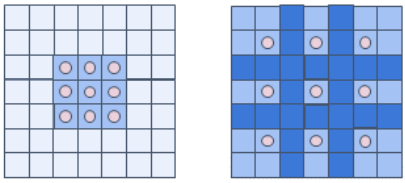

**Fig. 4**: Comparison of ordinary convolution and dilated convolution (dilation rate=2)

If the dilated convolution is used multiple times, the network may degenerate, so the standard convolution is used for the second dimension downsampling after the dilated convolution. The standard convolution stride used is 2, the convolution kernel size is 3×3, and the activation function is SiLU. Compared with Relu, it has the characteristics of no upper bound and lowers bound, smooth and non-monotonic.

## 3.3 R-Resblock for multi-scale feature fusion

In order to better learn multi-scale feature information and improve the detection performance of the network, this paper designs an R-resblock structure for multi-scale feature fusion based on the standard residual structure, as shown in Fig. 5.

First, the obtained feature map is extracted by convolution of 1×1, and then it is divided into three parts, namely IN1, IN2, and IN3. The first part

directly through the 3×3 convolution for feature extraction, and then transmit the feature image to OUT1; the second part inputs the output feature map of the previous part together with the input feature map of IN2 into a 3×3 convolution kernel for feature extraction, to obtain the output feature OUT2; the third part of the output feature map of the previous part and the input feature map of another group of input feature maps fuse information and input IN3 of the group into the convolutional kernel of 3×3 for feature extraction, to obtain output features OUT3.Finally, after obtaining the feature information of three scales, the network learning ability is enhanced by ECA attention, the convolutional kernel sent to 1×1 is sent for processing, the feature information is fully fused with the cross-layer connection, and the output feature map is finally obtained.

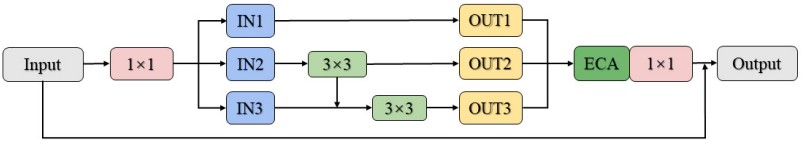


**Fig. 5**: R-Resblock structure of multi-scale feature fusion, 1×1, 3×3 represents the size of the convolution kernel

## 3.4 Lightweight RFB-SE

The traditional RFB and RFB-s models are designed based on the Inception module, which utilizes a multi-layer convolutional nested structure to improve accuracy. To make better use of the precision advantage of RFB and simplify the model parameters, the existing RFB model is improved. The improved RFB-SE structure is shown in Fig. 6.

The RFB-SE designed in this paper has a total of 5 branches, including a Shortcut connection branch and 4 feature extraction branches, and the convolution kernel settings of the specific 4 branches are shown in Fig. 6. After obtaining feature images of different scales in 4 branches, the images are stitched together and convoluted 1×1 to generate new feature images. The processed feature map is added to the Shortcut connection branch, and the result is reactivated by SE attention and fed into the next connection layer.

## 3.5 Loss function

To better determine the network output parameters, this paper tested a variety of regression loss functions, including MSE, L1, L2, and Smooth-L1, and finally found that the Smooth-L1 function performed better, so this paper uses Smooth-L1 as the regression loss function of the output value, and the loss function of the grasp model is defined as follows:

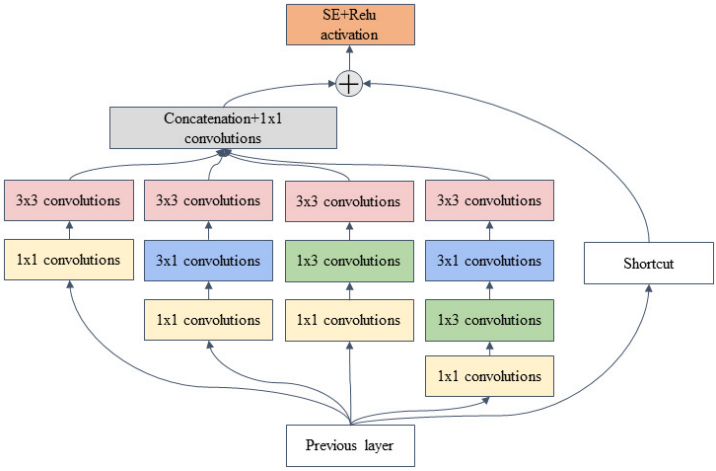


**Fig. 6**: RFB-SE structure

$$L_r(x,y) = \sum_{i}^{N} \sum\nolimits_{m\in\{q,\cos(2\theta),\sin(2\theta),W\}} Smooth_{L1}(x_i - y_i) \tag{3}$$

The $Smooth_{L1}(x_i - y_i)$ is calculated as follows

$$Smooth_{L1}(x_i, y_i) = \begin{cases} 0.5(y_i - x_i)^2, if|y_i - x_i| < 1 \\ |y_i - x_i| - 0.5, otherwise \end{cases} \tag{4}$$

where N is the number of grabbing candidate boxes, q and w represent the grabbing quality and the width of the grabber opening, respectively,$\cos(2\theta), \sin(2\theta)$ are the parameter representations of the grabbing angle. $y_i$ represents the output prediction box of the network, and $x_i$ represents the real grasp box.

# 4 Experiments and analysis

This paper trained and tested the proposed model on the datasets Cornell and Jacquard, which are publicly available in the field of grasping, and compared our method with the latest analysis methods. In addition, this paper conducted ablation experiments on the improvements proposed in this article and verified the effectiveness of each improvement. For the retail warehousing environment, this paper conducted a crawl simulation experiment to verify the effectiveness of the proposed crawling method.

## 4.1 Evaluation indicators

This paper adopts the currently popular Jaccard index to measure our grasp detection accuracy.

(1) The difference between the rotation angle of the predicted grasping rectangle and the ground-truth grasping rectangles is within $30^{o}$;

(2) The Jaccard index between the predicted grasping rectangles and the ground-truth grasping rectangle exceeds 0.25.

The Jaccard index is defined as follows:.

$$J(G_P, G_T) = \frac{G_P \cap G_T}{G_P \cup G_T} \tag{5}$$

Where $G_p$ represents the area of the predicted grasping rectangle, $G_T$ represents the area of the real grasping rectangle, $G_P \cap G_T$ represents the intersection of the two grasping rectangles, and $G_P \cup G_T$ represents the union of the two grasping rectangles.

## 4.2 Data analysis

The Cornell Grasping dataset contains 885 RGB-D images with a resolution of 640×480 pixels containing 240 different real objects. Each image in the dataset has several correspondingly labeled positive grasping rectangles (positive samples) and negative grasping rectangles (negative samples). The positive grasping rectangles represent feasible grasping boxes, and the negative grasping rectangles Represent a marker box that cannot be grasped. In the experiment, only the positive samples of the labeled boxes are used as training data.

This paper chose to carry out network training on the Cluster Engine equipped with 40G video memory, and the Pytorch1.10 and CUDA11.3 environments have been set up on the platform, and the CPU is set to 10 cores to ensure the computing power of the system.

Before network training, the image size is first converted from 640×480 to 320×320. This paper used the Adam optimizer for optimization training, the initial learning rate was set to 1e-3, and the batch size was 8 for training. During training, this paper created an augmented dataset using random cropping, scaling, and rotation, expanding the original dataset by a factor of 5 and splitting the dataset into training and validation sets in a 9:1 ratio. According to the above evaluation indicators, this paper compares the grasping accuracy and speed of the proposed method on the Cornell dataset with other popular grasping methods, and the results are shown in Table 1.

It can be seen from Table 1 that the RS-ConvNet model proposed in this paper has an accuracy rate of 97.8% and a speed of 78FPS. Although the detection accuracy of Stefan[24] is higher than that of the method proposed in this paper, the fusion of the two models leads to higher requirements on the hardware system and greater training difficulty, while the method in this paper is simple to training, and the parameter amount is only 711,430, which is less than half of the GR-ConvNet proposed by Kumra [22] and much smaller than other complex structures containing millions of parameters. Which are less computationally expensive, faster, and more suitable for application development in lightweight scenarios.

**Table 1**: Comparison of grasp accuracy and speed in Cornell dataset.

| Author | Algorithm | Grasp Accuracy(%) | Speed(FPS) |
|---|---|---|---|
| Asif et al.[19] | GraspNet(2018) | 90.6 | 41.67 |
| Guo et al.[20] | ZF-Net(2017) | 93.2 | - |
| Zhang et al.[21] | Resnet-101(2019) | 93.6 | 25.2 |
| Kumra et al.[22] | GR-ConvNet(2020) | 96.6 | 52 |
| Chu et al.[23] | Resnet-50(2018) | 94.4 | 8.33 |
| Stefan et al.[24] | Faster-RCNN(2022) | 98.2 | 63 |
| Our | RS-ConvNet | 97.8 | 78 |

The Jacquard grasping dataset consists of 54,000 RGB-D images and annotations of successful grasp locations performed in a simulated environment, for a total of 1.1 million grasp examples. The amount of data in this dataset is enough to train our model, so it is not augmented. The image size is preprocessed during training, and the dataset is trained in a 9.5:0.5 ratio, and the other conditions are consistent with the Cornell dataset training except for no data enhancement. The analysis of the grasping results is shown in Table 2.

**Table 2**: Comparison of grasp accuracy in Jacquard datasets

| Author | Algorithm | Grasp Accuracy(%) |
|---|---|---|
| Kumra et al.[22] | GR-ConvNet(2020) | 92.1 |
| Stefan et al.[24] | Faster-RCNN(2022) | 92.95 |
| Our | RS-ConvNet | 91.5 |

From the analysis in Table 2, it can be seen that this algorithm has a small number of model parameters, and is insufficient in the ability to obtain more detailed information. However, the multi-scale and large receptive field modules designed can ensure that the capture accuracy rate is above 90%. Compared with other network models with more complex structures, its accuracy can meet the daily grasping tasks. Fig. 7 shows some prediction results of RS-ConvNet in the Jacquard dataset.

Equations in LaTeX can either be inline or on-a-line by itself ("display equations"). For inline equations use the `$...$` commands. E.g.: The equation $H\psi = E\psi$ is written via the command `$H \psi = E \psi$`.

For display equations (with auto generated equation numbers) one can use the equation or align environments:

## 4.3 Ablation experiment

To prove the effectiveness of each improvement proposed in this paper, this paper uses GG-CNN as the base network model and uses ablation experiments

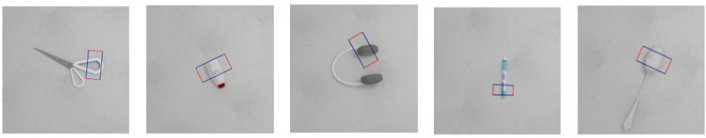

**Fig. 7**: RS-ConvNet prediction results

to verify the effectiveness of all improvements. The training set in the experiment adopts the 885-sheet version of the Cornell grasping dataset, of which 796 (90%) are used as the training set and 89 (10%) are used as the validation set. The training epoch is uniformly set to 500, the Adam optimizer is used to improve the training speed during the training process, and the initial learning rate is set to 1e-3. The Cluster Engine equipped with pytorch1.10 and CUDA11.3 environment is used for training. To better fit the samples, and considering that the total number of samples in the data set is not large, the batch size is set to 8 during training. The number of CPUs on the supercomputing platform is uniformly set to 20, and CUDA11.3 is used for accelerated training. After the training is completed, download the trained weight file, and perform verification and comparison on the experimental platform of Ubuntu 20.04. The experimental platform is equipped with AMD Ryzen 7 5800H CPU with a main frequency of 3.2GHz, and the GPU adopts NVIDIA Geforce RTX 3060-6G, which supports CUDA11. The PyTorch and CUDA versions are the same as during training. During the verification, the five weights with the highest IOU in the training process are selected for testing, and the average of the five sets of results is taken as the final result. The verification results are shown in Table 3.

**Table 3**: Experimental verification results of ablation based on GG-CNN network

| Base | Focus | DilatedConv | R-Resblock | RFB-SE | Accuracy(%) | Speed(ms) |
|---|---|---|---|---|---|---|
| ✓ | | | | | 79.5 | 19 |
| ✓ | ✓ | | | | 80.9 | 20 |
| ✓ | ✓ | ✓ | | | 82.9 | 20 |
| ✓ | ✓ | ✓ | ✓ | | 85.8 | 22 |
| ✓ | ✓ | ✓ | ✓ | ✓ | 88.3 | 23 |

As shown in Table 3, first of all, the basic model is the unimproved GG-CNN network model, which does not use cross-layer local connections, dilated convolution blocks, ECA attention, and Focus module. Although the detection speed of GG-CNN is fast, the detection effect is poor. After adopting Focus, the input RGB-D image is sliced, which retains more input features and improves the accuracy. The dilated convolution block is used to improve the receptive field, better learn the RGB-D image features, and achieve a good grasping

effect. On the other hand, after adding the multi-scale fusion R-Resblock, the feature information of multiple scales can be better integrated, and the use of ECA attention can make the network pay more attention to the grasped area for learning, which significantly improves the detection effect. Finally, a lightweight RFB-SE module is added, so that the model can obtain a larger receptive field and improve the accuracy of grasp detection.The training curve of the model is compared as shown in Fig. 8.

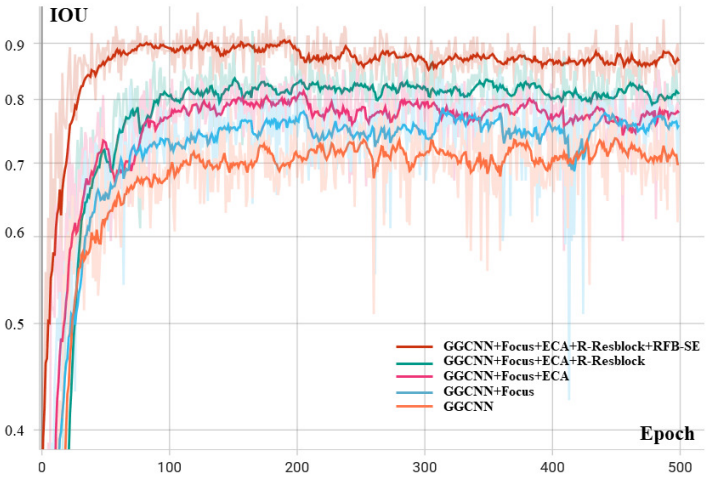


**Fig. 8**: Comparison of training IOUs in ablation experiments

When comparing the detection effects, the basic network and the two groups of models with the most improvements were selected for prediction experiments. As shown in Fig. 9, the first row is the prediction result of the GG-CNN model, and the second row adds all the improved model prediction effects. It can be seen that the improved model can make better use of image features, learn to grasp the angle information of the predicted frame and provide the robot with a better grasping position. It can be closer to the grasping center for rod-shaped objects, and the grasping success rate is also relatively higher.

Since the Cornell dataset is manually labeled, it cannot guarantee that all positive samples meet the grasping conditions. At the same time, considering that the mean square error (MSE) loss function is sensitive to negative samples, it will give negative samples a higher weight at the expense of other samples. The prediction results of the verification sample will reduce the overall model performance. Change the regression loss function to the Smooth-L1 loss function, which is more effective in the object detection field. It corrects the shortcomings of the L1 loss with inflection points and non-smoothness. At the same time, it does not sacrifice the prediction results of positive samples, which can ensure the performance of the overall model. The loss functions of

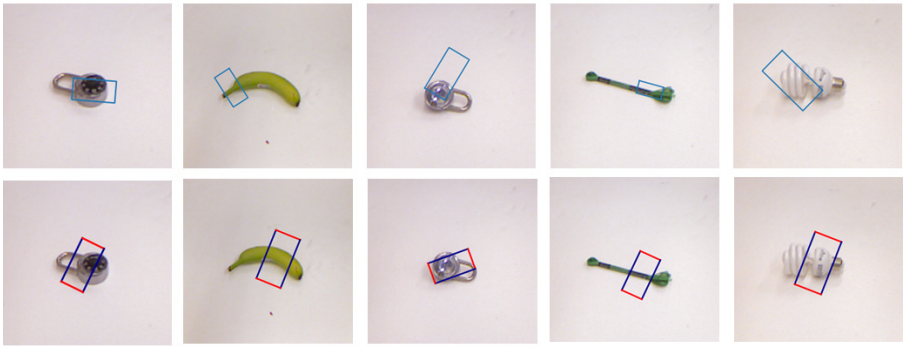

**Fig. 9**: Comparison of the prediction effect of the GG-CNN model and the model after adding all the tricks

the improved training and test sets drop faster and more smoothly, as shown in Fig. 10.

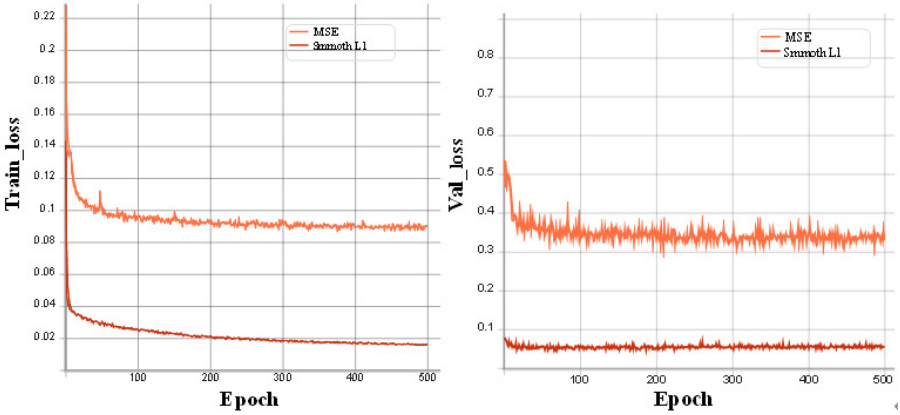


**Fig. 10**: Comparison of the prediction effect of the GG-CNN model and the model after adding all the tricks

It can be seen from Fig. 10 that the curve of the Smooth-L1 loss function is smoother and has less impact on the negative samples during training, indicating that the Smooth-L1 loss function can better fit the positive samples in the captured data set. It is more suitable for application in grasping detection.

## 4.4 Grasp Simulation Experiment

### 4.4.1 Experiment Platform

In the grasping experiment, this paper chooses to build a robotic object grasping system in the educational version of the V-rep simulation environment to

simulate grasping tasks. The robotic object grasping system consists of UR3, Kinect depth camera, and RG2 gripper. The robot is used to grasp objects to the target position within the right range. Kinect depth cameras are used to provide RGB and depth images of complex scenes and transmit them to DisplayPort. Experiments were carried out using a top-grasp strategy, with the camera mounted above the platform. The simulation platform is equipped with a CPU of AMD Ryzen 7 (5800H) with the main frequency of 3.2GHz, a graphics card of NVIDIA Geforce RTX 3060-6G, and an operating system of Windows 11. The simulation experiment is carried out by combining V-rep and Pycharm programming environments. The built grasping simulation platform is shown in Fig. 11.

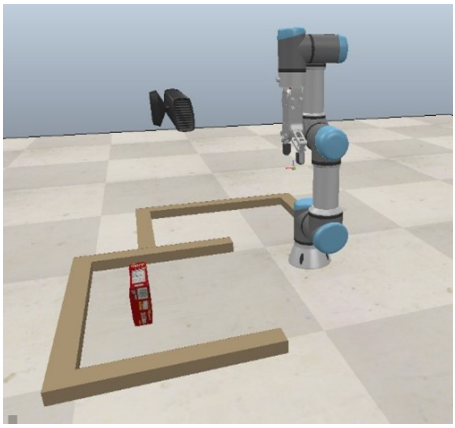

**Fig. 11**: Grasp simulation platform

### 4.4.2 Grasp Experiment

After obtaining the position information of the grasp target, coordinate transformation needs to be performed to convert it into the world coordinate system. After obtaining the target data to be grasped through the Kinect depth camera, use the deep learning model to generate the grasping position information, use the obtained pixel position information to perform the coordinate transformation, convert it to the world coordinate system, and obtain the position of the target to be grasped in the world coordinate system.Finally, the position information is input to the robotic arm for inverse kinematics solution, different joint values are obtained, and the grasping task is completed.

Fig. 12 shows the recognition process of a total of 6 items in a grasping experiment. The first row, the second row, and the third row in the Fig.12 is the grasping quality heatmap, grasping angle heatmap, and grasping width heatmap of the item to be grasped. After the robot arm recognizes the target to be grasped, it will preferentially select the target with the highest grasping

quality Q value for grasping. After the first grasping task is completed, the robotic arm will return to the original position and wait for the second grasping task. Finally, until there are no more grasping targets in the camera, the grasping task ends. Fig. 13 shows the process of the robotic arm grasping the can.

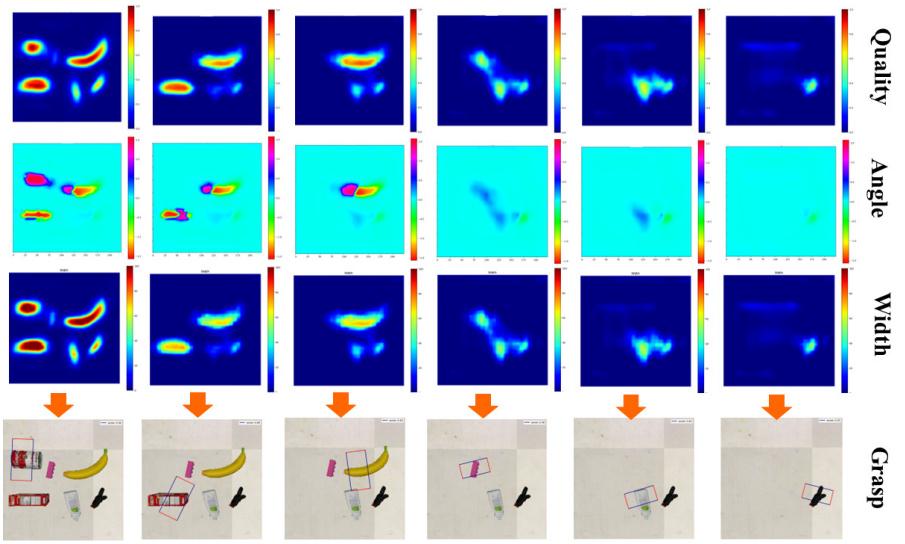


**Fig. 12**: The completion process of a grasping experiment

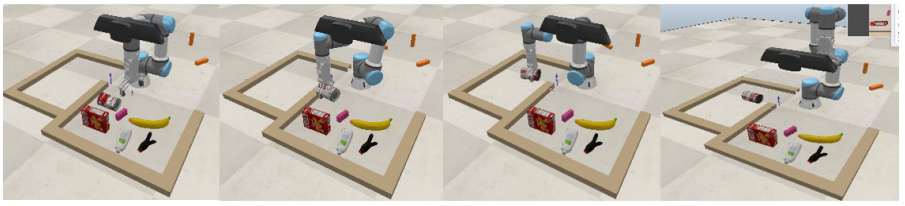

**Fig. 13**: Simulation of grasping cans

This paper carried out grasping simulation experiments on 6 kinds of objects, including bananas, cans, boxes, clips, detergents, and Lego blocks [25]. The objects were rearranged every 5 experiments during runtime, and each target was only grasped once. Take the opportunity, if it is not successfully grasped or dropped in the middle, it will be judged as a grasping failure. A total of 80 groups of grasping experiments were conducted, and the grasping success rate of each item is shown in Table 4.

**Table 4**: The number of successful grasps and the rate of grasping success for different items

| Class | Number of successful grasps | Grasp Accuracy(%) |
|---|---|---|
| Banana | 76/80 | 95.0 |
| Can | 70/80 | 87.5 |
| Box | 75/80 | 93.8 |
| Clip | 73/80 | 91.3 |
| Detergent | 75/80 | 93.8 |
| Lego block | 74/80 | 92.5 |
| Average | 74/80 | 92.3 |

During the experiment, due to the smooth surface of the pop-top can, there will be a phenomenon of falling off during the gripping process, so the gripping difficulty is high, and the success rate of gripping is only 87.5%. For other items, the success rate of grasping can be kept above 90%. Although our model is only trained on the Cornell grasping dataset, it still maintains a good grasping success rate for items not present in the dataset. Overall, this paper achieves an average grasping success rate of 92.3%, which meets the accuracy requirements for grasping tasks in a retail warehouse environment.

# 5 Conclusion

Aiming at the problem of grasping applications in retail warehousing scenarios, this paper proposes a lightweight grasping pose estimation model RS-ConvNet. The model first uses the Focus module to perform downsampling without information loss and learns each feature map of the previous layer through the dilated convolution block. The multi-scale fusion R-Resblock structure is used to fuse the information of each scale, and a lightweight RFB-SE module is designed to enrich the feature information. Finally, after upsampling and restoring the image, the target's grasping quality score, grasping angle, and grasping width are regressed to obtain the optimal grasping pose of the item. The number of parameters of the model this paper designed is only 711,430, which has good conditions for lightweight application. The experimental results show that the method can effectively obtain the grasping pose of the item. The grasping accuracy rate on the Cornell dataset can reach 97.8%, and the grasping speed is 78FPS, while the grasping success rate on the Jacquard dataset can reach 91.5%. In the grasping simulation experiment, the comprehensive grasping success rate for retail commodities is 92.3%, which meets the requirements of grasping accuracy and grasping speed in the retail warehousing environment. This paper focuses on the accuracy and speed of grasping objects of various shapes, and does not consider grasping tasks in complex environments for the time being, and will continue to consider grasping pose estimation in complex environments in the future.

# 6 Acknowledgment

Thanks for the computing support of the State Key Laboratory of Public Big Data, Guizhou University.

# 7 Declarations

## Ethical Approval

Not applicable.

## Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Authors' contributions

Qingni Yuan:Conceptualization, Methodology, Investigation, Formal analysis, Writing Original Draft, Visualization. Chen Wang:Data curation, Conceptualization, Methodology, Writing Review and Editing. Jianyou Qi:Investigation, Formal analysis, Visualization. Xiaoying Du:Software, Investigation.

## Funding

This work is partially supported by the National Natural Science Foundation of China (Project No.52065010 and No.52165063), Department of Science and Technology of Guizhou Province (Project No. [2022] G140 and No. [2020]4Y140), Graduate Innovative Talents Program of Guizhou University (2021), Research on Industrial Robot Technology based on Patent Analysis(K19-0204-001).

## Availability of data and materials

The datasets used or analysed during the current study are available from the corresponding author on reasonable request.

# References

[1] Sahbani, A., El-Khoury, S., Bidaud, P.: An overview of 3d object grasp synthesis algorithms. Robotics and Autonomous Systems **60**(3), 326–336 (2012)

[2] Caldera, S., Rassau, A., Chai, D.: Review of deep learning methods in robotic grasp detection. Multimodal Technologies and Interaction **2**(3), 57 (2018)

[3] Bohg, J., Morales, A., Asfour, T., Kragic, D.: Data-driven grasp synthesis—a survey. IEEE Transactions on robotics **30**(2), 289–309 (2013)

[4] Van Vuuren, J.J., Tang, L., Al-Bahadly, I., Arif, K.M.: A 3-stage machine learning-based novel object grasping methodology. IEEE Access **8**, 74216–74236 (2020)

[5] Ribeiro, E.G., de Queiroz Mendes, R., Grassi Jr, V.: Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation. Robotics and Autonomous Systems **139**, 103757 (2021)

[6] Du, G., Wang, K., Lian, S., Zhao, K.: Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. Artificial Intelligence Review **54**(3), 1677–1734 (2021)

[7] Cheng, B., Wu, W., Tao, D., Mei, S., Mao, T., Cheng, J.: Random cropping ensemble neural network for image classification in a robotic arm grasping system. IEEE Transactions on Instrumentation and Measurement **69**(9), 6795–6806 (2020)

[8] Park, D., Seo, Y., Shin, D., Choi, J., Chun, S.Y.: A single multi-task deep neural network with post-processing for object detection with reasoning and robotic grasp detection, 7300–7306 (2020). IEEE

[9] Zhu, H., Li, Y., Bai, F., Chen, W., Li, X., Ma, J., Teo, C.S., Tao, P.Y., Lin, W.: Grasping detection network with uncertainty estimation for confidence-driven semi-supervised domain adaptation, 9608–9613 (2020). IEEE

[10] Shang, W., Song, F., Zhao, Z., Gao, H., Cong, S., Li, Z.: Deep learning method for grasping novel objects using dexterous hands. IEEE Transactions on Cybernetics (2020)

[11] Zhang, Q., Gao, G.: Prioritizing robotic grasping of stacked fruit clusters based on stalk location in rgb-d images. Computers and electronics in agriculture **172**, 105359 (2020)

[12] Liu, H., Deng, Y., Guo, D., Fang, B., Sun, F., Yang, W.: An interactive perception method for warehouse automation in smart cities. IEEE Transactions on Industrial Informatics **17**(2), 830–838 (2020)

[13] Chiu, M.-C., Tsai, H.-Y., Chiu, J.-E.: A novel directional object detection method for piled objects using a hybrid region-based convolutional neural network. Advanced Engineering Informatics **51**, 101448 (2022)

[14] Yu, Y., Cao, Z., Liu, Z., Geng, W., Yu, J., Zhang, W.: A two-stream cnn

with simultaneous detection and segmentation for robotic grasping. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020)

[15] Chen, L., Huang, P., Li, Y., Meng, Z.: Edge-dependent efficient grasp rectangle search in robotic grasp detection. IEEE/ASME Transactions on Mechatronics **26**(6), 2922–2931 (2020)

[16] Xu, R., Chu, F.-J., Vela, P.A.: Gknet: grasp keypoint network for grasp candidates detection. The International Journal of Robotics Research, 02783649211069569 (2022)

[17] Ruan, J., Liu, H., Xue, A., Wang, X., Liang, B.: Grasp quality evaluation network for surface-to-surface contacts in point clouds, 1467–1472 (2020). IEEE

[18] Morrison, D., Corke, P., Leitner, J.: Learning robust, real-time, reactive robotic grasping. The International journal of robotics research **39**(2-3), 183–201 (2020)

[19] Asif, U., Tang, J., Harrer, S.: Graspnet: An efficient convolutional neural network for real-time grasp detection for low-powered devices. **7**, 4875–4882 (2018)

[20] Guo, D., Sun, F., Liu, H., Kong, T., Fang, B., Xi, N.: A hybrid deep architecture for robotic grasp detection, 1609–1614 (2017). IEEE

[21] Zhang, H., Lan, X., Bai, S., Zhou, X., Tian, Z., Zheng, N.: Roi-based robotic grasp detection for object overlapping scenes, 4768–4775 (2019). IEEE

[22] Kumra, S., Joshi, S., Sahin, F.: Antipodal robotic grasping using generative residual convolutional neural network, 9626–9633 (2020). IEEE

[23] Chu, F.-J., Xu, R., Vela, P.A.: Real-world multiobject, multigrasp detection. IEEE Robotics and Automation Letters **3**(4), 3355–3362 (2018)

[24] Ainetter, S., Fraundorfer, F.: End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb, 13452–13458 (2021). IEEE

[25] Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199 (2017)