Robotica (2025), **43**, pp. 4191–4211
doi:10.1017/S0263574725102750

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Research on robotic grasp detection using improved generative convolution neural network with Gaussian representation

Zhanglai Chen, Xu Zhang and Dawei Tu

School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China
Corresponding author: Dawei Tu; Email: tdw@shu.edu.cn

Received: 1 April 2025; **Revised:** 5 September 2025; **Accepted:** 27 September 2025;
First published online: 14 November 2025

Keywords: robot grasping; grasp detection; generative convolutional neural network; two-dimensional gaussian kernel

Abstract

Grasp detection is a significant research direction in the field of robotics. Traditional analysis methods typically require prior knowledge of the object parameters, limiting grasp detection to structured environments and resulting in suboptimal performance. In recent years, the generative convolutional neural network (GCNN) has gained increasing attention, but they suffer from issues such as insufficient feature extraction capabilities and redundant noise. Therefore, we proposed an improved method for the GCNN, aimed at enabling fast and accurate grasp detection. First, a two-dimensional (2D) Gaussian kernel was introduced to re-encode grasp quality to address the issue of false positives in grasp rectangular metrics, emphasizing high-quality grasp poses near the central point. Additionally, to address the insufficient feature extraction capabilities of the shallow network, a receptive field module was added at the neck to enhance the network's ability to extract distinctive features. Furthermore, the rich feature information in the decoding phase often contains redundant noise. To address this, we introduced a global-local feature fusion module to suppress noise and enhance features, enabling the model to focus more on target information. Finally, relevant evaluation experiments were conducted on public grasping datasets, including Cornell, Jacquard, and GraspNet-1 Billion, as well as in real-world robotic grasping scenarios. All results showed that the proposed method performs excellently in both prediction accuracy and inference speed and is practically feasible for robotic grasping.

1. Introduction

Robots are playing an increasingly important role in fields such as industrial production and social services. Among them, object grasping is one of the most common tasks for robots, but it remains challenging. This is because grasping an object with a robotic arm requires both the pose information of the object to adjust the hand's posture through inverse kinematics, as well as the shape information of the object to determine the grasp location of the end-effector on the object [1–4]. However, general object detection and pose estimation methods do not account for the specific shape features of the object when explaining and predicting grasp locations, which may lead to grasp failures [5, 6]. Early targeted grasp detection research selected grasp locations by searching for key points or analyzing geometric contour features [7, 8]. Although these methods improved grasp accuracy to some extent, they still suffer from issues such as low accuracy, inefficiency, and poor generalization. In recent years, the development of deep learning has provided new approaches for enhancing grasp detection capabilities [9–11]. Jiang et al. [12] proposed a rectangular representation of the grasp location, providing a unified definition of the ideal grasp position. LenZ et al. [13] were the first to introduce the grasp rectangle representation into neural networks for learning and prediction, improving the generalization of grasp detection. However, the complex and redundant network structure led to inefficient predictions. Redmon et al. [14] leveraged the powerful feature extraction capabilities of AlexNet to transform the prediction of grasp rectangles

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

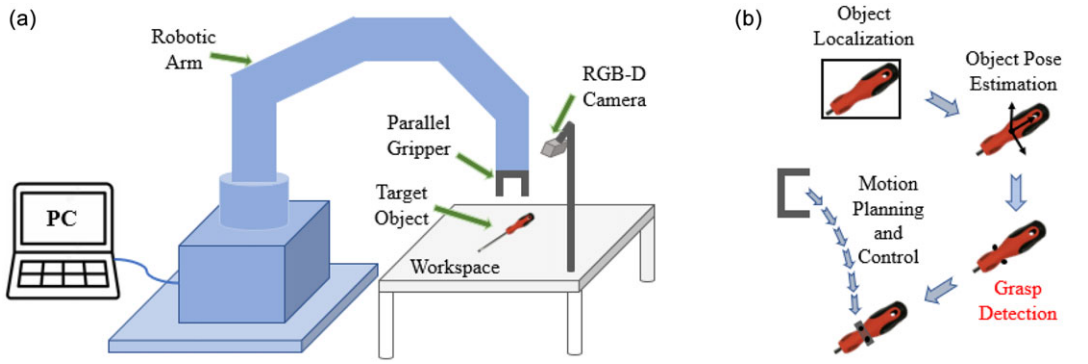


Figure 1. *Vision-based robot grasping system. (a) Hardware components of the system. (b) General grasping process.*

into a regression problem. However, this network can only predict a single grasp region, often resulting in an averaging effect in the grasp outcomes. Chu et al. [15] and Zhou et al. [16] independently proposed new models to address the single-region and single-angle issues identified in Ref. [14]. Morrison et al. [17], building upon Ref. [13], proposed a novel grasp detection method that directly generates pixel-level representations of grasp parameters through a generative convolutional neural network (GCNN), significantly enhancing both grasp accuracy and prediction efficiency. Based on Ref. [17], Kumra et al. [18, 19] introduced a residual module that can generate robust grasp predictions from n-channel inputs at real-time speeds. However, this method still has limitations when handling complex object shapes and scenes, and requires further improvements to enhance its accuracy and robustness.

The grasp rectangle representation proposed in Ref. [19] uses a binary method to learn grasp labels, assuming that any grasp with a predicted center falling within the correct box is considered correct. This assumption is flawed because the proximity of the center to the edge, combined with certain angular variations, may cause the gripper to collide with the object, leading to grasp failure. We referred to such cases, where the grasp theoretically succeeds but often fails in practice, as 'false positives'. To mitigate grasp failures caused by the 'false positives' issue and improve the model's ability to understand high-quality grasps [20, 21], we introduced a 2D Gaussian kernel to re-encode the grasp quality, emphasizing that the closer the grasp prediction is to the optimal grasp center, the higher the grasp quality score. Building upon this, to further improve the grasp prediction performance of the model, we proposed an improved structure based on the GCNN network, which consists mainly of five modules: input, encoding, bottleneck, decoding, and output prediction. To address the limitations of the original shallow network in extracting high-discriminative and robust features, we introduced a receptive field block (RFB) [22] at the bottleneck, which simulates the receptive field mechanism of the human visual system. This module enables the network to more precisely capture and extract high-discriminative and robust features, providing more effective feature support for subsequent grasp detection tasks. Additionally, the rich feature information obtained from feature extraction and reconstruction of the input image often contains redundant noise, which may affect subsequent grasp detection. We employed a global-local feature fusion module (GLFFM) [23] to suppress redundant features and highlight important object features. Finally, the proposed method was thoroughly evaluated on the widely used Cornell, Jacquard, and GraspNet-1 Billion datasets, followed by real-world robotic grasping experiments. The experimental results demonstrate the effectiveness and advancement of the proposed method.

2. Robot grasping system and problem modeling

2.1. Robot grasping system

As shown in Figure 1(a), a vision-based robotic grasping system typically consists of a robotic arm with a gripper, visual sensors, a work platform, and a host computer. Figure 1(b) illustrates the general grasping process. To complete the grasping task, the visual sensor (RGB-D camera) captures the scene images

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

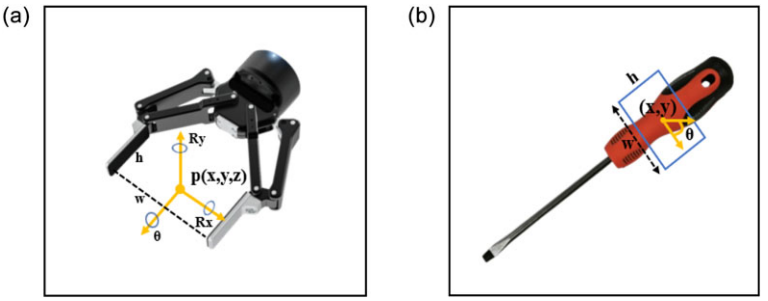


Figure 2. *Grasping pose representation. (a) Two-finger parallel gripper. (b) Schematic image of grasp representation.*

from the work platform and sends them to the host computer for analysis and prediction. The prediction results generate a control logic, which is then transmitted to the robot arm's joints and end-effector through the trajectory planning controller to control the arm in completing the grasping task.

2.2. Grasping problem modeling

Based on the aforementioned robot grasping system, defining an appropriate grasp representation model is crucial. To avoid grasp failures caused by false positives and enhance the model's ability to recognize high-quality grasps, we introduced an improved version of the grasp representation model proposed by Morrison et al. [17], incorporating a 2D Gaussian distribution. This approach emphasizes the grasp pose with the highest confidence to improve grasping accuracy.

2.2.1. Basic grasp representation

Equation (1) represents an earlier proposed five-dimensional grasp representation model [17], where (x, y) denotes the center of the gripper, θ represents the grasping angle, and w and h correspond to the width and height of the appropriate grasping rectangle. The five-dimensional grasp representation model, inspired by traditional object detection bounding boxes, effectively and simplistically represents the parallel gripper grasping approach shown in Figure 2(a) on the grasp example object in Figure 2(b). The grasp diagram is shown in Figure 2(b) and has been successfully used for grasp detection. However, this model is not entirely applicable.

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

To improve the accuracy and real-time performance of robotic grasp detection, a new simplified grasp representation model was reintroduced based on Equation (1) [17]:

$$g = \{p, \varphi, w, q\} \tag{2}$$

where p represents the center point (x, y, z) in Cartesian coordinates, φ and w denote the grasping angle and width, while q is a newly introduced quality coefficient used to assess the quality of the grasp and provides a measure for grasp correctness.

Furthermore, Equation (2) can be reformulated in two-dimensional space as Equation (3) to interpret the grasp representation model from the camera plane.

$$\hat{g} = \left\{\hat{p}, \hat{\varphi}, \hat{w}, \hat{q}\right\} \tag{3}$$

where $\hat{p} = (u, v)$ represents the center point in image coordinates, $\hat{\varphi}$ is the grasping angle in the camera coordinate system, $\hat{w}$ and $\hat{q}$ represent the grasp width and quality, respectively. Vision-based grasping predicts the grasp pose $\hat{g}$ of the target object from the camera image.

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

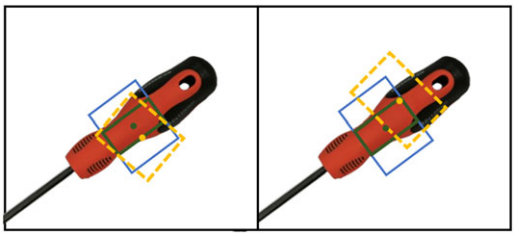


Figure 3. *False positive detection in binary cognition method.*

To remap the grasp pose $\hat{g}$ from the image coordinate system back to the robot coordinate system, calibration of the robotic grasping system is required. Using the calibration results, the grasp pose $\hat{g}$ is transformed into the world coordinate system, guiding the robot to effectively complete the grasping task. The transformation formula is given in Equation (4).

$$\begin{aligned} g &= T_{RC}\left(T_{CI}\left(\hat{g}\right)\right), \\ T_{CI} &= \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix}, T_{RC} = [R, T]. \end{aligned} \tag{4}$$

where T_{CI} is the transformation matrix that converts from image coordinates to camera coordinates and T_{RC}, which consists of the rotation matrix $R \in R^{3\times3}$and the translation matrix $T \in R^{3\times1}$, is the transformation matrix that converts from camera coordinates to world coordinates.

Since the robotic arm's grasp pose prediction for the target object may yield different results, we represented the set of all grasp poses as shown in Equation (5). Φ, W, and Q are computed using Equation (3) for each pixel in the image, resulting in three predicted images representing the grasp angle, grasp width, and grasp quality score.

$$G = (\Phi, W, Q) \in \mathbb{R}^{3\times h\times w} \tag{5}$$

By searching for the pixel coordinates of the highest grasp quality score $\hat{g} = max_Q G$, the grasp center point with the highest confidence can be obtained, along with the corresponding grasp angle and width.

2.2.2. Introducing a 2D Gaussian distribution

Since discrete rectangular boxes cannot cover all possible grasp positions on the target object, Morrison used a binary method for grasp labels to implement a heuristic model cognition [17]. As shown in Figure 3, if the predicted grasp center falls within the green box which is one-third the size of the correct grasping rectangle (blue box), it is considered valid, with the grasp quality marked as 1. If it does not fall within the green box, it is marked as 0, meaning invalid. This cognition is flawed. For example, the yellow dashed box is marked as a valid grasping frame in the binary method, but in real grasping scenarios, it is highly likely that the task cannot be completed due to a collision between the gripper and the object [20]. This situation is referred to as 'false positives' in our paper.

To address this flaw, we introduced a 2D Gaussian kernel (Equation (6)) to redefine the grasp quality label. Grasp quality points within the valid grasping frame are represented as a 2D Gaussian distribution, with the mean located at the center of the optimal grasp rectangle, emphasizing the importance of the optimal grasp. As the distance from the center increases, the grasp quality score gradually decreases, thus providing the model with a new heuristic understanding to avoid most 'false positives' and improve the accuracy of real grasping.

$$Q = K(x, y) = \exp\left(-\frac{(x-x_0)^2}{2{\sigma_x}^2} - \frac{(y-y_0)^2}{2{\sigma_y}^2}\right) \tag{6}$$

In the above equation, the grasp quality map predicted by the model is determined by the center point position (x_0, y_0), self-learning scale factor parameters $\sigma_x = \sigma_{x0} exp(\Delta s_x)$ and $\sigma_y \sigma_{y0} exp(\Delta s_y)$, with the peak

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

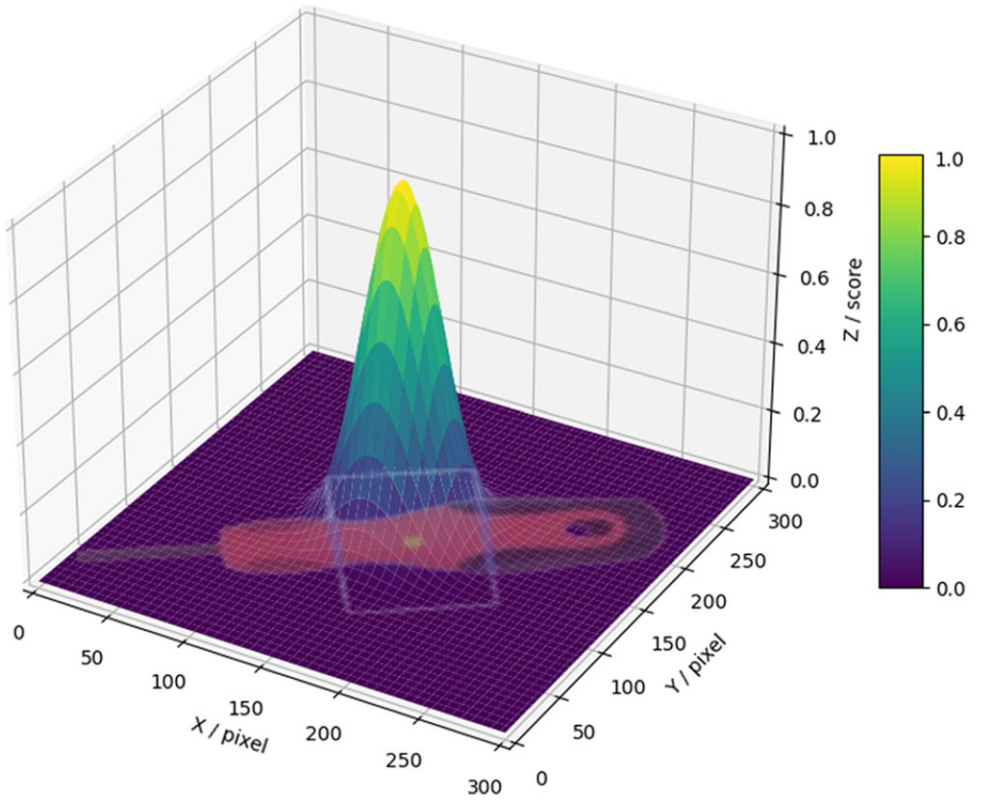


Figure 4. *Illustration of grasp quality distribution after the expansion of the 2D Gaussian function.*

of the Gaussian distribution located at the center of the optimal grasp rectangle. Here, σ_{x0} and σ_{y0} are hyperparameters determined empirically or through experimentation. In image processing and deep learning, σ is typically set to a power of 2 and is no larger than the image size. In this work, we trained and tested the grasp detection accuracy on the Cornell dataset using different values of σ from the set {4,8,16,32,64,128}, and selected the value of σ that achieves the highest accuracy as the initial values for σ_{x0} and σ_{y0} (set to $\sigma_{x0} = \sigma_{y0}$=32). Finally, we have introduced Δs_x and Δs_y as the adjustment terms learned by the network, which are used to optimize the initial scale factors. This is because a fixed-scale 2D Gaussian distribution limits the performance and adaptability of grasp detection, whereas introducing self-learned scale factors enables the Gaussian distribution of grasp quality to be more flexible and precise, adapting to complex and variable grasping environments, thereby significantly enhancing the performance of grasp detection. Finally, the aforementioned approach is integrated into the network's grasp quality branch for learning and prediction, thereby producing more accurate grasp quality maps. Figure 4 illustrates the grasp quality distribution after the expansion of the 2D Gaussian function.

3. Improved generative convolutional neural network

3.1. Basic network structure

Figure 5 illustrates the GCNN proposed by Kumra [18]. This network takes an n-channel image as input and uses an encoder-decoder architecture. Features are extracted through convolutional layers, residual layers, and deconvolutional layers. Based on these features, three images are generated to predict the grasp pose: grasp quality score, grasp angle, and grasp width. The efficient and simple network not only improves the accuracy of grasp detection but also ensures real-time prediction performance. However, such a network still has limitations: Firstly, the bottleneck module employs five residual layers, but an

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

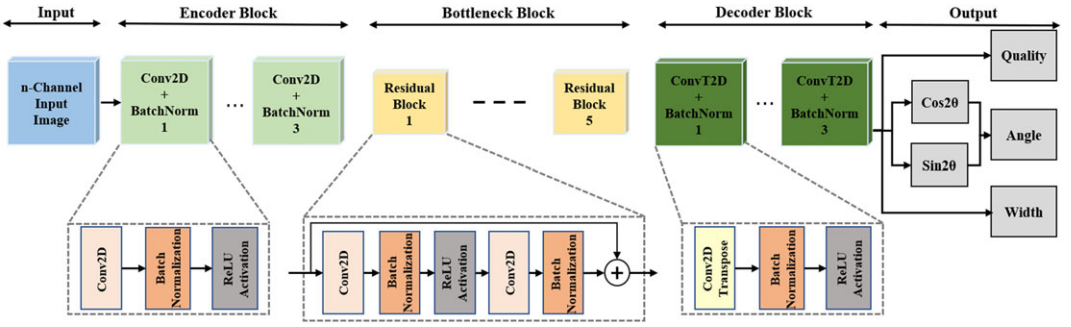


Figure 5. *Architecture of the generative residual convolutional neural network.*

excessive number of residual layers can lead to issues such as vanishing gradients and dimensional errors, causing saturation and a decline in accuracy [19]. Additionally, the simple bottleneck module fails to effectively focus on high-discriminative and robust features, which are crucial for improving the accuracy of network predictions. Secondly, the network uses deconvolution operations to reconstruct features during the decoding phase. However, due to the large amount of image feature information extracted during the encoder and bottleneck phases, which includes both valid features and redundant noise, the original network does not enhance the valid information or reduce the redundant noise, which negatively impacts the accuracy to some extent.

In summary, we proposed an improved GCNN to enhance the grab prediction performance of the original network. The structure is shown in Figure 6: n-channel input images pass through the encoder block to obtain feature maps with smaller dimensions, more channels, and richer semantic information. Subsequently, four residual layers were applied and a RFB [22] was introduced, enabling the network to extract more discriminative and robust features from the feature maps. The small-sized feature maps with rich information obtained so far are difficult to interpret into the desired information for grasp prediction. Therefore, a decoder block was used for feature reconstruction. During the reconstruction process, we introduced a GLFFM [23] to obtain more refined target information and reduce irrelevant noise, which is primarily used to effectively enhance and fuse the target features after feature reconstruction. Finally, four sub-channels with convolution layers and dropout layers (which help extract rare but useful feature information through regularization) were used to predict the pixel-level images of grasp quality scores, grasp angles and grasp widths, thus completing the grasp prediction. The calculation formula for the grasp angle is given by Equation (7).

$$g_{Angle} = \arctan\left(\frac{g_{\sin 2\theta}}{g_{\cos 2\theta}}\right)/2 \tag{7}$$

Here, $g_{sin2\theta}$ and $g_{cos2\theta}$ represent the network-predicted values of $sin2\theta$ and $cos2\theta$, respectively. Due to the $\pm\pi/2$ symmetry of antipodal grasps, this formulation avoids angular ambiguity. A unique grasp angle g_{Angle} is obtained through Equation (7).

3.2. Multi-scale receptive field block

The RFB (L8) introduced in Figure 6 is further illustrated in Figure 7. The design of the module is inspired by the receptive fields of the human visual system, particularly the characteristics of receptive field size and eccentricity variation across retinal locations. These characteristics enable the human visual system to focus on salient features in image tasks. The RFB module leverages these features to address the limitations in the feature extraction ability of shallow convolutional neural networks [22]. In robotic grasp detection tasks, multi-scale receptive fields enhance the deep feature extraction capabilities of neural networks, thereby yielding features with high discriminability and robustness in the feature maps.

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

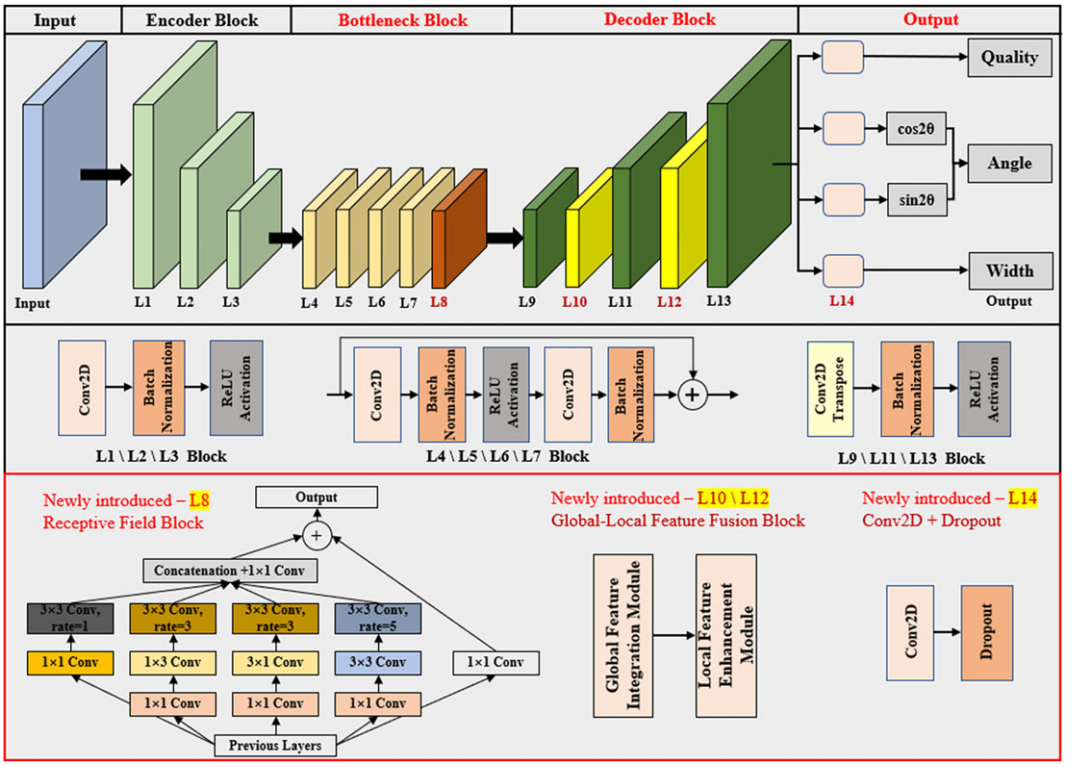


Figure 6. *The structure of the improved generative convolutional neural network, with improvements indicated in red.*

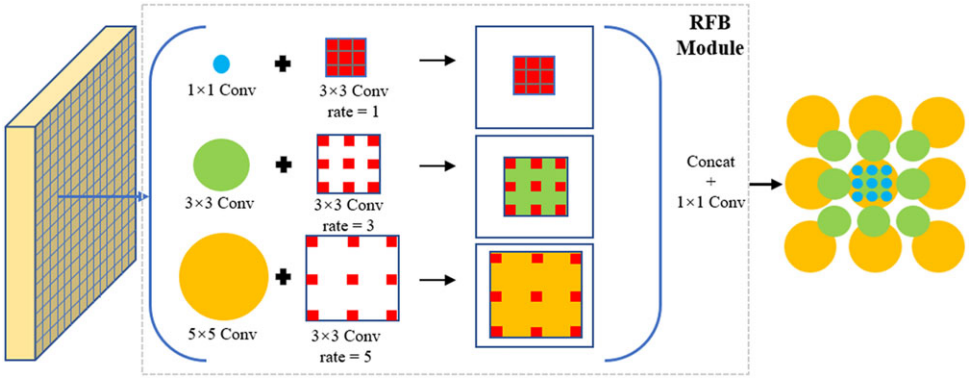


Figure 7. *Simulation process of the RFB module.*

Figure 7 illustrates the simulation process of the RFB, which is divided into three main components: (1) the use of multi-branch pooling to simulate receptive fields of various sizes, capturing features at different scales through convolution kernels and pooling layers of different sizes; (2) the use of dilated convolutions to simulate the variation of eccentricity by increasing the spacing between convolutional kernel elements, thus expanding the receptive field without increasing computational parameters; and (3) the use of feature recombination to mimic the receptive field of the human visual system, capturing rich image details.

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

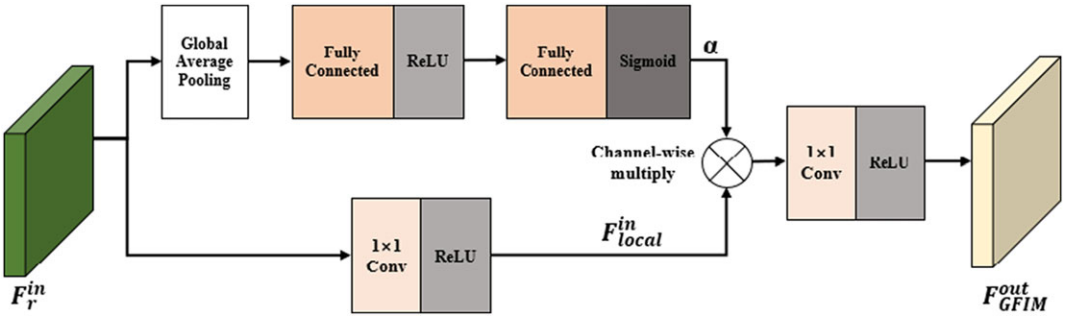


Figure 8. *The structure of the global feature integration module.*

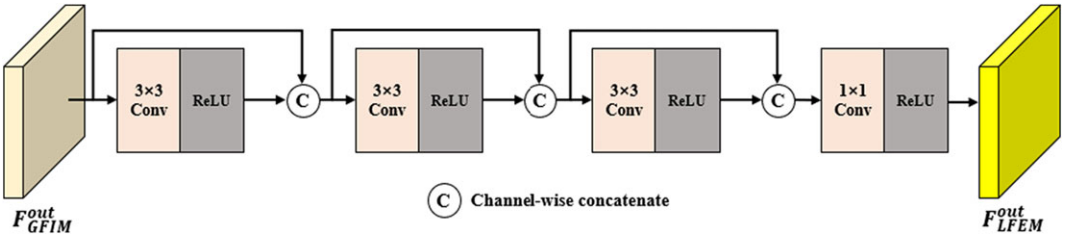


Figure 9. *Local feature enhancement module.*

3.3. Global-local feature fusion module

After passing through the encoder and bottleneck modules, the rich image feature information often contains redundant noise, which may affect the accuracy of grasp prediction. To minimize feature redundancy and ensure the model focuses on important target information rather than irrelevant noise, we proposed the integration of a GLFFM [23] in the decoder block. This module extracts useful information from the rich reconstructed feature data, eliminates redundant noise and fuses the remaining information into the output, thereby enhancing key feature information. The L10 and L12 modules introduced in Figure 6 will be further explained in Figures 8 and 9.

3.3.1. Global feature integration module (GFIM)

Figure 8 illustrates the GFIM module. The upper branch follows the squeeze-and-excitation module, performing global average pooling on the input feature map F_r, compressing it into a global descriptor for each channel, and obtaining the channel weights α through two fully connected layers. The lower branch uses a 1×1 convolutional layer followed by a ReLU activation to obtain the local features F_{local}. Finally, as defined in Equation (8), the result of multiplying the local features by the channel weights is passed through a 1×1 convolutional layer and a ReLU activation to yield the final output F_{GFIM}^{out}.

$$F_{GFIM}^{out} = CR\left(\alpha \times F_{local}^{in}\right) \tag{8}$$

where the $CR(\cdot)$ operation denotes performing a 1×1 convolution followed by a ReLU activation on the features inside the parentheses.

3.3.2. Local feature enhancement module (LFEM)

After the global feature aggregation operation, dense convolutions are applied to feature map F_{GFIM}^{out} to integrate features from different stages, aiming to obtain more refined features. The dense convolution consists of three identical parts: the result obtained by applying a 3×3 convolution and ReLU activation to the input is concatenated with the original input along the feature dimensions, and the concatenated result is then used as the input for the next part. Finally, a 1×1 convolution followed by a ReLU activation is applied to obtain the fused features F_{LFEM}^{out}, and the network architecture is shown in Figure 9.

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

Table I. *Key hardware and software parameters in the experimental environment.*

Name	Parameter
Central Processing Unit (CPU)	Intel Core I7-7700@3.60Hz
Graphic Processing Unit (GPU)	NVIDIA GeForce RTX 3060 12GB
Operating system	Ubuntu 20.04
Training framework	Pytorch 1.2.0+torchvision 0.4.0+Cuda 10.0
Programming software	Pycharm 2021.1.3
Programming language	Python 3.6

3.4. Definition of the loss function

The neural network model can be viewed as a method for approximating complex functions, learning the relationship between the input images $I = \{I_1, \ldots, I_n\}$ and the correct grasp labels $L = \{L_1, \ldots, L_n\}$, and inferring the corresponding grasp predictions $G = \{G_1, \ldots, G_n\}$ through $I \rightarrow G$. The model is then trained by minimizing the error between the grasp predictions G and the correct grasp labels L, thereby enabling accurate grasp prediction. Since the learning task is a regression problem, and to better handle gradient explosion, we used the smooth $L1$ loss as the loss function to optimize the model. The mathematical formula is given in Equation (9) [18].

$$L(G, I) = \sum_{i}^{N} \sum_{m \in (q, \cos 2\theta, \sin 2\theta, w)} l_1 (G_i - L_i) \tag{9}$$

the smooth loss l_1 is defined as,

$$l_1 (x) = \begin{cases} (\sigma x)^2 / 2 & if\ |x| < 1 \\ |x| - 0.5/\sigma^2, & otherwise \end{cases} \tag{10}$$

where N represents the number of grasp candidates, q and w represent grasp quality and width, respectively. The grasp angle is represented by $cos2\theta$ and $sin2\theta$, and σ is the hyperparameter in the smooth $L1$ loss function used to control the smoothing region. In this work, σ is set to 1.

4. Experimental results and analysis

To evaluate the performance of the proposed method, experiments were conducted on the widely used Cornell, Jacquard and GraspNet-1 Billion Grasp Dataset and real robot grasping. Section 4.1 describes the experimental setup, Section 4.2 introduces the above datasets and evaluation metrics, and Sections 4.3 to 4.5 discuss the experimental results, demonstrating the effectiveness and advancement of the method.

4.1. Experimental environment

In this experiment, the key hardware and software parameters involved and configured are shown in Table I. Other experimental settings, such as a batch size of 8, 50 training epochs, a learning rate of 10e-3, and the use of the Adam optimizer for backpropagation during training, are also provided.

4.2. Datasets and evaluation metrics

4.2.1. Introduction to the public grasp datasets

In this study, the Cornell [12], Jacquard [24], and GraspNet-1 Billion [25] datasets were primarily introduced. These grasping datasets are widely used for the validation and evaluation of robotic grasp

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

Table II. *A summary of the public datasets introduced.*

Dataset	Image format	Type	Number of objects	Number of images	Number of grasping labels
Cornell	RGB-D	Real	240	1035	8k
Jacquard	RGB-D	Sim	11k	54k	1.1M
GraspNet-1Billion	RGB-D	Real	88	97k	1.2B

detection methods, effectively quantifying the performance of the proposed grasp detection methods [1]. In general, a summary of the public datasets is presented in Table II.

The Cornell Grasp Dataset contains 1,035 RGB-D images with a resolution of 640 × 480, with a total of 5,110 positive grasps and 2,909 negative grasps. But the dataset size is insufficient to train the proposed network effectively, we employed data augmentation techniques such as random cropping, rotation, and scaling to create an expanded version of the Cornell dataset. Since only positive grasps are considered during training, data augmentation was applied solely to the positive grasp examples, resulting in a final total of 51.1k grasp labels for training the network [18].

The Jacquard dataset is a large-scale grasping dataset synthetically generated based on CAD models, and it is significantly larger in scale than the Cornell dataset. It consists of 1.1M grasp annotations generated from 11k different objects across 54k diverse scenes. For each scene, the dataset provides one rendered RGB image, one segmentation mask, two depth images, and a set of valid and feasible grasp labels. In addition, Jacquard includes the simulated grasp trials (SGTs) to facilitate performance comparison across different methods. This system executes grasping actions using a simulated robot and returns the final grasp success rate.

The GraspNet-1 Billion dataset is a large-scale grasping dataset containing a diverse set of everyday objects. It consists of 97,280 RGB-D images captured from 190 cluttered scenes, with over 1.2 billion annotated grasp labels in total. To utilize the original 1280 × 720 resolution rectangular images, we cropped each image into a 720 × 720 square centered at the average center of the ground-truth bounding boxes.

4.2.2. Grasp evaluation metrics

To ensure a fair comparison with other methods, the unified rectangular metric [18] was still used to evaluate our proposed method. Specifically, a grasp prediction is considered correct if it meets the following two conditions:

(1) Intersection over Union (IoU): The IoU between the ground truth grasp rectangle and the predicted grasp rectangle needs to exceed 25%, and the formula is shown in Equation (11).

$$IoU\left(g_p, g_t\right) = \frac{\left|g_p \cap g_t\right|}{\left|g_p \cup g_t\right|} \tag{11}$$

where g_p and g_t denote the area of the predicted grasp rectangle and the ground truth grasp rectangle, respectively.

(2) Angle offset: The difference between the grasp direction of the predicted grasp rectangle and the grasp direction of the ground truth rectangle is less than 30°.

4.3. Experiments on public datasets

4.3.1. Evaluation on the Cornell dataset

To effectively evaluate the grasp detection performance of the model on the Cornell dataset, we employed the 5-fold cross-validation setup used in Ref. [13]: The proposed network was trained and validated using two types of data split methods: image-wise (IW), where the training and test sets do not share the same images, and object-wise (OW), where the training and test sets do not share any images of the

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

Table III. *Performance evaluation of different methods on the Cornell dataset.*

Authors	Method	Input Modality	Detection Accuracy(%) IW	OW	Inference Time(ms)
LenZ [13]	SAE,struct.reg	RGB	73.9	75.6	1350
Kumra [26]	ResNet-50x2	RGB	89.2	88.9	103
Karaoguz [27]	GRPN	RGB	88.7	–	200
Chen [28]	Edge-based EG	RGB	93.5	92.1	146
Kumra [18]	GR-ConvNet	RGB-D	97.7	96.6	20
Yu [29]	SE-ResUNet	RGB-D	98.2	97.1	25
Zhai [30]	FANet-GPU	RGB	98.5	97.8	23
Wang [31]	TF-Grasp	RGB-D	97.9	96.7	42
Xi [32]	EGA-Net	RGB-D	97.8	98.9	24
Zhou [33]	DSC-GraspNet	RGB-D	98.3	97.7	14
Ours	Improved-GCNN	RGB-D	99.0	98.3	22

Table IV. *The model parameter comparison of different methods.*

Authors	Method	Parameters (Approx.)
Johns [34]	Grasp Function + CNN	60 million
Morrison [17]	GG-CNN2	0.066 million
Chu [15]	Deep Grasp	216 million
Kumra [18]	GR-ConvNet	1.9 million
Wang [31]	TF-Grasp	5.8 million
Guo [35]	MDETR	5.5 million
Ours	Improved-GCNN	2.0 million

same objects. The experimental results, as shown in Table III, demonstrated that the proposed method achieves detection accuracies of 99.0% and 98.3% for image-wise and object-wise, reaching state-of-the-art performance compared to other methods. Table IV presents a comparison of model parameters across different methods. As shown in Tables III and IV, our network has only about 2.0M parameters, making it more lightweight compared to other grasp detection methods with millions of parameters or complex network architectures. Its inference speed is fast, with an inference time of approximately 22 ms per image during evaluation.

To further validate the accuracy, robustness, and suppression effect of the model on false positives, we also evaluated the accuracy of the model at higher IoU thresholds, and the results are shown in Figure 10. Compared to other methods, our proposed method maintains a higher prediction accuracy even under higher threshold conditions, further demonstrating its excellent performance.

Figure 11 shows the grasp detection results for the example objects from the Cornell dataset. The result with the highest grasp quality score was selected as the final output. The first column displays the original input image, while the second, third, and fourth columns represent pixel prediction maps for grasp quality score, grasp angle, and width, respectively. The fifth column shows the optimal predicted grasp pose. As seen in Figure 11, our proposed method can perform accurate and reliable grasp detection on objects of varying shapes and poses.

In addition, our method was compared with other advanced methods for grasp detection on the scissor object in the Cornell dataset, and the qualitative comparison is shown in Figure 12. As illustrated in the figure, our method outperforms the others in the predicted grasp quality, angle, and width maps. This superiority is mainly reflected in the fact that the predicted information is more aligned with realistic grasping and contains richer features, thereby enabling more accurate grasp detection. It can be

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

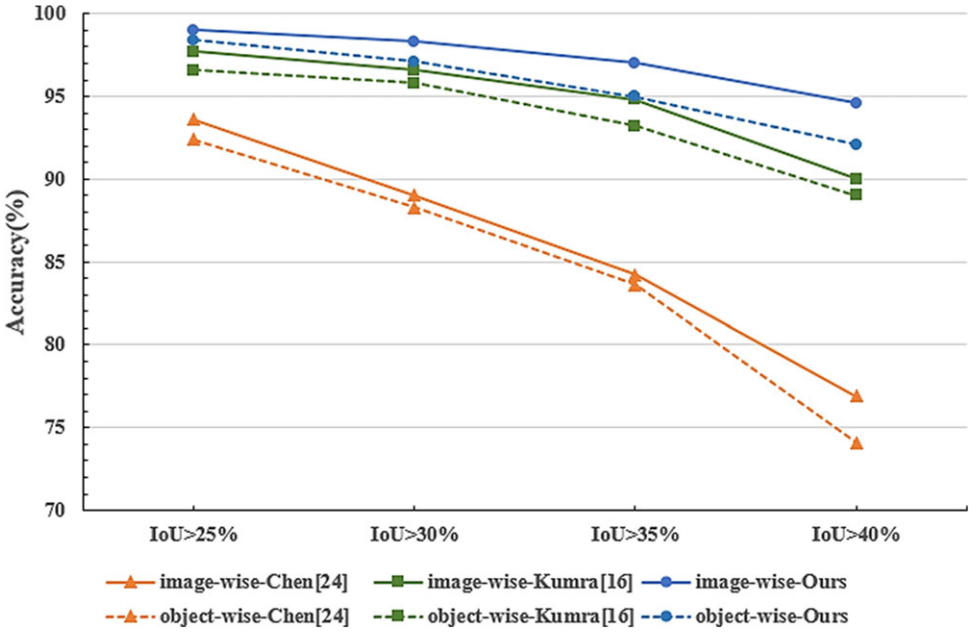


Figure 10. *Accuracy comparison on the Cornell dataset at different IoU thresholds.*

attributed to the introduction of a 2D Gaussian representation and a more effective network architecture for learning and predicting grasps. As shown in the last column, our method yields better predictions and a higher theoretical success rate for robotic grasping. This is because the first method attempts to grasp the less stable blade area, while the second method predicts an insufficient grasp width that may lead to collisions, both of which reduce the success rate.

Despite the many advantages of our proposed method demonstrated in the above experiments, it still encounters some failure cases during the prediction process. As shown in Figure 13, these failure cases occurred during the evaluation process. Although the real-world robot may occasionally succeed in grasping during some attempts, the overall theoretical success rate remains relatively low. Specifically, the white plastic bottle on the left has a color very similar to the background, which causes the network to struggle in accurately predicting the required grasp width and angle. On the right, the cluttered wire harness, with its uniform color and complex variations in texture, may introduce prediction errors. To effectively tackle these challenges, it is imperative to improve the standardization and precision of grasp label definitions while simultaneously advancing the network's capacity for robust feature extraction and understanding of targets within real-world environments.

4.3.2. Evaluation on the Jacquard and GraspNet-1 Billion dataset

For the Jacquard dataset, its large-scale image data effectively supports the training of deep neural networks. We split the dataset into training and testing sets using a 9:1 ratio for model training and validation. The evaluation results of our method compared with other approaches on the Jacquard dataset are summarized in Table V. In the experiments, we employed both the IoU metric and the SGTs system for grasp evaluation, achieving accuracies of 95.9% and 91.6%, respectively. Moreover, Figure 14(a) presents multiple grasp visualization results for several test objects from the Jacquard dataset, demonstrating that our method can effectively perform grasp detection on complex objects with diverse characteristics. These results demonstrate the superior performance of the proposed method and further validate its effectiveness.

For the GraspNet-1 Billion dataset, its large-scale training data effectively compensates for the lack of complexity in previous grasping datasets. However, the excessive volume of training samples may

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

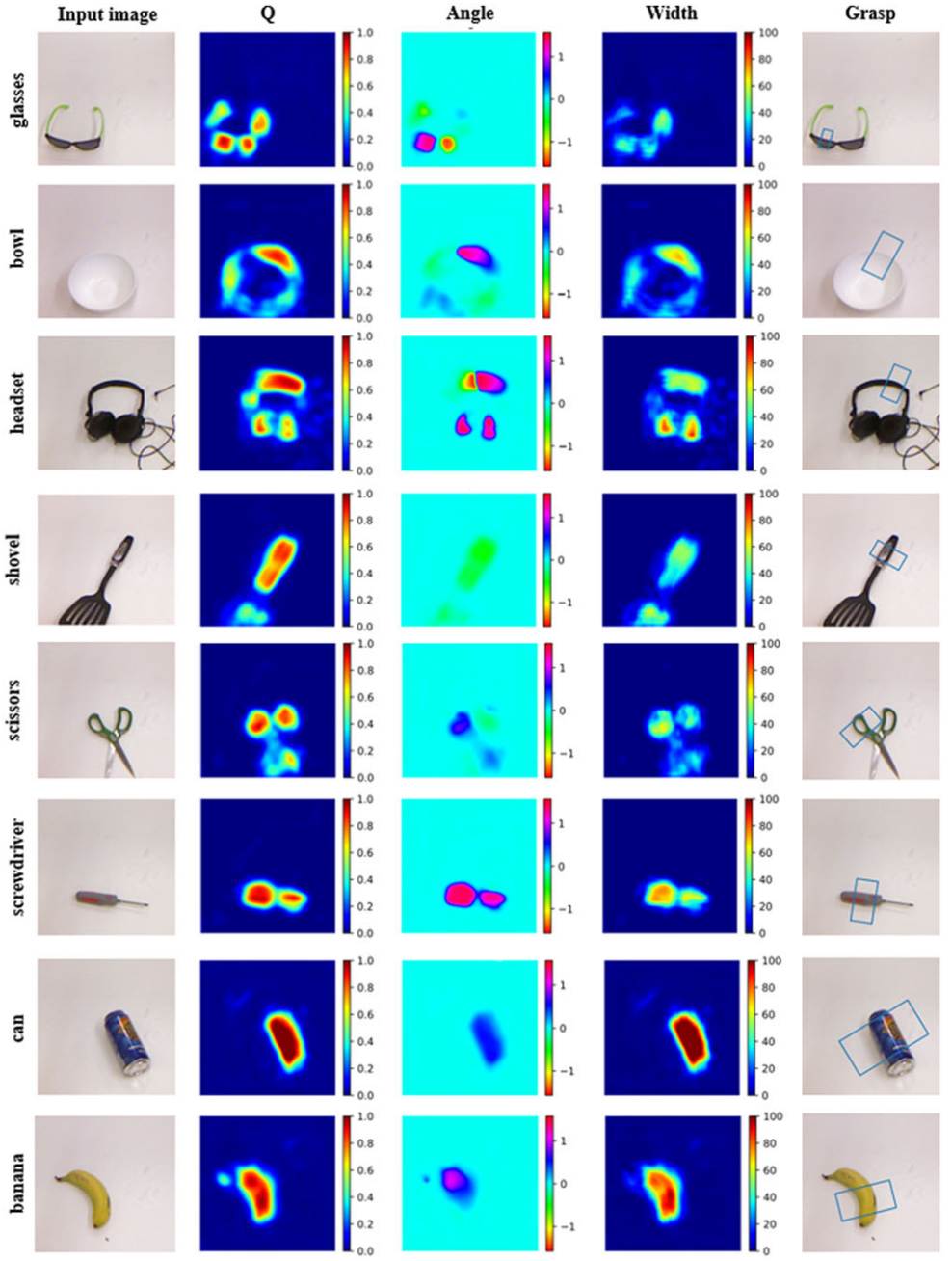


Figure 11. *Grasp detection results of the grasp network on example object.*

overload limited computational resources. To address this, we performed a preprocessing step by discarding grasp labels with a friction coefficient $\mu < 0.5$, thereby reducing the number of grasp annotations loaded per scene. Following the GraspNet-1 Billion evaluation protocol, we conducted training and testing on a total of 190 scenes: scenes 1–100 (after preprocessing) were used for training; scenes 101–130 served as test cases for seen objects; scenes 131–160 evaluated generalization to similar objects; and scenes 161–190 were used to test performance on unseen objects. The grasp prediction accuracy across the three scenes was compared against other methods, with the detailed results summarized in Table VI. Figure 14(b) shows the grasp detection results for several complex test scenarios from the

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

Table V. *Performance evaluation of different methods on the Jacquard dataset.*

Authors	Method	Accuracy(%)	
		IoU	SGTs
Morrison [17]	GG-CNN2	84.0	85.0
Zhou [16]	FCGN, ResNet-101	92.8	81.9
Kumra [18]	GR-ConvNet	94.6	89.5
Wang [31]	TF-Grasp-RGBD	94.6	–
Zhou [33]	DSC-GraspNet-D	94.7	90.3
Zhai [30]	FANet-CPU	95.3	–
Zhu [36]	CFRA	94.2	–
Ours	Improved-GCNN	95.9	91.6

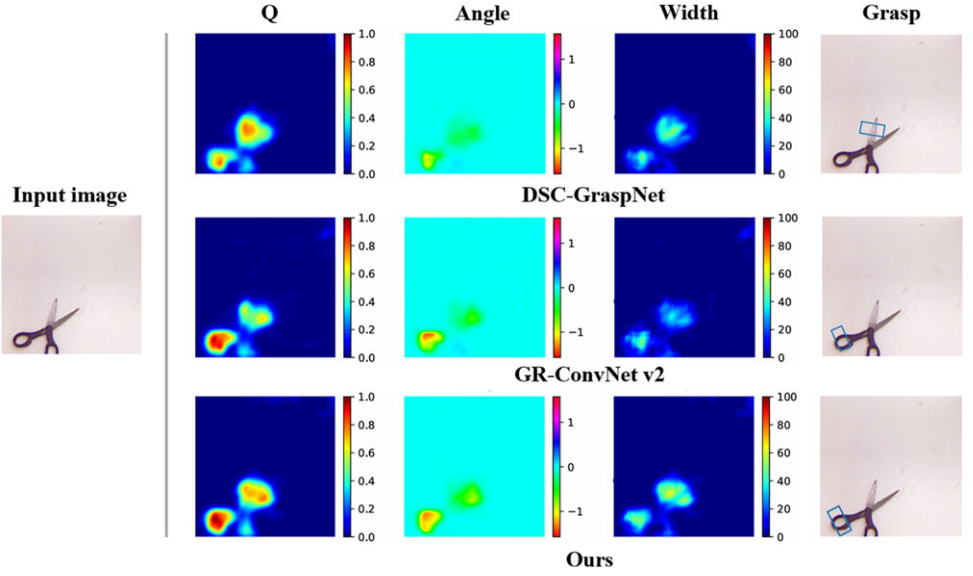


Figure 12. *Comparison of grasp detection results on the scissor object across different methods.*

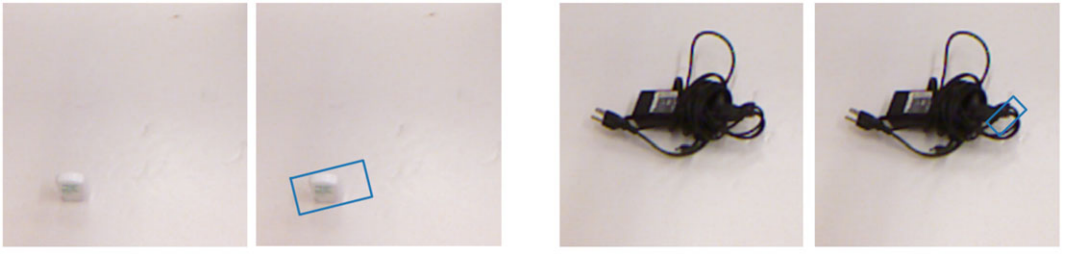


Figure 13. *Some failure cases in the evaluation on the Cornell dataset.*

GraspNet-1 Billion dataset. The results demonstrate that the proposed method achieves consistently strong performance in grasp detection, particularly in scenarios involving novel or previously unseen objects.

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

Table VI. *Performance evaluation of different methods on the GraspNet-1 Billion dataset.*

		Accuracy (%)		
Authors	**Method**	**Scenes 101–130**	**Scenes 131–160**	**Scenes 161–190**
Morrison [17]	GG-CNN2	83.0	79.4	76.3
Kumra [19]	GR-ConvNet	96.2	94.8	87.9
Yan [37]	SISG-Net	96.7	93.9	88.4
Ours	Improved-GCNN	98.0	95.8	90.6

Table VII. *The impact of different network configurations on grasp performance.*

+ Gaussian distribution		√	√	√
+ RBF module			√	√
+ GLFFM module				√
Average accuracy(%)	97.15	97.35	98.15	**98.65**

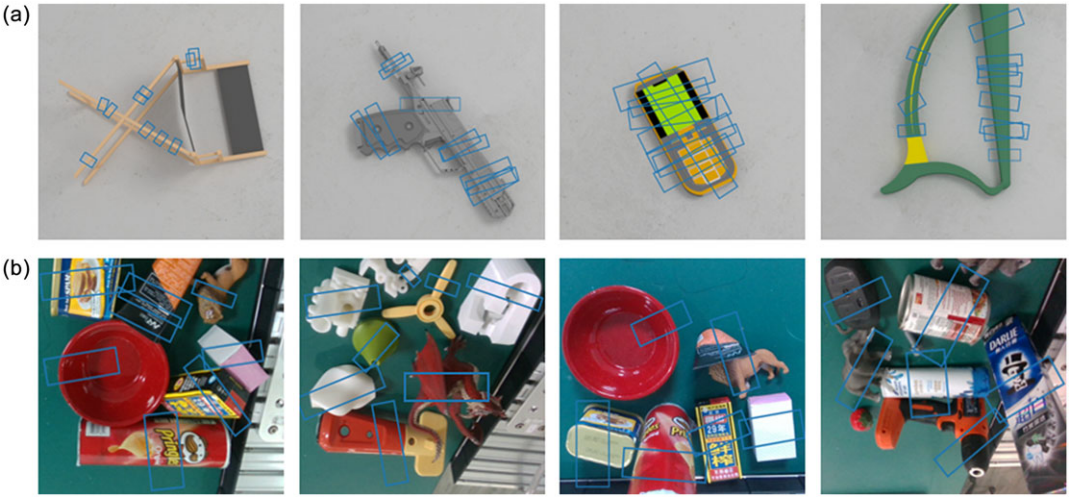


Figure 14. *Some visualization results from the evaluation on the Jacquard and GraspNet-1 Billion datasets. (a) Jacquard dataset. (b) Grasp-1 Billion dataset.*

4.4. Ablation studies

To further investigate the impact of different input data and improved modules on grasp detection accuracy, ablation studies were conducted. Models with different network configurations were trained and compared on the Cornell dataset. Figure 15 shows the impact of different input data on grasp detection accuracy. Among them, the RGB-D input data achieved the highest scores in both IW and OW validation methods. This demonstrated that combining RGB data, which contains rich color and texture information, with depth data for grasp prediction results in higher accuracy and robustness of the model.

Table VII shows the evaluation results of various improved modules on grasp detection accuracy. In this set of experiments, RGB-D data were used as input, and the average accuracy under IW and OW forms was calculated. The results indicated that each improvement contributed to enhanced grasp detection performance, with the combination of all improvements yielding the best grasp detection results. This further underscores the rationality and advancement of the proposed method.

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

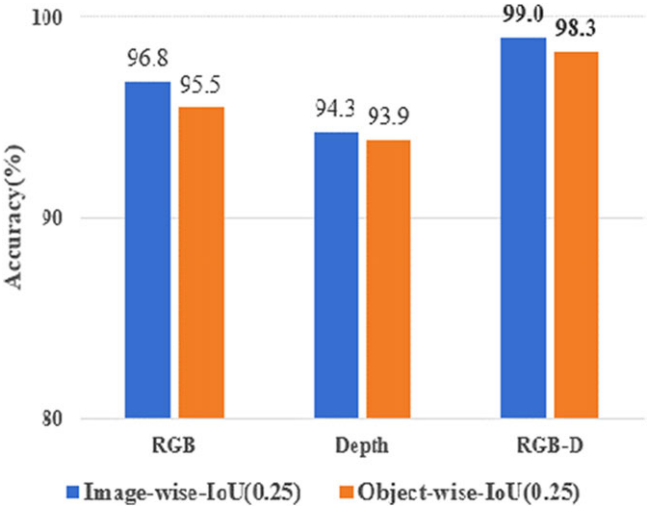


Figure 15. *Quantitative evaluation of the impact of different input data on grasp detection accuracy.*

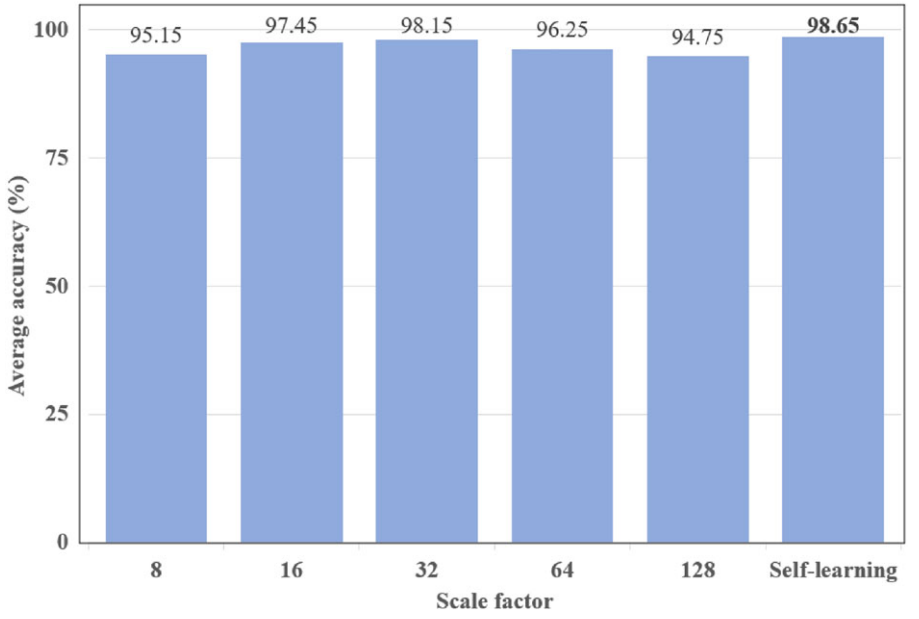


Figure 16. *Grasp detection accuracy on the Cornell dataset when using different scale factors of the Gaussian kernel.*

Finally, we conducted grasp detection accuracy experiments on the Cornell dataset with different scale factor settings in the Gaussian kernel. The set of fixed scale factor values was [8,16,32,64,128], and the initial values of the self-learning scale factors were set to the current optimal value (32). The average accuracy results under IW and OW settings are shown in Figure 16, where the self-learned scale factors achieved the best detection accuracy of 98.65%, further demonstrating that dynamically adjusting the scale factors for different objects can improve grasping performance.

4.5. Real robot grasping experiments

To further validate the performance of the proposed method in real robotic arm grasping scenarios, a grasping experimental platform was built as shown in Figure 17. The platform mainly consists of

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

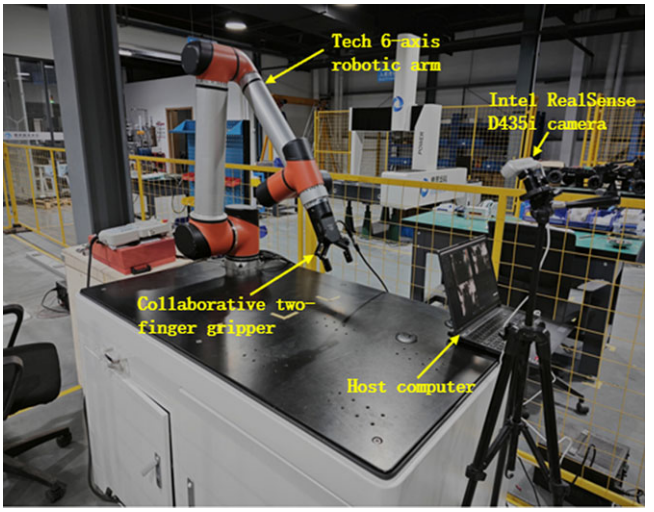


Figure 17. *Real robotic arm grasping experimental platform.*

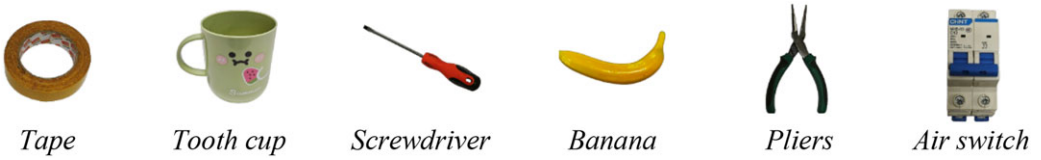


Figure 18. *Grasping experiment targets.*

the Tech TB6-R10 6-axis robotic arm, CTAG2F90-C collaborative two-finger gripper, Intel RealSense D435i camera, and a host computer.

In our study, common everyday objects, as shown in Figure 18, were selected as the grasping experiment targets. Real grasping experiments were conducted in two scenarios: (1) a single object in different poses (Scene 1) and (2) multiple objects placed randomly (Scene 2).

4.5.1. Scene 1 experiments

In this section, a tooth cup was selected as the experimental object for this scene. In the experiment, the grasp is considered successful when the robotic arm successfully grasps the target and effectively places it in the target area. If the object drops or is placed incorrectly, the grasp is considered failed. Finally, the object was placed in 20 different poses within the robotic arm's workspace, and a total of 18 successful grasps were achieved, yielding a success rate of 90%. The grasping process for several poses is shown in Figure 19: (a) Object to be grasped; (b) Grasp detection results from the proposed method, displaying the highest confidence grasp position; (c) Robotic arm grasping the target object; (d) Robotic arm placing the object in the designated area.

4.5.2. Scene 2 experiments

In Scene 2, all experimental objects were randomly placed on the work platform, and the D435i camera captured color and depth images, which were then transmitted to the grasp detection node for processing. After detection, the robotic arm sequentially performs the grasping tasks. Each object grasped is counted as one trial. The success criteria for the grasp are the same as in the first scenario. After all objects are grasped, their placement positions can be randomized for additional testing, allowing for the calculation of the grasp success rate over all trials. The captured images and grasp detection results are shown in Figure 20, and the grasping process is shown in Figure 21. Finally, to further evaluate the performance of

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

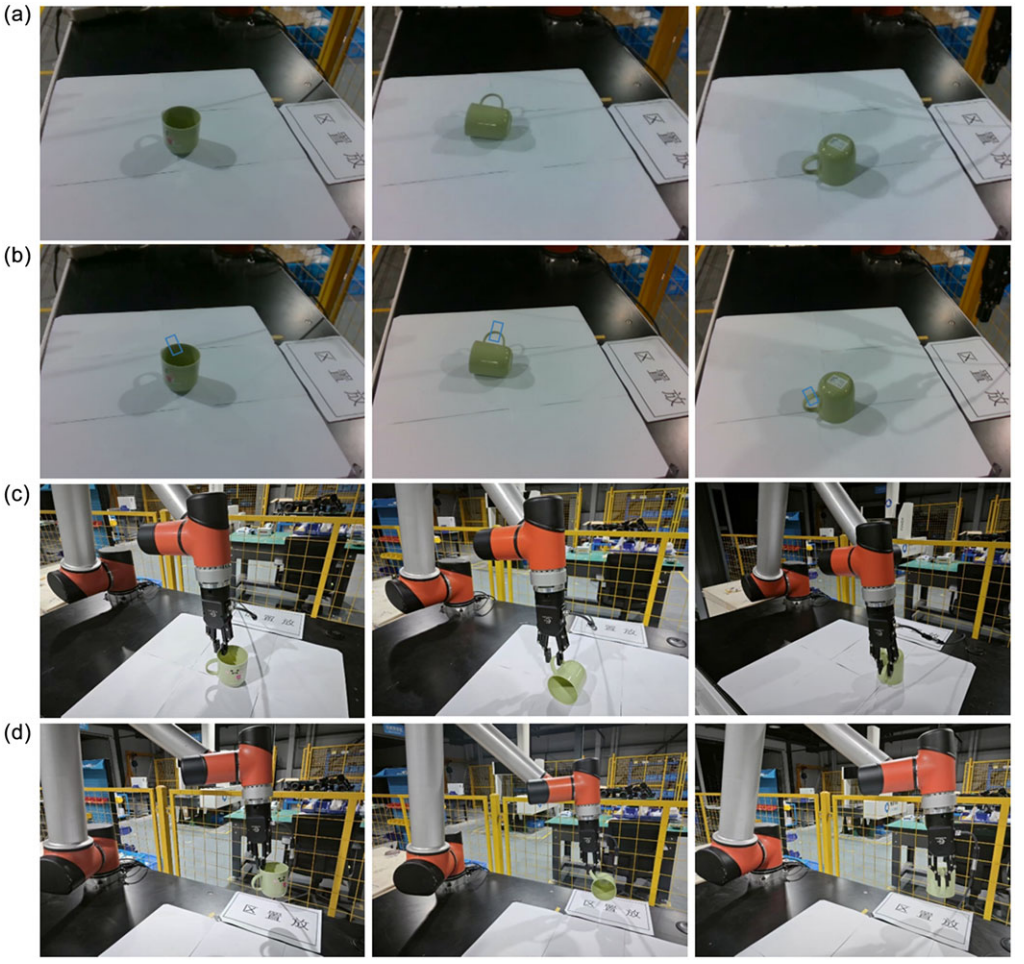


Figure 19. *Partial experimental process for scene 1. (a) Object to be grasped. (b) Grasp detection. (c) grasping. (d) placing.*

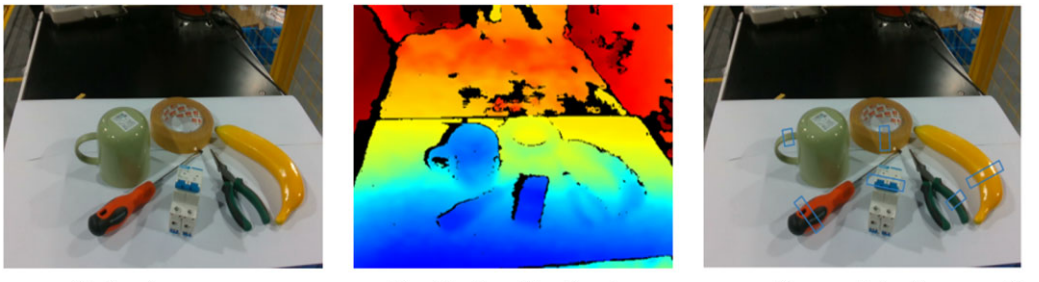


Figure 20. *Captured images and grasp detection results.*

our method, three additional randomly arranged cluttered scenes were tested, with experimental results showing an average grasp success rate of 83.3%.

Compared with simple scenarios, the grasp success rate in cluttered scenarios decreases, which may be attributed to the following reasons. For instance, objects that are densely and irregularly arranged tend to occlude one another or have overlapping edges, which can result in the loss of geometric and other visual features, thereby reducing the accuracy of grasp detection. In this study, a two-finger parallel

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

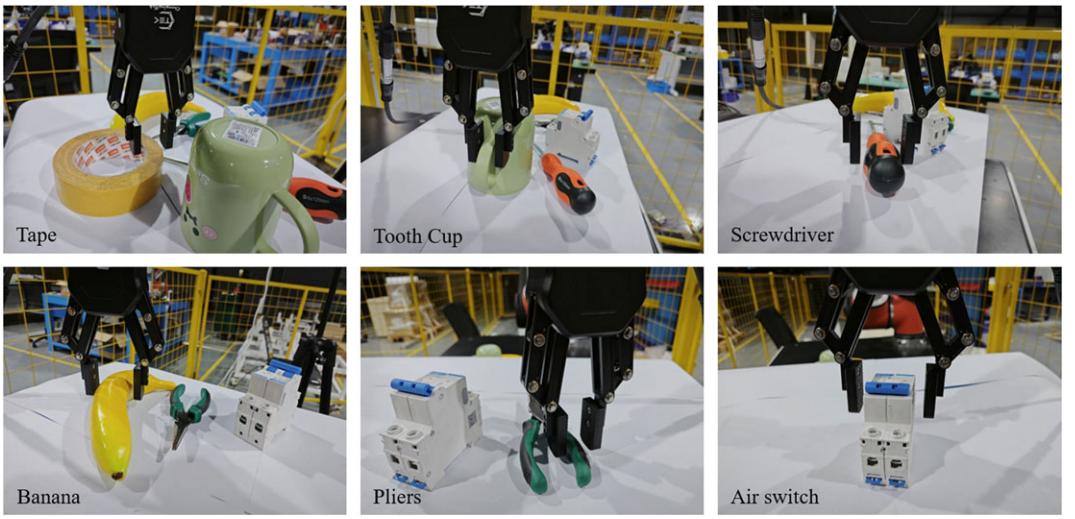


Figure 21. *Grasping process in Scene 2.*

gripper was used for grasping. When multiple objects are placed closely together in a cluttered manner, the limited spacing may lead to collisions during the grasping process, which further contributes to reduced success rates. Future research will focus on addressing these typical challenges to improve the grasping accuracy of robots in real-world cluttered environments.

The experimental results in Scene 1 and Scene 2 demonstrated that the proposed method effectively performs grasp detection on the same object in different poses as well as multiple objects placed randomly. The robotic arm has completed the grasping task based on the grasp detection results, further validating the rationality and effectiveness of the proposed method.

5. Conclusion

To enhance the robotic grasp detection ability for target objects, we addressed the issue of 'false positives' in traditional grasp representations by introducing a 2D Gaussian kernel to emphasize the grasp quality near the center, thereby improving the model's recognition of grasp tasks. Based on this, an improved generative grasp convolutional neural network with an encoder-decoder architecture was proposed, consisting of five stages: input, encoder, bottleneck, decoder, and output prediction. In the bottleneck stage, we introduced a RFB to extract highly discriminative and robust features. In the decoder stage, a GLFFM was introduced to eliminate redundant noise, enhance critical feature information, and guide the model to focus more on the features of the target object. In the experimental section, we first conducted experiments on the Cornell Grasp Dataset, where the image-wise (IW) and object-wise (OW) accuracies reached 99.0% and 98.3%, respectively. The inference time for a single image is only 22 ms. Additionally, to further validate the performance of our method, we conducted experimental evaluations on the Jacquard and GraspNet-1 Billion datasets. The results demonstrate a clear advantage over existing grasp detection methods. Finally, real robotic grasping experiments were also conducted, achieving success rates of 90% in scenarios with the same object in different poses and 83.3% in cluttered multi-object settings. All the above results showed that the proposed method performs excellently in both prediction accuracy and inference speed, enabling efficient grasp detection.

Author contribution. Zhanglai Chen conceived and designed the study. Xu Zhang and Dawei Tu provided relevant theoretical guidance and financial support. Zhanglai Chen carried out relevant experiments and primarily wrote the article, and Dawei Tu revised and polished it.

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

Financial support. This research was supported by National Natural Science Foundation of China (Grant No. 62176149 and No.52327805).

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.

Ethical approval. The authors declare that this work is original and does not include experiments with animals.

References

[1] M. Dong and J. Zhang, "A review of robotic grasp detection technology," *Robotica* **41**(12), 3846–3885 (2023).
[2] J. Yun, D. Jiang, L. Huang, B. Tao, S. Liao, Y. Liu, X. Liu, G. Li, D. Chen and B. Chen, "Grasping detection of dual manipulators based on Markov decision process with neural network," *Neural Netw.* **169**, 778–792 (2024).
[3] X. Huang, M. Halwani, R. Muthusamy, A. Ayyad, D. Swart, L. Seneviratne, D. Gan and Y. Zweiri, "Real-time grasping strategies using event camera," *J. Intell. Manuf.* **33**(2), 593–615 (2022).
[4] Q. M. Marwan, S. C. Chua and L. C. Kwek, "Comprehensive review on reaching and grasping of objects in robotics," *Robotica* **39**(10), 1849–1882 (2021).
[5] G. Du, K. Wang, S. Lian and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review," *Artif. Intell. Rev.* **54**(3), 1677–1734 (2021).
[6] L. Chen, M. Niu, J. Yang, Y. Qian, Z. Li, K. Wang, T. Yan and P. Huang, "Robotic grasp detection using structure prior attention and multiscale features," *IEEE Trans. Syst. Man Cybern. -Syst.* **54**(11), 7039–7053 (2024).
[7] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce and K. Daniilidis, "Single Image 3D Object Detection and Pose Estimation for Grasping," In: 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China (2014).
[8] A. Ten Pas and R. Platt, "Using geometry to detect grasp poses in 3d point clouds," Robotics Research. **2**, 307–324 (2018).
[9] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. C. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, D. Fox and A. Cosgun, "Deep learning approaches to grasp synthesis: A review," *IEEE Trans. Robot.* **39**(5), 3994–4015 (2023).
[10] V. Kushwaha, P. Shukla and G. C. Nandi, "Vision-based intelligent robot grasping using sparse neural network," *Int J Intell Robot Appl.* **9**, 1214–1227 (2025).
[11] D. Kim, A. Li and J. Lee, "Stable robotic grasping of multiple objects using deep neural networks," *Robotica* **39**(4), 735–748 (2021).
[12] J. Yun, S. Moseson and A. Saxena, "Efficient Grasping from RGBD Images: Learning Using a New Rectangle Representation," In: 2011 IEEE International Conference on Robotics and Automation, Shanghai, China (2011).
[13] I. Lenz, H. Lee and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.* **34**(4-5), 705–724 (2015).
[14] J. Redmon and A. Angelova, "Real-time Grasp Detection Using Convolutional Neural Networks," In: 2015 IEEE International Conference on Robotics and Automation (ICRA), Washington, USA (2015).
[15] F.-J. Chu, R. Xu and P. A. Vela, "Deep grasp: Detection and localization of grasps with deep neural networks," ArXiv. abs/1802.00520, (2018).
[16] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang and N. Zheng, "Fully Convolutional Grasp Detection Network with Oriented Anchor Box," In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain (2018).
[17] D. Morrison, P. Corke and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Robot. Res.* **39**(2-3), 183–201 (2020).
[18] S. Kumra, S. Joshi and F. Sahin, "Antipodal Robotic Grasping Using Generative Residual Convolutional Neural Network," In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, Nevada, USA (2020).
[19] S. Kumra, S. Joshi and F. Sahin, "GR-ConvNet v2: A real-time multi-grasp detection network for robotic grasping," *Sensors* **22**(16), 6208 (2022).
[20] W. Prew, T. Breckon, M. Bordewich and U. Beierholm, "Evaluating Gaussian Grasp Maps for Generative Grasping Models," In: 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy (2022).
[21] H. Cao, G. Chen, Z. Li, Q. Feng, J. Lin and A. Knoll, "Efficient grasp detection network with Gaussian-based grasp representation for robotic manipulation," *IEEE/ASME Trans. Mechatron.* **28**(3), 1384–1394 (2022).
[22] S. Liu, D. Huang and Y. Wang, "Receptive Field Block Net for Accurate and Fast Object Detection," In: Computer Vision – ECCV 2018, Cham, (2018).
[23] J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks," In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah (2018).
[24] A. Depierre, E. Dellandréa and L. Chen, "Jacquard: A Large Scale Dataset for Robotic Grasp Detection," In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain (2018).
[25] H. S. Fang, C. Wang, M. Gou and C. Lu, "GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping," In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, USA (2020).
[26] S. Kumra and C. Kanan, "Robotic Grasp Detection Using Deep Convolutional Neural Networks," In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada (2017).

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press

[27] H. Karaoguz and P. Jensfelt, "Object Detection Approach for Robot Grasp Detection," In: 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada (2019).
[28] L. Chen, P. Huang, Y. Li and Z. Meng, "Edge-dependent efficient grasp rectangle search in robotic grasp detection," *IEEE/ASME Trans. Mechatron.* **26**(6), 2922–2931 (2021).
[29] S. Yu, D. H. Zhai, Y. Xia, H. Wu and J. Liao, "SE-ResUNet: A novel robotic grasp detection method," *IEEE Rob. Autom. Lett.* **7**(2), 5238–5245 (2022).
[30] D. H. Zhai, S. Yu and Y. Xia, "FANet: Fast and accurate robotic grasp detection based on keypoints," *IEEE Trans. Autom. Sci. Eng.* **21**(3), 2974–2986 (2024).
[31] S. Wang, Z. Zhou and Z. Kan, "When transformer meets robotic grasping: Exploits context for efficient grasp detection," *IEEE Rob. Autom. Lett.* **7**(3), 8170–8177 (2022).
[32] H. Xi, S. Li and X. Liu, "A pixel-level grasp detection method based on efficient grasp aware network," *Robotica* **42**(9), 3190–3210 (2024).
[33] Z. Zhou, X. Zhang, L. Ran, Y. Han and H. Chu, "DSC-GraspNet: A Lightweight Convolutional Neural Network for Robotic Grasp Detection," In: 2023 9th International Conference on Virtual Reality (ICVR), Xianyang, China (2023).
[34] E. Johns, S. Leutenegger and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (2016).
[35] C. Guo, C. Zhu, Y. Liu, R. Huang, B. Cao, Q. Zhu, R. Zhang and B. Zhang, "End-to-end lightweight transformer-based neural network for grasp detection towards fruit robotic handling," *Comput. Electron. Agric.* **221**, 109014 (2024).
[36] Z. Zhu, S. Huang, J. Xie, Y. Meng, C. Wang and F. Zhou, "A refined robotic grasp detection network based on coarse-to-fine feature and residual attention," *Robotica* **43**(2), 415–432 (2025).
[37] Y. Yan, L. Tong, K. Song, H. Tian, Y. Man and W. Yang, "SISG-Net: Simultaneous instance segmentation and grasp detection for robot grasp in clutter," *Adv. Eng. Inform.* **58**, 102189 (2023).

Cite this article: Z. Chen, X. Zhang and D. Tu (2025). "Research on robotic grasp detection using improved generative convolution neural network with Gaussian representation", Robotica **43**, 4191–4211. https://doi.org/10.1017/S0263574725102750

https://doi.org/10.1017/S0263574725102750 Published online by Cambridge University Press