




Article

Research on Generation Method of Grasp Strategy Based on DeepLab V3+ for Three-Finger Gripper

Sanlong Jiang ¹, Shaobo Li ^{1,2,*}, Qiang Bai ¹, Jing Yang ¹, Yanming Miao ³ and Leiyu Chen ¹

¹ School of Mechanical Engineering, Guizhou University, Guiyang 520025, China; gs.sljiang19@gzu.edu.cn (S.J.); cme.qbai18@gzu.edu.cn (Q.B.); jyang23@gzu.edu.cn (J.Y.); cly199667@gmail.com (L.C.)

² State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China

³ Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, Guiyang 550025, China; docmym@163.com

* Correspondence: lishaobo@gzu.edu.cn; Tel.: +86-139-8505-3753

Abstract: A reasonable grasping strategy is a prerequisite for the successful grasping of a target, and it is also a basic condition for the wide application of robots. Presently, mainstream grippers on the market are divided into two-finger grippers and three-finger grippers. According to human grasping experience, the stability of three-finger grippers is much better than that of two-finger grippers. Therefore, this paper's focus is on the three-finger grasping strategy generation method based on the DeepLab V3+ algorithm. DeepLab V3+ uses the atrous convolution kernel and the atrous spatial pyramid pooling (ASPP) architecture based on atrous convolution. The atrous convolution kernel can adjust the field-of-view of the filter layer by changing the convolution rate. In addition, ASPP can effectively capture multi-scale information, based on the parallel connection of multiple convolution rates of atrous convolutional layers, so that the model performs better on multi-scale objects. The article innovatively uses the DeepLab V3+ algorithm to generate the grasp strategy of a target and optimizes the atrous convolution parameter values of ASPP. This study used the Cornell Grasp dataset to train and verify the model. At the same time, a smaller and more complex dataset of 60 was produced according to the actual situation. Upon testing, good experimental results were obtained.

Keywords: semantic segmentation; grasp strategies; atrous convolutions; three-finger gripper



Citation: Jiang, S.; Li, S.; Bai, Q.; Yang, J.; Miao, Y.; Chen, L. Research on Generation Method of Grasp Strategy Based on DeepLab V3+ for Three-Finger Gripper. *Information* **2021**, *12*, 278. <https://doi.org/10.3390/info12070278>

Academic Editors: Aldo Jonathan Muñoz-Vázquez and Juan Diego Sánchez Torres

Received: 3 June 2021

Accepted: 23 June 2021

Published: 8 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The generation of a grasping strategy is the first step in robot grasping, and it is also an important step [1]. The traditional grasping strategy is mostly used for determining the location and type of targets, and certain grasping points are artificially set as the grasping strategy. With the miniaturization and precision of control motors, the maturity of control modules, such as single-chip microcomputers, and the development and application of artificial intelligence technology [2], including the types of grasping targets and grasping positions, will become more diversified in future applications of grasping robots. A robot generates reasonable grasping strategies for different targets in different positions to make subsequent grasping operations more reliable and successful. Operating as information input, images obtain a larger amount of information with a lower amount of data in a short time; thus, images represent the most suitable choice for informational input in the generation of a grasp strategy. Only by generating a reasonable grasp strategy based on the target image collected by the camera can a robot accurately and rapidly grasp targets [3,4].

Due to the great potential and advantages of deep learning in semantic segmentation and target recognition [5,6], research on the direction of grasping strategy generation has gradually adopted deep learning as the core means of solving problems associated with grasping strategy prediction [4,7–9]. Deep learning can eliminate the large number of kinematics, dynamics, and geometry calculations [10] and directly accept images that need to be detected. After the convolution and pooling process of the neural network, the grasp

strategy is generated on the image to achieve end-to-end processing, which is faster and more accurate [10,11].

In recent years, thanks to the pixel-level representation of semantic segmentation and the analysis of the entire content of the image, the application of semantic segmentation in robot grasping has primarily focused on using 3D point cloud data to achieve 3D object pose estimation and data segmentation of the target; it then generates a grasp strategy for the target [12–14]. This algorithm is suitable for location and grasp detection in a chaotic environment in where a target is occluded. However, when the scene is relatively simple in artificial processing terms, such as in a factory, this algorithm has problems with the large amount of calculation and low accuracy. On the other hand, the grasping detection algorithm primarily focuses on the realization of a target generation grasping strategy based on target recognition and detection algorithms [15–17], which have a small number of parameters and calculations and can meet the requirements for the real-time generation of target-grasping strategies in simple scenarios.

This paper inherits the advantages of the previous research [1,3,12–19] regarding the pixel-level representation of the semantic segmentation algorithm, the detection ability for small objects, and the real-time operation and high accuracy of the grasp detection algorithm. Thus, this paper builds a neural network model with semantic segmentation as the core, generates some reasonable grasping strategies for the target [7,18], and divides the surface of the object into the part that can be grasped and the part that cannot be grasped. When the grasp point falls on the position that can be grasped, the grasp point is considered to be correct and is supplemented by the appropriate width and angle; only then is the generated grasp strategy is considered to be successful. This idea is similar to semantic segmentation, and the pixel-level output of semantic segmentation has a higher ability to recognize a given target. In general, this paper proposes a neural network based on DeepLab V3+, which can be used as the robot's brain to generate grasping strategies for known or unknown targets that need to be grasped in a scene in real time and that can guide the robot to grasp them.

Our work offers two main contributions:

1. It proposes a grasp strategy generation neural network (grasp network with atrous convolution) based on the idea of semantic segmentation and, based on DeepLab V3+ with atrous spatial pyramid pooling (ASPP), it achieves a good grasp accuracy rate and outputted pixel-level results in the current research;
2. It explores the effect of different rates of the atrous space convolution pooling pyramid on the recognition of the images in Cornell Grasp dataset, and analyzes the value of the rates of ASPP for different targets.

This paper is organized as follows. Section 2 discusses the difference between convolution kernel and atrous convolution kernel, the structure of DeepLab V3+, the neural network structure proposed in this article, and the difference between the commonly used five-dimensional representation and the oriented base-fixed triangle representation used in this paper. Section 3 introduces the experimental details and results, Section 4 discusses the results, and Section 5 offers some conclusions and future prospects.

2. Materials and Methods

2.1. Theoretical Analysis

As mentioned above, an advantage of semantic segmentation is that its output result operates at a pixel level, and it has good recognition ability for a global scene. Given its good performance in the field of semantic segmentation, DeepLab V3+ adopts an encoder–decoder structure and converts its convolution kernel into ASPP with different rates of atrous convolution. With high recognition speed and recognition accuracy, it can precisely process different sizes of objects in the background at the same time without a high amount of calculation. In this paper, the DeepLab V3+ model [20] is used as the feature extractor. The output of the DeepLab V3+ model is processed by the convolutional layer and the upsampling layer to generate the final grasp strategy, which represented by the pixel-level

representation and the oriented base-fixed triangle. In the structure of the feature extractor, the structure of ASPP consists of the atrous convolution kernel, and the encoder–decoder is the key. This article will examine the atrous convolution, DeepLab V3+, and the structure of GNAC in detail.

2.1.1. Convolution Kernel

A convolution kernel is one of the core elements of a neural network. The structure and function of an ordinary convolution kernel is shown in Figure 1a. When processing images, the result of a weighted average of the pixel values of a small area in a given input image is used as a corresponding pixel in the output image. This process is called convolution, where the weight is defined by a function; this function is called the convolution kernel. The convolutional kernel is developed from a single large 13×13 convolution kernel used in AlexNet [21]. Neural networks contain convolution kernels of different sizes. Under the action of the entire neural network, the depth of the feature map and the information contained in a single pixel will be improved.

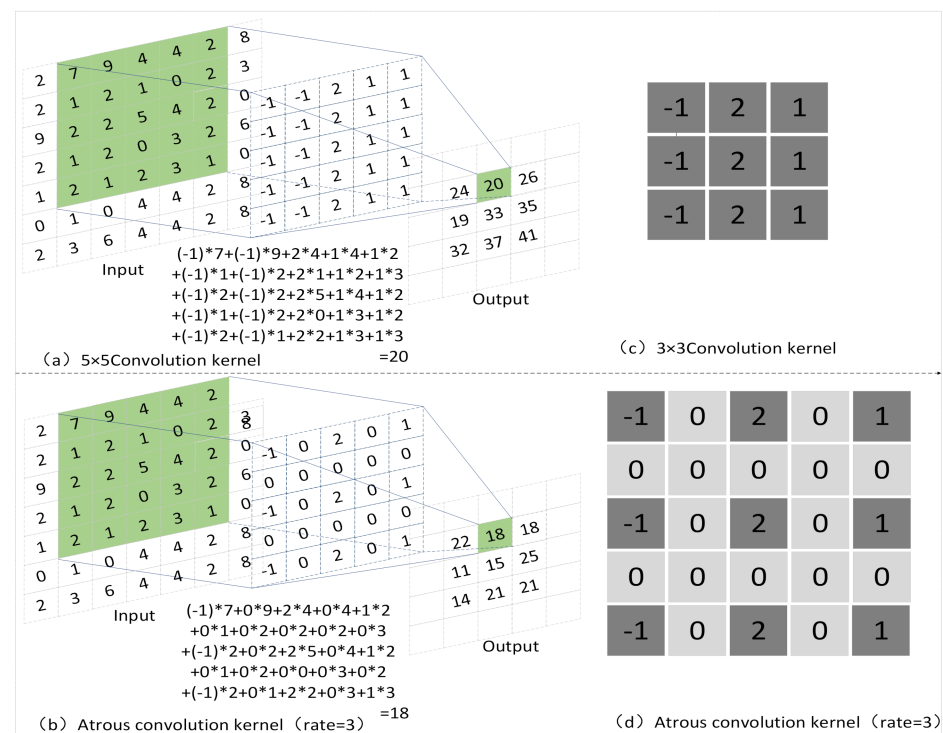


Figure 1. (a) The work process of ordinary convolution. (b) The work process of atrous convolution. (c) The size of ordinary convolution kernel. (d) The size of atrous convolution kernel.

After multiple convolutions, the receptive field of the feature map will rapidly decrease, resulting in a decrease in the final accuracy. In order to improve the receptive field of the feature map after convolution without increasing the parameters and the amount of calculation, we use the following hypothesis: the pixels that are closely adjacent to the target are almost the same, and all of them are included in the convolution operation, which will lead to excessive redundancy. It is better to skip one or more pixels to take information. It has been proposed to insert zero points in the adjacent parameters of the convolution kernel, which is called atrous convolution [22], as shown in Figure 1b. The use of atrous convolution kernels instead of some ordinary convolution kernels can improve the receptive field of the feature map with the same amount of calculation, and the overall trend of the feature map after convolution is consistent. As shown in Figure 1c,d, one should set the size of the input feature map as h_1 , the output feature map size as h_2 , the size of the convolution kernel as k , stride as s , padding as p , the rate of atrous convolution

kernel as r , and the equivalent convolution kernel of the atrous convolution kernel as k_e . From those parameters, we can calculate the size of the equivalent convolution kernel as:

$$k_e = k + (k - 1)(r - 1) \quad (1)$$

For the calculation of the same 9 significant figures, it is clear that the ordinary convolution kernel can only obtain the content of a 3×3 area, while the atrous convolution kernel with 9 significant figures can obtain the information of areas larger than 3×3 . The latter can be obtained by Formula (1):

$$k = \frac{k_e + r - 1}{r} \quad (2)$$

The size of the convolution kernel determines the amount of calculation during convolution, at least to a certain extent. As shown in Figure 1, when $rate = 2$ while performing a 5×5 convolution, and $k = 5$ for ordinary convolution and $k_e = 5$ for atrous convolution, the receptive field is the same, but the amount of calculation can be reduced when controlled.

2.1.2. DeepLab V3+ NETWORK

The emergence of neural networks solves some image processing problems, such as target recognition and semantic segmentation. However, the multiple convolutions of deep convolutional neural networks reduce the size of the feature map and make the feature map too small, limiting the size and accuracy of the feature map and restricting the effect of model. The DeepLab series [20,23–25], researched by Google, replaces ordinary convolution with hole convolution and improves the size and accuracy of the feature map without increasing the amount of calculation, thereby improving the model's hierarchical reliability, computing power, and ability to acquire micro-frame details. Aiming at the problem of detecting objects of different scales, atrous spatial pyramid pooling has been proposed (ASPP), as shown in Figure 2a. ASPP can obtain multi-scale image text information by performing different rates of atrous convolution on the picture at the same time, thus improving the ability to obtain different targets. In order to better obtain the information of all targets in the images, DeepLab V3+ [20] takes the encoder–decoder as the main structure, as shown in Figure 2b. The encoder convolution adopts atrous separation convolution, which can significantly reduce the computational complexity of the model and maintain similar convolution performance. Moreover, the last layer of convolution is replaced with ASPP, and the high-level feature device analyzes global semantics to facilitate the use of the atrous separation convolutional layer to extract features at any resolution. Then, through the decoder for two consecutive quadruple upsamplings, the boundary information is gradually restored and semantic segmentation is realized. The overall structure is shown in Figure 2c. The DeepLab series introduces the atrous convolution kernel, constructs atrous spatial pyramid pooling, and uses it in the encoder–decoder structure to perform semantic segmentation on images with different size and color targets with a low amount of calculation. The research on target capture strategy generation not only requires the algorithm model to have the ability to accurately identify and segment a target's details, but it also requires the model to have a faster calculation speed. From people's grasp experience, we know that tiny details greatly influence grasp strategy and success. On the other hand, each target has its own characteristics, and the algorithm model needs to quickly output a reasonable strategy. The DeepLab V3+ algorithm has high semantic segmentation accuracy, low calculation volume, and fast running speed; thus, it has great application potential in the field of target capture strategy generation.

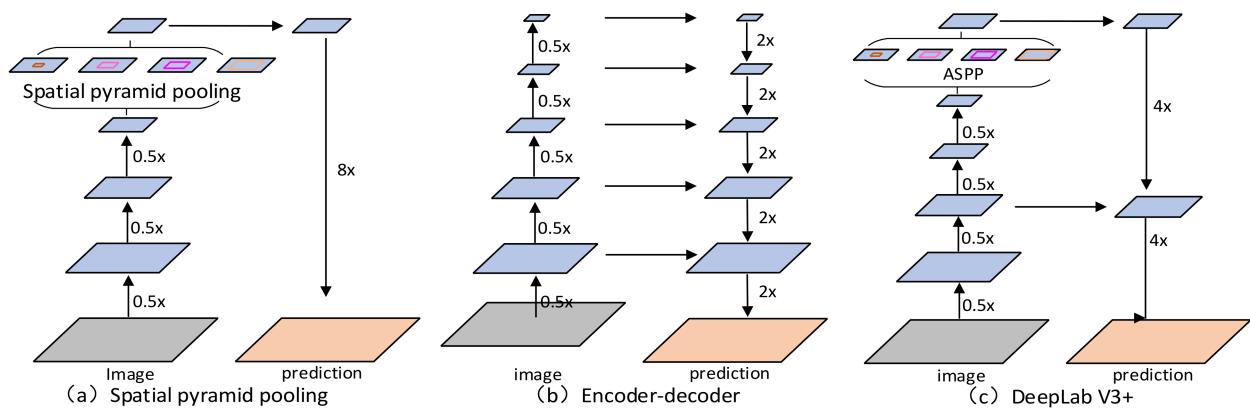


Figure 2. The structure of DeepLab V3+ and relative networks.

The grasping strategy of this article divides the entire surface area of the target to be grasped into a graspable and a non-graspable area, and converts the problem into a semantic segmentation problem. Additionally, based on the excellent performance of DeepLab V3+ in semantic segmentation, this paper applies DeepLab V3+ to the generation of grasping strategy by adjustments and training, and finally realizes the prediction of the target grasping strategy.

2.1.3. Network Architecture

The network structure of this article consists of two parts. The first is the convolution feature extraction layer, which is based on the structure of DeepLab V3+, that takes a 320×320 pixel RGB picture as input. After convolution and spatial pyramid pooling, a feature map with a size of 80×80 and a channel of 304 is generated. This feature map is used as the input for the second component of the prediction. As shown in Figure 3, the network structure designed in this paper is composed of a CNN layer, an ASPP layer, a pooling layer, a connected layer, and an upsampling layer.

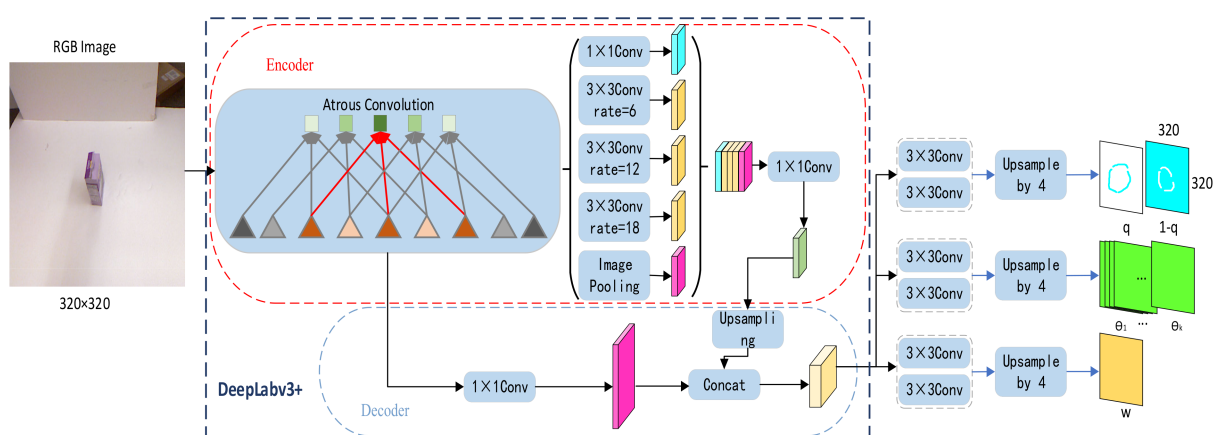


Figure 3. The construction of GNAC.

The CNN layer is composed of 4 convolution groups. The images that have passed through the CNN layer are input to the ASPP layer, which comprises a parallel 1×1 convolution, three 3×3 convolution layers with rates of 2, 4, 6, and a pooling layer. The output of this parallel layer is combined to perform 1×1 convolution and, finally, to perform an upsampling value of 4 to obtain the feature map. Simultaneously, the output of the CNN layer is connected to the 1×1 convolutional feature map and the upsampling feature map to obtain the final feature map. The grasping strategy consists of three output values

(grasp confidence, angle, and width), but DeepLab V3+ only outputs one predicted value, the semantic segmentation image. As such, the network model in this paper adjusts the prediction part that performs the 80×80 feature map with channel 304, and it does so simultaneously through 3 parallel CNN layers and 4 upsampling layers to output the final prediction. The CNN layer composed of two parallel 3×3 convolution groups generated the prediction of grasp confidence, angle, and width.

The neural network proposed in this paper outputs three predictors: grasp confidence, width, and angle. The grasp confidence predictor produces two output values—the possibility of being grasped and the possibility of not being grasped. In this series of grasp confidence, the highest possibility of being grasped, oriented base-fixed triangle representation, is the final output. The angle predictor will generate k output values, where the i th value represents the probability that the i th grasp angle can be grasped, and the output width predictor will output a value that is the graspable width. These three predictors also adopt different loss functions because of different predicted values and defined positive labels. The label of grasping confidence divides all areas of the target into graspable and non-graspable positions, so the prediction of confidence is a binary classification problem. Predicting grasp confidence $Loss_{abl}$ adopts a softmax cross-entropy function as a loss function:

$$Loss_{abl} = \sum_{n=1}^N p_n \log_2 \frac{1}{p_n} \quad (3)$$

$$p_n = -\log x_n \quad (4)$$

with x_n denoting the grasp confidence. Predicting the grasp angle is a multi-label, multi-classification problem, so loss function for predicting the grasp angle ($Loss_{ang}$) takes a sigmoid cross-entropy function as the loss function:

$$Loss_{ang} = -\frac{1}{N} \sum_{n=1}^N [p_n \log(\widehat{p}_n) + (1 - p_n) \log(1 - \widehat{p}_n)] \quad (5)$$

$$p_n = \text{sigmod}(x_n) = \frac{1}{1 + e^{-x_n}} \quad (6)$$

Predicting grasp width is a regression problem, so the mean square error function is selected as the loss function ($Loss_{wid}$).

$$Loss_{wid} = \frac{1}{N} \sum_{n=1}^N (x_n - \widehat{x}_n)^2 \quad (7)$$

with N denoting the number of the predicted grasp angle, x_n denoting the n th predicted angle, and \widehat{x}_{nth} denoting the ground truth label. The sum of loss ($Loss_{all}$) is:

$$Loss_{all} = Loss_{abl} + w_1 \times Loss_{ang} + w_2 \times Loss_{wid} \quad (8)$$

2.2. Generating Grasp Strategy

In the past, most research on grasp detection has been based on target recognition and target classification using RGB images as input, thus generating a grasp rectangular frame on the image.

The neural network established by Joseph Redmon et al. [26] makes adjustments in AlexNet [21] and establishes a network that directly return to the grasp detection through RGB-D images, and can directly output coordinates, such as width, height, and angle. Based on YOLO, Xu et al. [18] divided the entire input image into small 13×13 small, returning the center of each target to a small 13×13 small and using the point in the small grid as the center to return to a circle, with a calibrated diameter as the grasp trajectory; this method achieved target detection at the same time as the grasp strategy was generated. The research of these scholars has opened the way for grasping strategy prediction with

deep learning, and has greatly improved the prediction speed and accuracy of grasping strategy generation. In the past, scholars' research mostly used the five-dimensional grasp representation [4], as shown in Figure 4a:

$$G = \{x, y, \theta, h, w\} \tag{9}$$

with (x, y) representing the coordinates of the graspable center position, θ representing the angle between the grasp and the horizontal line, h representing the height of the grasp rectangle, and w representing the width of the grasp rectangle. The parallel plate gripper is represented by the rectangle [4,7].

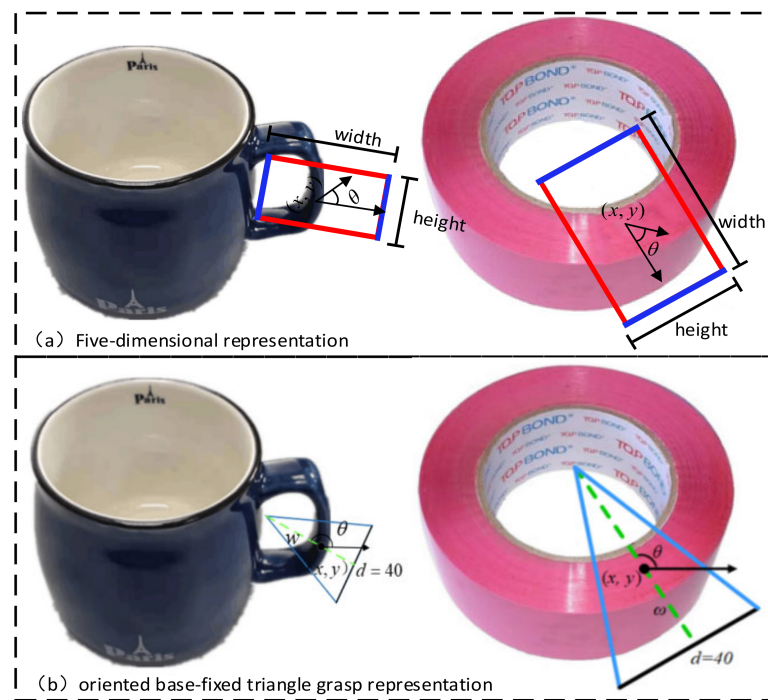


Figure 4. Grasp representation.

When the parallel plate gripper actually grasps the target, as in grasping a sphere, contact with the target theoretically occurs on only two points, and it only generates force in two directions or two points. Overall, the grasp stability is poor, and the five-dimensional representation used above is a region-level representation, which does not provide enough accuracy. In response to this problem, we used an asymmetric three-finger gripper to grip the target. Compared to the parallel plate gripper, the asymmetric three-finger gripper was more stable and practical, and it was divided into two ends: one finger at one end and two fingers at the other end. It could be used for a smaller location on one side of the target and a larger location on the other side of the target. There were also more applicable scenarios for this gripper. The parallel plate gripper had a symmetrical structure, so its grasp angle was $[0, \pi]$, while the three-finger gripper was not symmetrical in structure, so the range of the grasp angle was $[0, 2\pi]$. The three-finger gripper had three action points, which could be completely expressed by pixel-level representation and the oriented base-fixed triangle; thus, the directional triangle representation was used as the representation method, as shown in Figure 5.

$$G = \{x, y, \theta, d, w\} \tag{10}$$

where (x, y) represents the position coordinate of oriented base-fixed triangle, θ is the angle between the vertical line of the triangle and the horizontal line, d represents the width of the two finger directions, and w represents the position of the horizontal line connected by the two fingers from the vertex.

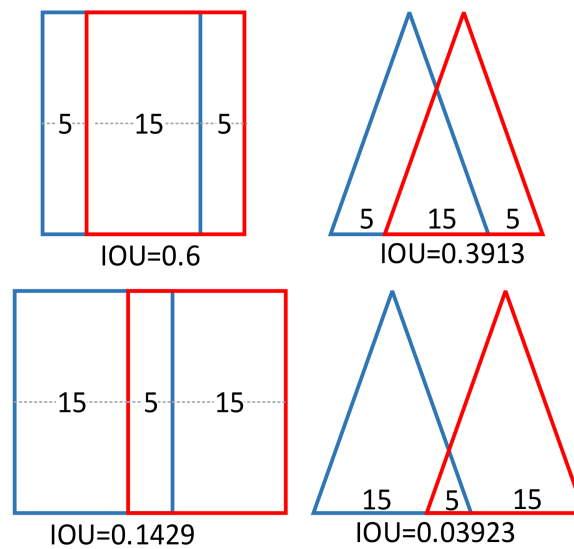


Figure 5. The difference between the IOU of the rectangle (**left**) and the IOU of the triangle (**right**) at the same grasping point.

2.3. Experiment Setup

2.3.1. Dataset

We used the Cornell Grasp dataset as a benchmark to train and verify the model. The Cornell Grasp dataset includes 885 RGB-D images, 5110 human-labeled positive labels, and 2909 negative labels. The label of the Cornell Grasp dataset is oriented to neural networks that use five-dimensional representation, and the label is multiple rectangles at positions that can be grasped on the image. This kind of label was not suitable for the asymmetric three-finger gripper used in this article. As the blueprint, we adopted the dataset used by Wang et al. [1] from the Cornell Grasp dataset, and we modified it to the oriented base-fixed triangle labeled dataset.

Since the Cornell Grasp dataset only includes 885 RGB images, it is relatively small compared to other deep learning datasets. In order to solve this problem, we adopted two methods. The first used data enhancement technology to expand the dataset, such as random rotation and cropping of the images in the dataset, which allowed us to enlarge or reduce picture size and adjust the picture to 320×320 pixels. The second used ResNet-101 pretrained by ImageNet as the feature extractor to reduce the training scale.

2.3.2. Implementation Details

For training and testing, our models ran on a laboratory workstation (CPU:i9-9900X, GPU:2080Ti*2, RAM:128G, ROM:512G SSD+6T HHD). Due to the memory limitation of the GPU, the batch size was set to 8, and each model was trained end-to-end for 1500 epochs. We used the Adam optimizer to optimize GNAC, and the learning rate (lr) was 0.001. Moreover, lr decayed stepwise at a rate of 0.5 times in the range [200, 500, 800, 1000] of epochs.

2.3.3. Test Methods Metric

The general benchmark consisted of intersection over union (IOU) and the difference between the predicted grasping angle and the ground grasping angle. When the IOU value was greater than 0.25, and the difference between the predicted grasping angle and the ground grasping angle was less than 30° , the predicted grasping strategy was considered correct. IOU is shown in Formula (11):

$$IOU = \frac{SR \cap GT}{SR \cup GT} \quad (11)$$

SR represents the predicted grasp area and GT represents the ground grasp area. As shown in Figure 5, when the positions of the action points were roughly similar, the asymmetric three-finger gripper used in this article had a smaller IOU value.

As in the past grasp detection methods, we randomly selected 75% of the data in the dataset as a training set and the remaining data as a test set. There were two test methods:

1. Image-wise split (IW): The image dataset was randomly divided into a test set and a training set. This method was mainly used to test the neural network's ability to recognize previously seen targets in new positions and new directions;
2. Object-wise split (OW): We divided the dataset at an object instance level. All the images of an instance were put into the same set. This method was mainly used to test the neural network's ability to recognize new targets.

3. Results

In the feature extraction component of the neural network, we found that the atrous convolution value in ASPP determined the ability to obtain the features of differently sized targets in the image, and determined the ability of the neural network to obtain and analyze the small edges of the target. In order to apply the ability of ASPP in the generation of grasping strategies more appropriately, this paper experimented with multiple sets of different values of ASPP for training, and the final model was obtained and tested with two different standards. As different ASPP values had the same calculation amounts, their speeds were all the same and could not be repeated. The results are shown in Table 1. The neural network with different ASPP ratios was trained, and the total loss value and the changes in the three sub-loss values during the training process are shown in Figure 6. Among them, $2 \times n$ represented the atrous convolution ratio in ASPP [1,2,4,6], $3 \times n$ represented [1,3,6,9], $6 \times n$ represented [1,6,12,18], and NO_ASPP represented the skipping of the ASPP module; the loss value of the neural network with different ratio values had the same trend and value in the training process. The total loss value and the three sub-loss values (grasp confidence, grasping width, and grasping angle) all steadily decreased with a certain shock. We knew that when the training was performed for 500 epochs, the decrease in the loss value slowed down. We also knew that when the training period was performed for 1400 cycles, the differential decrease in the loss value approached 0. During the training process, the real-time test results of the model and the loss value trend were roughly the same. When the training reached 500 epochs, the speed of the grasp strategy increased but the accuracy generated by the model decreased. The models trained under the three sets of rates gradually reached a peak of about 1500 epochs. Among them, the neural network with the ratio of [1,2,4,6] after 1470 epochs of training reached the best test accuracy of 0.97, and the result was also the best in the subsequent verification experiments. It is worth noting that the loss value of the neural network that skipped the ASPP module during the training process was significantly lower than the training loss value of the neural network with the ASPP module.

Table 1. Grasp accuracy of different rates of ASPP.

The Rates of ASPP	500 Epochs		1500 Epochs	
	IW (%)	OW (%)	IW (%)	OW (%)
[NO_ASPP]	92.27	94.06	95.45	94.97
[1,2,4,6]	91.74	89.54	97.71	96.82
[1,3,6,9]	91.32	90.00	96.35	96.82
[1,6,12,18]	90.41	90.00	95.43	95.00

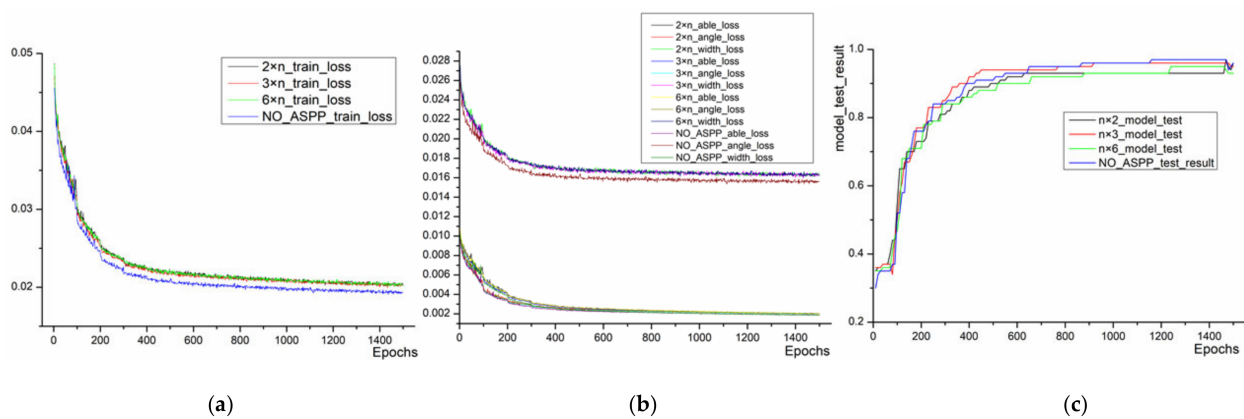


Figure 6. (a) The change process of the total loss value of different rates of ASPP; (b) the change process of the three sub-loss values of different rates of ASPP; (c) the training model test result in the training process of different rates of ASPP.

After a later verification experiment, as shown in Table 1, the research showed that, when the convolution ratio of the rate of ASPP was [1,2,4,6], the results obtained were the best. The result of the image-wise split was 97.71% and the result of the object-wise split was 96.82%.

4. Discussion

As shown in Table 1 above, we have adopted multiple sets of ASPP with different ratios. It can be seen that, in the later test, [1,2,4,6] had the best effect. We suggest that a reason for this is that the target of the Cornell Grasp dataset is large, as shown in Figure 7. Furthermore, the overall background is relatively simple. The object in the Cornell Grasp dataset is shown in Figure 7. The overall background is relatively simple, and the object belongs to the big object. When using the ASPP ratio [1,2,4,6], which was not too large, the atrous convolution value could make the neural network have a suitable receptive field, and it could achieve a better recognition effect. The effect of the atrous convolution is mentioned in Section 2. It can increase the receptive field without a large amount of calculation but, when the ratio of the selected ASPP is too large, it will cause the atrous convolution kernel to become a 1×1 convolution kernel. By adding too many 0 values to the convolution kernel, the numerical trend of the feature map could also be affected after convolution, resulting in a reduction in the accuracy of the generated grasp strategy. After 500 epochs of training, the neural network that trained without an ASPP module had the best test result, and its total loss was also relatively small, as shown in Figure 6a; however, after 1500 cycles of training, the model test result was not satisfactory. Our analysis suggests that this result is due to the fact that, because there was no ASPP module, the overall parameter was smaller, meaning a certain amount of overfitting resulted in a low loss value. That said, the effects of training and testing in the later test were poor. Different values of ASPP had different feature analyses and acquisition capabilities for targets of different sizes. In practical applications, the object may be large or small in a given scene. Further research needs to be carried out in this area. ASPP with an adaptive scale value can achieve a better grasping prediction accuracy for targets of different sizes in real time.

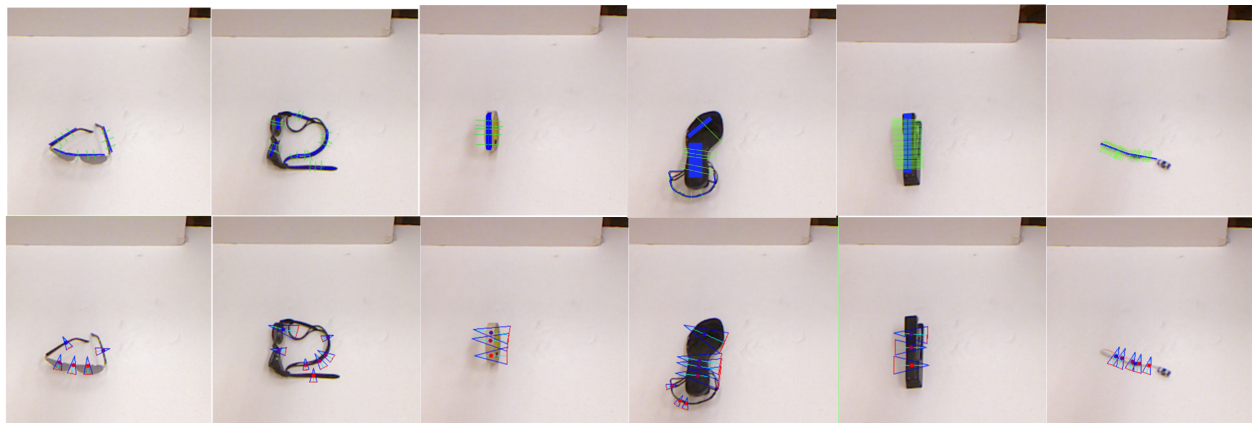


Figure 7. Grasp detections in the Cornell Grasp dataset. The first column is the label image and the second column is the predicted grasp strategy.

The final model trained in this paper was evaluated by the two detection standards noted above. A comparison between the test accuracy of the training model in this paper and the test accuracy of previous studies is shown in Table 2. The test results obtained in this article were compared with previous studies. This paper primarily uses the semantic segmentation algorithm as its core, and produces pixel-level results, which are clearer than the region-level output studied in the past; thus, this paper improves the accuracy of grasp detection. In this paper, the semantic segmentation of 3D point cloud data was mainly applied to grasp detection in relation to image grasp strategy generation, which broadens the way for future investigations into grasp strategies.

Table 2. Test results on the Cornell Grasp dataset.

Approach	Year	Grasp Representation	Algorithm	Accuracy (%)		Speed (ms)
				IW	OW	
Jiang [4]	2011	Five-dimensional representation	Fast Search	60.5	58.3	5000
Lenz [7]	2015		SAE, struct.reg	73.9	75.6	1350
Guo [3]	2017		ZF-net	93.2	89.1	-
Zhang [19]	2019		ROI-GD, ResNet-101	93.6	93.5	39.75
Chu [17]	2018		ResNet-50, Deep Grasp	96.5	96.1	20
Kumra [16]	2020		GR-ConvNet-RGB-D	97.7	96.6	20
Yanan [15]	2020		ResNet-50(RGD)	95.6	97.1	-
Wang [1]	2020	Oriented base-fixed triangle	SGDN	96.8	92.27	19.4
OURS	2021		GNAC	97.71	96.82	19.1

The research in this article was based on an asymmetric three-finger gripper. When using this model, there were three points of action on the target and, when facing a target with a curved surface, the gripping stability and practicability of the three-finger gripper were both stronger. In addition, as shown in Figure 5 above, when the points of action were similar, the IOU of the directional triangle grasp notation used in this article was smaller. Due to this disadvantage, the accuracy of GNAC still had to be improved in its two judgments. At the same time, the 19.1ms grasp strategy generation speed also met the requirements for image processing during the actual grasping. As it uses a three-finger gripper, which is different from mainstream grippers, as well as pixel-level oriented base-fixed triangle representation, this representation is only used in the Cornell Grasp dataset.

The main defect of this article is its lack of training and verification regarding datasets. For instance, Jacquard datasets and Dex-Net datasets could also have been used. Secondly, further experiments at a later period, as shown in Figure 8, show that our model struggled to generate a successful grasping strategy due to the interference of light when facing transparent bottles, stainless steel, and other objects that reflect or transmit light.



Figure 8. The above eight images are visualizations of detection in more realistic and complex scenes.

As shown in Figure 7, we demonstrated some of the grasp strategies that predicted success, namely the behavior label map and the second behavior model, both of which generated grasp strategies. Compared with the previous networks that used regression as their main idea, the grasping strategies generated by the model trained in this article were more focused on a target area suitable for grasping rather than a number of locations. At the same time, the grasping strategy was not only applicable to the position of the main body of the target, but it also successfully generated a suitable grasping strategy in relation to the target's artificially accustomed grasping position; shoelaces are a good example of this. The idea and advantages of semantic segmentation were successfully transferred to the robot grasping strategy.

In order to test the generalization ability of our model on an object-wise split in a complex environment, in which some targets were partially contacted, fully contacted, or partially covered, we collected some everyday objects for use in a dataset and tested them. Figure 8 shows the partially successful result. From Figure 8, we can see that our model can successfully generate grasp strategies for objects that are not available in the Cornell Grasp dataset. When the capture target had a small area of occlusion, the model in this paper still worked.

5. Conclusions

Compared with previous grasp detection algorithms based on object recognition algorithms and five-dimensional representations, this paper proposed a grasp detection algorithm based on semantic segmentation, and further explored ASPP with different ratios. An asymmetric three-finger gripper and an oriented base-fixed triangle representation—one that could cope with more grasping situations—achieved a more stable and accurate grasping effect on the target, realized the robot's grasping of multiple targets in different positions, and improved the intelligence, stability, and accuracy of the grasping robot. It promoted the development of grasping robots facing a variety of industrial products.

Although the current grasp strategy generation algorithm has good predictive accuracy, it mostly generates a predicted grasp position for a single object under a simple background. In actual application scenarios, there may be situations in which the grasped target is occluded, and there are multiple targets in the scene. Pertinent research in this

area can promote greater development of grasp detection strategies. As is commonly understood in this field, the reasonable use of multi-finger grippers can greatly improve the stability and applicability of a gripping action. This article is aimed at the development of three-finger grippers. The grasp detection of four-finger and five-finger humanoid grippers may be a future research direction.

Author Contributions: All authors contributed to this work. S.J. designed the research. Y.M. and L.C. processed the corresponding data. S.J. and Q.B. wrote the first draft of the manuscript. J.Y. gave some guidance about methods. S.L. revised and edited the final version. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of science and technology, China No. 2018AAA0101803; the Major Science and Technology Project of Guizhou Province, China No. [2019]3003; and the Science and Technology Project of Guizhou Province, Chin No. [2015]4011, [2017]5788.

Data Availability Statement: The data used to support this study's findings are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, D. SGDNet: Segmentation-Based Grasp Detection Network For Unsymmetrical Three-Finger Gripper. *arXiv* **2020**, arXiv:2005.08222.
2. Chan-Viquez, D.; Hasanbarani, F.; Zhang, L.; Anaby, D.; Turpin, N.A.; Lamontagne, A.; Feldman, A.G.; Levin, M.F. Development of vertical and forward jumping skills in typically developing children in the con-text of referent control of motor actions. *Dev. Psychobiol.* **2020**, *62*, 711–722. [[CrossRef](#)] [[PubMed](#)]
3. Guo, D.; Sun, F.; Liu, H.; Kong, T.; Fang, B.; Xi, N. A hybrid deep architecture for robotic grasp detection. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1609–1614.
4. Jiang, Y.; Moseson, S.; Saxena, A. Efficient Grasping from RGBD Images: Learning Using a New Rectangle Representation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3304–3311.
5. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. *SSD: Single Shot MultiBox Detector*; Leibe, B., Matas, J., Sbebe, N., Welling, M., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
6. Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; Rother, C. Learning 6D Object Pose Estimation Using 3D Object Coordinates. *Trans. Petri Nets Other Models Concurr. XV* **2014**, *2*, 536–551. [[CrossRef](#)]
7. Lenz, I.; Lee, H.; Saxena, A. Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* **2015**, *34*, 705–724. [[CrossRef](#)]
8. Kumra, S.; Kanan, C. Robotic grasp detection using deep convolutional neural networks. In Proceedings of the 2017 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 769–776.
9. Bicchi, A.; Kumar, V. Robotic grasping and contact: A review. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 348–353. [[CrossRef](#)]
10. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [[CrossRef](#)]
11. Bai, Q.; Li, S.; Yang, J.; Song, Q.; Li, Z.; Zhang, X. Object Detection Recognition and Robot Grasping Based on Machine Learning: A Survey. *IEEE Access* **2020**, *8*, 181855–181879. [[CrossRef](#)]
12. Chen, J.; Kira, Z.; Cho, Y.K. LRGNet: Learnable Region Growing for Class-Agnostic Point Cloud Segmentation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2799–2806. [[CrossRef](#)]
13. Monica, R.; Aleotti, J. Point Cloud Projective Analysis for Part-Based Grasp Planning. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4695–4702. [[CrossRef](#)]
14. Zhuang, C.; Wang, Z.; Zhao, H.; Ding, H. Semantic part segmentation method based 3D object pose estimation with RGB-D images for bin-picking. *Robot. Comput. Manuf.* **2021**, *68*, 102086. [[CrossRef](#)]
15. Song, Y.; Gao, L.; Li, X.; Shen, W. A novel robotic grasp detection method based on region proposal networks. *Robot. Comput. Manuf.* **2020**, *65*, 101963. [[CrossRef](#)]
16. Kumra, S.; Joshi, S.; Sahin, F. Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network. *Comput. Sci.* **2019**, arXiv:1909.04810v4.
17. Chu, F.-J.; Xu, R.; Vela, P.A. Real-World Multiobject, Multigrasp Detection. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3355–3362. [[CrossRef](#)]
18. Xu, Y.; Wang, L.; Yang, A.; Chen, L.; Ynag, A. GraspCNN: Real-Time Grasp Detection Using a New Oriented Diameter Circle Representation. *IEEE Access* **2019**, *7*, 159322–159331. [[CrossRef](#)]

19. Zhang, H.; Lan, X.; Bai, S.; Zhou, X.; Tian, Z.; Zheng, N. ROI-based Robotic Grasp Detection for Object Overlapping Scenes. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4768–4775.
20. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *6*, 84–90. [[CrossRef](#)]
22. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2015**, arXiv:1511.07122.
23. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *Comput. Sci.* **2014**, *4*, 357–361.
24. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]
25. Chen, L.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
26. Redmon, J.; Angelova, A. Real-time grasp detection using convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1316–1322.