Proceedings of the 2023 International Conference on Mechatronics and Smart Systems
DOI: 10.54254/2755-2721/10/20230194

Planar grasp detection using generative multi-column convolutional neural networks

Haikun Yuan[1, 3, †], Haipeng Huang[2, †]

[1]Robotic Engineering, Xidian University, 10701, China
[2]Computer science, Nanjing University of Posts and Telecommunications, 10293, China

[3]20049200117@stu.xidian.edu.com
[†]These authors contributed equally

Abstract. This paper presents an accurate, real-time generative multi-column convolutional neural network for two-dimension planner grasp detection. A multi-column structure is used to improve the ability of the network to extract features of different scales. Three parallel channels with three different reception fields can make the network learn different scales of features, which result in the network is more adaptable to a complex environment. This network overcomes some shortcomings like long computing period and by using a generative method rather than sampling grasping candidate. Our network has short computing time because of its light structure, which can be deployed in some close-loop situations. While training the network, we find that some labels in Cornell database is not suitable for the planner grasping detection training, for some specific labels represent different angel of grasping. By comparison with other models, our model's accuracy on Cornell grasping dataset is higher, reaching 94%, and our model runs at 13 frames per second.

Keywords: grasp detection, generative method, multi-column CNN, Cornell grasping dataset.

1. Introduction

To reduce the labour costs, robotic manipulators have been widely deployed in many situations like warehouses and factories, etc. With the development of artificial intelligence and intelligent robots, robotic manipulators have been able to make decisions according to their "own experiences." The grasping task is one of the most significant parts of robotic manipulators. To complete this task, the robots should generalize the most suitable grasping rectangle while the reacting speed should also be quick, for the environments where robots work are always dynamic, and the robots should have the ability to infer fast and robust grasps for any objects it might encounter. The robotic manipulators with long computation time can only be deployed in some static environments. So, the reaction speed is also an essential factor that determines whether the robotic manipulators can be used in dynamic environments. Robotic grasping has been researched for decades. Generally, the grasping approach can be divided into two main categories, analytic [1][2] and data-driven [3][4] method.

The analytic method analyses the mechanical characteristics of target objects according to their geometric characteristics. However, this method is hard to deploy in a dynamic environment because of

DOI: 10.54254/2755-2721/10/20230194

its complexity. Some parts of the input images might be covered and there might be some noises too, which is hard to generalize a correct grasping rectangle.

Data drive algorithms are among the popular tools in robotics perception, planning and control [5][6]. There are two strategies of machine learning (data-driven method) for robotic grasping. One is discriminate approaches [7][8], and the other is generative approaches [9][10]. Discriminate approaches train the CNN to give a score of each potential grasping point, and these potential grasping points are called grasping candidates. These approaches discriminate several grasp candidates, give each candidate a grasping score, and choose the highest one to grasp [10]. However, the reacting time is not short (over 2 seconds per image) [7] because of its multiple forward passes and massive parameters. The basic strategies of generative approaches are generating grasp rectangles and grasp angles. And jaw grippers are able to grasp the target objects according to the angles and rectangles generated by the neural networks.

A generative convolutional neural network is utilized to tackle the problems of the dynamic environment and recognize the best grasping points for robotic manipulators. We choose the generative neural network because of its short reaction time. Unlike previous approaches [11][12] which generate grasping rectangles, our model generates three grasping images to represent the grasping results. The reason why this CNN model is quick is that it does not sample and rank the gasping candidates but generates grasping poses on a pixel-wise basis. Pixel-wise basis means that the CNN network will evaluate each pixel's grasping quality. The multi-column structure with different reception fields can characterize different scales of features, improving the accuracy of the network.

The contributions of this paper are as follows. 1) We present a generative multi-column convolutional neural network (GM-CNN) structure. 2) We applied a Huber loss function instead of MSE loss function. 3) A dropout layer is used in our network, which is useful to avoid overfitting.

2. Problem formulation

Two dimensions planar grasping means the object is put on the working plane and the grasp is from one direction. The problem being addressed is how to effectively grasp unfamiliar objects using an antipodal gripper. To simplify the grasping action, we define that the gripper is perpendicular to the working plane (Figure 1).

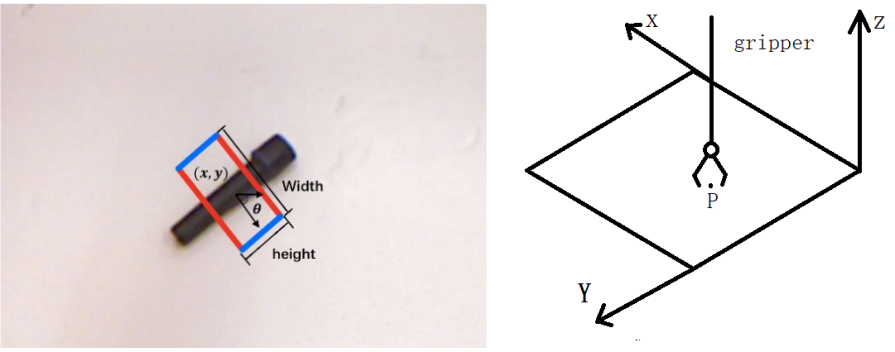

Figure 1. Four grasping parameters and the robot gesture.

We denote the grasping pose as:

$$G_r = (P, \theta_r, W_r, Q) \tag{1}$$

where $P = (x, y)$ is the center of the grasping rectangle, also the gripper's center position. The values of θ_r are expressed in the range of $\pm \pi/2$, as the antipodal grasp is symmetric around these angles. The variable W_r denotes the width of the gripper's opening. The variable Q takes a value within the range of 0 and 1, it imply a probability of a successful grasp execution.

Because the input of our network is n-channels images $I = \mathbb{R}^{n \times h \times w}$ whose height are h and width are w, the grasping pose can be defined as:

DOI: 10.54254/2755-2721/10/20230194

$$G_i = (x, y, \theta_r, W_r, Q) \tag{2}$$

where (x, y) represents the location of the grasping point within the image coordinate system, and the other parameters are similar to those in Equation 1. According to the image frame to the robotic base frame transformation, G_i can be converted to G_r and robot can know how to grasp target object.

3. Planar grasp detection

Firstly, we use Cornell and Jacquard grasping dataset for network training. Since the quantity of Cornell grasping dataset is small, we augment the dataset to a bigger one. After the dataset is prepared, a pre-processing program converts the depth data in dataset to the depth images, then it chops both RGB and depth image to 300×300 pixels. Each image comes with a file which record the coordinate of the grasping rectangles. According to the coordinate information, three images are generated. And they are used to calculate the loss function.

Secondly, 300×300 RGB and depth images are fed to GM-CNN which includes convolutional layer, pooling layer, dropout layer, upsampling layer etc. The front several convolutional layers extract features of the input images and the other convolutional layers create the predicted pose images. The difference between the predicted grasping image and the image generated by pre-processing program is called a 'loss', which is propagated backward. The 'loss' function is important and will be discussed in following sections.

Finally, the GM-CNN outputs four 300×300 images which are quality image, width image, COS image and SIN image. A post-processing program predicts a best oriented grasp rectangle according to these images, compare it with the correct grasping rectangle in dataset, and calculate an accuracy of the correct grasp.

3.1. Grasp representation

In our experiment, the input images consist of 4 channels: RGB images and depth image. These images are denoted as $R(red\ channel), G(green\ channel), B(blue\ channel), D(depth\ channel)$. The output of our network includes 3 channels, which represent grasping quality (Q), rotation angle (Φ), and grasping width (W). And the scales of output and input channels are the same. Image Φ is the set of the angle of grasp rectangle of each point. Image Q is the set of the grasp quality of each point. Image W is the set of the grasp width of each point.

3.2. Grasp evaluation

It is important to define the quality of the network's prediction. Because the grasp rectangles in the Cornell database are defined manually, so there is no standard to define the grasp quality. In previous works, the grasp quality is defined by the distance between the center of the predicted rectangle and the grasping rectangle of ground truth. If the distance between centers falls below the specified threshold value, it is deemed successful. But this approach is not sensible because it does not take into consideration the grasp width and grasp angle, which are also significant in antipodal grasping. So, we use another approach to evaluate the grasping point. The predicted point will be defined as a good grasp if the given condition is satisfied with both 1) the angle between a and b is less than 30°, and 2) the given index J is less than 0.25, shown as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

where A is predicting grasps and B is ground truth. In the experiment process, the grasping accuracy of our network is defined by the above method.

DOI: 10.54254/2755-2721/10/20230194

3.3. Multi-column architecture

To make our network reach a higher accuracy, we apply multi-column architecture to our network (shown in Figure 2). The original image is sent to three parallel branches whose kernel sizes are $3 \times 3, 5 \times 5, 11 \times 11$ separately. These three parallel convolutional branches extract features of different scales and a shared decoder to transform the deep semantic information into four grasping images.

Our model can capture different levels of information with various reception fields, which could learn features of different sizes in raw pictures. They correspond to filters of different sizes. Compared to the previous model [7] (single path CNN) our model can increase detection accuracy.

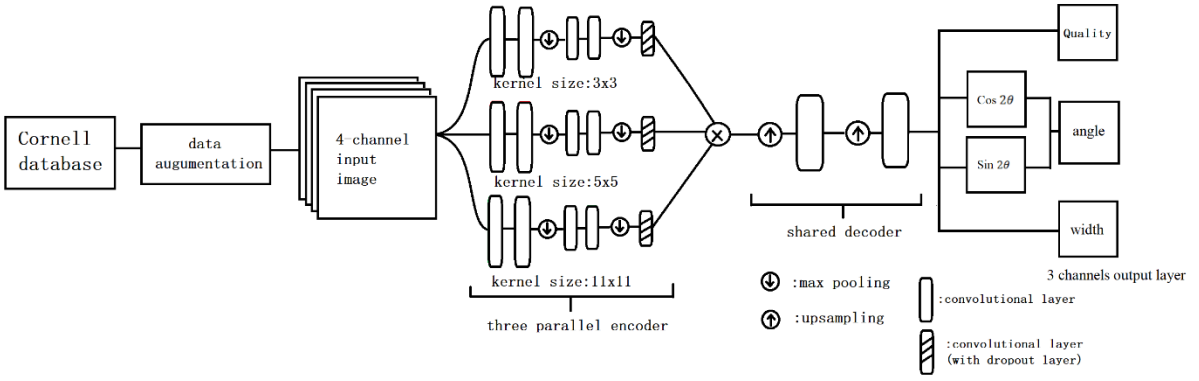

Figure 2. Our framework.

3.4. Data augmentation

The original Cornell dataset [13] only comprise 1035 images from 280 objects. Due to a low number of samples, we applied data augmentation to generate the training data by transforming the original input data, such as randomly rotating an image by a certain angle or flipping it horizontally or vertically. Thus, the model can be trained to recognize objects or patterns from different perspectives and orientations, making it more adaptable to different scenarios.

3.5. Dropout layer

To make the model more robust, we also applied the dropout layer [14] to our model. The dropout layer means that during each training process, a certain proportion of neurons will not be activated randomly (in our network, the rate is 0.1). The neurons which are not activated mean the forward pass and back-propagation are blocked. In other words, the "dead" neurons are removed from the network. So, each time an input is presented, the neural network has a different architecture. These approaches reduce the co-adaptations between each neuron, which has a random connection with each other and cannot rely on just one particular neuron.

3.6. Loss function

We applied Huber loss function to our model, which inherits from both squared error loss's (MSE) and absolute error loss (MAE)'s advantages. the Huber loss function is often used in situations where the data contains outliers that can have a significant impact on the performance of the model. The Huber loss function can help prevent these outliers from dominating the training process and thus lead to a more robust and accurate model. The Huber loss function is defined as:

$$\mathcal{L}(G, \hat{G}) = \begin{cases} \delta(G - \hat{G})^2 \ \ if \left|G - \hat{G}\right| < \delta \\ \delta\left|G - \hat{G}\right| - \frac{1}{2}\delta^2 \ \ if \left|G - \hat{G}\right| \geq \delta \end{cases} \tag{4}$$

The hyperparameter δ is the boundary of MAE and MSE loss. When the value of delta is small, the Huber loss function behaves like the MSE loss. However, when the value of delta is large, the function

DOI: 10.54254/2755-2721/10/20230194

behaves like the MAE loss. With the application of Huber loss function, our model can be more robust and accuracy.

4. Results

This section will cover the experimental results obtained from our network. In this process, we just execute the network on the computer and do not deploy it on the robotic arm. We evaluate our testing result with a series of objects, some from daily life, chosen from the Cornell grasping database. There are 11k objects, 54K images and 1.1M grasps in Jacquard dataset. Since this dataset is big enough, so data augmentation is not necessary. Table 2 is the test results comparing with other networks.

Table 1. Comparison between different networks and ours on the Cornell dataset.

Algorithm	Accuracy
Chance [7]	6.7
Jiang et al [7]	60.5%
Lenz et al. [7]	73.9%
MultiGrasp Detection [12]	88.0%
GM-CNN (our)	94.2%

Table 2. Comparison between different networks and ours on the Jacquard dataset.

Authors	Algorithm	Accuracy
Depierre [15]	Jacquard	74.2%
Morrison [16]	GG-CNN2	84%
Our	GM-CNN	85%

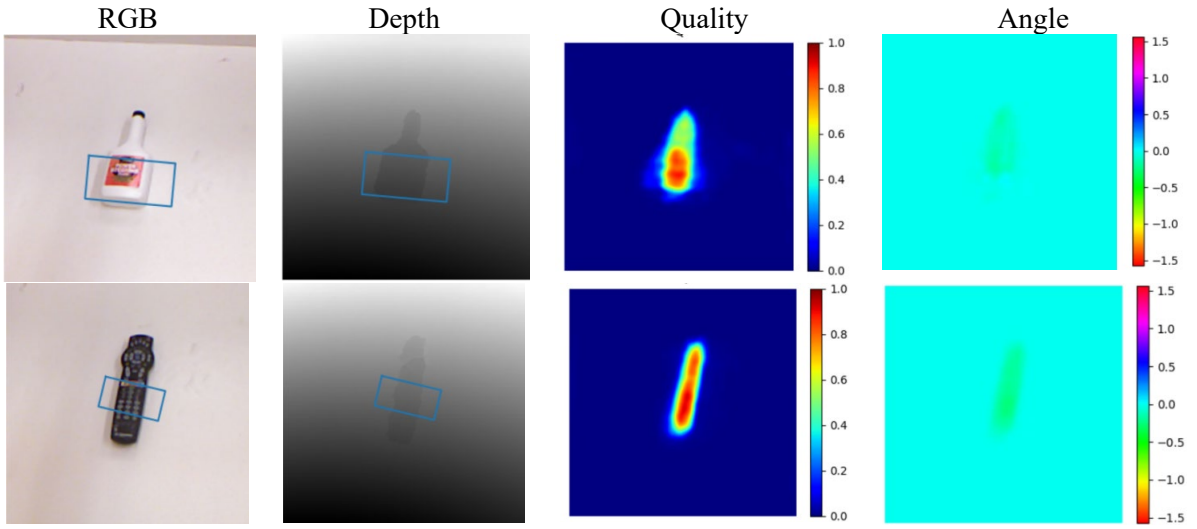

Figure 3. The correct grasp prediction images of our network and input images.

4.1. Training and testing process

Our implementation is with CUDA 12.0 and Python 3.6. The hardware is NVIDIA RTX2060 and CPUi7 3.6GHz, and the batch size is 8. And training process is 50 epochs till it converges. Training time is 45ms per batch and evaluation time is 75ms per image. Each epoch contains 8000 pictures. generated

DOI: 10.54254/2755-2721/10/20230194

from the Cornell database. The Cornell grasping database only has 885 RGB-D images of real objects with 5110 human labelled grasps. After zooming and cropping them (data augmentation), we finally increase the number of data up to 8840 pictures. The training results are shown in Figure 3, Table 1 and Table 2. In Table 1, we use image-wise (IW) evaluation. Table 1 shows the performance of our network and other networks. We obtained state-of art accuracy on object-wise of 94.2% in Cornell dataset.

In the RGB and Depth Images, the blue rectangles are the grasping rectangles obtained by the network. In quality images, the pixels' colour represents the value of the image. The more the colour of pixel is close to red, the more the value of this pixel is close to 1. In the angle image, the pixels' colour represents the angle of grasping.

4.2. Problem occurred on the cornell grasping dataset

The Cornell Database contains some labels that may not be suitable for our grasp training, as it provides images and labels from different grasping angles. For example, in Figure 4, the red rectangles represent the grasping rectangle in horizontal direction, but the other rectangles represent the grasping rectangle in vertical direction, which will mislead the network. Specifically, our model is set to detect grasps from a top-down view, then some of the labels in the Cornell Database may be incorrect or misleading. So we modified the Cornell grasping dataset and removed the grasping rectangles in horizontal direction.

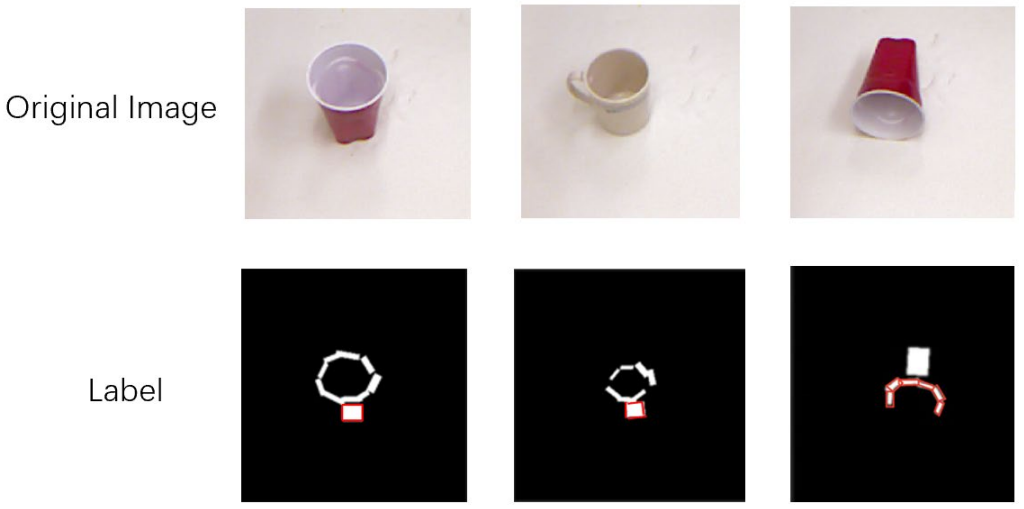

Figure 4. The red rectangles are the grasping rectangles whose angles are not perpendicular to the working plane.

5. Conclusions

Generative Grasping Multi-Column Convolutional Neutral network is presented in our essay, which directly generates grasping positions in pixel wises with a depth image and an RGB image. We apply 3 parallel channels with different filter sizes to extract features with various respective fields and spatial resolutions. Our structure overcomes the shortcoming that single-channel networks may ignore too large or too small semantic information in the input images. We also apply the dropout layer to our model to make it more robust. Our model achieves 94% accuracy when testing on the Cornell dataset. In future work, we will deploy our GM-CNN on a robotic arm and verify the network performance

References

[1] Anis S, Sahar E and Philippe B 2012 An Overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, vol 3 p 326–336.

[2] Bicchi A and Kumar V 2000 Robotic grasping and contact: a review *Icra Millennium Conference*

DOI: 10.54254/2755-2721/10/20230194

IEEE International Conference on Robotics & Automation Symposia vol 1.
[3] Bohg J, Morales A, Asfour T and Kragic D 2014 Data-driven grasp synthesis – a survey *IEEE Transactions on Robotics* vol 2 p 289-309.
[4] Lenz I, Lee H and Saxena 2013 A deep learning for detecting robotic grasps *The International Journal of Robotics Research* vol 4-5 p 34.
[5] Chen J, Xie Z and Dames P 2022 The semantic PHD filter for multi-class target tracking: From theory to practice *Robotics and Autonomous Systems* 149 103947.
[6] Chen J and Dames P 2022 Multi-class target tracking using the semantic phd filter In Robotics Research: The 19th International Symposium ISRR p 526-541.
[7] Mahler J , Liang J ,Niyaz S, Laskey M, Doan R, Liu X, Ojea JA and Goldberg K 2017 Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics.
[8] Morrison D, Corke P and Leitner J 2018 Closing the loop for robotic grasping: A Real-time, generative grasp synthesis approach Robotics: Science and Systems (RSS).
[9] Yun J, Moseson S and Saxena A 2011 Efficient grasping from RGBD images: learning using a new rectangle representation *2011* IEEE International conference on robotics and automation 3304-3311.
[10] Pinto L and Gupta A 2016 Supersizing self-supervision: learning to grasp from 50K tries and 700 Robot hours *2016* IEEE International Conference on Robotics and Automation (ICRA) 3406—3413.
[11] Redmon J and Angelova A 2014 Real-time grasp detection using convolutional neural networks Proceedings IEEE International Conference on Robotics & Automation 1316-1322.
[12] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition *Computer Scienc*e CoRR abs/1409.1556.
[13] Krizhevsky A, Sutskever I and Hinton G 2012 ImageNet classification with deep convolutional neural networks *Advances in neural information processing systems* vol 2 p 25.
[14] Alex K, Ilya S and Geoffrey E June 2017 ImageNet classification with deep convolutional neural networks *Communications of the ACM* vol 60 p 84–90.
[15] Amaury D, Emmanuel D and Liming C 2018 Jacquard: a large-scale dataset for robotic grasp detection RSJ International Conference on Intelligent Robots and Systems p 3511-3516.
[16] Morrison D, Corke P and Leitner J 2019 Learning robust, real-time, reactive robotic grasping *The International Journal of Robotics Research*. 39(2-3) 183-201.