



Improved GR-Convnet for Antipodal Robotic Grasping

Kyosuke Shibasaki^(✉), Keisuke Hamamoto, and Huimin Lu

Kyushu Institute of Technology, 1-1 Sensui-Cho, Tobata-Ku, Kitakyushu-shi, Fukuoka, Japan
shibasaki.kyosuke602@mail.kyutech.jp

Abstract. This paper introduces a robot system designed to address the problem of performing antipodal robot grasping for unknown objects. We focus on the high-level approach of GR-Convnet for the task and propose a neural network with high robustness while maintaining real-time performance. The three improvements include introducing Squeeze and Excitation (SE) blocks, removing Dropout in the final layer, and using Residual Block and Concurrent Spatial and Channel Squeeze and Channel Excitation (scSE) Block. We evaluate the proposed network on the Jacquard dataset containing information on various household objects. As a result, we achieved an approximately 7.2% improvement in accuracy compared to GR-Convnet. Additionally, using a real robot, we demonstrated a grasp success rate of 93.3% and 92.5% for household and adversarial objects, respectively.

Keywords: Antipodal Robotic Grasping · Deep learning · Grasping Point Estimation

1 Introduction

In recent years, Japan has faced a social problem of a declining working population due to rapid aging and a low birthrate after peaking in 2008. [1] This has led to a reduction in the size of the economy and a decline in international competitiveness. Thus, it is necessary to create more added value with a limited labor force. Industrial robots are an effective solution to this problem since they can perform tasks in place of humans, stabilize production efficiency, operate for long hours, and improve quality by preventing human error [2, 3]. As a result, the demand for industrial robots has been growing and is expected to continue to grow in the future. By 2025, the Japanese domestic robotics market will be worth approximately 5.3 trillion yen and 9.7 trillion yen by 2035 [4, 5]. However, conventional industrial robots require predefined movements, which is an obstacle to their widespread use. To overcome this difficulty, the use of deep learning to make industrial robots intelligent has been attracting attention. Therefore, many network models for grasp position estimation by deep learning have been proposed. In the early studies of grasp position estimation, rule-based methods [6, 7] and object detection-based methods [8, 9] were mainly used. Therefore, a generative convolutional neural network [10, 11] was applied and succeeded in reducing the weight. Furthermore, a suitable grasping posture can be predicted from extracted pixel-by-pixel features, making the

method more suitable for grasping tasks. [12–14] Recently, GR-ConvNet [15] achieved state-of-the-art grasping detection accuracy by introducing a residual structure [16] into the network model.

2 Method

In this chapter, we use a method to represent the grasping position that is similar to the one used in GR-Convnet. This helps us detect the grasping position accurately. Next, we introduce an even better architecture. In the bottleneck, we merge the ResNet block and the SE block composite module together at the same time, and also bring in the csSE block to the decoder. In the output layer, we create a network that predicts the grip’s quality, angle, and width separately, by utilizing the features we have extracted, and without needing to use dropout.

2.1 Formulation of Grasping Position

The position of a gripper in 3D space is expressed as

$$G_r = (P, \theta_r, W_r, Q) \quad (1)$$

where P is the center position of the gripper tip, θ_r is the rotation of the gripper, W_r is the gripper width, and Q is the probability of a successful grasp. The position of the grasp in the image is expressed as

$$G_i = (x, y, \theta_i, W_i, Q) \quad (2)$$

where (x, y) is the center point of the object in the image, θ_i is the rotation angle, W_i is the width of the gripper in the image, and Q is the probability of a successful grasp. The transformation of the grasping position information from the image to the camera space and then to the world coordinates is expressed as

$$G_r = T_{rc}(T_{ic}(G_i)) \quad (3)$$

The grasping positions of multiple objects are expressed as

$$G = (\theta, W, Q) \in \mathbb{R}^{n \times h \times w} \quad (4)$$

in Eq. (4), where θ is a quality map representing the success rate of grasping, W is a width map representing the width of the end-effector, and Q is an angle map representing the angle of the end-effector.

2.2 Proposed Network Model

This is the network architecture proposed in Fig. 1 we constructed a network model with reference to the GR-convnet. The E block and D block have the same conventional structure and are shown at the bottom of Fig. 1. The B block, F block, and output layer will be described in a later section.

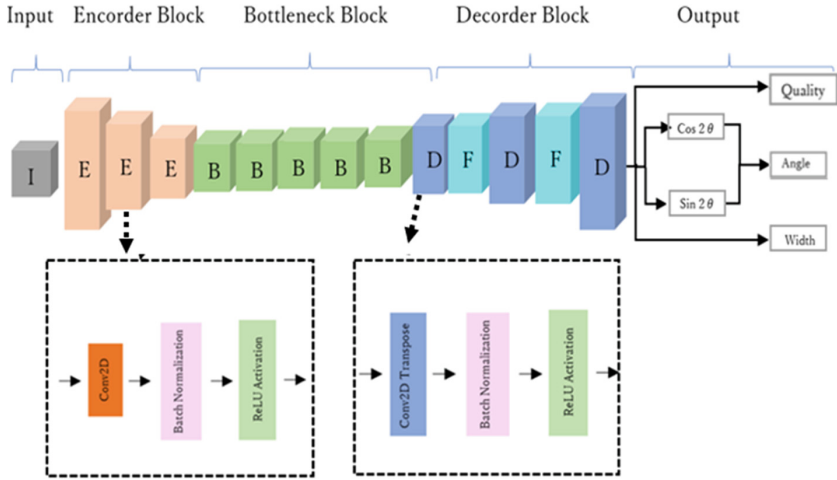


Fig. 1. Network model of the proposed method

Bottleneck Module: The bottleneck module shown in Fig. 1 utilizes the module shown in Fig. 2 and combines SE blocks with ResNet to emphasize pixel-level information by weighting each channel with a sigmoid function. In [17], the optimal placement of the SE block in the Residual Block [16] is investigated. Among the tested patterns, this paper uses the SE-Identity Block [17]. Similarly, it has been applied to ResNet-50 [18] and inception-resnet [19], achieving top scores in the ILSVRC 2017 competition.

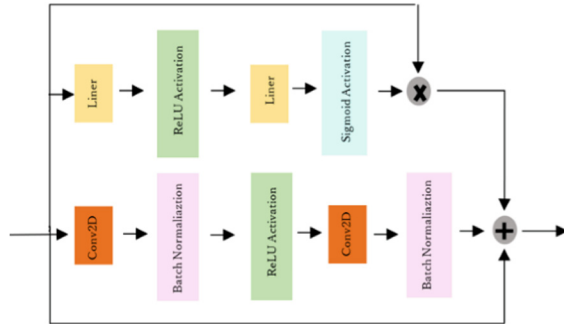


Fig. 2. Improved bottleneck block, combining Resnet with SE block in parallel.

Deletion of Output Layer Dropout: Dropout [20] and Batch Normalization [21] are techniques to prevent overfitting in deep learning models. Dropout randomly selects certain neurons to be unused during each mini-batch, while Batch Normalization normalizes each node's output to prevent bias. Combining these techniques may lead to decreased accuracy [22], so Dropout is not used in the paper according to previous research.

Decoder Module: The decoder module, shown in Fig. 1, also includes the F block (Fig. 3) that combines the Concurrent Spatial and Channel Squeeze and Channel Excitation (scSE) Block [23] with ResNet. The scSE Block was designed for segmentation and emphasizes pixel-level information by convolving each channel and calculating weighting for each pixel. The module aims to reduce noise and information loss when restoring the image to its original size.

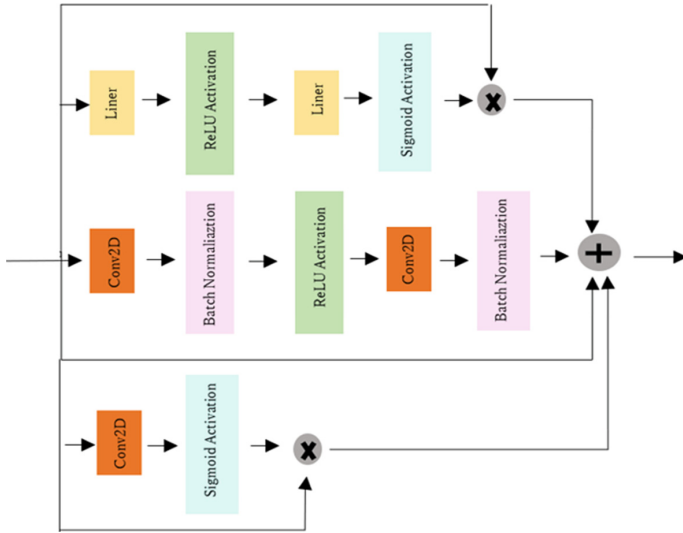


Fig. 3. Improved decoder module.

3 Evaluation of Network Models

In this chapter, we define assumptions and evaluation methods for inference and perform inference. Furthermore, we compare the performance of the proposed method with that of previous research [16], and discuss the results.

3.1 Learning Environment

In this section, we used Nvidia's 32[GB], RTX2080Ti. Each network is trained for 100 epochs with a learning rate of 10^{-3} and a batch size of 8.

3.2 Dataset

The Jacquard dataset [24] is annotated using a simulation environment and CAD model, without human intervention. It was split into training, validation, and test data with an 8:1:1 ratio using two methods: IW (image-wise) for evaluating generalization ability for unknown object postures, and OW (object-wise) for evaluating generalization performance for unknown objects. In GR-ConvNet [15], training and validation were divided 9:1.

3.3 Optimization Function

We used Radam [25] as our optimization function in this study, which was proposed by Liu et al. in 2020. Adam is a commonly used optimization function for deep learning, but Liu et al. found that it has a high variance in the early stages of learning. To address this, they proposed a method that uses SGD [20] with momentum in the initial stages and corrects Adam [26] with a correction term afterwards. This method outperforms Adam on tasks such as ImageNet image classification.

3.4 Loss Function

In this study, we use the Smooth L1 Loss used in Gr-ConvNet [8].

$$\text{loss}(G_t - G_p) = \begin{cases} 0.5(G_t - G_p)^2, & \text{if } |G_t - G_p| < \beta \\ |G_t - G_p| - 0.5, & \text{otherwise} \end{cases} \quad (5)$$

Here, G_t represents the true value and G_p the predicted value, where β is the threshold value, and the upper and lower equations are used separately. MAE Loss and MSE Loss are common loss functions in deep learning. MAE Loss is good for outliers because it treats errors as absolute values, but it cannot be differentiated when the true and predicted values are equal. Smooth L1 Loss uses MSE Loss when the absolute error is smaller than the threshold β , and uses MAE Loss when the absolute error is larger than the threshold β to reduce the effect of outliers. In this study, the threshold β is set to 1.0.

3.5 Evaluation Index

This is an evaluation index similar to GR-ConvNet [15]. The condition for considering the grasping position estimated by the network model as correct is when the following two conditions are satisfied:

1. The IOU between the bounding box of the inferred grasping position and the ground truth value is greater than 0.25.
2. The error between the inferred grasp angle and the ground truth value is less than 30° .

The evaluation index is calculated using the following Eq. (6).

$$\text{Accuracy rate}[\%] = \frac{\text{The number of correct answers}}{\text{number of datasets}} \quad (6)$$

3.6 Object to be Grasped

This section describes the grasping object in the grasping experiment using the actual machine.

Household Test Objects: Twelve objects of different shapes and sizes from each other were prepared. For example, there are objects that are similar to the objects in the dataset, smaller, transparent, reflective, soft, reflective, and black, which are considered to be difficult to visualize. The household test objects are shown on the left in Fig. 4.

Adversarial Test Objects: These are objects in the dataset Dex-Net 2.0 [27] used by Mahler et al. to validate the performance of the Grasping Quality CNN, and are objects that are considered difficult to grasp. In this study, eight objects were prepared. The adversarial test objects are shown on the right in Fig. 4



Fig. 4. Left: Test object for home use, right: Adversarial test object

4 Result

In this chapter, we present the results of inference performed with each network model, as well as experimental results using a physical robot arm, and demonstrate the process of inference during the experiments.

4.1 Inference with Network Models

We made three different versions of the model. Modification (1) improved only one part, the bottleneck. Modification (2) improved two parts, the bottleneck and output layer. The proposed method improved three parts, the bottleneck, output layer, and decoder. We compared the original model with the three improved models to show how effective each improvement was. Results are shown in Table 1.

Table 1. Results for each network model

Method	Accuracy (%)		Inference time (ms)	Parameters
	Object-Wise	Image-Wise		
GR-ConvNet [8]	86.54	85.22	18.1	1,900,900
Modification (1)	90.34	90.89	18.4	1,904,108
Modification (2)	90.40	92.09	18.4	1,904,108
Proposed Method	91.37	92.62	20.3	1,992,862

4.2 Experiments with Robotic Arms

Experiments were conducted on GR-ConvNet [15] and the proposed method, which was the most accurate according to Table 3. The experimental results are shown in Table 2. 86.5% for GR-ConvNet[15] and 93% for the proposed method.

Table 2. Experimental results

Object	Method	
	GR-ConvNet [15]	Proposed Method
Home test objects	104/120	112/120
Adversial test objects	69/80	74/80
total amount	176/200(86.5%)	186/200(93%)

4.3 Visualization of Inference

To show the results of grasping position inference, a Quality Map is used as an image to indicate that red has a high probability of grasping. Table 3 shows the inference results.


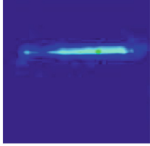
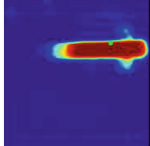

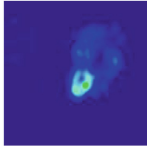
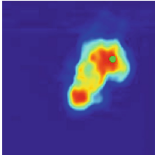

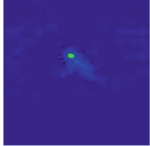


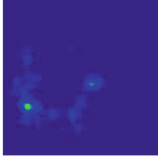
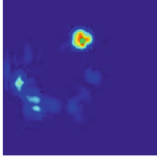
5 Discussion and Conclusion

The effectiveness of the proposed method was demonstrated through network evaluation and gras experiments using a robot [28, 29]. As shown in Table 1, each improvement was found to improve the accuracy of the network. Additionally, it is believed that the accuracy was improved without significantly increasing the number of parameters. Furthermore, as shown in Table 2, it was confirmed that improving the accuracy of the network also improves the accuracy of the physical grasp experiment using a robot arm.

According to Table 3, the success rate of grasping difficult-to-recognize black objects has improved. In the case of failure with GR-ConvNet, it was found that the grasping angle was not appropriate for the remote control. In addition, for wristwatches, there were cases where the reflection part was not detected at the appropriate grasping position. However, the proposed method makes it clear where the expected position for high success rate of grasping is located. Moreover, both methods showed low accuracy for small and transparent objects. One of the reasons for grasp failure is that these objects are difficult for humans to visually recognize. Due to their low visibility, the amount of image information obtained is limited, and accurate inference needs to be made from this limited information.

In the future, efforts will be focused on improving the grasp grip of objects that are difficult for humans to visually recognize [30, 31], by improving the network model and changing evaluation criteria. Preprocessing techniques such as removing reflections and enlarging small objects, as well as other approaches, will also be considered to obtain more information.

Table 3. Inference by actual equipment

	Original image	GR-Convnet[15]	Proposed-method
Only the proposed method succeeded			
			
Both failed			
			

References

1. Ministry of Internal Affairs and Communications. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd101200.html>. Last accessed 9 Jan 2022
2. KEYENCE. https://www.keyence.co.jp/ss/products/vision/fa-robot/industrial_robot/merit.jsp. Last accessed 2022
3. Lu, H., Li, Y., Nakashima, S., et al.: Underwater image super-resolution by descattering and fusion. *IEEE Access* **5**, 670–679 (2017)
4. Ministry of Internal Affairs and Communications. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h27/html/nc241330.html>. Last accessed 2022
5. Lu, H., Li, Y., Chen, M., et al.: Brain Intelligence: go beyond artificial intelligence. *Mobile Netw. Appl.* **23**, 368–375 (2018)
6. Pokorny, F.T., Bekiroglu, Y., Kragic, D.: Grasp moduli spaces and spherical harmonics. *ICRA* (2014)

7. Lu, H., Yang, R., Deng, Z., et al.: Chinese image captioning via fuzzy attention-based DenseNet-BiLSTM. *ACM Trans. Multimedia Comput. Commun. Appl.* **17**(1s), 1–18 (2021). <https://doi.org/10.1145/3422668>
8. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. *CoRR* (2015)
9. Lu, H., Yu, T., Sun, Y.: DRRS-BC: decentralized routing registration system based on blockchain. *IEEE/CAA J. Autom. Sinica* **8**(12), 1868–1876 (2021)
10. Morrison, D., Corke, P., Leitner, J.: Learning robust, real-time, reactive robotic grasping. *The Int. J. Robot. Res.* **39**, 183–201 (2020)
11. Lu, H., Wang, D., Li, Y., et al.: CONet: a cognitive ocean network. *IEEE Wireless Commun.* **26**(3), 90–96 (2019)
12. Wang, S., Jiang, X., Zhao, J., Wang, X., Zhou, W., Liu, Y: Efficient fully convolution neural network for generating pixel wise robotic grasps with high resolution images. *CoRR* (2019)
13. Lu, H., Qin, M., Zhang, F., et al.: RSCNN: a CNN-based method to enhance low-light remote-sensing images. *Remote Sens.* **13**(1), 62 (2020)
14. Xu, X., Lu, H., Song, J., et al.: Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval. *IEEE Trans. Cybern. Cybern.* **50**(6), 2400–2413 (2020)
15. Kumra, S., Joshi, S., Sahin, F.: Antipodal robotic grasping using generative residual convolutional neural network (2019)
16. He, K., Zhang, X., Ren, S., Su, J.: Deep residual learning for image Recognition. In: *CVPR770–778* (2016)
17. Hu, J., et al.: Squeeze-and-Excitation Networks. In: *CVPR* (2018)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR* (2015.)
19. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inceptionv4, inception-resnet and the impact of residual connections on learning. In: *AAAI Conference on Artificial Intelligence* (2015)
20. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors (2012)
21. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International Conference on Learning Representations* (2015)
22. Li, X., Chen, S., Hu, X., Yang, J.: Understanding the disharmony between dropout and batch normalization by variance shift (2018)
23. A. G. Roy, N. Navab, and C. Wachinger. Concurrent spatial and channel ‘squeeze & excitation’ in fully convolutional networks. In: *International Conference on Medical Image Computing and Computer Assisted Intervention 2018*. https://doi.org/10.1007/978-3-030-00928-1_48
24. Depierre, A., Dellandrea, E., Chen, L.: Jacquard: A large scale dataset for robotic grasp detection. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2018)
25. Liu, L., He, J., Chen, P., Liu, W., Gao, J., Han, J.: On the variance of the adaptive learning rate and beyond. In: *ICLR* (2020)
26. Diederik, J.B., Kingma, P.: Adam A method for stochastic optimization. In: *International Conference for Learning Representations* (2015)
27. Nitanda, A.: Stochastic proximal gradient descent with acceleration techniques. In: *Advances in Neural Information Processing Systems* (2014)
28. Mahler, J., et al.: Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics (2017)
29. Lu, H., Zhang, Y., Li, Y., et al.: User-oriented virtual mobile network resource management for vehicle communications. *IEEE Trans. Intell. Transport. Syst.* **22**(6), 3521–3532 (2021)
30. Lu, H., Li, Y., Mu, S., et al.: Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE Internet Things J.* **5**(4), 2315–2322 (2018)
31. Lu, H., Zhang, M., Xu, X., et al.: Deep fuzzy hashing network for efficient image retrieval. *IEEE Trans. Fuzzy Syst.* **29**(1), 166–176 (2021)