Real-Time Grasp Detection Using Efficient Channel Attention

Ribo Lan[1], Yong Xu[1], Manping Qin[1], Yuanji Chen[1], Guanyuan Ming[1], and Kui Fu[1,2(✉)]

[1] School of Artificial Intelligence and Smart Manufacturing, Hechi University, Yizhou 546300, China
kui13968361888@163.com
[2] Guilin University of Electronic Technology, Guilin 541004, China

Abstract. To realize the intelligence of robots, robots need to have the ability to grasp unknown objects. In this paper, a novel grasp detection network is proposed, named Efficient Channel Attention Grasp Network (ECA-GraspNet). ECA-GraspNet uses an encoder-decoder architecture to fuse feature information from different layers using a Feature Pyramid Network (FPN). The network is able to actively focus on convolutional layers that are more useful for grasp detection by embedding Efficient Channel Attention (ECA). The proposed network is able to generate pixel-by-pixel grasp poses from RGB-D images at real-time speed. We evaluate the proposed network on the public Cornell dataset and achieve an accuracy of 97.7%.

Keywords: Grasp Detection · Efficient Channel Attention · Feature Pyramid Network

1 Introduction

Robotic grasping technology is widely used in factory assembly, agriculture, and logistics. For intelligent robots, the primary task is to use sensors to perceive the environment and predict graspable poses. However, it has been a challenge for robots to predict the grasp pose of unknown objects in unstructured environments. Traditional methods utilize analytical approaches, such as computing graspable poses through geometry, kinematics, and dynamics [1]. However, these methods heavily depend on the shape of the object to be grasped and require a lot of computation. In this paper, we mainly focus grasp detection algorithm is applied on planar unknown objects.

With the development of deep learning technology, most studies have proposed grasp detection methods based on neural networks. These methods can be simply divided into classification-based methods [2–5] and detection-based methods [6, 7]. Classification-based methods first generate a large number of candidate grasp poses from the input data, and then filter out feasible grasps. Such methods usually have high accuracy. Due to real-time limitations, it is difficult for most studies to generate dense grasp poses. Detection-based approaches utilize convolutional neural networks (CNNs) to generate

Y. Zhang et al. (Eds.): CENet 2023, LNEE 1126, pp. 92–99, 2024.
https://doi.org/10.1007/978-981-99-9243-0_10

grasps end-to-end from input data. These methods build lightweight grasp detection models by stacking multiple convolutional layers, which can quickly generate graspable poses. However, the lightweight design affects the detection accuracy of the model.

In order to keep the model lightweight and further improve the accuracy of model detection, this paper proposes a novel grasp detection network ECA-GraspNet. ECA-GraspNet adopts the encoder-decoder structure, and uses the Feature Pyramid Network (FPN) [8] to fully utilize the feature information extracted by different layers. The proposed grasp detection network embeds Efficient Channel Attention (ECA) [9], focusing on convolutional layers that are more useful for grasp detection. Compared with existing work [6, 7], the proposed network uses FPN to make full use of the feature information extracted by different layers, and uses ECA to make the network focus on features useful for grasp detection.

2 Related Work

For robotic grasping, the primary task is to estimate the grasp poses of objects to be grasped from the input data. Jiang et al. [11] proposed to utilize a 7-dimensional grasp rectangle to represent the configuration, including 3D position, 3D orientation and required grasp width. They use a two-stage SVM to find grasp rectangles from the input data. With the development of deep learning technology, some studies have begun to use neural networks to detect grasp rectangles. Lenz et al. [10] first used deep learning methods to detect grasp rectangles, and they used a sliding window detector to estimate. Although sliding window can improve the accuracy of grasp detection compared with SVM, it requires a lot of computational operations. To improve the accuracy and speed of grasp detection, other studies also use direct regression grasp detectors. Song et al. [5] used a region proposal network to regress grasp rectangles from RGB images and predict their categories. Chen et al. [4] used a densely connected feature pyramid network feature extractor and multiple two-stage detection units for predicting dense grasp poses. Their experimental results show that the proposed method is real-time, but the detection accuracy needs to be further improved.

Morrison et al. [6] designed a generative grasp detector, which can realize real-time grasp detection through a convolutional neural network. They employ a generative grasp detection method to directly predict pixel-wise grasp configurations end-to-end from input data. Due to their simple use of multiple stacked convolutional layers, their models still have some limitations in terms of grasp detection accuracy. In order to improve the accuracy of grasp detection, Kumra et al. [7] proposed a novel grasp algorithm. They achieved higher accuracy by designing a deeper network structure. Similarly, many subsequent studies also adopted this generative grasp detection method [12, 13].

3 Problem Statement

In this paper, a five-dimensional rectangle is used to represent the grasp configuration of the gripper [6]. We follow the grasp generation pipeline of Morrison et al. [6] to generate pixel-wise grasp poses from input images. We generate a grasp configuration from an N-channel image $\mathrm{I} \in \mathbb{R}^{N \times H \times W}$ with height H and width W, which can be expressed as:

$$\mathbf{g}_i = (\mathbf{p}_i, q_i, \theta_i, w_i) \tag{1}$$

where $\mathbf{p}_i = (u, v)$ is the parallel gripper's center position, q_i is the scalar quality measure, θ_i denotes the rotation in camera's frame of reference, w_i denotes required width in image coordinates.

The grasp quality q_i represents the grasp score. It is a scalar value between 0 and 1 where a value closer to 1 indicates higher grasp quality. θ_i represents the rotation grasp angle in the camera coordinate. Due to the grasp angle θ_i is symmetrical, the angle is represented as a value in the range [-π/2, π/2]. To avoid learning the grasp angle directly, the angle is encoded as two components of a unit vector, $\cos(2\Theta_I)$ and $\sin(2\Theta_I)$. The final grasp angle is calculated by $\theta_i = atctan(\sin(2\Theta_I)/\cos(2\Theta_I))/2$.

All grasp configurations in image space can be expressed as:

$$\mathbf{G}_I = (\mathbf{Q}_I, \mathbf{W}_I, \mathbf{\Theta}_I) \in \mathbb{R}^{3\times H\times W} \tag{2}$$

where $\mathbf{Q}_I \in \mathbb{R}^{H\times W}$,$\mathbf{W}_I \in \mathbb{R}^{H\times W}$ and $\mathbf{\Theta}_I \in \mathbb{R}^{H\times W}$ represent three images and contain values of q_i, w_i and θ_i at each pixel, respectively.

The optimal grasp configuration in image space can be computed as follows:

$$\mathbf{g}_i^* = \max_{\mathbf{Q}_I}(\mathbf{G}_I) \tag{3}$$

4 Method

Robots need to be able to grasp unseen objects in an unstructured environment. For a grasp model, it is necessary to consider the uncertainty of the shape, pose and size of the object to be grasped. Generally, the grasp model uses stacking multiple convolutional layers to extract the feature information of the object to be grasped from the input image. Shallow networks usually extract low-level feature information of objects (i.e. texture and edge), and deeper networks usually extract more abstract high-level semantic information. Therefore, considering the feature information of different layers extracted by the model may be more beneficial for the grasp detection task. In addition, it also needs to consider how different convolutional layers can effectively extract features useful for grasp detection.

In this paper, a grasp detection model named ECA-GraspNet is proposed. ECA-GraspNet employs an encoder-decoder architecture. In order to utilize the feature information extracted by different layers, we refer to the idea of Feature Pyramid Network (FPN) [8] to make skip connections between some convolutional layers in the encoder and decoder. Extensive studies have shown that models with deeper and wider layers generally demonstrate better performance for the same task. For grasp tasks with potential in industry, a lightweight and efficient grasp model is the primary consideration. In order to effectively extract the feature information of objects to be grasped, an efficient channel attention network is used in FPN. The network architecture is shown in Fig. 1.

4.1 Grasp Detector

The grasp detector is designed with a generative end-to-end grasp pipeline. In the encoder part, the RGB-D image first passes through three convolutional layers [C1, C2, C3] to

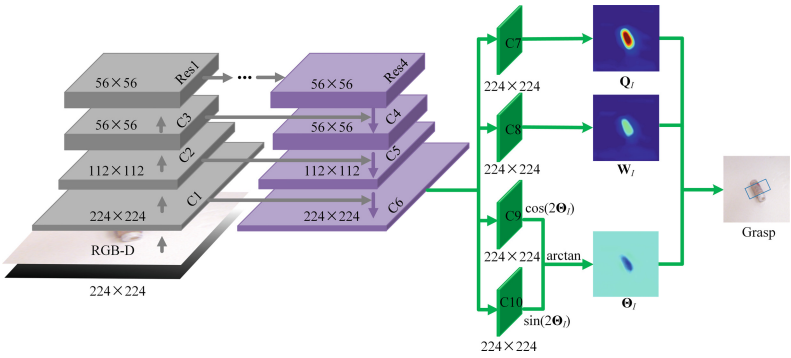

Fig. 1. Network architecture. The encoder consists of 3 convolutional layers [C1, C2, C3] and 4 residual blocks [Res1, Res2, Res3, Res4]. The decoder consists of 3 convolutional layers [C4, C5, C6] for feature extraction and 4 convolutional layers [C7, C8, C9, C10] for generating grasps.

extract features, and the output sizes are 32 × 224 × 224, 64 × 112 × 112 and 128 × 56 × 56. Each convolutional layer consists of a convolutional operation, a batch normalization (BN) layer and a ReLU respectively. The kernel size and stride of C1 are 9 and 1, respectively. C2 and C3 have the same kernel size and stride, which are 4 and 2, respectively. To utilize the convolutional layers more effectively, ECA is inserted after C2 and C3 to focus on the effective convolutional layers for grasp detection. It is well known that the performance of the model will increase as the number of layers increases at a certain number of layers. However, when exceeding a certain number of layers, the performance of the model will decrease. To extract high-level feature information that may be more useful for grasp detection at deeper layers, four residual blocks [Res1, Res2, Res3, Res4] are used. Each residual block is composed of two convolutional layers and a skip connection, and the output size is 128 × 56 × 56. The kernel size and stride of the convolutional layers in each residual block are both 3 and 1.

In the decoder part, three convolutional layers are used for sequential up-sampling. The size of each convolutional layer output is 128 × 56 × 56, 64 × 112 × 112 and 32 × 224 × 224, respectively. In this process, the corresponding convolutional layers in the encoder and decoder are skip-connected, and ECA is used to focus on the convolutional layers that are effective for grasp detection. ECA can make full use of limited convolutional layers to extract effective convolutional layers for grasp detection. C4 and C5 have the same kernel size and stride, which are 4 and 2, respectively. The kernel size and stride of C6 are 9 and 1, respectively. Furthermore, ECA is inserted after both C4 and C5 to further focus on convolutional layers that are more effective for grasp detection. The final output of the network includes grasp quality ($\mathbf{Q}_I$), grasp width ($\mathbf{W}_I$) and grasp angle ($\mathbf{\Theta}_I$). The grasp angle includes two parts $\cos(2\Theta_I)$ and $\sin(2\Theta_I)$. Each component of the network output is obtained using a convolutional layer with kernel size 1 and stride 1.

4.2 Loss Function

In order to deal with outliers more stably, we use smooth L1 (Huber) loss function. This loss function combines the advantages of L1 loss and L2 loss, which is not easily disturbed by other outliers and has good robustness. Given a generated grasp G_i and a ground-truth grasp $\tilde{G}_i$, the loss function is defined as follows:

$$L(G_i, \tilde{G}_i) = \frac{1}{n} \sum_k z_k \tag{4}$$

where z_k is represented as follows:

$$z_k = \begin{cases} 0.5(G_{i_k}, \tilde{G}_{i_k})^2, & if\, |G_{i_k}, \tilde{G}_{i_k}| < 1 \\ |G_{i_k}, \tilde{G}_{i_k}| - 0.5, & otherwise \end{cases} \tag{5}$$

5 Experiments and Results

5.1 Dataset and Metric

Models are trained and evaluated using the public Cornell Grasping Dataset [10]. For the Cornell dataset, there are 885 RGB and depth images of 240 different objects, of which there are 5,110 positive and 2,909 negative grasps. The dataset is divided in two ways: image-wise (IW) and object-wise (OW). The training set and test set divided by IW may have images of the same object. Objects do not overlap in the OW division way, which is closer to the real scene. We generate training and test sets using OW way.

This paper uses the standard rectangle metric to evaluate whether the grasp rectangle is correct. For a correctly predicted rectangle, the angular difference between the predicted grasp angle and the ground truth angle is within 30° and the Jacquard metric is greater than 0.25. The rectangle metric is calculated as follows:

$$\begin{cases} \left|\theta_p - \theta_{GT}\right| & < 30^\circ \\ J|G_p, G_{GT}| = \frac{G_p \cap G_{GT}}{G_p \cup G_{GT}} & > 0.25 \end{cases} \tag{6}$$

where θ_p represents the predicted grasp angle and θ_{GT} represents the ground truth grasp angle. G_p represents the predicted grasp rectangle and G_{GT} denotes the ground truth grasp rectangle.

5.2 Training Details

The model is implemented by PyTorch, and the model parameters are optimized using the Adam optimizer. We employ random cropping, scaling and rotation to create 8,840 RGB-D images of input size 224 × 224 and 51,100 grasp positive samples. The experiment is conducted using a 2.4GHz Intel Xeon Silver 4210R CPU and NVIDIA GeForce RTX3090 graphics card, and is trained for 50 epochs. The initial learning rate is set to 0.001 and the batch size is 8.

5.3 Experiments on the Dataset

Quantitative Results. Table 1 compares the results of different algorithms on the Cornell dataset. OW is used to demonstrate the accuracy of grasp detection. The proposed method achieves 97.7% accuracy on OW. Compared with other algorithms, the proposed method achieves better accuracy. Compared with the method of Kumra et al. [7], the proposed method has a relative improvement of 1.1% in accuracy. Moreover, the proposed method has an inference speed of 213 FPS, which can be used for real-time applications. The grasp detection time is 20 ms. It should be noted that the grasp detection time includes the sum of the time of all steps of image pre-processing, model inference and post-processing. Compared with the method of Tian et al. [17], the accuracy of the proposed method is reduced by 1.2%. However, the proposed method is faster in detection speed. Compared with other algorithms, our method can make a better trade-off between accuracy and detection time. For future industrial applications, grasp estimation algorithms need to perform well in both accuracy and efficiency. The proposed method maintains real-time performance while improving the accuracy of grasp detection. In addition, the proposed model uses multi-layer convolution to build a lightweight grasp detection network, which is more advantageous than others using large networks (*i.e.* ResNet-50) for deployment on actual AI embedded devices.

Table 1. Results of different algorithms on the Cornell dataset.

Authors	Algorithm	OW (%)	Time (ms)
Jiang [11]	Fast Search	58.6	5000
Lenz [10]	SAE	75.6	1350
Redmon [14]	AlexNet, MutiGrasp	87.1	76
Asif [15]	STEM-CaRFs	87.5	–
Kumra [16]	ResNet-50x2	88.9	103
Morrison [6]	GG-CNN	69.0	19
Kumra [7]	GR-ConvNet-RGB-D	96.6	20
Tian [17]	ResNet-50	98.9	26
Ours	ECA-GraspNet	97.7	20

Qualitative Results. Figure 2 shows the results of partial grasp detection. The first row to the third row in Fig. 2 represent the grasp quality, grasp angle and grasp width respectively. Locations that are darker red in the grasp quality image have higher grasp scores. The last row of Fig. 2 represents the predicted grasp rectangle. From the visualization results, the proposed method can better predict the grasp rectangles of different objects. Experimental results demonstrate that the proposed ECA-GraspNet is able to generate reliable grasps.

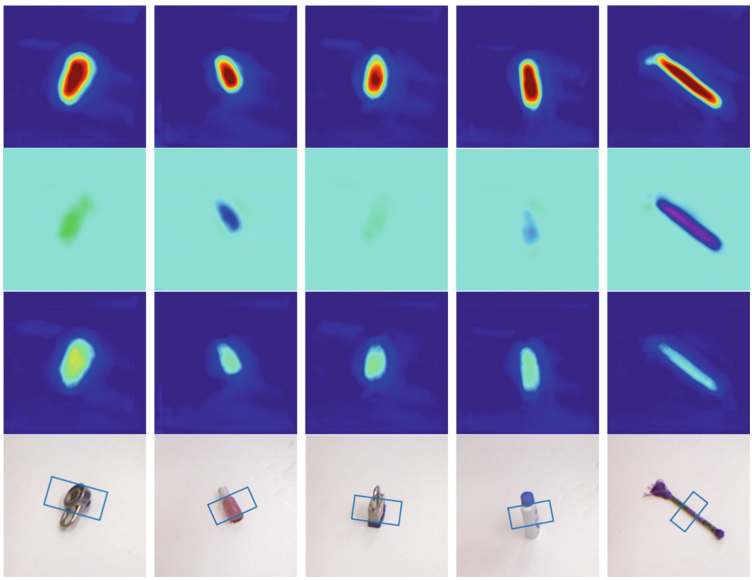

Fig. 2. Grasp detection results on the Cornell dataset.

Limitations. The method proposed in this paper only generates grasp rectangles for isolated objects. In the actual cluttered object scene, this will affect the accuracy of grasp detection to a certain extent. In addition, for potential grasping applications, the detection efficiency of the proposed method needs to be improved.

6 Conclusion

In this paper, we propose a novel grasp detection network ECA-GraspNet. The feature information extracted by different layers is fused through the feature pyramid network, and an efficient channel attention mechanism is embedded to make full use of the convolutional layers that are beneficial to grasp detection. We evaluated ECA-GraspNet on the Cornell dataset and achieved decent accuracy. Moreover, the lightweight design enables our proposed grasp detection network to be used in real-time robotic applications. In the future, we will focus on grasp detection in cluttered scenes and improve detection efficiency.

Acknowledgments. This work was supported in part by young and middle-aged Teachers in Guangxi Universities (ID: 2022KY0607), and School-level commissioned project of Hechi University (ID: 2022YLXK003), National-level Innovation and Entrepreneurship Training Program for College Students (ID: 202310605044). This research was financially supported by First-class Discipline Construction Project of Hechi University, Guangxi Colleges and Universities Key Laboratory of AI and Information Processing (Hechi University), Education Department of Guangxi Zhuang Autonomous Region.

References

1. Rubert, C., Kappler, D., Morales A., et al.: On the relevance of grasp metrics for predicting grasp success. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 265–272 (2017)
2. Cheng, H., Meng, M.Q.H.: A grasp pose detection scheme with an end-to-end CNN regression approach. In: 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 544–549 (2018)
3. Chu, F.J., Xu, R., Vela, P.A.: Real-world multiobject, multigrasp detection. IEEE Robot. Autom. Lett. **3**(4), 3355–3362 (2018)
4. Cheng, H., Wang, Y., Meng, M.Q.H.: A vision-based robot grasping system. IEEE Sens. J. **22**(10), 9610–9620 (2022)
5. Song, Y., Gao, L., Li, X., et al.: A novel robotic grasp detection method based on region proposal networks. Robot. Comput.-Integr. Manuf. **65**, 101963 (2020)
6. Morrison, D., Corke, P., Leitner, J.: Learning robust, real-time, reactive robotic grasping. Int. J. Robot. Res. **39**(2–3), 183–201 (2020)
7. Kumra, S., Joshi, S., Sahin, F.: Antipodal robotic grasping using generative residual convolutional neural network. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9626–9633 (2020)
8. Lin, T.Y., Dollár, P., Girshick, R., et al.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
9. Wang, Q., Wu, B., Zhu, P., et al.: ECA-Net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11534–11542 (2020)
10. Jiang, Y., Moseson, S., Saxena, A.: Efficient grasping from rgbd images: learning using a new rectangle representation. In: 2011 IEEE International Conference on Robotics and Automation, pp. 3304–3311 (2011)
11. Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasp. Int. J. Robot. Res. **34**(4–5), 705–724 (2015)
12. Song, Y., Wen, J., Liu, D., et al.: Deep robotic grasping prediction with hierarchical rgb-d fusion. Int. J. Control. Autom. Syst. **20**(1), 243–254 (2022)
13. Liu, D., Tao, X., Yuan, L., et al.: Robotic objects detection and grasping in clutter based on cascaded deep convolutional neural network. IEEE Trans. Instrum. Meas. **71**, 1–10 (2022)
14. Redmon, J., Angelova, A.: Real-time grasp detection using convolutional neural networks. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1316–1322 (2015)
15. Asif, U., Tang, J., Harrer, S.: GraspNet: an efficient convolutional neural network for real-time grasp detection for low-powered devices. In: International Joint Conference on Artificial Intelligence, vol. 7, pp. 4875–4882 (2018)
16. Kumra, S., Kanan, C.: Robotic grasp detection using deep convolutional neural networks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 769–776 (2017)
17. Tian, H., Song, K., Li, S., et al.: Lightweight pixel-wise generative robot grasping detection based on rgb-d dense fusion. IEEE Trans. Instrum. Meas. **71**, 1–12 (2022)