Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 5512728, 12 pages
https://doi.org/10.1155/2021/5512728

Research Article

A Single Target Grasp Detection Network Based on Convolutional Neural Network

Longzhi Zhang and Dongmei Wu

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Dongmei Wu; wdm@hit.edu.cn

Received 21 May 2021; Accepted 10 July 2021; Published 20 July 2021

Academic Editor: Nian Zhang

Grasp detection based on convolutional neural network has gained some achievements. However, overfitting of multilayer convolutional neural network still exists and leads to poor detection precision. To acquire high detection accuracy, a single target grasp detection network that generalizes the fitting of angle and position, based on the convolution neural network, is put forward here. The proposed network regards the image as input and grasping parameters including angle and position as output, with the detection manner of end-to-end. Particularly, preprocessing dataset is to achieve the full coverage to input of model and transfer learning is to avoid overfitting of network. Importantly, a series of experimental results indicate that, for single object grasping, our network has good detection results and high accuracy, which proves that the proposed network has strong generalization in direction and category.

1. Introduction

Over recent years, deep learning has gained huge breakthroughs in computer vision [1, 2]. Unlike traditional hand-engineered features, deep learning can autonomic learning features from images, to acquire highly abstract and robust visual features via making use of image information to the most extent. Naturally, as one of the most representative deep learning models, convolutional neural network has become a research hotspot in computer vision, with easy training, high performance, few parameters, and strong generalization. Particularly, researchers have attempted to introduce it into research on robotic grasp detection, since its remarkable achievements in target detection [3–12].

Literature [13] innovatively used convolutional neural network for robotic grasps. More importantly, a deep neural network with four layers was proposed, which could effectively express multimodal features of grasping position, to achieve accurate detection of suitable grasping position on object [13]. Furthermore, a three-stage convolutional neural network was adopted to detect the grasping position of objects in depth image [14], where the first-level convolutional neural network was used for performing preliminary location of grasping position, the second-level convolutional neural network was utilized for acquiring the preselected grasping boundary, and the third-level convolutional neural network was to reevaluate the preselected grasping boundary. To perform operations, a two-step robotic grasp detection system was proposed [15].

Distinct from above thoughts, although convolutional neural network was also adopted to identify the grasping region of the object, the entire image of the object was taken as the input of network, to directly generate the position of the possible grasping region on object [16]. Reference [17] evaluated the possible position to be grasped of the target via predicting the grasping function learned from the convolutional neural network. In addition, researchers converted the grasp detection into an 18-channel binary classification [18] and adopted a convolutional neural network to learn the clamping rule of the two-finger gripper to obtain the optimal grasping position on the target. Xia et al. proposed a planar grasping pose detection method of the robot based on the cascaded convolutional neural network

[19]; they established a cascaded two-stage convolution neural network model with position and attitude from coarse to fine to estimate the optimal grasping position and angle. In order to perform grasping new unknown model objects, visual feature points of an object in the process of being grasped were exacted via a convolutional neural network model, and a grasp strategy was constructed based on these visual feature points [20].

For actual robotic grasping, some grasp detection methods based on convolutional neural network models were put forward. Literature [21] proposed a hybrid deep architecture combining visual and tactile sensing for robotic grasp detection. An efficient framework of hierarchical cascaded forests to perform recognition and grasp detection of objects from RGB-D images of real scenes was proposed [22]. Ribeiro et al. [23] addressed the problems of grasp detection and visual servoing using deep learning and applied them as an approach to the problem of grasping dynamic objects. To acquire satisfactory grasp detection results, a self-supervised learning method was applied to learn grasping data directly collected by a robot [24]. To recognize and detect grasp rectangles on images of an object to be held by two-plates parallel grippers, a dictionary learning and sparse representation framework was proposed [25]. Also, unsupervised feature-learning methods were proposed for grasp detection [26–28]. In literature [26], a network model was proposed for predicting the 6 DOF pose of the target to confirm the position to be grasped. A beneficial attempt was conducted via using tactile sensors and an unsupervised feature-learning approach to predict whether a grasp is successful [27]. To clean water surface by aquatic robots, researchers came up with an unsupervised grasp detection method for water-surface object collection [28].

Additionally, for actual robotic grasping, another category of prediction approach is based on reinforcement learning. Zhang et al. proposed a reinforcement learning method for grasp detection to define a grasp as a point in a 2D image plane [29] via Q network [30] to perform target reaching after training in simulation. In literature [31], an asynchronous deep reinforcement learning approach was presented for learning robotic grasping policies, which can be trained on real physical robots. To perform complex sequences of pushing and grasping on a real robot, a method that combines deep reinforcement learning with affordance-based manipulation was put forward for detecting grasps [32]. Furthermore, to improve the flexibility of robotic detection for grasps, a curriculum-based reinforcement learning approach was conducted to learn reactive policies for the task of real picking [33]. Obviously, unlike above methods, grasp detection based on reinforcement learning mainly focuses on learning grabbing strategy for detecting grasps, rather than involving the network architecture itself.

However, with some success of grasp detection based on convolutional neural network in theories and applications, for grasp detection network inherence itself, overfitting in multilayer convolutional neural network still exists and leads to poor detection precision. To achieve highly accurate detection for grasps, a single target grasp detection network with high detection accuracy is proposed, which generalizes the fitting of angle and position.

The remainder of this paper is organized as follows. Section 2 introduces our preliminary work to provide a theoretical basis for this research. Section 3 gives an exhaustive formulation of our thoughts. Experimental results are shown in Section 4 to demonstrate the superiority of the proposed network. Ultimately, Section 5 concludes the paper and looks forward to the future work.

2. Related Work

2.1. Overview and Analysis of Components in Convolutional Neural Network. Objective of exploring each component in convolutional neural network is to deepen the understanding of network structure, so as to carry out our research. As a matter of fact, convolutional neural network is a feed-forward neural network, but distinct from ordinary neural networks, it is generally composed of a convolution layer, activation layer, pooling layer, and fully connected layer. Following, each component is overviewed and analyzed.

2.1.1. Convolutional Layer. Convolutional layer is the core module in a convolutional neural network, which is usually composed of several convolution kernels with different sizes. After image input into the convolutional neural network, the convolution kernel performs convolution operations successively on the width and height of the image with a certain step length, to obtain a convolved feature vector.

Unlike connection ways of neurons in the ordinary neural network, convolution operation adopts sparse connection, which means that only neurons calculated with convolution kernels are connected to each other. Thus, this connection mode could increase the sparsity of the network to greatly reduce the number of network parameters and also could avoid overfitting of the network. In addition, convolutional neural network has the characteristic of weight sharing; that is, different positions of an image could be processed via the same convolution kernel, which could also reduce the number of network parameters.

Furthermore, the relation between input and output in a convolutional neural network is determined by convolution operation and selection of hyperparameters.

Assuming that the input image size is $H \times W \times C$, the convolution kernel size is $F \times F \times C$, the number is N, the convolution step is S, the unilateral filling size is P, and the

output eigenvector is $H \times W \times C$, then the output could be expressed as

$$\begin{cases} H' = \dfrac{H - F + 2P}{S} + 1, \\ W' = \dfrac{W - F + 2P}{S} + 1, \\ C' = N. \end{cases} \tag{1}$$

Apparently, the output height and width of the convolutional neural network are determined by input, convolution kernel size, filling size, and step, while the output channel number is determined by convolution kernel number.

2.1.2. Activation Function. Activation function plays an important role in the convolutional neural network. In fact, the inexistence of activation function will lead to the output that is a linear expression of input, which means that the network could only deal with linear problems, thereby greatly weakening the expression ability of the network model. As a result, to increase the nonlinear expression ability of network, activation function is usually added after convolutional layer.

Sigmoid function is one of the typical activation functions [34], and its expression is

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{2}$$

In Sigmoid activation function, definition domain is $(-\infty, +\infty)$, and value ranges at $(0, 1)$, as shown in Figure 1.

Sigmoid function was formerly widely used in the shallow neural network, but when the input is large, its gradient approaches 0, and with the increasing depth of the network, gradient dissipation is easy to occur in backpropagation, leading to failure of network training. Moreover, the output value of the Sigmoid function is not centered at 0.

Another typical activation function is the tanh function [35], which could be expressed as

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{3}$$

Similar to the Sigmoid function, the definition domain of the tanh function is $(-\infty, +\infty)$, and the value also ranges at $(0, 1)$. However, different from the Sigmoid function, the output value of the tanh function is centered at 0, as shown in Figure 2.

Although the output value of the tanh function is centered at 0, it still has not solved the problem that the network could not effectively backpropagate in case output or initial value is large. Hence, applications of above two activation functions tend to drop off.

Subsequently, a linear rectifier function called ReLU was proposed [36]; the expression is

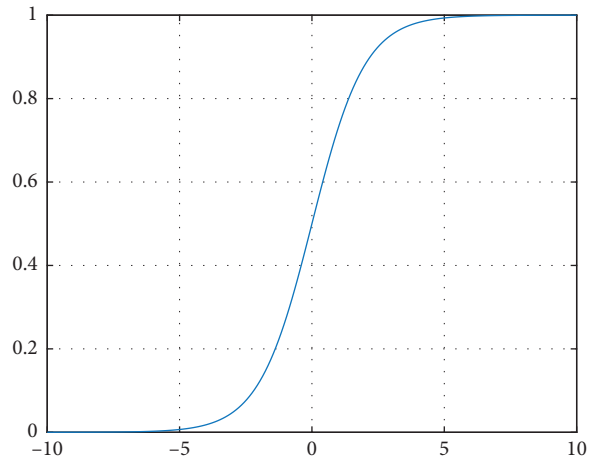

Figure 1: Sigmoid activation function.

$$f(x) = \begin{cases} 0, & x < 0, \\ x, & x \gg 0. \end{cases} \tag{4}$$

ReLU function is simple and easy to derive, which does not increase the difficulty in process of backpropagation and greatly accelerates the training speed. Even though the function could not be differentiated at 0, it has left derivatives and right derivatives around 0 and any of them could be selected since values exactly falling at 0 are minor and hardly affect the overall results. The image of this activation function is shown in Figure 3.

In ReLU activation function, the gradient saturation phenomenon is inexistent and the gradient is always 1, leading to fast convergence. Simultaneously, there is low computation due to nonexponential operations. Furthermore, neurons with output less than 0 do not work, which greatly increases the sparse expression ability of the network, to improve the network generalization performance. Thus, ReLU activation function is most widely used in current deep neural networks.

2.1.3. Pooling Layer. Pooling layer is also called downsampling layer and is commonly located behind the convolutional layer to reduce parameters number and computational complexity. Meanwhile, pooling layer could compress eigenvectors to exact main features and avoid overfitting. Generally, polling layer could compress the sizes of eigenvectors but could not change their depth.

Typical pooling methods include average pooling and maximum pooling; their calculation principles are, respectively, shown in Figures 4 and 5.

In Figures 4 and 5, the size of convolution kernels is the same, since above convolution kernel size is the most universally used in the convolutional neural network. It can be clearly seen that the average pooling takes the average value of convolution kernel region size and blurs the eigenvectors; thus, it is not conducive to feature extraction. However, maximum pooling takes the maximum value of convolution kernel region size and retains remarkable features. Accordingly, maximum pooling is mostly used at present.

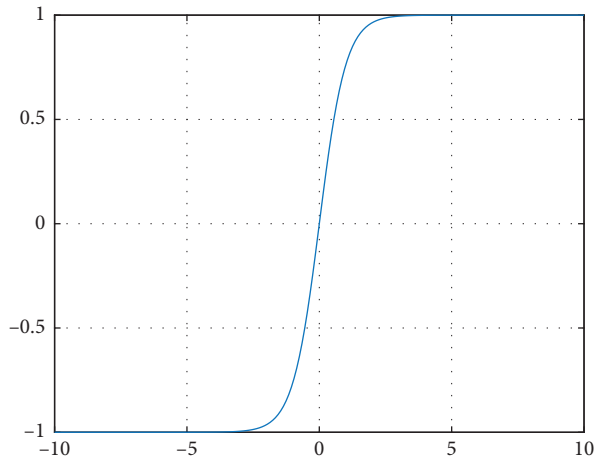

Figure 2: Tanh activation function.

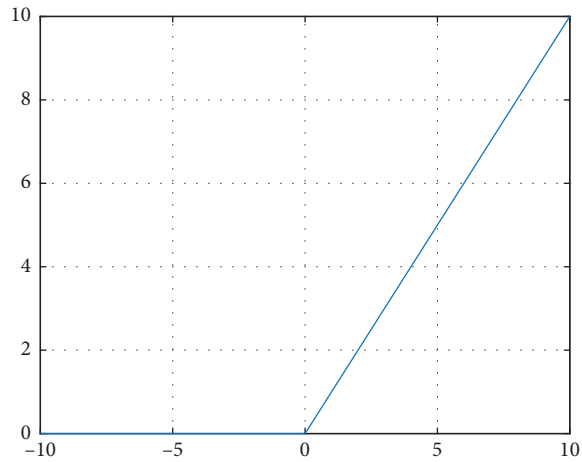

Figure 3: ReLU activation function.

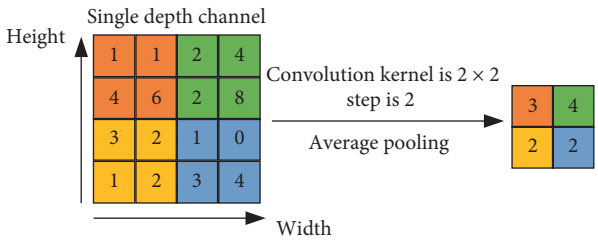

Figure 4: Calculation principle of average pooling.

2.1.4. Fully Connected Layer. Fully connected layer is similar to the ordinary neural network, without weight sharing and sparse connection of convolutional layer, and each neuron in it is interconnected. In a convolutional neural network, the input of the fully connected layer is eigenvectors extracted from the convolutional layer, and the output layer is selected based on completed task, such as Softmax output layer and logistic regression layer.

However, fully connection gives rise to a large number of parameters. If the number of data is too small, the network will easily fall into overfitting. Thus, in convolutional neural network, the emergence of the fully

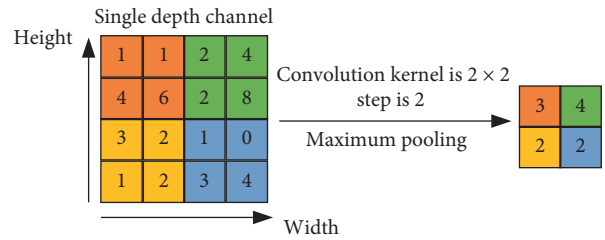

Figure 5: Calculation principle of maximum pooling.

connected layer is generally accompanied by the dropout layer. The dropout layer could stochastically discard some neurons to make them ineffective in fully connected layer. That is, the dropout layer is to imitate the sparse connection of the convolutional layer to prevent the overfitting of the network. In fact, the coefficient of dropout is confirmed by the specific application scenarios and network models, whose value is usually between 0.5 and 0.8 during training.

2.2. Performance Comparison of End-to-End Target Detection Algorithms Based on Convolutional Neural Network. Among target detection based on convolutional neural network, end-to-end networks directly detect the results from the image output, leading to a good performance in real time. Accordingly, we compare and analyze the performance of nowadays commonly used end-to-end networks to provide a theoretical foundation for our research.

In our implementation, VOC07 + 12 dataset is divided into a training set and test set, where the test set is 2007 test set, and the rest are training set. Detection results of different algorithms on test set are shown in Table 1.

It can be concluded from Table 1 that YOLOv2 is superior to YOLOv1 in accuracy and real time, and compared with YOLOv2-tiny, YOLOv2 gets a significant increase in accuracy at the expense of certain speed. Moreover, YOLOv2 is lower than SSD-300 in accuracy only at 1 percent, but more than four times faster in real time. Compared with

Table 1: Detection results of different end-to-end algorithms.

Algorithm	Training set	Test set	mAP	FPS
YOLOv1	VOC07 + 12 trainval	VOC07 test	48.1	71
YOLOv1-tiny	VOC07 + 12 trainval	VOC07 test	33.5	282
YOLOv2	VOC07 + 12 trainval	VOC07 test	75.4	100
YOLOv2-tiny	VOC07 + 12 trainval	VOC07 test	41.1	250
SSD-300	VOC07 + 12 trainval	VOC07 test	75.5	18
SSD-512	VOC07 + 12 trainval	VOC07 test	79.0	15

SSD-512, YOLOv2 is 6.7 times faster than it, while being lower than it in accuracy only at 3.6%.

Through comparative analysis of above results, it can be seen that YOLOv2 has superiority over others in high accuracy and better real time. Hence, this paper introduces it into research on grasp detection and makes use of its end-to-end detection thought to conceive a single grasp detection network, which takes an image as input and grasp parameters as output. Also, the proposed network has a great generalization ability to fit in angle and position and has high detection accuracy.

3. Constructing Single Target Grasp Detection Network

3.1. Modeling Grasping Parameters. Indeed, the essence of grasp detection based on a convolutional neural network is to find grasping parameters that could achieve stable grasping. Hence, establishment of an appropriate grasping parameter model to achieve stable grasping is the key to the research of grasp detection based on a convolutional neural network.

Saxena et al. adopted a 2D grasping point as a grasp parameter model [37], and Le et al. utilized a pair of grasping points as a grasp parameter model [38]. However, the limitation of above grasp parameter models lies in that they could not fully represent the seven dimension parameters in grasping operation of the robot, and the other parameters need to be estimated separately.

Due to this, Jiang et al. proposed a seven-dimensional representation method combining 2D grasping rectangle and 3D point cloud [39], which described the 3D position, attitude, and size of the end-gripper. However, 3D point cloud data need to be calculated, which means that the extracted point cloud data require high precision and large amounts of computation.

To deal with above problem, Redmon and Angelova [16] simplified the model of literature [39]. Their contribution simplified the grasping in three-dimensional into planar grasping and proposed a five-dimensional parameter representation method based on the 2D grasping rectangle, which brought inspiration to our research.

Obviously, simplifying 3D grasping into 2D planar grasping and using a grasping rectangle to express the grasp parameters could effectively reduce the computation, and the issue of grasp detection becomes relatively simple. Particularly, the grasping rectangle is used to describe the grasp parameters, which makes grasp detection quite similar to object detection, while the distinction between the two is that the direction of the gripper needs to be considered in grasp detection.

Consequently, we utilize the strong learning ability of the convolutional neural network on image features to convert the grasp detection of the robot into target detection and adopt a 2D grasping rectangle to confirm the appropriate grasp parameters. More importantly, in order to enable 2D grasping to be fully mapped into 3D space and directly utilized by the robot to accomplish grasping operations, in this work, we assume that the gripper is always perpendicular to the z-axis to grasp vertically downward.

To sum up, we build up a model of grasp parameters with the manner of five-dimensional representation. More precisely, we use the position of the gripper (x, y), the direction of gripper θ, the opening size of the gripper before grasping objects w, and the size of gripper h to constitute a grasping rectangle, as exhibited in Figure 6.

The grasp parameters model could be expressed as

$$M = \{x,\ y,\ h,\ w, \theta\}, \tag{5}$$

where (x, y) is the center coordinates of grasping rectangle, θ represents the rotation angle of grasping rectangle relative to the horizontal axis of the image (counterclockwise is positive), w means the width of grasping rectangle, and h refers to the height of grasping rectangle.

As displayed in Figure 6, a grasping rectangle of a remote device is composed of five grasp parameters defined by formula (5), where blue is on behalf of the gripper, red represents the distance between the two ends of the gripper before grasping, (x, y) is the center coordinates of grasping rectangle, and θ represents the rotation angle of grasping rectangle relative to the horizontal axis.

3.2. Modeling Grasp Detection Network. As mentioned above, YOLOv2 has obvious advantages in detection accuracy and real time. Thus, we introduce it into the research of grasp detection and utilize its "end-to-end" detection manner to establish a grasp detection network model with the proposed 5 grasp parameters as output. Accordingly, it is necessary to comprehend and analyze the network structure of YOLOv2 before modeling the grasp detection network.

Darknet19 as the framework of YOLOv2 is composed of 19 convolutional layers and 5 maximum pooling layers. In darkent19, largely 3×3 convolutional kernels are used for feature exaction, and after each maximum pooling layer, channels are doubled to prevent information loss. Simultaneously, 1×1 convolutional kernels are added after 3×3 convolutional kernels to compress eigenvectors. Lastly, global average pooling is adopted to reduce dimension, and the Softmax layer is utilized for prediction. Furthermore, batch normalization is used for improving the stabilization and accelerating the convergence of the model in process of training. The network model of darknet19 is shown in Figure 7.

In fact, darknet19 has good performance in target detection, and the established grasp detection network model in this paper only needs the output 5 grasp parameters. Thus, in order to simplify the process of training network,

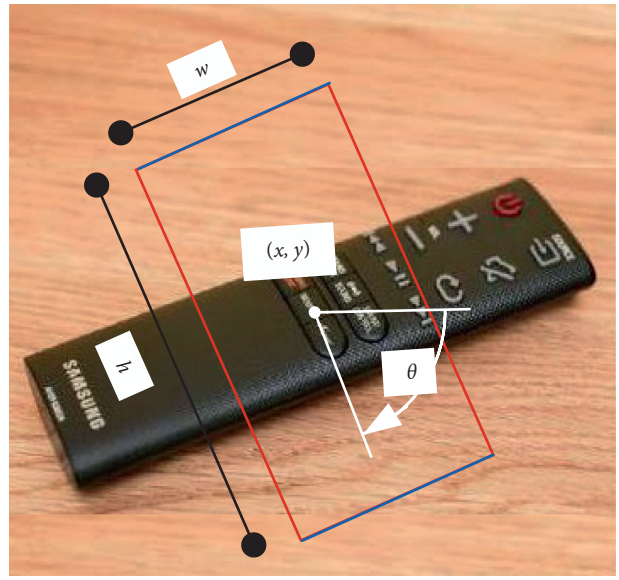

Figure 6: Schematic of grasp parameter model.

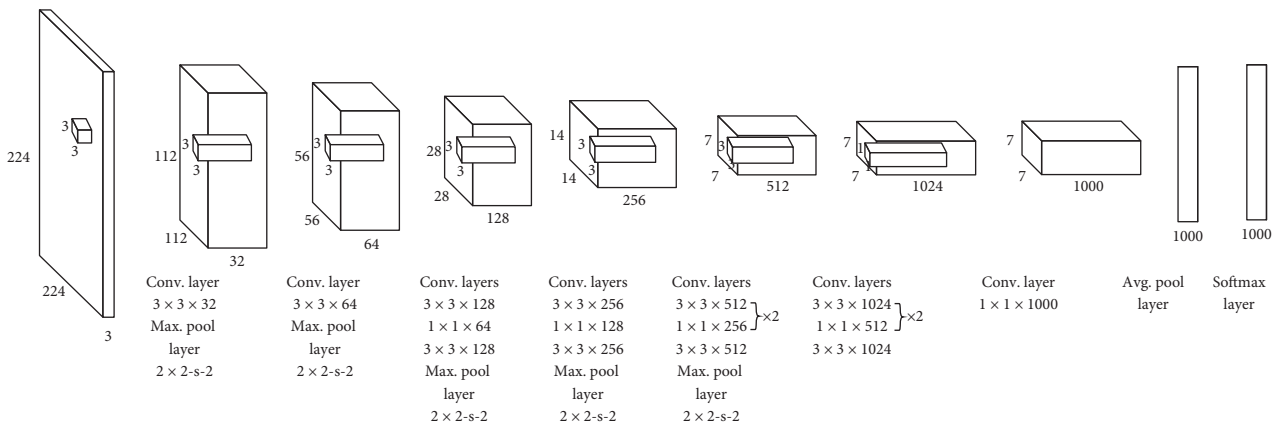

Figure 7: Network model of darknet19.

meanwhile shortening the process of forward reasoning and backpropagation, thereby to avoid the occurrence of overfitting, we construct a grasp detection network model based on the network structure of darknet19, which has a relatively simple structure and could adapt to the proposed grasp parameters.

On the other hand, both accuracy and real time in grasp detection are taken into consideration; the constructed grasp detection network model should be able to make full use of powerful learning ability and extraction ability of convolutional neural network on image features and could avoid multiple time-consuming classification calculations in a small part of the whole image. Hence, the established grasp detection network model should be able to carry out bounding box regression on the whole image to acquire the appropriate grasping rectangle.

In summary, based on the network architecture of darknet19, we put forward a grasp detection network model with the whole image as input and five grasp parameters as output, whose structure is displayed in Figure 8.

As shown in Figure 8, compared with darknet19, the grasp detection network model established in this paper prunes the 1×1 convolutional kernel used for compressing eigenvectors, which was connected with 3×3 convolutional kernel, and removes the 3×3 convolutional kernel used for learning higher-level features, which was between 1×1 convolutional kernel and maximum pooling layer. The eigenvectors of $7 \times 7 \times 1024$ are obtained after six convolutional layers and pooling layers and without connection of pooling layers behind the last convolutional layer. In addition, the 1×1 convolutional layer, fully connected layer, and Softmax output layer used for classification tasks are replaced by three fully connected layers with 1024, 512, and 5 neurons, respectively, where fully connected layers with 1024 and 512 neurons are used to deal with $7 \times 7 \times 1024$ eigenvectors extracted by convolutional layer, and the last 5 neurons are used to output the grasp parameters.

When the original image is input into the network model, the convolutional layer is used to extract features from the image, and the fully connected layer of the last 5 neurons is used

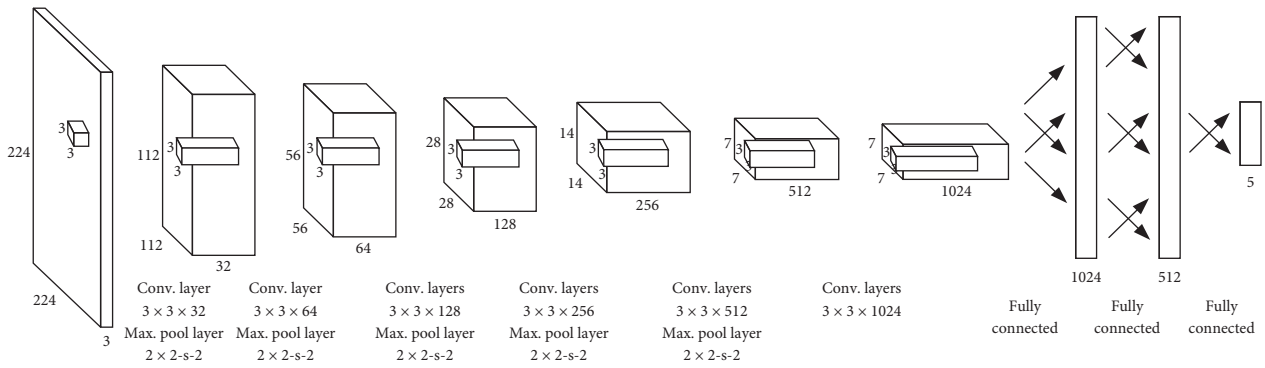

Figure 8: Structure of constructed grasp detection network model.

as the output layer corresponding to the coordinates of grasp parameters, where four neurons correspond to the position, width, and distance of the gripper. The grasping angle is symmetric; thus, $\theta \in (-\pi/2, \pi/2)$, but $\tan\theta$ is monotone increasing in this interval. Accordingly, the last neuron corresponds to the $\tan\theta$ value of the gripper relative to z-axis rotation angle. Although θ between $(-\pi/2, \pi/2)$ is reasonable, $\tan\theta$ is closer to these two thresholds, and the value of $|\tan\theta|$ is greater, which is quite disadvantageous to the calculation of regression and even leads to difficulty in continuing training the network model. To avoid the emergence of this situation, we further limit the range of θ. Since $\tan \pm 85^\circ \approx \pm 11$, in this paper, the angle range is limited to $\theta \in (-85^\circ, 85^\circ)$, namely, only loss of 5°, and $\tan\theta$ is limited to a small range, which is convenient for regression calculation of the model.

Indeed, the constructed network model is for single object grasp detection; hence, each object only needs to predict once grasp. That is, as long as the image is input into the model, our model could directly make a global regression prediction of the image.

During training the proposed grasp detection network model, the model randomly selects a real value as a label to carry our regression with the predicted value. Since label value is always changing each time, the network model is uneasy to overfit in grasp parameters of an object. In order to better training the proposed network model, we define the loss function, which could be expressed as

$$F_{\text{coord}} = \lambda_{\text{coord}}\left((x-\widehat{x})^2 + (y-\widehat{y})^2 + (h-\widehat{h})^2 + (w-\widehat{w})^2\right), \tag{6}$$

$$F_{\text{angle}} = \lambda_{\text{angle}}(\tan\theta - \tan\widehat{\theta})^2, \tag{7}$$

$$F_{\text{total}} = F_{\text{coord}} + F_{\text{angle}}, \tag{8}$$

where λ_{coord} is the trade-off parameter of coordinate values losses, λ_{angle} is the trade-off parameter of angle values losses, F_{coord} is the coordinate values losses of the network, F_{angle} is the angle values losses of the network, and F_{total} is the total loss of network.

It can be seen from formula (6) and formula (7) that the paper adopts a sum of square errors to construct loss function, but different weight factors are used for different parameters to ensure that the contribution of each parameter to the loss is approximately consistent. Through statistics, rectangular center coordinates x and y are mostly between 100 and 150 pixels , as well as h and w are mostly between 20 and 30 pixels. Obviously, it is unreasonable to add directly and proportionately to the loss. Indeed, grasp position is quite important, but the opening and closing size of the gripper is also equally important. Hence, the regulator of coordinate values λ_{coord} is added before error losses of x and y, whose value is 0.1. Similarly, since the value of $\tan\theta$ is limited at the range of $(-11, 11)$, to adjust to the same level, the adjustment factor λ_{angle} is added before angular losses, whose value is 10. Through the above manners, losses of all parameters are basically guaranteed to account for the same proportion in total loss, which are conducive to the training network to obtain good results.

3.3. Selection and Preprocessing of Dataset. In order to verify the effectiveness of the proposed network model, it is necessary to select an appropriate dataset for the training model. At present, Cornell dataset is a widely used grasping dataset, which contains 240 common objects and 885 images obtained from different angles of these objects [1, 39]. In this dataset, numerous images contain the same kind of object, but the position and direction of the object in the image are different, which is extremely important for improving the robustness of the network model to the position and direction of the object during training. Consequently, this paper selects the Cornell grasping dataset to verify the validity of the proposed grasp detection network model.

However, in current data labels, cases that could not completely cover overall grasp positions and directions still exist. Thus, it is essential to preprocess the dataset to adapt the input of the model. In other words, we expand the dataset to achieve full coverage of input.

For the entire dataset, in order to prevent some objects in subsequent steps which are cut off, we primarily intercept pixel-sized areas of 321 × 321 from the center in each image and utilize a filling algorithm to fill in the neighboring pixels to pixel-sized areas of 501 × 501. Then we randomly spin the image five times with a certain angle. Namely, the image is randomly, respectively, moved five times within 100 pixels in

x and y directions. Lastly, pixel-sized area of 320 × 320 from center in each image is cut out and scaled to the pixel-sized area of 240 × 240 that the network model needs to input. At the same time, label values also need to be synchronized to match the changes of each image. The whole process of the data preprocessing algorithm is shown in Figure 9.

After preprocessing, the dataset is expanded 125 times, including 110625 images, which satisfies the requirements of the following network training.

In our implementation, we use a 50-fold cross-validation method to test our model. Meanwhile, we adopt two ways to segment the image. The first one is to randomly segment all the images in the dataset, which means that the most likely occurrence of the test set is objects seen during training, but the direction is random and unseen. This image segmentation method tests the sensitivity of the network model to angle. The other is to randomly segment each category of the object in data; that is, all images of the same object are in the same cross-validation set, which means that objects in the test set are unseen during training, but the direction is seen. This segmentation manner has higher requirements and greater difficulty for the model, which is to test the generalization ability of the network model. In fact, generalization ability is exactly what we expect the proposed model should have.

3.4. PreTraining Grasp Detection Network. As a matter of fact, the dataset used in this paper contains a limited amount of data; directly training the network model easily leads to network overfitting. Yet pretraining a large-scale convolutional neural network model could greatly shorten training time and avoid overfitting [40]. Hence, it is essential to pretraining the network model to avoid overfitting during training.

Due to data similarity between grasp detection and target detection is high, and the training set has 88500 images after expansion of the whole dataset via preprocessing, whose amount is large. Thus, we could use transfer learning to extract image features from networks trained by datasets in target detection for grasp detection.

Nevertheless, transfer learning has different processing manners for diverse application scenarios. Consider that the only distinction between grasp detection and target detection is the output of grasp detection which has an extra gripper angle. Therefore, after data classification in the network, we use parameters of six convolution layers to send the extracted eigenvectors to the following fully connected layer for processing and predicting results. The three fully connected layers are trained from scratch and only one initialization value is given.

3.5. Training Grasp Detection Network. After pretraining the network model, we adopt a small-batch gradient descent algorithm to training the network 100 times with the manner of end-to-end, where the value of each batch is 128. We set the learning rate α to 0.0005, the weight attenuation coefficient λ to 0.01, and the dropout parameter among three fully connected layers to 0.5. The loss of training processing is exhibited in Figure 10.

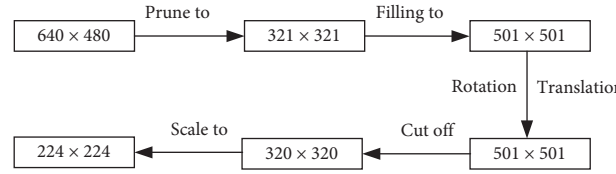

Figure 9: Flow of data preprocessing algorithm.

In Figure 10, the abscissa represents the number of training steps, and the ordinate refers to the corresponding loss value. Apparently, the total loss is decreasing with the increasing of iterative steps, but a short oscillation occurs when it decreases to a certain extent, and then it continues decreasing to a certain value, which indicates that the performance of the model for the training set tends to be stable at this time. Hence, in general, the model is reliable for the training set.

4. Experiments

4.1. Select and Determine the Evaluation Index of Proposed Grasp Detection Network. Point coordinates and rectangular coordinates are currently two general indexes to evaluate the performance of a grasp detection network [16, 41]. Indeed, point coordinates are to judge the quality of grasping via comparing the distance between the predicted coordinates of the center point in grasping rectangle and center points coordinates of all real grasping values, whereas this evaluation method does not consider the impact of grasping angle on accuracy, but angle value is particularly important in actual grasping. In addition, point coordinates also need to set another threshold to evaluate the results of point coordinates, which also affects the accuracy of calculation to a certain extent.

Rectangular coordinate is to judge the quality of grasping by comparing the difference between the predicted grasping angle and real grasping value. When the difference is less than 30° and the Jaccard similarity coefficient between the predicted grasping rectangle and real grasping value is greater than 25%, the grasping is considered to be effective [42]. In this paper, the Jaccard similarity coefficient is similar to Intersection-over-Union in target detection, which is defined as follows:

$$J\left(M_g, M_p\right) = \frac{\left|M_g \cap M_p\right|}{\left|M_g \cup M_p\right|}, \tag{9}$$

where M_g represents actual values of grasping rectangle and M_p refers to the predicted values of grasping rectangle.

Obviously, the value of the Jaccard similarity coefficient is larger, which indicates that the effect of grasp detection is better.

From above analysis, it can be concluded that the rectangle index considers both position and angle, which is more comprehensive than point coordinates and more convincing in judging the quality of grasping. Accordingly,

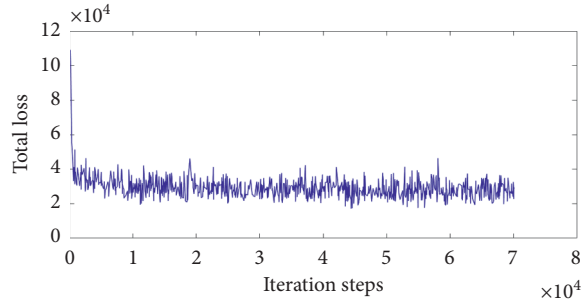

Figure 10: Changing curve of total loss.

in this paper, we adopt rectangle index to evaluate the performance of the proposed grasp detection network.

4.2. Experimental Results and Analysis. In order to validate the effectiveness of our network model, we conduct experimental verification on Cornell dataset. The image is input into the proposed network model, and the output result is the prediction grasping rectangle box of each input image. Some of the visual detection results are exhibited in Figure 11.

Obviously, detection results in Figure 11 illustrate that our model could detect the grasping region. Thus, to further illustrate the effectiveness of prediction, we calculate the Jaccard similarity coefficient of each prediction rectangle in Figure 11, and calculation results are shown in Table 2.

It can be clearly seen from Table 2 that all Jaccard similarity coefficients are greater than 0.25, which indicates that our grasp detection is effective, and grasp detection results for single object grasping could be regarded as good.

Through analysis of established network model, it can be known that acquired good detection results lie in two reasons. The first one is that our model adopts directly calculation of the loss and carry out global boundary regression on image to acquire the appropriate grasping rectangle. The other is that our model randomly selects a label value for each image during model training, which means that, after multiple training of dataset, the model predicts an average value for each object. Thus, for single object grasping, the predicted average value still has a good detection effect.

Additionally, to further verify the performance of the proposed network model, we make a comparison with other models based on convolutional neural networks, and the results are exhibited in Table 3.

It can be seen from above table that, in terms of detection accuracy, the prediction accuracy of our network model for image segmentation is 88.7%, and prediction accuracy for object segmentation is 87.2%; both of them stay at the third, belonging to an upper level. On the other hand, our research is inspired by literature [16, 39], and the comparison results in Table 3 show that our model is superior to the above two in detection accuracy, indicating that our research is meaningful even though it is not the best in above comparisons.

In summary, above experimental results demonstrate that the constructed network model has good detection results and high accuracy in single object grasping. Also, these results validate that our model is effective with strong generalization in direction and category.

Table 2: Jaccard similarity coefficients of grasp detection in Figure 11.

Image no.	Jaccard similar coefficient
a	0.83
b	0.85
c	0.82
d	0.51
e	0.81
f	0.68
g	0.85
h	0.75
i	0.86
j	0.67

Table 3: Grasping prediction accuracy of different algorithms on Cornell dataset.

Algorithms	Image segmentation accuracy (%)	Object segmentation accuracy (%)
Jiang et al. [39]	60.5	58.3
Lenz et al. [13]	73.9	75.6
Redmon and Angelova [16]	88.0	87.1
Wang et al. [15]	81.8	N/A
Guo et al. [21]	93.2	89.1
Asif et al. [22]	88.2	87.5
Ribeiro et al. [23]	94.8	86.9
Trottier et al. [25]	87.7	86.6
Ours	**88.7**	**87.2**

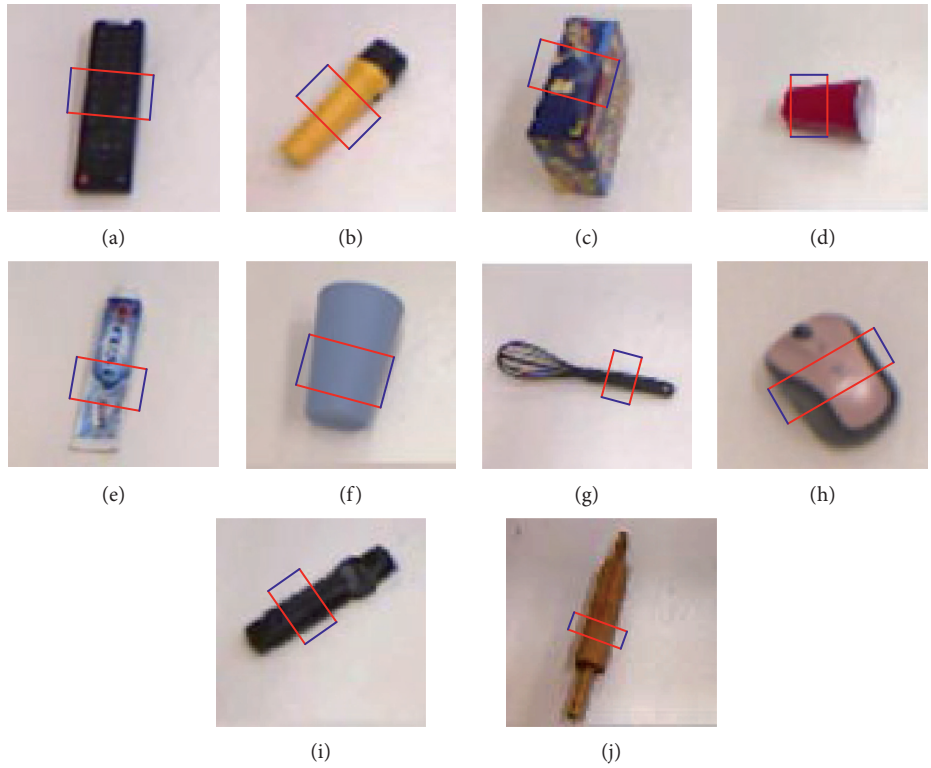

Figure 11: Some grasp detection results.

5. Conclusions

In this work, a single target grasp detection network based on a convolutional neural network is put forward, which generalizes the fitting of angle and position with high detection accuracy. Specifically, we simplified 3D space grasping into 2D planar grasping and modeled grasping parameters with the manner of five-dimensional representation. Afterward, we adopted end-to-end detection ways to construct a grasp detection network model with the image as input and five grasping parameters as output. In order to verify the effectiveness of the proposed grasp detection network model, the Cornell grasp dataset is selected and expanded to match the input of the model. Furthermore, a 50-fold cross-validation method was adopted to test our network model, and the image was split into two ways. Moreover, for the sake of avoiding overfitting of the network in training, the constructed network model was pretrained via transfer learning. Ultimately, experimental results indicate that, for single object grasping, the proposed grasp detection network has good detection results and high prediction accuracy, which demonstrates that our detection model has strong generalization in direction and category.

Particularly, in the future, using other datasets to further optimize and validate our model is a beneficial work to be finished. Also, applying the proposed network to actual grasping operation is worth being deeply researched.

Data Availability

In this paper, the dataset is the Cornell dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Valuable suggestions given by Research Associate Baoshi Cao of the Harbin Institute of Technology are also acknowledged. This work was supported by the Self-Planned Task of the State Key Laboratory of Robotics and Systems under Grant No. SKLRS201910B.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, 2014.

[4] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[6] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems NIPS'16*, Barcelona, Spain, December 2016.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR'2017*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969, Venice, Italy, October 2017.

[9] D. Yoo, S. Park, J.-Y. Lee, A. S. Paek, and I. So Kweon, "Attentionnet: aggregating weak directions for accurate object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2659–2667, Santiago, Chile, December 2015.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[11] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot MultiBox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 21–37, Amsterdam, The Netherlands, October 2016.

[12] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[13] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[14] Q. Yu, W. Shang, and C. Zhang, "Object grasp detecting based on three-level convolution neural network," *Jiqiren/Robot*, vol. 40, no. 5, pp. 762–768, 2018.

[15] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Advances in Mechanical Engineering*, vol. 8, no. 9, pp. 1–12, 2016.

[16] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1316–1322, Seattle, WA, USA, May 2015.

[17] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4461–4468, Daejeon, South Korea, October 2016.

[18] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3406–3413, Stockholm, Sweden, May 2016.

[19] J. Xia, K. Qian, X. Ma, and H. Liu, "Fast planar grasp pose detection for robot based on cascaded deep convolutional neural networks," *Robot*, vol. 40, no. 6, pp. 794–802, 2018.

[20] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519, Stockholm, Sweden, May 2016.

[21] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1609–1614, Singapore, May 2017.

[22] U. Asif, M. Bennamoun, and F. A. Sohel, "RGB-D object recognition and grasp detection using hierarchical cascaded forests," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 547–564, 2017.

[23] E. G. Ribeiro, R. De Queiroz Mendes, and V. Grassi, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robotics and Autonomous Systems*, vol. 139, no. 2, Article ID 103757, 2021.

[24] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[25] L. Trottier, P. Giguère, and B. Chaib-draa, "Dictionary learning for robotic grasp recognition and detection," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016)*, Daejeon, South Korea, October 2016.

[26] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3 dmatch: learning the matching of local 3d geometry in range scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, vol. 1, no. 2, p. 4, Honolulu, HI, USA, July 2017.

[27] D. Cockburn, J.-P. Roberge, T.-H.-L. Le et al., "Grasp stability assessment through unsupervised feature learning of tactile images," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2238–2244, Singapore, May 2017.

[28] S. Kong, X. Chen, Z. Wu et al., "An unsupervised grasp detection for water-ssurface object collection," in *Proceedings of the 38th Chinese Control Conference*, pp. 4421–4426, Guangzhou, China, July 2019.

[29] F. Zhang, J. Leitner, M. Milford et al., "Towards vision-based deep reinforcement learning for robotic motion control," in *Proceedings of the Australasian Conference on Robotics and Automation*, Canberra, Australia, December 2015.

[30] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[31] S. Gu, E. Holly, T. Lillicrap et al., "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3389–3396, Singapore, May 2017.

[32] A. Zeng, S. Song, S. Welker et al., "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligence Robots and Systems*, pp. 4238–4245, Madrid, Spain, October 2018.

[33] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, "Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1549–1556, 2019.

[34] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proceedings of the 1995 International Workshop on Artificial Neural Networks*, vol. 930, pp. 195–201, Torremolinos, Spain, 1995.

[35] W. Malfliet, "The tanh method: a tool for solving certain classes of nonlinear evolution and wave equations," *Journal of Computational and Applied Mathematics*, vol. 164-165, pp. 529–541, 2004.

[36] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[37] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pp. 1209–1216, Vancouver, Canada, December 2006.

[38] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5062–5069, Anchorage, AK, USA, May 2010.

[39] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: learning using a new rectangle representation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2011*, pp. 3304–3311, Shanghai, China, May 2011.

[40] D. Guo, T. Kong, F. Sun, and H. Liu, "Object discovery and grasp detection with a shared convolutional neural network," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA) 2016*, pp. 2038–2043, Stockholm, Sweden, May 2016.

[41] J. Donahue, Y. Jia, O. Vinyals et al., "A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on International Conference on Machine Learning*, Beijing China, June 2014.

[42] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 57, 2018.