Efficient Grasp Detection Network With Gaussian-Based Grasp Representation for Robotic Manipulation

Hu Cao, Guang Chen, *Member, IEEE*, Zhijun Li, *Fellow, IEEE*, Qian Feng, Jianjie Lin, and Alois Knoll, *Fellow, IEEE*

***Abstract*—Deep learning methods have achieved excellent results in the field of grasp detection. However, deep learning-based models for general object detection lack the proper balance of accuracy and inference speed, resulting in poor performance in real-time grasp tasks. This work proposes an efficient grasp detection network with n-channel images as inputs for robotic grasp. The proposed network is a lightweight generative structure for grasp detection in one stage. Specifically, a Gaussian kernel-based grasp representation is introduced to encode the training samples, embodying the maximum center that possesses the highest grasp confidence. A receptive field block is plugged into the bottleneck to improve the model's feature discriminability. In addition, pixel-based and channel-based attention mechanisms are used to construct a multidimensional attention fusion network to fuse valuable semantic information, achieved by suppressing noisy features and highlighting object features. The proposed method is evaluated on the Cornell, Jacquard, and extended OCID grasp datasets. The experimental results show that our method achieves excellent balancing accuracy and running speed performance. The network gets a running speed of 6 ms, achieving better performance on the Cornell, Jacquard, and extended OCID grasp datasets with 97.8, 95.6, and 76.4% accuracy, respectively. Subsequently, an excellent grasp success rate in a physical environment is obtained using the UR5 robot arm.**

***Index Terms*—Efficient grasp detection, fully convolutional neural network, Gaussian-based grasp representation (GGR), multidimension attention fusion, receptive field block (RFB).**

I. Introduction

Intelligent robots are crucial in human–robot cooperation, robot assembly, and robot welding [1]. The robots need an effective automated manipulation system to complete the task of picking and placing [2], [3]. However, grasping is a straightforward action for humans but challenging for robots because it involves perception, planning, and execution. Grasp detection is a crucial procedure for robots to perform grasp and manipulation tasks in the real world environment. Therefore, it is necessary to develop a robust perception algorithm to improve the performance of the robotic grasp.

Early grasp detection algorithms were mainly based on search algorithms. Unfortunately, these algorithms are inefficient in complex real-world scenarios [4]. Recently, deep learning-based approaches have achieved excellent results in robotic grasp detection [5], [6], [7], [8]. A five-dimensional (5-D) grasp configuration is proposed to represent a grasp rectangle based on two-dimensional (2-D) space projected into three-dimensional (3-D) space to guide the robot to grasp [5]. Due to the simplification of the grasp object dimension, the deep convolutional neural network can learn to extract more suitable features for specific tasks than hand-engineered features by taking 2-D images as inputs. According to the literature, training neural networks to predict grasp with the highest probability score from multiple grasp candidates is the best grasping result [9], [10], [11]. Currently, excellent general object detection models have been introduced in the grasp detection task, such as one-stage and two-stage deep learning methods [12], [13], [14]. Similarly, the idea of Faster R-CNN is to perform robotic grasp detection by taking RGB-D images as inputs [15]. While in [16] and [17], achieving excellent grasp detection accuracy is based on single-stage object detection methods, YOLO [12], and SSD [13]. However, these methods are challenging to balance accuracy and inference speed due to their complex network structures. The authors in [18], [19] improved the performance of grasp

Manuscript received 17 July 2022; revised 15 September 2022; accepted 29 October 2022. Recommended by Technical Editor C. Liu and Senior Editor N. G Tsagarakis. This work was supported in part by the National Natural Science Foundation of China under Grant 61906138, in part by Shanghai Municipal Science and Technology Major Project under Grant 2018SHZDZX01, in part by ZJ Laboratory, in part by the Shanghai Center for Brain Science and Brain-Inspired Technology, and in part by the Shanghai Rising Star Program under Grant 21QC1400900. *(Corresponding author: Guang Chen.)*

Hu Cao, Jianjie Lin, and Alois Knoll are with the Chair of Robotics, Artificial Intelligence and Real-time Systems, Technische Universität München, 80333 München, Germany (e-mail: hu.cao@tum.de; jianjie.lin@tum.de; knoll@in.tum.de).

Guang Chen is with the Tongji University, Shanghai 201804, China (e-mail: guangchen@tongji.edu.cn).

Zhijun Li is with the University of Science and Technology of China, Hefei 230026, China (e-mail: zjli@ieee.org).

Qian Feng is with the Chair of Robotics, Artificial Intelligence and Real-time Systems, Technische Universität München, 80333 München, Germany, and also with Agile Robots AG, 81477 Gilching, Germany (e-mail: qian.feng@tum.de).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2022.3224314.

Digital Object Identifier 10.1109/TMECH.2022.3224314

detection by employing an oriented anchor box mechanism to match the grasp rectangles. These methods have improved the detection accuracy, but the computational loads are still too large to be suitable for real-time applications.

A new grasp representation was proposed to solve these mentioned problems using the method of sampling grasp candidate rectangles, applying convolutional neural networks to regress grasp points directly [20]. This approach simplifies the definition of grasp representation and achieves high real-time performance based on lightweight architecture. Inspired by [20], the authors of [21], [22] use the key idea of algorithms in vision segmentation tasks to predict robotic grasp poses from extracted pixelwise features. Recently, the residual structure was introduced into the generated neural network model [8], achieving better grasp detection accuracy on the common grasp datasets. However, the shortcoming is the failure to highlight the importance of the largest grasp probability at the center point.

This work uses a 2-D Gaussian kernel to encode training samples to emphasize the highest grasp confidence score at the center point position. Based on Gaussian-based grasp representation (GGR), we developed a lightweight generative architecture for robotic grasp pose estimation. Referring to the human visual system's receptive field structure, the combination of residual and receptive field blocks (RFBs) in the bottleneck layer can enhance the feature's discriminability and robustness. Furthermore, low-level features and deep features in the decoder are fused to reduce the information loss caused in the sampling process. Specifically, a multidimensional attention network composed of pixel and channel attention networks is used to suppress redundant features and highlight significant object features in the fusion process. Experimental results demonstrate that the proposed algorithm achieves excellent performance in balancing accuracy and inference speed. The main contributions are summarized as follows:

1) We propose a GGR, reflecting the maximum grasp score at the center point location.
2) We developed an efficient generative architecture for robotic grasp detection.
3) A RFB is embedded in the network's bottleneck to enhance its feature discriminability and robustness. A multidimensional attention fusion network (MDAFN) has been developed to suppress redundant features and improve object features in the fusion process.
4) Experimental results demonstrate that the proposed method performs well on the public Cornell [23], Jacquard [24], and extended OCID [25] grasp datasets.

This work is an extension of a conference paper published at ICRA 2021 [26]. We improved and extended the conference version in several important aspects as follows:

1) We add more detailed illustrations about grasp representation.
2) We provide a new grasp detection algorithm to improve performance. Specifically, an RFB is embedded in the network's bottleneck to enhance the model's deep features, and an MDAFN is introduced to suppress noise features and highlight the object features.
3) We conduct additional experiments and analysis to provide a more elaborate presentation. To validate the effectiveness of our method on the more complex scenes, we further perform experiments on the extended OCID dataset and real robot experiments on three multiple object grasp scenes: a multiple object scene, an occluded object scene, and a cluttered object scene.

The rest of this article is organized as follows. The proposed robotic grasp system is introduced in Section II. A detailed description of the proposed grasp detection method is illustrated in Section III. Experiments based on the public grasp datasets are discussed in Section IV. Finally, Section V concludes this article.

II. Robotic Grasp System

This section gives an overview of the robotic grasp system settings and illustrates the principles of GGR.

A. System Setting

A robotic grasp system consists of a robot arm, perception sensors, grasping objects, and workspace. To complete the grasping task successfully, the subsystem of planning and control is involved along with the grasping pose of objects. In grasp detection, we consider limiting the manipulator to the normal direction of the workspace to become a goal for perception in 2-D space. Most graspable objects are flat through these settings by placing them reasonably on the workbench. As opposed to building 3-D point cloud data, the whole grasp system can reduce the cost of storage and calculation and improve its operational capacity. The grasp pose of flat objects can be treated as a rectangle. Since the size of each plate gripper is fixed, we use a simplified grasp representation mentioned in Section II-B to perform grasp pose estimation.

B. Gaussian-Based Grasp Representation

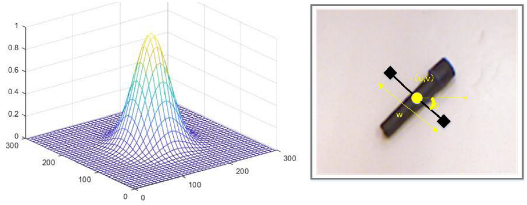
The grasp detection model should take RGB or depth images as inputs to generate grasp candidates for subsequent manipulation tasks. Works from literature built their grasp detection model for grasp pose prediction based on 5-D grasp representation [15], [18], [27]

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

where the center point is denoted as (x, y). θ is the grasp angle and (w, h) is the weight and height of the grasp rectangle, respectively. The 5-D grasp representation is borrowed from conventional object detection, but is not perfectly suited for robotic grasp detection. The simplified grasp representation introduced in [20] for fast robotic grasp detection can be formulated as in the following:

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

where $\mathbf{p}$ is the position of the center point expressed in Cartesian coordinates as $\mathbf{p} = (x, y, z)$. The φ and w denote the grasp angle and grasp width, respectively, and q is a scale factor for measuring the grasp quality. Furthermore, the new grasp



(a) (b)

Fig. 1. GGR: The 2-D Gaussian kernel is applied to the grasp quality map to highlight the max grasp quality of its central point position. (a) Schematic diagram of grasp quality weight distribution after 2-D Gaussian function deployment. (b) Schematic diagram of grasp representation.

representation in 2-D space is represented in the following:

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

where $\hat{p}$ is the center point in the image coordinates denoted as $\hat{p} = (u, v)$. $\hat{\varphi}$ represents the grasp angle in the camera frame. $\hat{w}$ and $\hat{q}$ denote the grasp width and the grasp quality, respectively. After obtaining the grasp system calibration results, matrix operations transform the grasp pose $\hat{g}$ into world coordinates g, as in the following:

$$g = T_{\text{RC}}(T_{\text{CI}}(\hat{g})) \tag{4}$$

where T_{RC} is the transformation matrix from camera frames to world frames and T_{CI} is the transformation matrix from 2-D image space to camera frames. The grasp map in image space is denoted in the following:

$$\mathbf{G} = \{\Phi, W, Q\} \in \mathbb{R}^{3\times W\times H} \tag{5}$$

where the pixels of grasp maps, Φ, W, Q, are filled with the corresponding values of $\hat{\varphi}, \hat{w}, \hat{q}$. The central location can be found by searching the pixel coordinate with the maximum grasp quality $\hat{g^*} = \max_{\hat{Q}} \hat{G}$. The authors in [20] filled a rectangular area around the center with 1, indicating the highest grasp quality and other pixels as 0. The training model learns the maximum grasp quality of the center. Because all pixels in the rectangular area have the best grasping quality, it leads to a limitation that the importance of the center point is not highlighted, resulting in ambiguity in the model. In this work, we use a 2-D Gaussian kernel to regularize the grasp representation to indicate where the object center might exist, as shown in Fig. 1. The novel GGR is represented as g_k. The corresponding Gaussian-based grasp map is defined in the following:

$$G_K = \{\Phi, W, Q_K\} \in \mathbb{R}^{3\times W\times H}$$

where

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

where

$$\sigma_x = T_x, \sigma_y = T_y. \tag{6}$$

In the abovementioned equation, the generated grasp quality map is decided by the center point location (x_0, y_0), the parameter σ_x and σ_y, and the corresponding scale factor T_x and T_y. In this method, the peak of the Gaussian distribution is the center coordinate of the grasp rectangle. This work discusses the detailed effects of parameter settings in the Section IV-F.

III. Method

In this section, we introduce a lightweight generative architecture for robotic grasp detection. Fig. 2 presents the overall structure of our grasp detection model. The input data are downsampled into feature maps with smaller sizes, more channels, and richer semantic information. ResNet [28] and the multiscale RFB are combined in the bottleneck to extract more discriminability and robustness features. Furthermore, an MDAFN consisting of pixel-based and channel-based attention subnetworks is used to fuse shallow and deep semantic features. The proposed model suppresses redundant features and enhances the object features during the fusion process based on the attention mechanism. Finally, based on the extracted features, four task-specific subnetworks are added to predict grasp quality, angle (in the form of sin(2θ) and cos(2θ)), and width, respectively. A detailed illustration of each component of the proposed grasp network is depicted in the following sections.

A. Basic Network Architecture

The proposed generative grasp architecture comprises of the down-sampling block, bottleneck layer, MDAFN, and upsampling block, as shown in Fig. 2. The down-sampling block consists of a convolutional layer with a kernel size of 3×3 and a maximum pooling layer with a kernel size of 2×2, which can be represented in the following:

$$x_d = f_{\text{maxpool}}(f_{\text{conv}}^n(f_{\text{conv}}^{n-1}(\ldots f_{\text{conv}}^0(I)\ldots))). \tag{7}$$

In this article, we use two down-sampling blocks and two convolutional layers in the down-sampling process. The first down-sampling block comprises four convolutional layers (n = 3) and one maximum pooling layer. The second down-sampling layer comprises two convolutional layers (n = 1) and one maximum pooling layer. After the downsampled data passes through two convolutional layers, it is fed into a bottleneck layer consisting of three residual blocks (k = 2) and one RFB to extract features. Since RFB comprises various scale convolutional filters, it is possible to acquire richer image details. More illustrations of RFB are presented in Section III-B. The output of the bottleneck can be formulated in the following:

$$x_b = f_{\text{RFBM}}(f_{\text{res}}^k(f_{\text{res}}^{k-1}(\ldots f_{\text{res}}^0(f_{\text{conv}}^1(f_{\text{conv}}^0(x_d)))\ldots))). \tag{8}$$

The output x_b of the bottleneck is fed into an MDAFN and upsampling block. The MDAFN composed of pixel attention and channel attention subnetworks can suppress the noise features and enhance the valuable features during the fusion of shallow features and deep features. The detailed illustration of the MDAFN is presented in Section III-C. In the upsampling block, the pixshuffle layer [29] increases feature resolution with the scale factor set to 2. In this work, the number of MDAFN and upsampling blocks is two, and the output is represented in

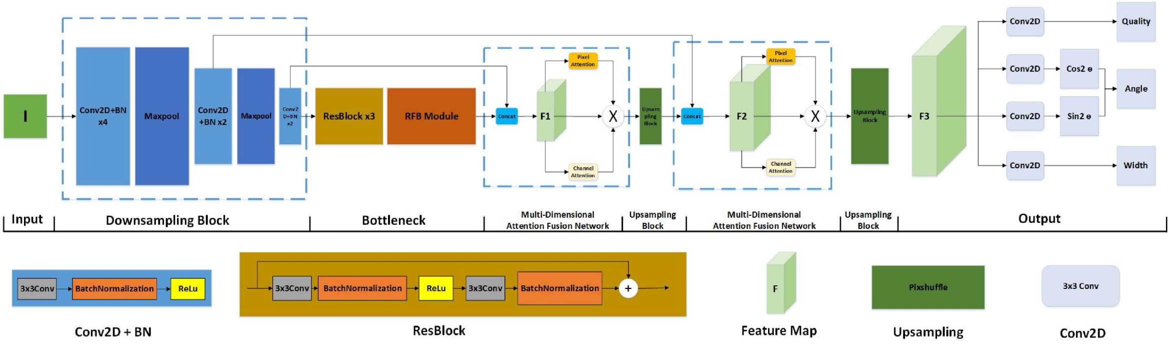


Fig. 2. Architecture of our generative grasping detection model. I and Conv denote the input data and convolution filter, respectively. The proposed method consists of the down-sampling block, the bottleneck layer, the MDAFN, and the upsampling block.

the following:

$$x_u = f^1_{\text{pixshuffle}}(f^1_{\text{MDAFN}}(f^0_{\text{pixshuffle}}(f^0_{\text{MDAFN}}(x_b)))). \tag{9}$$

The final layer consists of four convolutional filters with a kernel size of 3×3. The corresponding outputs can be expressed in the following:

$$\begin{aligned} g_q &= \max_q(f^0_{\text{conv}}(x_u)) \\ g_{\cos(2\theta)} &= \max_q(f^1_{\text{conv}}(x_u)) \\ g_{\sin(2\theta)} &= \max_q(f^2_{\text{conv}}(x_u)) \\ g_w &= \max_q(f^3_{\text{conv}}(x_u)) \end{aligned} \tag{10}$$

where the center point is located by searching the pixel coordinate with the highest grasp quality g_q. g_w denotes the grasp width, and the grasp angle is calculated by $g_{\text{angle}} = \arctan\left(\frac{g_{\sin(2\theta)}}{g_{\cos(2\theta)}}\right)/2$.

B. Multiscale RFB

In neuroscience, researchers have discovered a particular function in the human visible cortex that regulates the size of the visible receptive area [30], [31]. This mechanism can help to emphasize the importance of the area near the center. For robotic grasping tasks, multiscale receptive fields can enhance the neural network's deep features. We hope to enhance the model's receptive field to improve its feature extraction capability for multigrasp objects. In this work, we introduce a multiscale RFB [32] to assemble the bottleneck layer to improve the model's receptive field capability. The RFB comprises multibranch convolutional layers with different kernels corresponding to the receptive fields of various sizes. The dilated convolution layer controls the eccentricity, and the features extracted by the branches of the different receptive fields are recombined to form the final representation, as shown in Fig 3. In each branch, the convolutional layer follows a dilated convolutional layer. The kernel sizes are a combination of (1×1, 3×3, 7×1, 1×7). The features extracted from the four branches are concatenated and then added to the input data to obtain the final multiscale feature output.

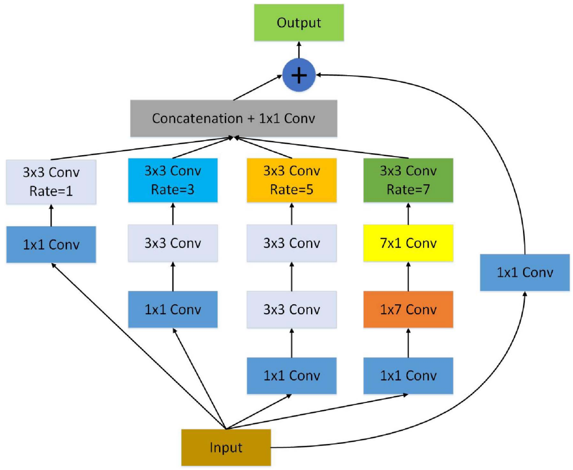


Fig. 3. Multiscale RFB. The RFB consists of multibranch convolutional layers with different kernels corresponding to the receptive fields of various sizes.

C. Multidimensional Attention Fusion Network

When humans look at an image, not much attention is paid to everything; instead, more focus is paid to what is interesting. In computer vision, attention mechanisms with few parameters, fast speed, and excellent effects have been developed [33], [34], [35], [36], [37]. The motivation for MDAFN is to effectively perceive grasping objects against a complex background. This attention mechanism can suppress the noise features and highlight the object features. As shown in Fig. 4, the shallow and deep features are concatenated together. The concatenated features are fed into MDAFN to perform representation learning at pixel-level and channel-level. The feature map F passes through a 3×3 convolution layer in the pixel attention subnetwork to generate an attention map by a convolution operation. The attention map is computed with a sigmoid to obtain the corresponding pixelwise weight score. SENet [34] is then used as the channel attention subnetwork, which accepts 1×1×C features through global average pooling. It then uses two feedforward layers and the corresponding activation function ReLU to build the correlation between channels and finally outputs the weight score of the feature channel through the sigmoid operation. Both pixelwise and channelwise weight maps are multiplied with the

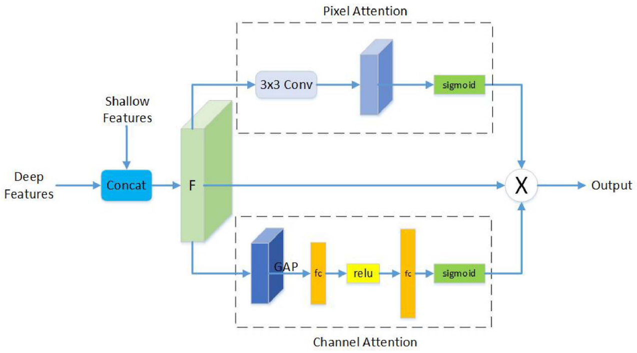


Fig. 4. MDAFN. The top branch is the pixel-level attention subnetwork, and the bottom is the channel-level attention subnetwork.

feature map F to obtain a novel output with reduced noise and enhanced object information.

D. Loss Function

The neural network model can be considered as a method to approximate the complex function $F : I \longmapsto \hat{G}$ for input images $I = \{I_1 ... I_n\}$ and corresponding grasp labels $L = \{L_1 ... L_n\}$. F is the proposed grasp model and I is the input image. $\hat{G}$ denotes the grasp prediction. Specifically, the model is trained on the dataset to learn the grasp detection function F by optimizing minimum errors between grasp predictions $\hat{G}$ and the corresponding labels L. This task is a regression problem, so the Smooth L1 loss is deployed as the loss function to optimize our model. The loss function L_r is formulated in the following:

$$L_r(\hat{G}, L) = \sum_{i}^{N} \sum_{m \in \{q, \cos 2\theta, \sin 2\theta, w\}} \mathrm{Smooth}_{L1}(\hat{G}_i^m - L_i^m) \tag{11}$$

where Smooth_{L1} is defined as

$$\mathrm{Smooth}_{L1}(x) = \begin{cases} (\sigma x)^2/2, & \text{if } |x| < 1 \\ |x| - 0.5/\sigma^2, & \text{otherwise} \end{cases}$$

where N represents the count of grasp candidates, the grasp angle is defined as the form of $(\cos(2\theta), \sin(2\theta))$. q, w denote the grasp quality and grasp width, respectively. σ is the hyperparameter in the Smooth_{L1} function, which controls the smooth area.

IV. Experiment

To verify the generalization capability of the proposed lightweight generative model, we conducted experiments on three public grasp datasets, Cornell [23], Jacquard [24], and extended OCID [25], [38]. Experimental results indicate that the proposed algorithm has high inference speed while achieving high grasp detection accuracy. In addition, we further explore the impact of different network designs on algorithm performance and discuss the shortcomings of the proposed method.

A. Evaluation Metrics

Similar to previous works [8], [15], [26], the metric used in this article to evaluate our model on the Cornell [23], Jacquard [24], and extended OCID [25] grasp datasets is the rectangle metric. Specifically, a predicted grasp is regarded as a correct grasp when it meets the following two conditions.

- *Angle difference:* The difference of orientation angle between the predicted grasp and corresponding grasp label is less than 30°.
- *Jaccard index:* The Jaccard index of the predicted grasp and corresponding grasp label is greater than 25%, which can be formulated in the following:

$$J(g_p, g_l) = \frac{|g_p \cap g_l|}{|g_p \cup g_l|} \tag{12}$$

where g_p and g_l denote the predicted grasp rectangle and the area of the corresponding grasp label, respectively. $g_p \cap g_t$ represents the intersection of the predicted grasp and the corresponding grasp label. The union of predicted grasp and the corresponding grasp label is represented as $g_p \cup g_t$.

B. Implementation Details

Data preprocessing: The experiments for this work were performed on the Cornell [23], Jacquard [24], and extended OCID [25] grasp datasets. Due to the small data size of Cornell and OCID, online data augmentation is conducted to train the network. Specifically, random crops, zooms, and rotations are used to improve the diversity of the Cornell and OCID grasp datasets. Meanwhile, the Jacquard dataset has sufficient data and the network is trained directly without any data augmentation. In addition, the data labels are encoded for training. A 2-D Gaussian kernel is used to encode each ground-truth positive grasp so that the corresponding region satisfies the Gaussian distribution, where the peak of the Gaussian distribution is the coordinate of the center point. We also use $\sin(2\theta)$ and $\cos(2\theta)$ to encode the grasp angle, where $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. The resulting corresponding values range from –1 to 1. Using this method, ambiguity can be avoided in the angle learning process, which is beneficial to the convergence of the network. Similarly, the grasp width is scaled to a range of 0–1 during the training.

Training configuration: The grasp network is achieved using Pytorch 1.7.0 with Cudnn-7.5 and Cuda-10.0 packages. During the training period, the model is trained end-to-end on an Nvidia RTX2080Ti GPU with 11GB of memory.

C. Experiments on the Cornell Grasp Dataset

The images of the Cornell dataset are resized to 224×224 to feed into the network. Following [20], an imagewise split method is used to test our network, where the images of the dataset are randomly divided and the images of each object in the training set and test set are different. The average of the 5-fold cross-validation is used as the final results.

Training schedule: The famous Adam optimizer [39] is used to optimize the network for backpropagation during the training

TABLE I
EVALUATION RESULTS (%) OF DIFFERENT METHODS ON THE CORNELL DATASET

Author	Method	Input modality	Input size	Accuracy(%)	Time (ms)
Jiang‡ [23]	Fast Search	RGB-D	227 × 227	60.5	5000
Lenz† [5]	SAE	RGB-D	227 × 227	73.9	1350
Chu† [15]	FasterR-CNN	RGB	227 × 227	96.0	120
Zhang† [40]	Multimodal Fusion	RGB-D	224 × 224	88.9	117
Zhou‡ [18]	FCGN	RGB	320 × 320	97.7	117
Redmon† [9]	AlexNet, MultiGrasp	RGB-D	224 × 224	88.0	76
Kumra‡ [11]	ResNet-50	RGB-D	224 × 224	89.2	103
Kumra* [8]	GR-ConvNet	RGB-D	300 × 300	97.7	7
Asif† [41]	GraspNet	RGB-D	224 × 224	90.6	24
Morrison* [20]	GGCNN	D	300 × 300	73.0	4
Wang† [21]	GPWRG	D	400 × 400	94.4	8
Ours	Efficient Grasp	D	224 × 224	94.6	6
		RGB		95.3	6
		RGB-D		**97.8**	6

Runtime results for the methods †are referred to in [21], the runtime results for the methods ‡are referred to in [8], and the runtime results for the methods * are tested by ourselves.

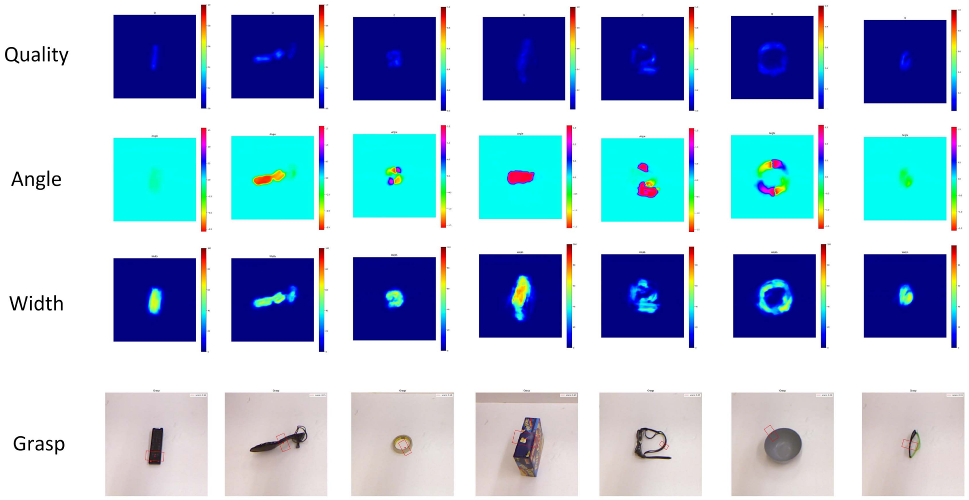


Fig. 5. Detection results of the grasp network on the Cornell dataset. The first three rows are the grasp quality, angle, and width maps. The last row is the best grasp output for several objects.

process. The initial learning rate is defined as 0.001 and the batch size is set as 8. The network is trained for a total of 50 epochs to get the final training weights.

Results: The comparison of the grasp detection accuracy of our model and other methods on the Cornell dataset [23] is presented in Table I. Since the grasping scene in the Cornell dataset is simple (single object grasping scene), the proposed grasp detection model achieves high detection accuracy of 97.8% with an inference time of 6ms. The model maintains better accuracy and running speed performance than other state-of-the-art algorithms. By changing the mode of input data, the generated grasp detection architecture achieves excellent performance with the input of depth data. The results demonstrate that the combination of depth data and RGB data with rich color and texture information enables the model to have a more robust generalization ability to unseen objects. Fig. 5 shows the plot of the grasp detection results of some objects for display. Only the grasp prediction with the highest quality score is selected as the final output, and the top-1 grasp is visualized in the last row. The first three rows are the grasp quality, angle, and width maps. It can be seen that the proposed algorithm provides reliable grasp candidates for objects with different shapes and poses.

D. Experiments on the Jacquard Grasp Dataset

The images of the Jacquard dataset are resized to 300×300 to feed into the network. We use an imagewise split to test our network, where 90% of the data are used as a training set, and the rest of the data are used as a test set.

Training schedule: Similar to training the network on the Cornell dataset, we train the model end-to-end on the Jacquard dataset with a learning rate of 0.001 and a batch size of 8. Adam [39] is used as the default optimizer. Since the data size of the Jacquard dataset is larger than the Cornell dataset, the network is trained for a total of 150 epochs to get the final training weights.

Results: Similarly, the network is trained on the Jacquard dataset [24] to perform grasp pose estimation. The results are

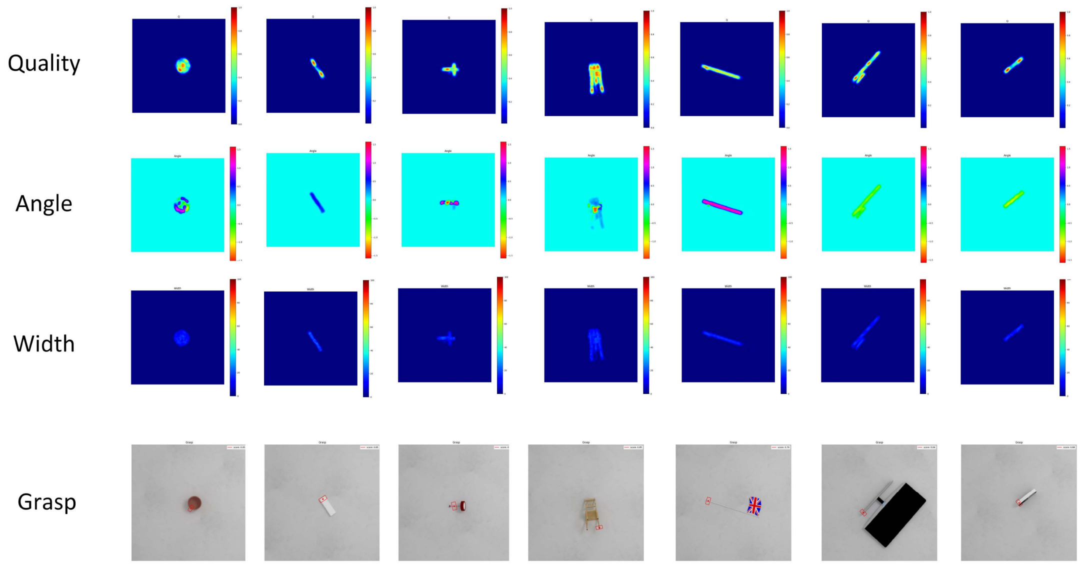


Fig. 6. Detection results of the grasp network on the Jacquard dataset. The first three rows are the grasp quality, angle, and width maps. The last row is the best grasp output for several objects.

TABLE II
EVALUATION RESULTS (%) OF DIFFERENT METHODS ON THE JACQUARD DATASET

Author	Method	Accuracy(%)	Time(ms)
Depierre [24]	Jacquard	74.2	-
Morrison* [20]	GG-CNN2	84.0	4
Kumra* [8]	GR-ConvNet	94.6	7
Ours	Efficient Grasp-D	**95.6**	6
	Efficient Grasp-RGB	91.6	6
	Efficient Grasp-RGB-D	93.6	6

The runtime results for the methods * are tested by ourselves.

TABLE III
EVALUATION RESULTS (%) OF DIFFERENT METHODS ON THE EXTENDED OCID DATASET

Author	Method	Accuracy(%)	Time(ms)
Stefan* [25]	Det_Seg	89.0	22
Morrison* [20]	GG-CNN2	63.4	4
Kumra* [8]	GR-ConvNet	74.1	7
Ours	Efficient Grasp-D	72.7	6
	Efficient Grasp-RGB	74.7	6
	Efficient Grasp-RGB-D	76.4	6

The runtime results for the methods * are tested by ourselves.

summarized in Table II. Taking depth data as input, the proposed approach obtains excellent performance with a detection accuracy of 95.6%, which exceeds the existing methods and reaches the best result on the Jacquard dataset. The experimental results in Table I and Table II demonstrate that our algorithm achieves excellent performance on the Cornell grasp dataset and outperforms other methods on the Jacquard grasp dataset. Detection examples are displayed in Fig. 6. Specifically, grasp quality, angle, width, and the best detection results are presented in the figure.

E. Experiments on the OCID Grasp Dataset

The images of the extended OCID dataset are resized to 224×224 to pass through the network. The imagewise method is used to split the dataset. Specifically, 1411 selected images are divided into training set and 352 selected images are used as test set. We report the average of the 5-fold cross-validation as the final results.

Training schedule: The network is trained end-to-end on the extended OCID dataset with a learning rate of 0.001 and a batch size of 8. Adam [39] is used as the default optimizer, and the network is trained for a total of 400 epochs to get the final training weights.

Results: To verify the effectiveness of the proposed method on the complexity scenes, we test our method on the extended OCID [25] grasp dataset. The experimental results are shown in Table III. The grasp detection accuracy of our method is better than contact point-based methods [8], [20] and the running speed of our method is faster than detection-based method [25]. Our method provides an excellent balance between accuracy and speed.

Objects in clutter: To validate the generalization ability of the proposed method in the cluttered scene, the model trained on the Cornell dataset is used to test it in a more realistic multiobject environment. The detection results are the first two rows presented in Fig. 7. The model is trained on a single object dataset but can still predict the grasp pose of multiple objects. Moreover, the last two rows presented in Fig. 7 are the test results of our model trained on the extended OCID dataset. The results show that the proposed method can simultaneously output grasp poses of various objects in complex scenarios.

F. Ablation Study

Influence of the different components: To further explore the impact of different components on grasp pose learning, we trained our models with varying network settings on the Cornell

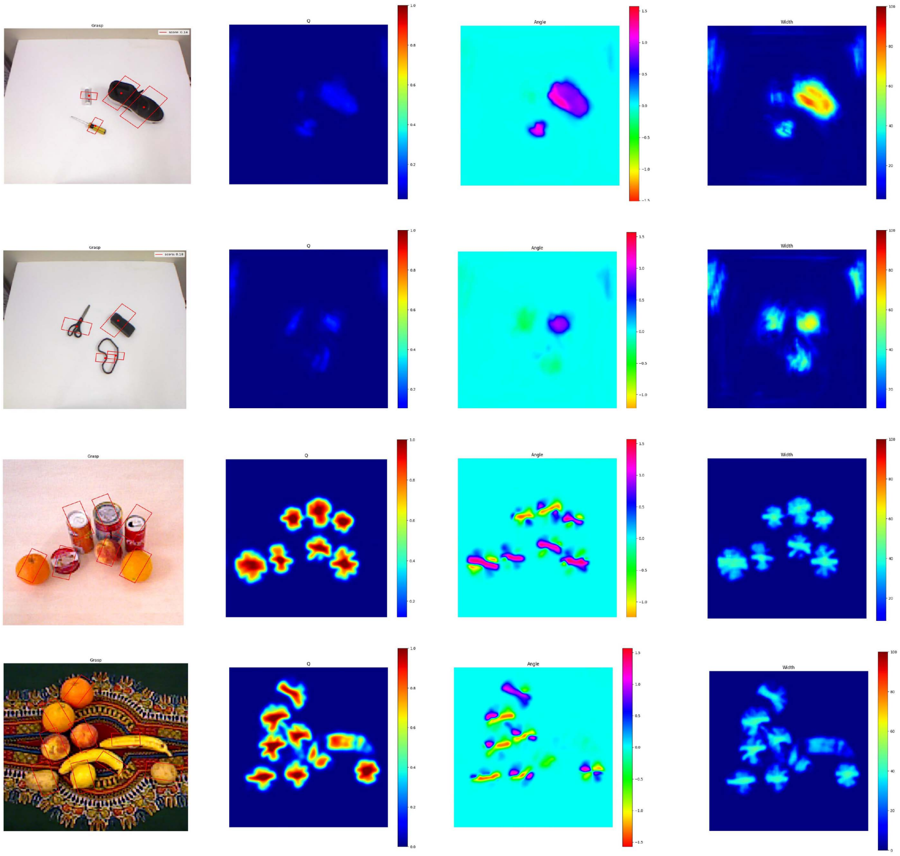


Fig. 7. Multiple grasped object detection results The first column is the grasp outputs of corresponding RGB images for several objects. The last three columns are the maps for grasp quality, angle, and width.

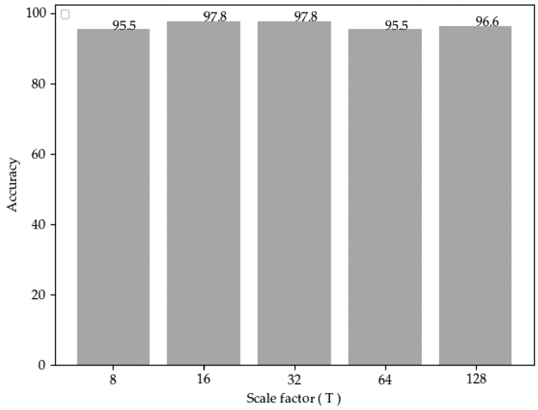


Fig. 8. Grasp detection accuracy when using different scale factors of the Gaussian kernel.

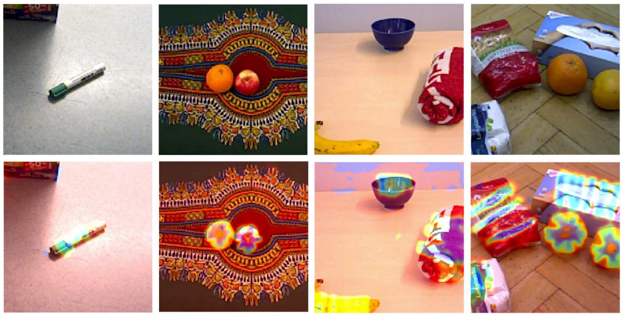


Fig. 9. Visualization of feature heatmaps.

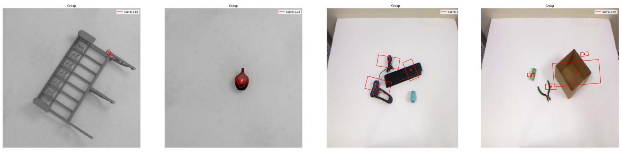


Fig. 10. Failed detection cases with single and multiple objects.

dataset [23] with RGB-D data as input. The experimental results are summarized in Table IV. It can be obtained from the detection accuracy evaluation results in the Table IV that GGR, RFB, and MDAFN can all bring performance improvement to the network, and all components combined can get the best grasp detection performance.

Effect of the scale factor: We also discuss the impact of different scale factor settings (T) on the model, as shown in Fig. 8. In this work, the scale factors T_x and T_y mentioned in Section II-B are set to $Tx = Ty = T$ with values ranging from $\{8, 16, 32, 64, 128\}$. When $T = 32$, the model training on the Cornell dataset reaches the best detection accuracy of 97.8. During the experiment, it is found that the different densities

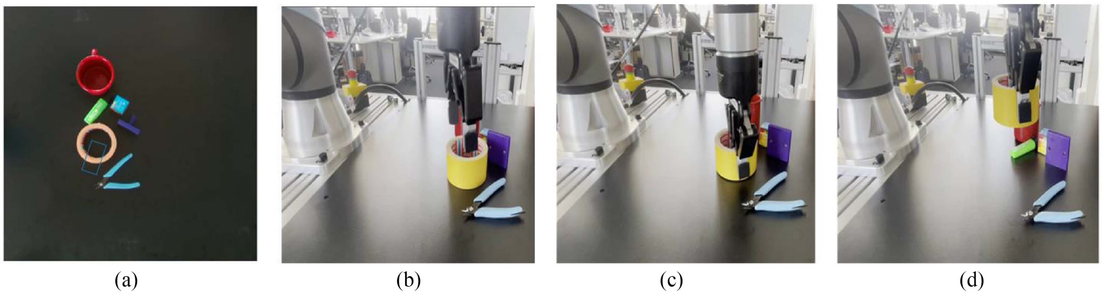

(a) (b) (c) (d)

Fig. 11. Process of physical grasp experiments. (a) Grasp detection output. (b) Robot approaches the object. (c) Robot grasps the object. (d) Robot completes the successful grasp.

TABLE IV
IMPACT OF DIFFERENT NETWORK SETTINGS ON DETECTION PERFORMANCE

+ GGR	✓				✓	✓	✓
+ RFB		✓		✓		✓	✓
+ MDAFN			✓	✓	✓		✓
Acurracy (%)	94.4	95.5	95.5	96.6	95.5	96.6	**97.8**

TABLE V
EFFICIENCY COMPARISON OF DIFFERENT METHODS (APPROX)

Methods	**Params**	**FLOPs**	**Time (GPU)**	**Time (CPU)**
Levine† [42]	1M	–	0.2–0.5s	–
Morrison* [20]	70.6 k	1.0G	4ms	57ms
Kumra* [8]	1.9M	10.9G	7ms	473ms
Ours	1.2M	5.7G	6ms	86ms

The results for the methods †are referred to in [20]. The results for the methods * are tested by ourselves.

TABLE VI
EXPERIMENTAL RESULTS FOR THE DIFFERENT GRASP SCENES

Scenes	**Successes**	**Total Grasps**	**Grasp Success Rate(%)**
Single object	94	100	94.0
Multiple object	142	156	91.0
Occluded object	114	128	89.1
Cluttered object	122	142	85.9

of the annotation for a particular dataset should be set to the size of the corresponding scale factor value, which can slow the instability of the network learning caused by labels' overlap.

Comparison of network efficiency: In Table V, parameters, FLOPs, and the model's inference time (GPU and CPU) are used as efficiency evaluation metrics. To improve the real-time performance of the grasp algorithm, we developed a lightweight generative grasp detection architecture that achieves better detection accuracy and faster running speed. The experimental results show that the proposed method achieves excellent efficiency when executed on both GPU and CPU hardware.

Feature visualization: To help better understand the effectiveness of the proposed grasp model, we visualized the heatmaps of the feature maps, as shown in Fig. 9. The first row is the original images selected from the extended OCID dataset, and the second row is the corresponding heatmap visualization results of the feature maps. As can be seen from the figure, our grasp model can effectively focus on the object while suppressing unimportant background information.

Failure cases discussion: The experimental results show that the proposed method achieved excellent detection performance but still had some cases of detection failure, as shown in Fig. 10. The model does not work well for objects with complex shapes. Furthermore, in the clutter scenes, smaller objects among multiple objects are often missed by the model, and the detection quality of the model for large boxes is relatively insufficient. However, these shortcomings can be alleviated by adding more challenging data to the training set.

G. Verification on Real Robot

To evaluate the efficiency of the proposed model, a Universal Robot 5 (UR5) attached to a Robotic Gripper 2F-85 is chosen as our experimental instrument. The UR5 offers a real-time data exchange interface with an update rate of 8ms, making it possible to achieve the real-time properties. We deploy the robotic library [43] as our primary platform to communicate with the robot. Furthermore, to build a compact system, the OPC-UA mechanism is integrated into the robotic library so that the camera can publish the images to the component, which can further utilize this information for object detection. Together with the OPC-UA, the robotic library shares a similar structure as ROS1, but much faster, since ROS1 lacks real-time properties. The whole experimental process is illustrated in Fig. 11.

We use the Intel Realsense camera to perceive the environment. The output of the Realsense camera will be fed to the proposed network, which can generate a bunch of grasp configurations, and a final grasp configuration will be selected based on our predefined criteria (the grasp candidate with the highest grasp quality score is selected as the final grasp configuration.) The coordinate transformation is necessary to apply the grasp configuration described in the image coordinate. After the transformation, the grasp configuration in the world coordinate is specified.

As a consequence, the robotic arm joints can be calculated using the analytical inverse kinematic approach. Therefore, a trajectory in joint space can be generated using the trajectory planning block from the robotic library. The novel objects are evaluated, with different and complex shapes. We summarize the results of single object grasp scene in Table VI and indicate the effectiveness of our method with the 94% grasp success rate.

To further test the performance of our method on more complex scenes, we performed real robot experiments on three multiple object grasp scenes: a multiple object scene, an occluded object scene, and a cluttered object scene. The robot attempts multiple grasps until all objects are grasped, and then grasped objects are removed. As shown in Table VI, our method has a grasping success rate of 91.0, 89.1, and 85.9% on the multiple object scene, occluded object scene, and cluttered object scene, respectively.

V. Conclusion

In this article, we introduced GGR to highlight the maximum grasp quality at the center position. Based on GGR, a lightweight generative architecture with an RFB and an MDAFN was developed for grasp pose estimation. Experiments on three common datasets, the Cornell [23], Jacquard [24], and extended OCID [25] datasets, demonstrate that the proposed method achieves a fast running speed of 6 ms while having an excellent grasp detection accuracy of 97.8, 95.6, and 76.4%. In the physical grasp experiment, the proposed method achieves good performance with the application of the UR5 robot arm and robotic gripper.

References

[1] G. Du, K. Wang, and S. Lian, "Vision-based robotic grasping from object localization, pose estimation, grasp detection to motion planning: A review," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1677–1734, 2021.

[2] S. Liu, F. Wang, Z. Liu, W. Zhang, Y. Tian, and D. Zhang, "A two-finger soft-robotic gripper with enveloping and pinching grasping modes," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 1, pp. 146–155, Feb. 2021.

[3] K. Wen and C. Gosselin, "Static model based grasping force control of parallel grasping robots with partial cartesian force measurement," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 2, pp. 999–1010, Apr. 2022.

[4] F. T. Pokorny, Y. Bekiroglu, and D. Kragic, "Grasp moduli spaces and spherical harmonics," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 389–396.

[5] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4–5, pp. 705–724, 2015.

[6] L. Chen, P. Huang, Y. Li, and Z. Meng, "Edge-dependent efficient grasp rectangle search in robotic grasp detection," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 6, pp. 2922–2931, Dec. 2021.

[7] B. Li, H. Cao, Z. Qu, Y. Hu, Z. Wang, and Z. Liang, "Event-based robotic grasping detection with neuromorphic vision sensor and event-grasping dataset," *Front. Neurorobot.*, vol. 14, p. 51, 2020.

[8] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9626–9633.

[9] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, Seattle, 2015, pp. 1316–1322.

[10] U. Asif, J. Tang, and S. Harrer, "Densely supervised grasp detector (DSGD)," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 8085–8093, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4816

[11] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 769–776.

[12] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2016, *arXiv:1612.08242*.

[13] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[15] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3355–3362, Oct. 2018.

[16] G. Wu, W. Chen, H. Cheng, W. Zuo, D. Zhang, and J. You, "Multi-object grasping detection with hierarchical feature fusion," *IEEE Access*, vol. 7, pp. 43884–43894, 2019.

[17] D. Park, Y. Seo, and S. Y. Chun, "Real-time, highly accurate robotic grasp detection using fully convolutional neural networks with high-resolution images," 2018, *arXiv:1809.05828*.

[18] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7223–7230.

[19] Y. Song, L. Gao, X. Li, and W. Shen, "A novel robotic grasp detection method based on region proposal networks," *Robot. Comput.- Integr. Manuf.*, vol. 65, 2020, Art. no. 101963. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736584519308105

[20] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Robot. Res.*, vol. 39, no. 2–3, pp. 183–201, 2020.

[21] S. Wang, X. Jiang, J. Zhao, X. Wang, W. Zhou, and Y. Liu, "Efficient fully convolution neural network for generating pixel wise robotic grasps with high resolution images," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2019, pp. 474–480.

[22] D. Wang, "SGDN: Segmentation-based grasp detection network for unsymmetrical three-finger gripper," 2020, *arXiv:2005.08222*.

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

[24] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3511–3516.

[25] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from RGB," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13452–13458.

[26] H. Cao, G. Chen, Z. Li, J. Lin, and A. Knoll, "Residual squeeze-and-excitation network with multi-scale spatial pyramid module for fast robotic grasping detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13445–13451.

[27] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1609–1614.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[29] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.

[30] S. Liu, D. Huang, and Y. Wang, in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 385–400.

[31] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

[32] S. Liu, D. Huang, and A. Wang, "Receptive field block net for accurate and fast object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 385–400.

[33] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[35] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[36] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 510–519.

[37] H. Cao, G. Chen, J. Xia, G. Zhuang, and A. Knoll, "Fusion-based feature attention gate component for vehicle detection based on event camera," *IEEE Sensors J.*, vol. 21, no. 21, pp. 24540–24548, Nov. 2021.

[38] M. Suchi, T. Patten, D. Fischinger, and M. Vincze, "EasyLabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 6678–6684.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations (Poster)*, 2015, *arXiv:1412.6980*.

[40] Q. Zhang, D. Qu, F. Xu, and F. Zou, "Robust robot grasp detection in multimodal fusion," in *Proc. MATEC Web Conf.*, 2017, vol. 139, Art. no. 00060, doi: 10.1051/matecconf/201713900060.

[41] U. Asif, J. Tang, and S. Harrer, "GraspNet: An efficient convolutional neural network for real-time grasp detection for low-powered devices," in *Proc. 27th Int. Joint Conf. Artif. Intell., Int. Joint Conf. Artif. Intell. Org.*, 2018, pp. 4875–4882.

[42] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4-5, pp. 421–436, 2018.

[43] M. Rickert and A. Gaschler, "Robotics library: An object-oriented approach to robot applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 733–740.

Hu Cao received the M.Eng. degree in vehicle engineering from Hunan University, Changsha, China, in 2019. He is currently working toward the Ph.D. degree in computer science with the Chair of Robotics, Artificial Intelligence, and Real-time Systems, Technische Universität München, München, Germany.

His research interests include computer vision, neuromorphic engineering, robotics, and deep learning.

Mr. Cao is a member of the Informatics-6.

Guang Chen (Member, IEEE) received the B.S. and M.Eng. degrees in mechanical engineering from Hunan University, Changsha, China, in 2008 and 2011, respectively, and the Ph.D. degree from the Faculty of Informatics, Technical University of Munich, Munich, Germany, in 2016.

He is currently a Research Professor with Tongji University, Shanghai, China, and a Senior Research Associate (Guest) with the Technical University of Munich. His research interests include computer vision, image processing and machine learning, and the bioinspired vision with applications in robotics and autonomous vehicle. Dr. Chen was a recipient of the program of Tongji Hundred Talent Research Professor 2018.

Zhijun Li (Fellow, IEEE) received the Ph.D. degree in mechatronics from Shanghai Jiao Tong University, Shanghai, China, in 2002.

From 2003 to 2005, he was a Postdoctoral Fellow with the Department of Mechanical Engineering and Intelligent systems, The University of Electro-Communications, Tokyo, Japan. From 2005 to 2006, he was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, and with Nanyang Technological University, Singapore. Since 2017, he has been a Professor with the Department of Automation, University of Science and Technology, Hefei, China. Since 2019, he has been the Vice Dean of School of Information Science and Technology, University of Science and Technology. His research interests include wearable robotics, tele-operation systems, nonlinear control and neural network optimization.

Dr. Li has been the Co-Chairs of IEEE SMC Technical Committee on Bio-mechatronics and Bio-robotics Systems (B2S), and IEEE RAS Technical Committee on Neuro-Robotics Systems. He is an Editor-at-Large of Journal of Intelligent & Robotic Systems, and an Associate Editor for several IEEE Transactions.

Qian Feng received the B.S. degree in mechatronics from Zhejiang University, Hangzhou, China, in 2015, and the M.S. degree in mechatronics and information technology from the Technical University of Munich, Munich, Germany, in 2019. He is currently a Ph.D. degree in computer science candidate with the Chair of Robotics, Artificial Intelligence and Real-time Systems, Technical University of Munich.

He is a Research Scientist at Agile Robots AG, Gilching, Germany. His research interests include computer vision, robotic grasping, deep learning, and tactile sensing.

Jianjie Lin received the B.Sc. degree in electrical engineering and information technology from the Technical University of Kaiserlautern, Kaiserlautern, Germany, in 2015, and the master's degree in electrical engineering and information technology from the Technical University of Munich, Munich, Germany, in 2017 where he has been working toward the Ph.D. degree in computer science.

He was a Research Scientist with fortiss GmbH, Research Institute of the Free State of Bavaria for software-intensive systems. His research interests include deep learning algorithms, computer vision in 3-D point cloud and trajectory planning, and grasp planning.

Alois Knoll (Fellow, IEEE) received the diploma (M.Sc.) degree in electrical/communications engineering from the University of Stuttgart, Stuttgart, Germany, in 1985, and the Ph.D. (summa cum laude) degree in computer science from the Technical University of Berlin, Berlin, Germany, in 1988.

He was with the faculty of the Computer Science department of TU Berlin until 1993. He was with the University of Bielefeld, as a Full Professor and the Director of the research group Technical Informatics until 2001. Since 2001, he has been a Professor with the Department of Informatics, Technical University of Munich (TUM), Munich, Germany. He was also on the board of directors of the Central Institute of Medical Technology at TUM (IMETUM). From 2004 to 2006, he was an Executive Director with the Institute of Computer Science, TUM. His research interests include cognitive, medical and sensor-based robotics, multiagent systems, data fusion, adaptive systems, multimedia information retrieval, model-driven development of embedded systems with applications to automotive software and electric transportation, as well as simulation systems for robotics and traffic.