APSIPA Transactions on Signal and Information Processing, 2023, 12, e26

Original Paper

GP-Net: A Lightweight Generative Convolutional Neural Network with Grasp Priority

Yuxiang Yang[1,2], Yuhu Xing[1], Jing Zhang[2] and Dacheng Tao[2*]

[1] *School of Electronics and Information, Hangzhou Dianzi University, Hangzhou, China*
[2] *School of Computer Science, The University of Sydney, Australia*

ABSTRACT
Grasping densely stacked objects may cause collisions and result in failures, degenerating the functionality of robotic arms. In this paper, we propose a novel lightweight generative convolutional neural network with grasp priority called GP-Net to solve multiobject grasp tasks in densely stacked environments. Specifically, a calibrated global context (CGC) module is devised to model the global context while obtaining long-range dependencies to achieve salient feature representation. A grasp priority prediction (GPP) module is designed to assign high grasp priorities to top-level objects, resulting in better grasp performance. Moreover, a new loss function is proposed, which can guide the network to focus on high-priority objects effectively. Extensive experiments on several challenging benchmarks including *REGRAD* and *VMRD* demonstrate the superiority of our proposed GP-Net over representative state-of-the-art methods. We also tested our model in a real-world environment and obtained an average success rate of 83.3%, demonstrating that GP-Net has excellent generalization capabilities in real-world environments as well. The source code will be made publicly available.

Keywords: Robotic grasping, Generative convolutional network, Calibrated global context module, Grasp priority.

*Corresponding author: Dacheng Tao, dacheng.tao@gmail.com.

Received 13 January 2023; Revised 19 March 2023
ISSN 2048-7703; DOI 10.1561/116.00000002

1 Introduction

The ability to grasp objects is one of the most important and fundamental capabilities of intelligent robots [1, 30–32, 50]. As deep learning techniques have made great progress in visual perception, various deep learning methods have been applied to grasp techniques [9, 11, 28, 37, 43]. Six-degree-of-freedom (6DoF) grasp pose estimation methods [15, 33, 40] focus on constructing point cloud images of objects and diverse 6DoF grasp parameters in the simulation environment. In the real grasping environment, this method filters the grasp parameters with the aid of the positional estimation of the target object point cloud and finally achieves the selection of the optimal grasp parameters. These 6DoF pose estimation methods rely on a known point cloud image of the target object, which limits their performance in practical applications.

Facing the above problems, researchers simplified the process of robotic grasping by using 4DoF parameters, i.e., the x-coordinate of the grasp, the y-coordinate of the grasp, the angle of the grasp, and the width of the grasp. Mahler *et al.* [23] first proposed a two-stage 4DoF grasp detection network. The two-stage grasp detection network first generates the candidate regions through a deep network, and then evaluates the feature vectors of the candidate regions to generate grasp representation. However, these two-stage networks [9, 23] bring significant computational overhead, which impairs real-time efficiency. Recently, Morrison *et al.* [25] proposed a lightweight generative grasping convolutional neural network (GG-CNN) for real-time robotic grasping. This method generated pixel-level grasp images mapped to 4DoF grasp parameters, thus solving the real-time problem in actual grasping. Kumra *et al.* [18] added a residual module to GG-CNN [25], which significantly improved the grasping effect with less impact on the real-time efficiency. Chalvatzaki *et al.* [6] focused on the problem of grasp direction to make the network more concerned with the grasp direction while maintaining real-time efficiency. Xu *et al.* [41] proposed a key point detection algorithm that can reduce the actual detection difficulty of the network and further improve the real-time efficiency of the network. However, these methods are all trained in simple scenarios with a single object. Grasping densely stacked objects may cause collisions and result in failures, degenerating the functionality of these methods.

In fact, the grasp order is particularly important in complex multiobject stacking scenes. Recently, visual manipulation relationship detection methods [27, 45] have been proposed to predict the grasp order in multiobject stacking scenes, which consist of multiple stages, i.e., object detection, grasp detection, and relational inference. Such a multistage framework reduces the real-time efficiency of these methods. Moreover, the generalization ability of object detection and relational inference in complex multiobject stacking scenarios is a bottleneck, especially for unknown objects. Hence, it remains a challenge to obtain a highly robust grasp performance while maintaining real-time efficiency in complex multiobject stacking environments.

Figure 1: Comparisons of GG-CNN[25] and our GP-Net.The grasp representations of different networks in complex stacking scenarios from the *VMRD* dataset [46]. It can be seen that our GP-Net can focus on the top-level objects.

To address the above issues, we propose a lightweight generative convolutional neural network with grasp priority for the real-time grasping of multiple objects in complex environments, called GP-Net. As shown in Figure 1, our GP-Net can focus on the top-level objects and obtain more reasonable grasping results in complex scenes. Specifically, a calibrated global context (CGC) module is devised, which enables our model to have a global understanding of the visual scene by capturing long-range dependencies with a smaller computational effort. Moreover, a novel grasp priority prediction (GPP) module is designed to obtain the grasp order of multiple objects by generating pixel-level grasp priorities. In addition, a new loss function is constructed using the pixel-level grasp priority mask, which guides the network to efficiently focus on the top-level objects with high grasp priorities.

The contributions of this study can be summarized as follows:

- We devise a novel generative grasping convolutional neural network with grasp priority for real-time multiobject grasping in complex stacking environments.

- A calibrated global context module is proposed to obtain a more salient feature representation, which enables a global understanding of the visual scene by capturing long-range dependencies.
- A new grasp priority prediction module is developed to obtain the pixel-level grasp priority, which can efficiently guide the network to focus on the top-level objects together with a specially designed loss function.

The remainder of the paper is organized as follows: Section 2 reviews related works. In Section 3, we present the details of the proposed method, including the CGC module, the GPP module, and the loss function. The experimental results and analysis are given in Section 4. Finally, we conclude the paper in Section 5.

2 Related Work

2.1 Robotic Grasping

In past research, most robotic grasping methods relied on known information about the environment as well as the object model to obtain the optimal grasp poses [17, 24]. With the development of neural networks, deep learning methods have been applied to the field of robotic grasping, which can be broadly divided into two categories, namely, 6DoF grasp and 4DoF grasp. In 6DoF grasp, early works [2, 15, 40] focused on generating 6DoF grasp parameters through physical simulation and matching the appropriate grasp parameters by estimating the object's poses in the evaluation environment. However, these methods require a 3D model of the target object, which is difficult to acquire in many practical situations [20].

In 4DoF grasp, the number of parameters can be reduced by specifying a top-down grasp. A two-stage 4DoF grasp detection network was proposed in [23], which applied a region proposal network (RPN) to generate the region of interest (ROI) [19]. Then, ROIs generated in the first stage were used to crop the corresponding features and predict the grasp parameters. The two-stage method achieves satisfactory performance in 4DoF grasp detection, but the high inference latency of the two-stage framework significantly limits its application in practical scenarios. Recently, lightweight generative grasping networks were proposed for real-time 4DoF grasping [4, 18, 25, 41] by directly generating the images of geometric grasp parameters. These methods effectively improve the real-time efficiency of robotic grasping. However, these methods are trained in single-object environments. In multiobject stacking environments, unreasonable grasp predictions lead to collisions, thus greatly affecting the actual effectiveness of these methods. Different from these methods, in this paper a novel GP-Net is devised for real-time multiobject grasping in complex stacking environments by predicting the grasp priority.

2.2 Multitask Learning

Multitask learning (MTL) can improve the performance of the primary task through collaborative training on auxiliary tasks [12, 21, 52, 53]. By sharing features across multiple tasks, the network is guided to learn a common representation among them, which may reduce overfitting and thus better generalize the original task [39]. Hence, a growing number of MTL methods have been used in the field of robotic grasping to improve performance. For example, Prew *et al.* [29] achieved higher grasp detection performance using depth prediction as a secondary task. Nguyen *et al.* [26] improved the grasp detection performance with the aid of the bounding box generation task. Yu *et al.* [44] proposed a grasp task implemented through a secondary task of segmenting objects. In this paper, we design a novel grasp priority prediction auxiliary task for the 4DoF grasp detection network, which can obtain more accurate grasp parameters by sharing parameters among the tasks. Moreover, the auxiliary task in our network has little impact on the network inference time.

3 The Proposed Method

3.1 Problem Reconfiguration

In the top-down grasping model, the robotic grasping problem can be defined as a parameter estimation problem with four variables [18, 25]:

$$G_r = (\mathbf{S}, \Theta_r, W_r, Q)\,, \tag{1}$$

where $\mathbf{S} = (x, y, z)$ indicates the 3D coordinates of the gripping center. Θ_r indicates the angle of rotation of the gripping end around the z-axis. W_r is the width of the clamping jaw opening and Q is the grasp quality score.

However, these methods [18, 25] are trained in simple scenarios with a single object. Grasping densely stacked objects may cause collisions and result in failures, degenerating the functionality of these methods. In this paper, we propose an efficient grasp priority constraint for multi-object grasping task in cluttered stacked scenes. The higher the grasping priority of the region, the higher its grasp quality score should be. Hence, the equation can be expressed as:

$$G_r = (\mathbf{S}, \Theta_r, W_r, Q^*)\,, \tag{2}$$

where $Q^* = (Q \cdot P)$. Q indicates the original grasp quality score. P indicates our new grasp priority in a multiobject scene. A larger value of P indicates a higher priority. Q and P are multiplied together to obtain Q^*. A larger value of Q^* indicates a higher grasp success rate in the multiobject scene.

Specifically, our GP-Net predicts a set of images $\left\{\mathbf{Q_i}, \mathbf{\Theta_i^{cos}}, \mathbf{\Theta_i^{sin}}, \mathbf{W_i}, \mathbf{P_i}\right\}$. The size of each predicted image is 224×224. $\mathbf{Q_i}$ is the grasp quality score

image, $\mathbf{P_i}$ is the grasp priority image, $\mathbf{\Theta_i^{sin}}$ is the grasp angle sin component, $\mathbf{\Theta_i^{cos}}$ is the grasp angle cos component. Then, as shown in Figure 2, the dot product of $\mathbf{Q_i}$ and $\mathbf{P_i}$ gives our new grasp quality score $\mathbf{Q_i^*}$. $\mathbf{\Theta_i^{sin}}$ and $\mathbf{\Theta_i^{cos}}$ combined to obtain the grasp angle image $\mathbf{\Theta_i}$.

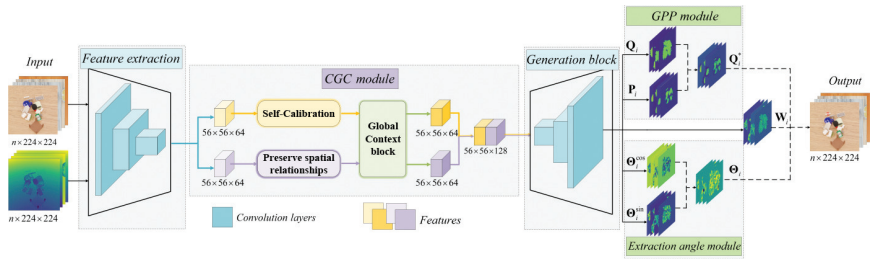

Figure 2: GP-Net framework. The input RGB-D image needs to go through three parts: feature extraction, attention block, and generation block. The grasp quality score image $\mathbf{Q_i}$, the grasp priority image $\mathbf{P_i}$, the grasp cosine angle image $\mathbf{\Theta_i^{cos}}$, the grasp sine angle image $\mathbf{\Theta_i^{sin}}$, and the grasp width image $\mathbf{W_i}$ are obtained from the generation block. These image features are fused to obtain the final valid grasp parameters.

We can get the coordinate (u, v) of the pixel with the maximum value in $\mathbf{Q_i^*}$. The x-axis and y-axis of $\mathbf{S}$ of Equation (2) can be obtained from (u, v) using a coordinate transformation, and the z-axis of $\mathbf{S}$ can be obtained from the depth map. Θ_r of Equation (2) can be obtained from the corresponding pixel (u, v) in the grasping angle image $\mathbf{\Theta_i}$. W_r of Equation (2) can be obtained from the coordinates pixel (u, v) of the grasping angle image $\mathbf{W_i}$. Thus, we can get all the grasp parameters.

3.2 GP-Net

In this paper, we propose a lightweight generative convolutional neural network with grasp priorities (GP-Net) for multiobject grasping in complex stacking environments. GP-Net is mainly composed of feature extraction, a calibrated global context (CGC) module, a generation block, a grasp priority prediction (GPP) module, and an extraction angle module. Among them, the feature extraction is mainly composed of convolutional layers, BN layers and ReLU activation layers. The feature map size changes $224 \rightarrow 112 \rightarrow 56 \rightarrow 56$. The CGC can enlarge the perceptual field and extract effective visual information. The generation block mainly consists of transpose convolution layer, BN layers and ReLU activation layers connected in series. The feature map size changes $56 \rightarrow 56 \rightarrow 112 \rightarrow 224$. The generation block is mainly responsible for the generation of feature maps. The GPP realizes the prediction of grasp priority, and the angle extraction module obtains the grasp angle information.

Figure 2 shows the diagram of our GP-Net. We input the RGB image and depth map of size 224×224 of n channels into the feature extraction

network to obtain the feature map of size 56×56. We then feed the feature map into the long-range attention module. Specifically, to obtain more spatial feature information while keeping the network lightweight, we divide the obtained feature map into two parts. One part is self-calibrated, and the other part undergoes a convolution operation. The results of the two parts are concatenated after passing through a self-attention module. The concatenated feature maps are then passed through transposed convolution to generate images containing grasp information [3, 5, 16, 34].

Finally, the grasp quality score $\mathbf{Q_i}$ and the grasp priority $\mathbf{P_i}$ are combined to obtain our new grasp quality score $\mathbf{Q_i^*}$ with more prominent features. We extract the angle in the form of two elements $\mathbf{\Theta_i^{cos}}$ and $\mathbf{\Theta_i^{sin}}$ that output distinct values that are combined to form the required angle $\mathbf{\Theta_i}$. The point with the largest pixel value in the grasp quality score image $\mathbf{Q_i^*}$ is the 2D coordinate of the grasp center, and the pixel value at the same position in the grasp width image $\mathbf{W_i}$ and the angle image $\mathbf{\Theta_i}$ is the grasp width and angle centered on that 2D coordinate in the image coordinate system.

3.2.1 Calibrated Global Context Module

To extract more meaningful visual features, most improvements in convolutional neural networks have focused on tuning the architecture of the network model to produce a rich finite element analysis [7, 8, 13, 38, 53]. There are two problems that make the extracted feature maps not very distinguishable: (a) Each output feature map is calculated by summing all channels, and all feature maps are generated uniformly by repeating the same formula several times. (b) The perceptual field of each spatial location is mainly controlled by the predefined convolutional kernel size.

In the CGC module, we use multiple convolutional kernels of different sizes and consider the spatial updown relationship to have a larger perceptual field. We put more emphasis on local contextual relationships, thus enabling more accurate positioning of grasp detection. Adaptive operations encode multiscale information, which provides rich features for grasp detection tasks.

Furthermore, long-range dependencies are particularly important in most vision tasks [10, 22, 35, 36, 42, 48, 49, 51]. To capture long-range dependencies, two types of approaches have been proposed. The first one uses a self-attention mechanism to model the relationship of query pairs, while the second one models the global context in a query-independent manner. But, the approach that uses a self-attentive mechanism to model the relationship of query pairs is computationally intensive and the approach that models the global context in a query-independent manner does not take full advantage of the global context information. Different from these methods, we design the CGC module to generate the global attention map, which is shared by all locations. On the one

hand, we do not create query pairs, which reduces the amount of computation. On the other hand, our module generates an attention weight for each point of the feature map. Long-range dependencies built in this way can obtain more global information while keeping the network lightweight. Hence, our CGC is able to model effective long-range dependencies such as SNL blocks [5, 35] and save computations such as SE blocks [13]

Specifically, the architecture of our CGC is given in Figure 3. We divided the feature map obtained from the feature extraction module into two parts. The purpose of this design is that on the one hand we need to extract deeper information and on the other hand we need a branch to preserve the feature representation from the upstream feature extraction module. In the branch above as shown in Figure 3, convolution kernels W_2, W_3, and W_4 are applied to extract the deeper feature representations. In terms of details, in the self-calibration module, we conduct convolutional feature transformation in two different scale spaces to efficiently gather informative contextual information for each spatial location, i.e., an original scale space and a small latent space after down-sampling. The embeddings after W_3 in the small latent space have large fields-of-view and are used as references to guide the original feature space. Our self-calibrating convolution can achieve the purpose of enlarging the receptive field through the intrinsic communication of features, which enhances the diversity of output features. In the global context block, we generate the global attention map by context modeling, which is shared with all locations. The implementation of context modeling relies on the 1×1 convolution kernel to extract the weights on the feature map. All locations share an attention map, which is less computationally intensive and allows global information to be encoded. The branch below as shown in Figure 3, is designed to preserve the original spatial context information.

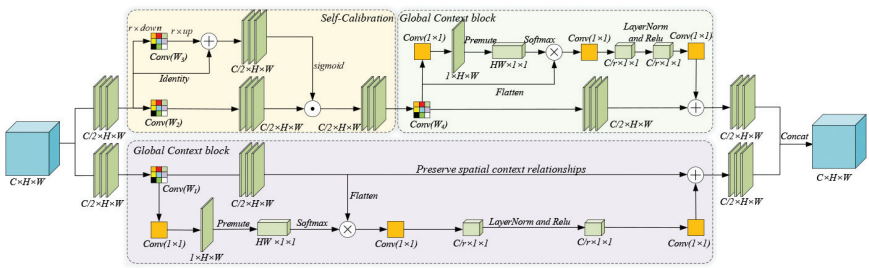

Figure 3: Illustration of the CGC module.

3.2.2 Grasp Priority Prediction Module

In previous grasping works [6, 18], depth information is often used to obtain the order of the entire grasp task in the face of multiobject stacking environments,

and there are also many studies that obtain the grasp order in terms of visual operational relationships [45]. However, visual operational relationships can only have some effect on some secondary grasping networks. For real-time closed-loop lightweight grasping networks, it is still a challenge to make judgments about the grasp order.

We constructed the grasp priority prediction (GPP) module by drawing upon the human experience of grasping in real life. Specifically, when facing the grasp challenge of multiple objects stacked in real scenarios, humans tend to prioritize the topmost objects for grasping, thus making the whole grasping process more stable and avoiding collisions. Therefore, the topmost object in a multiobject stacking environment has the highest grasp priority. For complex stacking environments, we represent the topmost object in a multiobject stacking scene by constructing a mask image P, i.e., the grasp priority. In the grasp priority mask P, the larger the pixel value is, the higher the object is in the top layer of the stacked scene, and the higher the capture priority of the object.

3.3 The Loss Function

The loss function of GG-CNN [25] is the sum of the mean squared loss of the output image in the space of four parameters $\mathbf{Q_i}$, $\mathbf{\Theta_i^{cos}}$, $\mathbf{\Theta_i^{sin}}$, and $\mathbf{W_i}$.

In this paper, we propose a new loss function to suppress regions of background and regions of non top-level objects. The network should learn to output values close to a default value of 0 for angle and width in such regions. To this end, we define the value of grasp priority to solve this problem. The scaling of the loss in this manner focuses the learning of the network on the grasp quality score, the grasp angle and width of the top object, and the grasp priority.Our new loss function is defined as follows. Assuming that the predicted grasp parameters are $\left\{\mathbf{Q_i}, \mathbf{\Theta_i^{cos}}, \mathbf{\Theta_i^{sin}}, \mathbf{W_i}, \mathbf{P_i}\right\}$ and the true grasp ground-truth are $\left\{\mathbf{Q_{gt}}, \mathbf{\Theta_{gt}^{cos}}, \mathbf{\Theta_{gt}^{sin}}, \mathbf{W_{gt}}, \mathbf{P_{gt}}\right\}$, the proposed loss function is defined as:

$$\begin{aligned} L_{GP-Net} = &\left\|\mathbf{Q_i} - \mathbf{Q_{gt}}\right\|^2 + \left\|\mathbf{P_{gt}}\left(\mathbf{\Theta_i^{sin}} - \mathbf{\Theta_{gt}^{sin}}\right)\right\|^2 \\ &+ \left\|\mathbf{P_{gt}}\left(\mathbf{\Theta_i^{cos}} - \mathbf{\Theta_{gt}^{cos}}\right)\right\|^2 + \left\|\mathbf{P_{gt}}\left(\mathbf{W_i} - \mathbf{W_{gt}}\right)\right\|^2 \\ &+ \left\|\mathbf{P_i} - \mathbf{P_{gt}}\right\|^2 \end{aligned} \tag{3}$$

We get the $\mathbf{Q_{gt}}$, $\mathbf{\Theta_{gt}^{cos}}$, $\mathbf{\Theta_{gt}^{sin}}$, and $\mathbf{W_{gt}}$ ground-truths as described in [25]. In the *REGRAD* grasp dataset [47], the pixel-level segmentation information between each object is given, and the parent-child relationship between the stacked objects is also available. Based on these two relevant information, we

can obtain pixel-level segmentation of the top-level objects and get the priority ground-truth $\mathbf{P_{gt}}$.

4 Experiment

4.1 Datasets

In this paper, the *REGRAD* dataset [47] is used to train the network. *REGRAD* is a simulation dataset that consists of 50K kinds of objects with 100M grasping labels. In addition, the *REGRAD* dataset also contains the operational relationships between different objects and their segmentation. Using this information, we construct the grasp priority masks, as shown in Figure 4, which are used as the grasp priority ground-truth $\mathbf{P_{gt}}$. We can also obtain the grasp ground-truth of center $\mathbf{Q_{gt}}$, angle $\mathbf{\Theta_{gt}^{cos}}$, angle $\mathbf{\Theta_{gt}^{sin}}$, width $\mathbf{W_{gt}}$ by mapping grasping labels as described in [25]. Then we evaluated our method on the test set of *REGRAD* and the *VMRD* dataset [46], as well as in our constructed real scenarios.

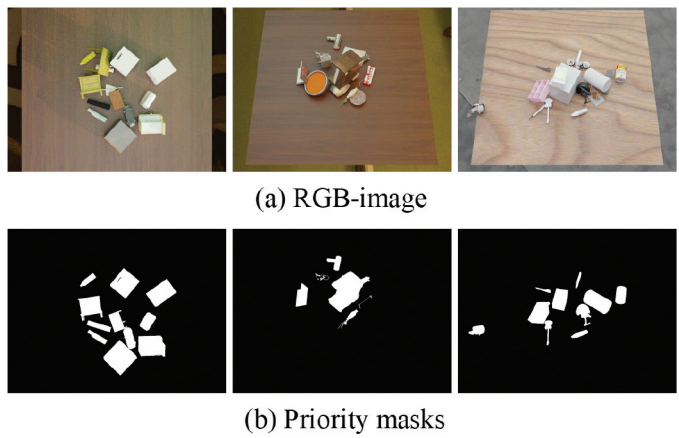

Figure 4: The grasp priority masks of our method.

4.2 Grasp Detection Metric

The rectangle metric [14] is used to report the performance of different methods. Specifically, a grasp is considered valid when it satisfies the following two conditions: (1) the intersection over union (IoU) score between the ground truth and the predicted grasp rectangle is more than 25%, and (2) the offset between the predicted grasp orientation and the ground truth is less than 30°.

4.3 Implementation Details

The execution time of our proposed GP-Net is measured on a system running Ubuntu 16.04 with an Intel Core i7-7800X CPU clocked at 3.50 GHz and an NVIDIA GeForce GTX 1080 Ti GPU with CUDA 10. We use the Adam optimizer with an initial learning rate of 0.001 and a gradual decay to 0.0001. Because of the memory limitation, the size of the mini-batch in this paper is set to 16 and the network is trained for a total of 50 epochs.

To comprehensively evaluate the robustness of the network in different settings, we trained our network in three settings, i.e., RGB image input only, depth image input only, and RGB-D input. The 2D grasping part of the *REGRAD* dataset was used as the training set in all three settings. The training set contains 5.3K RGB images, depth images, operation relation labels, image segmentation labels, and approximately 6984.9K grasp labels. For testing, we used the rest of the *REGRAD* dataset and *VMRD* dataset as the test sets and tested the model in real-world complexly stacked scenes. For a fair evaluation, we retrained GG-CNN [25], GG-CNN2 [25], GR-ConvNet [18], and ORANGE [6] networks on the same *REGRAD* training set.

4.4 Quantitative Evaluation

We evaluated our network on the *REGRAD* dataset [47], recording the grasp success rate in different scenarios. To compare with previous work, we evaluated some grasp networks on the *REGRAD* dataset as well. For a fair comparison, we trained these networks on the *REGRAD* dataset. Our method improves the grasping success rate by 29.3% over the GG-CNN network on the *REGRAD* dataset. Compared with GR-ConvNet, it improved the grasp accuracy from 67.1% to 82.6%. Our method achieves state-of-the-art grasp performance on the *REGRAD* dataset compared with other networks of the same type. To evaluate the effectiveness of different modules in our model, we performed ablation experiments in two cases, i.e., removing the CGC module and removing the GPP module. After removing the CGC module and GPP module, the overall grasping performance of the algorithm showed a relatively large drop. It is also noteworthy that the removal of the GPP module resulted in a larger drop in performance and a more pronounced impact on the overall system. The results in Table 1 show that the auxiliary task of grasp priority significantly enhances the robustness of grasping, with a 19.4% improvement in the success rate of grasping in the same test set, compared with the network without the GPP module. The experimental results show that the CGC module and GPP module play an important role in the whole system and can help achieve a better grasp performance in complexly stacked multiobject scenes.

In addition, we evaluate the performance of the network with different input modalities. The modalities that the model was tested on included unimodal

Table 1: Evaluation results on the *REGRAD* dataset.

Algorithm	Input	Accuracy(%)	Time (ms)
GGCNN [25]	RGB-D	53.3	19
GGCNN2 [25]	RGB-D	56.4	20
GR-ConvNet [18]	RGB-D	67.1	20
ORANGE [6]	Depth	64.6	-
Ours (No CGC)	Depth	70.1	20
	RGB	72.3	21
	RGB-D	74.7	22
Ours (No GPP)	Depth	66.3	20
	RGB	67.8	20
	RGB-D	69.2	22
Ours	Depth	75.3	21
	RGB	78.2	21
	RGB-D	82.6	23

input, such as depth-only and RGB-only input images, as well as multimodal input, such as RGB-D images. Table 1 shows that our network performs better on multimodal data than on unimodal data because the multiple input modalities provide abundant information for learning better features. We evaluated the latency of GP-Net and other methods and report the results in Table 1. Compared with other grasp detection methods, GP-Net achieves the most advanced detection results on the *REGRAD* dataset with a small extra latency of only 1–2 ms. These results indicate that our method can perform well for real-time robotic grasping tasks.

We compare the success rate of grasping by using different loss functions to train our GP-Net on the *REGRAD* dataset. Figure 5 shows that using the priority loss function improves the grasp success rate by approximately 5% compared with using the GG-CNN loss function. This is because the proposed priority loss function can suppress the background area of non top-level objects in multiobject stacking scenes, thereby delivering better grasp results.

To qualitatively illustrate the effectiveness of our method, we also present the visual results of GP-Net on the *VMRD* dataset [46]. All the methods in Table 2 are trained using the *REGARD* grasp datastet and directly tested on the *VMRD* grasp dataset. Our method obtains a success rate of 70.2% and 20ms latency on the *VMRD* dataset. Compared with other methods, our model achieves the best performance. As shown in Figure 6, the grasp quality score map $\mathbf{Q}^*$ indicates grasping of top-level objects, while the grasp angle and width show a significant suppression effect on the region of non top-level objects.

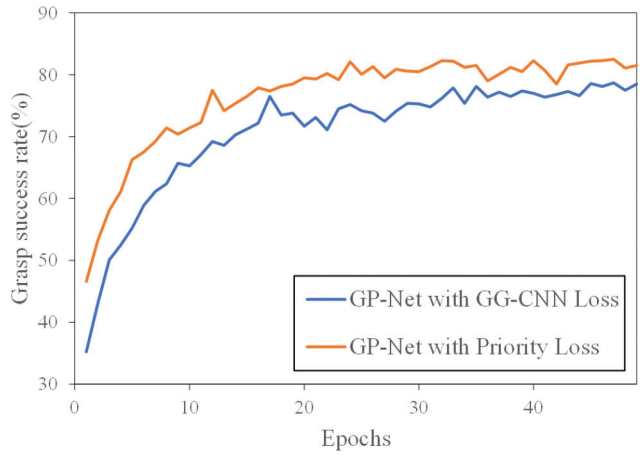

Figure 5: The comparison of using different loss functions to train GP-Net. In multi-object scenario, the priority loss function obtains the higher grasp success rate.

Table 2: Evaluation results on the *VMRD* dataset.

Algorithm	Input	Accuracy(%)	Time (ms)
GGCNN [25]	RGB	52.1	19
GGCNN2 [25]	RGB	53.7	19
GR-ConvNet [18]	RGB	62.4	20
Ours (No CGC)	RGB	66.9	20
Ours (No GPP)	RGB	63.3	20
Ours	RGB	70.2	21

The results demonstrate that our GP-Net can effectively grasp the topmost object, which is an extremely reasonable grasp in a multiobject stacked scene, and can effectively avoid collisions. The evaluation results on the *VMRD* dataset also demonstrate that our GP-Net generalizes well to new objects that it has never seen before.

4.5 Evaluation Results in Real-world Scenarios

We built a real robot arm test environment, where the robot arm is a UR10 and the camera is an Inter Realsense D435, and the objects used in testing are shown in Figure 7 (a). During testing, incorrect grasping is defined as shown in Figure 7 (b) where the grasped object is not a top-level object. Successful grasping is defined as shown in Figure 7 (c), where the topmost object is grasped.

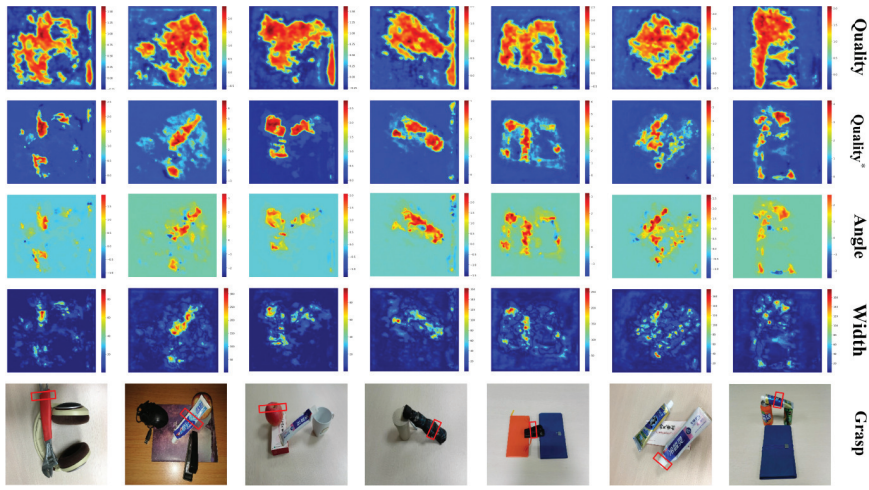

Figure 6: Visual results on the *VMRD* dataset [46]. "Quality" is the original grasp quality score $\mathbf{Q}$, while "Quality*" is the grasp quality score $\mathbf{Q}^*$ after combining with the predicted priority mask. It can be seen that $\mathbf{Q}^*$ has more prominent features on the top-level objects. Besides, the angle and width feature maps show a strong suppression effect on the background region of non-top-level objects with the supervision of the priority-optimized loss function. GP-Net has a more reasonable grasp performance for grasp detection in complex stacking scenes.

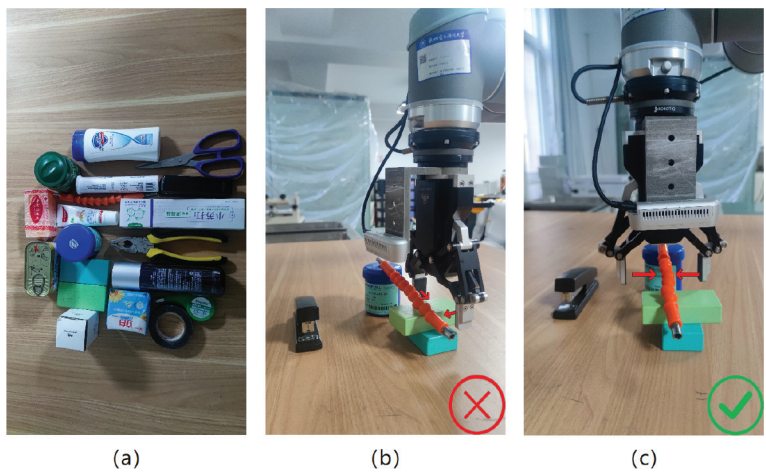

Figure 7: Real scenario testing. (a) The objects used in our testing, (b) Incorrect grasping where the grasped object is not a top-level object, and (c) Successful grasping.

In the real environment grasping test, we constructed grasping scenarios with different levels of difficulty. The number of objects and the number of pairs with stacking relationships between objects were used as the criteria

to measure the difficulty of grasping in the grasping experiments. Figure 8 shows the results of our grasp method in real scenarios. The new grasp quality score $\mathbf{Q}^*$ gives higher grasp priority to the top-level objects. This indicates that our method has high stability in a real stacking environment. Next, we selected 3-7 objects for grouping, and in each group, we performed 18 rounds of grasping experiments. To better illustrate the effect of different stacking scenarios on the grasping success rate, we measured the real environment grasping performance as the ratio of the number of rounds that successfully completed all the grasps of the whole group to the total number of grasping rounds in each group of 18 rounds of grasping experiments. From Table 3, it

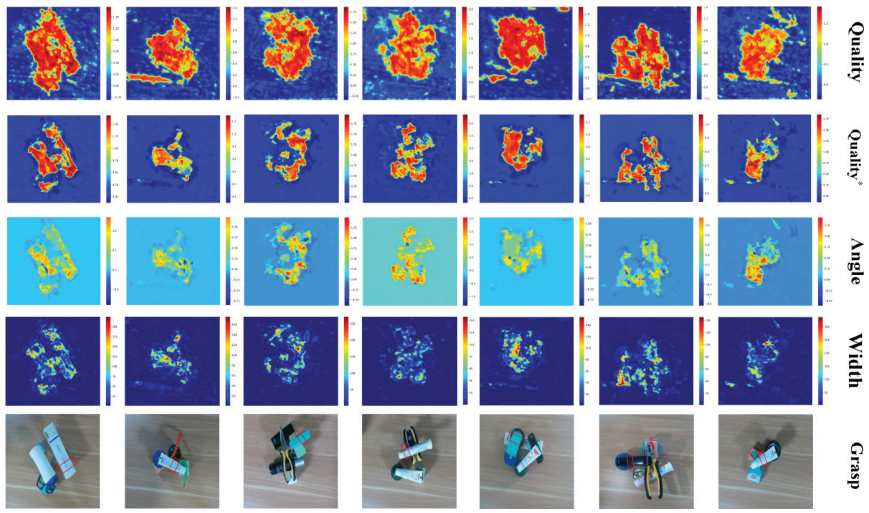

Figure 8: Qualitative analysis of real scenarios. "Quality" is the original grasp quality score $\mathbf{Q}$, and "Quality*" is the grasp quality score $\mathbf{Q}^*$ after combining with the predicted priority. Note that $\mathbf{Q}^*$ gives the topmost objects a higher grasp priority.

Table 3: Results of our method in real scenario testing.

Number of objects	Number of pairs of objects with stacking relationships	Accuracy(%)
3	2	94.4
4	3 ± 1	83.3
5	4 ± 1	88.8
6	5 ± 1	77.8
7	6 ± 2	72.2

can be seen that our method obtains satisfactory grasping results in real-world complex stacking scenarios, with an average grasping success rate of 83.3%.

5 Conclusions

In this paper, a novel neural network GP-Net is proposed to generate grasp parameters with grasp priority, which improves the rationality of the grasp representation for complex stacking scenarios. We evaluate our GP-Net on public grasping datasets and real-world complex stacking scenarios. Extensive experiments demonstrate the superiority of our method over previous representative methods and its good generalization ability to deal with unseen objects. Moreover, our GP-Net also enjoys a fast inference speed, which can meet the real-time requirement in practical applications.

Acknowledgements

Part of this work was done during Yuxiang Yang's visit at The University of Sydney. This work was supported by the Zhejiang Provincial Natural Science Foundation Key Fund of China (LZ23F030003).

References

[1] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *2000 IEEE International Conference on Robotics and Automation*, Vol. 1, 2000, 348–53.

[2] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, "Volumetric Grasping Network: Real-time 6 DOF Grasp Detection in Clutter," in *Conference on Robot Learning*, PMLR, 2021, 1602–11.

[3] R. Brooks, Y. Yuan, Y. Liu, H. Chen, *et al.*, "DeepFake and its Enabling Techniques: A Review," *APSIPA Transactions on Signal and Information Processing*, 11(2), 2022.

[4] H. Cao, G. Chen, Z. Li, J. Lin, and A. Knoll, "Lightweight Convolutional Neural Network with Gaussian-based Grasping Representation for Robotic Grasping Detection," *arXiv preprint arXiv:2101.10226*, 2021.

[5] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *2019 IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[6] G. Chalvatzaki, N. Gkanatsios, P. Maragos, and J. Peters, "Orientation Attentive Robotic Grasp Synthesis with Augmented Grasp Map Representation," *arXiv preprint arXiv:2006.05123*, 2020.

[7] H.-S. Chen, S. Hu, S. You, C.-C. J. Kuo, *et al.*, "Defakehop++: An enhanced lightweight deepfake detector," *APSIPA Transactions on Signal and Information Processing*, 11(2), 2022.

[8] P.-R. Chen, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "DSNet: An efficient CNN for road scene segmentation," *APSIPA Transactions on Signal and Information Processing*, 9, 2020.

[9] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, 3(4), 2018, 3355–62.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2020.

[11] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *The International Journal of Robotics Research*, 39(2-3), 2020, 202–16.

[12] B. Hou, Y. Liu, N. Ling, Y. Ren, L. Liu, *et al.*, "A Survey of Efficient Deep Learning Models for Moving Object Segmentation," *APSIPA Transactions on Signal and Information Processing*, 12(1), 2022.

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 7132–41.

[14] Y. Jiang, 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 (ICRA)*, 2011, 3304–11.

[15] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *2017 IEEE International Conference on Computer Vision (CVPR)*, 2017, 1521–9.

[16] D. P. Kingma, M. Welling, *et al.*, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, 12(4), 2019, 307–92.

[17] D. Kragic and H. I. Christensen, "Robust visual servoing," *The International Journal of Robotics Research*, 22(10-11), 2003, 923–39.

[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)*, 2020, 9626–33.

[19] E. A. D. Lagmay, M. M. T. Rodrigo, *et al.*, "Enhanced Automatic Areas of Interest (AOI) Bounding Boxes Estimation Algorithm for Dynamic Eye-Tracking Stimuli," *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[20] W. Lin, S. Lee, *et al.*, "Visual Saliency and Quality Evaluation for 3D Point Clouds and Meshes: An Overview," *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[21] T. Liu, D. Tao, M. Song, and S. J. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2), 2016, 227–41.

[22] B. Ma, J. Zhang, Y. Xia, and D. Tao, "Auto learning attention," *2020 Advances in Neural Information Processing Systems*, 33, 2020, 1488–500.

[23] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[24] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, 2308–15.

[25] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International Journal of Robotics Research*, 39(2-3), 2020, 183–201.

[26] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, 5908–15.

[27] D. Park, Y. Seo, D. Shin, J. Choi, and S. Y. Chun, "A single multi-task deep neural network with post-processing for object detection with reasoning and robotic grasp detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, 7300–6.

[28] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta, "The curious robot: Learning visual representations via physical interactions," in *European Conference on Computer Vision*, 2016, 3–18.

[29] W. Prew, T. Breckon, M. Bordewich, and U. Beierholm, "Improving Robotic Grasping on Monocular Images Via Multi-Task Learning and Positional Loss," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, 9843–50.

[30] A. Rakshit, A. Konar, and A. K. Nagar, "A hybrid brain-computer interface for closed-loop position control of a robot arm," *IEEE/CAA Journal of Automatica Sinica*, 7(5), 2020, 1344–60.

[31] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, 27(2), 2008, 157–73.

[32] K. B. Shimoga, "Robot grasp synthesis algorithms: A survey," *The International Journal of Robotics Research*, 15(3), 1996, 230–66.

[33] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *2019 IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 3343–52.

[34] H.-P. Wang, W.-H. Peng, and W.-J. Ko, "Learning priors for adversarial autoencoders," *APSIPA Transactions on Signal and Information Processing*, 9, 2020.

[35] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 7794–803.

[36] K. Li-Wei, "Special issue on deep learning based detection and recognition for perceptual tasks with applications," *APSIPA Transactions on Signal and Information Processing*, 8, 2019.

[37] T. Weng, A. Pallankize, Y. Tang, O. Kroemer, and D. Held, "Multi-modal transfer learning for grasping transparent and specular objects," *IEEE Robotics and Automation Letters*, 5(3), 2020, 3791–8.

[38] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *2018 European Conference on Computer Vision (ECCV)*, 2018, 3–19.

[39] Y. Wu, K. Yoshii, *et al.*, "Joint Chord and Key Estimation Based on a Hierarchical Variational Autoencoder with Multi-task Learning," *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

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

[41] R. Xu, F.-J. Chu, and P. A. Vela, "Gknet: grasp keypoint network for grasp candidates detection," *The International Journal of Robotics Research*, 2022.

[42] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, "Vitae: Vision transformer advanced by exploring intrinsic inductive bias," *Advances in Neural Information Processing Systems*, 34, 2021, 28522–35.

[43] Y. Yang, Z. Ni, M. Gao, J. Zhang, and D. Tao, "Collaborative Pushing and Grasping of Tightly Stacked Objects via Deep Reinforcement Learning," *IEEE/CAA Journal of Automatica Sinica*, 9(1), 2021, 135–45.

[44] Y. Yu, Z. Cao, Z. Liu, W. Geng, J. Yu, and W. Zhang, "A Two-Stream CNN With Simultaneous Detection and Segmentation for Robotic Grasping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[45] H. Zhang, X. Lan, L. Wan, C. Yang, X. Zhou, and N. Zheng, "Rprg: Toward real-time robotic perception, reasoning and grasping with one multi-task convolutional neural network," *arXiv preprint arXiv:1809.07081*, 2018, 1–7.

[46] H. Zhang, X. Lan, X. Zhou, Z. Tian, Y. Zhang, and N. Zheng, "Visual manipulation relationship network for autonomous robotics," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, 118–25.

[47] H. Zhang, D. Yang, H. Wang, B. Zhao, X. Lan, J. Ding, and N. Zheng, "REGRAD: A Large-Scale Relational Grasp Dataset for Safe and Object-Specic Robotic Grasping in Clutter," *IEEE Robotics and Automation Letters*, 2022.

[48] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, 7151–60.

[49] J. Zhang, Z. Chen, and D. Tao, "Towards high performance human keypoint detection," *International Journal of Computer Vision*, 129(9), 2021, 2639–62.

[50] J. Zhang and D. Tao, "Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet of Things Journal*, 8(10), 2020, 7789–817.

[51] Q. Zhang, Y. Xu, J. Zhang, and D. Tao, "Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond," *International Journal of Computer Vision*, 2023.

[52] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," *IEEE Transactions on Knowledge and Data Engineering*, 34(12), 2022, 5586–609.

[53] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *2018 IEEE Conference on Computer vision and pattern recognition (CVPR)*, 2018, 8697–710.