machines

MDPI

Article

Grasping Pose Estimation for Robots Based on Convolutional Neural Networks

Tianjiao Zheng [1,2,†], Chengzhi Wang [1,†], Yanduo Wan [1], Sikai Zhao [1], Jie Zhao [1], Debin Shan [2] and Yanhe Zhu [1,*]

[1] State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China; zhengtj@hit.edu.cn (T.Z.); 20b908016@stu.hit.edu.cn (C.W.); 8200880114@stu.hit.edu.cn (Y.W.); 16b908056@stu.hit.edu.cn (S.Z.); jzhao@hit.edu.cn (J.Z.)
[2] School of Materials Science and Engineering, Harbin Institute of Technology, Harbin 150001, China; shandb@hit.edu.cn
* Correspondence: yhzhu@hit.edu.cn
† These authors contributed equally to this work.

Abstract: Robots gradually have the ability to plan grasping actions in unknown scenes by learning the manipulation of typical scenes. The grasping pose estimation method, as a kind of end-to-end method, has rapidly developed in recent years because of its good generalization. In this paper, we present a grasping pose estimation method for robots based on convolutional neural networks. In this method, a convolutional neural network model was employed, which can output the grasping success rate, approach angle, and gripper opening width for the input voxel. The grasping dataset was produced, and the model was trained in the physical simulator. A position optimization of the robotic grasping was proposed according to the distribution of the object centroid to improve the grasping success rate. An experimental platform for robot grasping was established, and 11 common everyday objects were selected for the experiments. Grasping experiments involving the eleven objects individually, multiple objects, as well as a dark environment without illumination, were performed. The results show that the method has the adaptability to grasp different geometric objects, including irregular shapes, and it is not influenced by lighting conditions. The total grasping success rate was 88.2% for the individual objects and 81.1% for the cluttered scene.

Keywords: robot grasping; pose estimation; convolutional neural network; deep learning

Citation: Zheng, T.; Wang, C.; Wan, Y.; Zhao, S.; Zhao, J.; Shan, D.; Zhu, Y. Grasping Pose Estimation for Robots Based on Convolutional Neural Networks. *Machines* **2023**, *11*, 974. https://doi.org/10.3390/machines11100974

Academic Editor: Bao Kha Nguyen

Received: 13 September 2023
Revised: 12 October 2023
Accepted: 18 October 2023
Published: 20 October 2023

1. Introduction

Traditional robotic grasping mostly performs repetitive actions, and the end pose of the robot is obtained through manual teaching or kinematic derivation. It is relatively mature and widely applied in environments with certain pose objects. However, the robot is limited by the external objects and the environment. As more complex application scenes such as logistics sorting and service robots have been required, situations with various poses of objects and inevitable stacking occlusions have appeared. As the perception ability of robots has been greatly improved, the robot can autonomously plan the manipulation of actions based on the perceived environmental information. By learning the operation of typical scenes, the robot has adaptability in unknown scenes, analyzes unknown scenes by using known frameworks, and plans grasping actions autonomously; so, the robot is no longer bound by the constraints of traditional scenes.

There are two main kinds of methods for the robot grasping task. One is based on the pose of the object, which requires the robot to have some prior knowledge. Suitable grasping poses can be provided in advance for different types of objects; so, the issue of robot grasping can be transformed into the classification and pose estimation of the object [1]. The idea of the traditional 6D pose estimation method for objects is to match the feature information of the object between the scene and the known template. Point Feature Histograms [2], Fast Point Feature Histograms [3], Point Pair Features [4], and the linemod method [5]

are the representative feature description methods. With significant breakthroughs in deep learning in computer vision, scholars have extended two-dimensional images to three-dimensional objects, and convolutional neural networks (CNN) have achieved many research results in object pose estimation. Representative methods have been proposed, like the SSD-6D method [6], the Pose CNN method [7], the Real-Time Seamless Single Shot 6D method [8], the DenseFusion method [9], the PVN3D method [10], and the FFB6D method [11]. For objects in the dataset, the above methods can accurately estimate the pose of the object and grasp the object, combining prior knowledge. However, the object pose estimation method has poor generalization for unknown objects that do not exist in the dataset.

The other method is an end-to-end method with a better generalization, which directly processes the input image or point cloud to obtain a suitable grasping pose. Jiang et al. [12] proposed to determine the graspable position as a directional rectangle. Six-dimensional grasping is simplified and becomes grasping on the plane; so, the grasping modeling is directly carried out on the input RGB image. Furthermore, not only color images but also three-dimensional point clouds are considered to obtain appropriate grasping poses. Ten Pas et al. [13] proposed to detect grasping poses in point clouds. This method takes the point cloud as input and randomly generates N candidate grasping poses near the object. The neural network scores each candidate grasping pose and outputs whether it is a suitable grasping prediction. This method is able to generate candidate grasping poses on arbitrary visible surfaces. The GG-CNN method proposed by Morrison et al. [14] outputs the appropriate grasping pose and grasping quality at each position in the input depth map, in order to overcome the limitations of deep learning in six-dimensional grasping. The PointnetGPD method proposed by Liang et al. [15] performs grasping actions on random samples and scores grasping effects based on force closure and grasping space.

The above grasping pose estimation methods are also trained for specific datasets. Although they have certain generalization capabilities, their accuracy for unknown objects is still greatly reduced compared to known objects. Since the learning-based methods heavily rely on the scale of the training dataset, which can be tedious for human to collect from physical grasping experiments, researchers from NVIDIA proposed the 6-DoF graspnet method [16,17]. The dataset is completely generated by using a physical simulator. For the input point cloud, the grab samples are randomly selected by the variational autoencoder, and the grab sampling evaluation model is used to evaluate and optimize. The method has an 88% grasping success rate across different appearance size scales. Another prominent work of such data-driven learning is the Dexterity Network (Dex-Net) series [18–21]. In Dex-Net 1.0, cloud computation and big data are firstly employed to accelerate the object classification task based on a Multi-View Convolutional Neural Network. Later, Dex-Net 2.0 and 3.0 are extended to use a Grasping-Quality Convolutional Neural Network (GQCNN) for a parallel-jaw gripper and vacuum suction cup separately, and achieve grasping success rate over 90% and 82% each, but they still take more than 2.5 s and 3.0 s to plan grasps. The next version, 4.0, combines previous work and introduces an "ambidextrous" policy learning method to enable the intelligent switch of end-of-arm tools (EoAT), which leads to 93% successful grasping. Methods based on deep learning have high requirements for computational ability. Although accurate results can be obtained, it is difficult to meet the real-time requirements for the ordinary hardware, and also, they are not robust enough to handle complex real-world conditions.

One important research field concentrates on the robustness of grasp and pick algorithms in uncontrolled cluttered environments where there are package piles and partial occlusion, in order to deploy robotic grasping in industrial applications. The learned pick quality system used in the Robin induction fleet of Amazon.com, Inc. (Seattle, USA) [22] is regarded as the first large-scale deployment of such a method in real production system and nowadays can sort several million packages per day. Its shallow machine learning model trained on historical pick outcomes makes full use of prior experience to learn which feathers are most important for prediction, and to rank and determine the most promising

picks based upon that very effectively. The VGN method [23] proposed by ETH Zurich represents a breakthrough in computing speed. This method does not select point clouds as input but converts objects into 3D voxels and uses the voxels as the input of the CNN model. The method has an 80% grasping success rate. Meanwhile, each planning only takes 10 ms in this method, and a real-time grasping pose estimation of the robot can be possible. The corresponding dataset production rule in the VGN method is to attempt to capture the six fixed directions of the point cloud's outer normal. Although the dataset generated in this way can be successfully captured in a simulation environment, the corresponding grasping pose may not be reasonable and stable.

In this paper, we present a stable grasping pose estimation method for robots, in which a CNN model for grasping pose estimation is based on the VGN model. This model can output the grasping success rate, approach angle, and gripper opening width for each voxel in the scene to be captured. The grasping dataset was produced based on common sense, and the model was trained in the physical simulator. In addition, a position optimization of the robotic grasping is proposed according to the distribution of the object centroid in order to improve the success rate of the robotic grasping.

The remainder of this paper is organized as follows. Section 2 presents the grasping pose estimation method for robots based on the CNN model. Section 3 explains the grasping experiments and results. Finally, conclusions are provided in Section 4.

2. Method

2.1. Architecture of the CNN Model for Grasping Pose Estimation

The purpose of the convolutional neural network (CNN) model of the grasping pose estimation in this paper is to establish a mapping relationship that can output the grasping success rate, approach angle, and gripper opening width at each voxel grid for an arbitrary input 3D voxel space by training the dataset and optimizing the parameters. The architecture of the CNN model is shown in Figure 1. Three-dimensional convolutional layers are employed as the basic structure in this architecture.

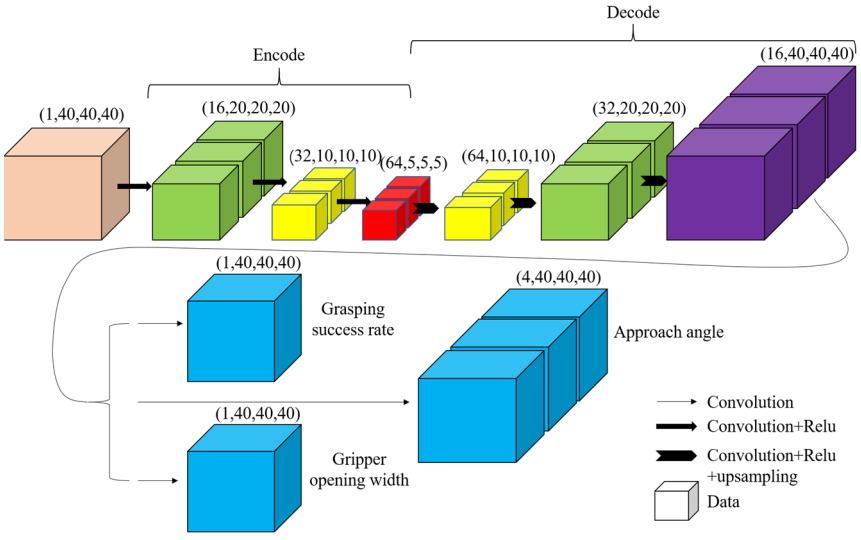

Figure 1. The architecture of the CNN model for grasping pose estimation.

For the encoding section of the model, a perception module consisting of 3 convolutional layers with 16, 32, and 64 filters, maps the input volume V to a feature map of dimension 64×5^3. ReLu (Rectified Linear Unit) is employed as the activation function to improve the nonlinear fitting ability of the model.

For the decoding section of the model, the neural network consists of 3 convolutional layers interleaved with $2\times$ bilinear upsampling, followed by three separate heads for predicting the grasping success rate, approach angle, and gripper opening width.

2.2. Loss Function of the CNN Model

For the value of the grasping success rate, the labels of the data contain two types of 0 and 1, which are failure and success, respectively. The output should be in the interval [0, 1]. The closer the value is to 1, the higher the success rate of grasping at this pose. Therefore, the binary cross-entropy loss can be considered as the loss function of the grasping success task, as shown in Equation (1). It has been proved to have a fine training effect in the binary classification task.

$$L_q = -\sum_{i=1}^{N}(\hat{q}_i \log(q_i) + (1 - \hat{q}_i) \log(1 - q_i)) \tag{1}$$

Here, N is the batch size, $\hat{q}_i$ is the actual label of the input data, and q_i is the predictive value of the model.

For the approach angle, Quaternion is employed to represent angle information. The inner product of the Quaternion represents the cosine value of the angle between two vectors in a four-dimensional space. When the two vectors are consistent, the cosine value of the angle is 1, and the corresponding loss is 0. Otherwise, the corresponding loss value will be larger when the two approach angles have a significant difference. The loss function of pose is shown in Equation (2).

$$L_r = 1 - (\hat{r} \cdot r) \tag{2}$$

Here, $\hat{r}$ is the actual label of the Quaternion, and r is the predictive value of the model.

However, there is a specific situation due to the symmetry of a parallel-jaw gripper. A configuration rotated 180° around the gripper's wrist axis corresponds effectively to the same grasp but leads to inconsistent loss signals as the model is penalized for regressing to one of the two alternative 3D rotations. Therefore, it is necessary to determine the corresponding situation when it rotates 180° around the gripper's wrist axis, and the minimum value of these two values should be the actual output, as shown in Equation (3).

$$L_r = \min(1 - (\hat{r} \cdot r), 1 - (\hat{r} \cdot r_\pi)) \tag{3}$$

For predicting the gripper opening width, the mean square error is used as the loss function. When the two values are consistent, the corresponding output is 0; otherwise, the corresponding loss will be larger, as shown in Equation (4).

$$L_w = (\hat{w} - w)^2 \tag{4}$$

Here, $\hat{w}$ is the actual opening width of the gripper, and w is the predictive output of the gripper opening width.

Combining the above three tasks, the overall loss function value is determined by the above loss functions L_q, L_r, and L_w. Among the three kinds of losses, the grasping success rate is the basic representation for deciding the generated grasping pose. Therefore, the predicted value of the grasping success rate is employed as the proportional coefficient of the other two loss functions. In addition, the ranges of the loss functions L_q and L_r are within [0, 1], and a proportional coefficient is used for the loss function of the gripper opening width, in order to ensure the same order of magnitude for the output value of the three loss functions, as shown in Equation (5).

$$Loss = L_q + q(L_r + kL_w) \tag{5}$$

2.3. Grasping Dataset and Model Training

In this work, a dataset for the CNN model was created, because the model's input is a 3D voxel grid. The grasping dataset based on virtual 3D objects in the simulation environment can contribute to improve the generalization ability of the model. Thus, based

on the simulation physics engine of PyBullet, a 6-DoF grasping dataset with approximately 200,000 times grasping attempts was created considering both the gripper pose and the force between the gripper and the object.

The objects to be grasped in this dataset are 3D models with regular geometric shapes such as cuboids and cylinders. These 3D models are the regularized expression of the objects in daily life, as shown in Figure 2. In a simulation environment, n kinds of random object models, after being arbitrarily scaled and rotated, constitute a stacking scene. Grasping attempts were performed 60 times in each scene, until 200,000 grasping attempts were generated.

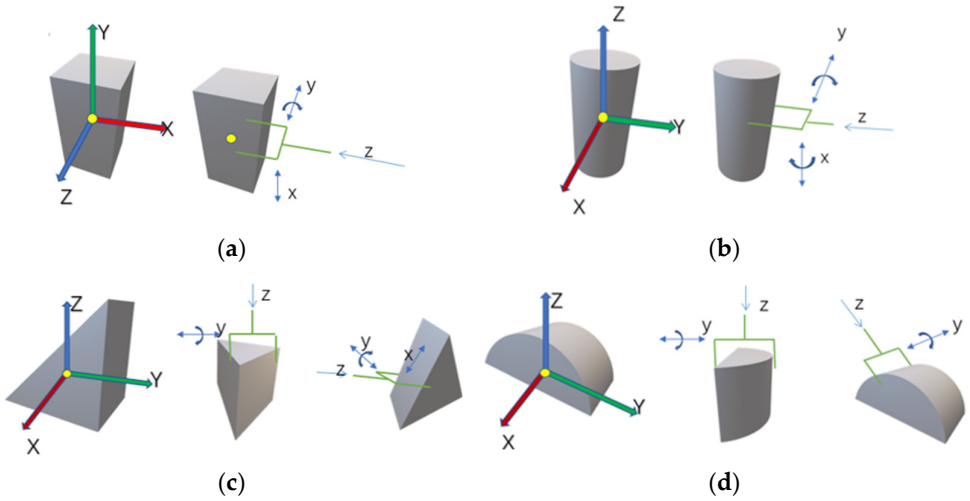

Figure 2. Grasping schematic diagram of objects along with the grasping rules. (**a**) Cuboid; (**b**) cylinder; (**c**) tri-prism; (**d**) hemicylinder.

The setting rule of the grasping dataset is based on common sense in daily life. For two-finger grasping, people are accustomed to approaching the surface of an object in the direction of its normal, and the surface normal of the object is located on the line connecting the contact points of two fingers. In addition, the grasping position is located near the center of mass of the object in general. This kind of object grasping is stable and meets the force closure conditions. Therefore, the grasping pose can be determined through the above rule in the simulation.

As shown in Figure 2, taking the cuboid as an example, the directions of its coordinate system represent the directions of three surface normals. The object posture is random in the scene. When the coordinate axis of the object points to a positive component, the z-axis direction of the grasping is the opposite of the object coordinate axis. Otherwise, the z-axis of the grasping and the object coordinate axis are in the same direction. Therefore, the z-axis, y-axis, and x-axis directions of the grasping are determined first. To improve the generalization of the dataset and consider the possible situations, a rotation of the gripper around its y-axis could be allowed, which is $r_y \sim N(0,\ \pi/18)$. The offset distances $d_z \sim U(0.015 - l_z/2,\ 0.035 - l_z/2)$ of the gripper along the z-axis, which are related to the geometric size of the object and depth of the gripper, could be allowed and are determined by the simulation experiments. In addition, an offset distance of the gripper along the x-axis is allowed, which is $d_x \sim N(0, l/6)$. Similarly, the setting rules of the grasping dataset for cuboid, cylinder, tri-prism, and hemicylinder are shown in Figure 2 and Appendix A.

In each scene of the simulation, grasping attempts for all objects were performed in accordance with the above grasping rules. In the entire grasping process, if the object was grasped and displaced successfully, and the gripper did not collide with other objects or boundaries, the grasping pose was marked as a positive label. Otherwise, with the occurrence of a collision or the failure to displace the object, the grasping pose was marked as a negative label. The ratio of positive labels to negative labels in the generated dataset is approximately 1:3.

Based on the established dataset, the training for the CNN model was performed. The CPU of the hardware platform used in the training process was Intel i7-10700 (Santa Clara, CA, USA), and the graphics card was NVIDIA's GTX1650 (Santa Clara, CA, USA). Meanwhile, the operating system was 64-bit Windows 10, and the programming language of the application was Python 3.8. The deep learning framework was built based on PyTorch-ignite-0.4.6.

The data used for training and testing were randomly selected from the dataset in a 9:1 ratio. The Adam (Adaptive Moment Estimation) optimizer was employed to update the parameters in backpropagation during the training process. Meanwhile, the learning rate was set to 0.0003. In order to improve the training efficiency, the batch size was set to 48, and 30 epochs were performed on the training samples. The losses during the training process were calculated by Equation (5) and are shown in Figure 3. The loss in the initial epochs of training decreased rapidly and gradually converged to a smaller value. The effectiveness of the trained CNN model was verified.

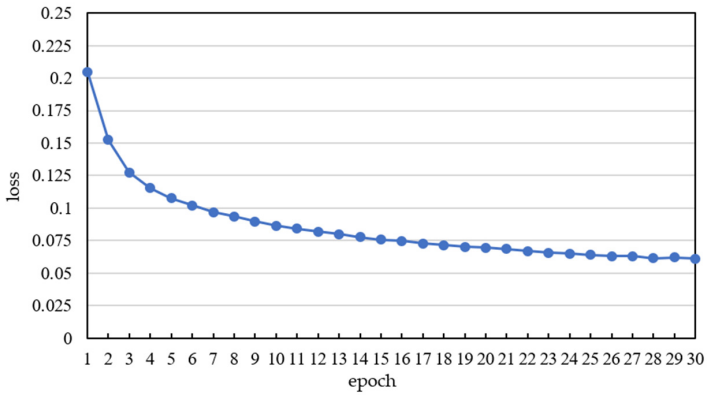

Figure 3. The losses during the training process.

After each epoch, the testing dataset was input into the trained CNN model and compared with the outputs. The accuracy was calculated by Equation (6) and is shown in Figure 4. The accuracy of the testing dataset in the initial epochs of training increased rapidly and gradually stabilized around 0.95.

$$\text{acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Here, *TP* represents that the label is positive, and the prediction result is also positive; *TN* represents that the label is negative, and the prediction result is also negative; *FP* represents that the label is negative, but the prediction result is positive; and *FN* represents that the label is positive, but the prediction result is negative.

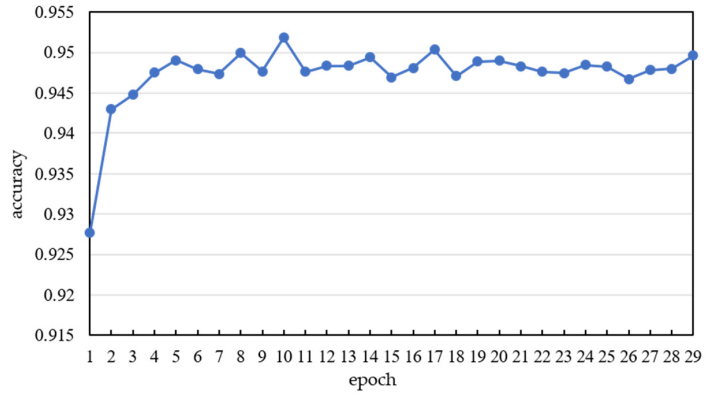

Figure 4. The accuracy of the testing dataset.

2.4. Optimization of the Grasping Pose Estimation Method

The grasping position has a significant impact on the grasping results. Taking a cuboid object as an example, the outputs of the CNN model show that the surfaces of the cuboid are probability positions that can be successfully grasped. However, when the line connecting the two grasping points of the fingers and the object is perpendicular to and intersects with the gravity direction of the object, the friction force between the gripper and the object can be equivalent to a force acting on the center of mass of the object. And the friction force only needs to balance the object's own gravity in this situation. Otherwise, a torque causing object rotation will be generated inevitably, and the grasping success rate will be obviously reduced [24], as shown in Figure 5. Therefore, the optimal grasping position should be located near the centroid of the object.

Figure 5. The influence of the grasping position. (**a**) Grasping in the centroid position; (**b**) grasping in the non-centroid position.

Therefore, the DBSCAN (density-based spatial clustering of applications with noise) method [25] was employed to determine the centroid position of the object. By clustering the scene point cloud, each object was classified as a separate category, and then the centroid position of each category was determined. In addition, the RANSAC (random sampling consensus) method [26] for the plane segmentation was employed to avoid the influence of the ground point cloud on the clustering.

For the RANSAC method of the plane segmentation, the corresponding plane model can be expressed as Equation (7).

$$ax + by + cz + d = 0 \tag{7}$$

Let us assume that we have a point cloud $P = \{p_i \in R^3, i = 1, 2 \cdots N\}$ with N-many points, and that the number of samples in the minimum sampling set S is 3. Firstly, a minimum sampling set is randomly selected from the point cloud and input into Equation (7) to obtain a plane model M. For all other points in the point cloud, if the distance from the point to the plane is less than the threshold, the point is in the plane; otherwise, the point is out of the plane as an abnormal point. The normal point set in the point cloud for the plane model and the abnormal point set are distinguished. In addition, the least squares method was used to perform fitting on the normal point set to obtain a new model M^*. Iterative optimization of the plane model was repeated until it met the deduction conditions. It is worth emphasizing that the RANSAC method has limitations due to its reliance on threshold setting, specifically for real-world applications where the features of various object are less distinctive, such as piles of deformable polybags or semi-rigid containers.

To avoid the influence of the ground point cloud on the clustering, the threshold for plane segmentation was set to 10 mm, and the maximum number of iterations was 100. The 3D voxel information of the scene and objects was converted into point cloud information. As shown in Figure 6, the yellow point cloud represents the segmented plane point cloud, and the red represents the point cloud of the out-of-plane object.

(**a**) (**b**)

Figure 6. The results of plane segmentation. (**a**) Original image; (**b**) plane segmentation.

The DBSCAN method is a density-based clustering algorithm. By setting a range and the minimum number of points in the neighborhood, several high-density regions in the point cloud were set to continuously search. The range of the neighborhood was set to 0.02 m, and the minimum number of points in the neighborhood was set to 100, after experimental attempts. Meanwhile, in order to prevent the nonuniform density of some areas due to the camera visual field, each point cloud cluster was required to contain at least 500 points to be identified as a category. The results are shown in Figure 7. It is worth emphasizing that the RANSAC method has limitations segmenting the objects with occlusions and overlaps.

(**a**) (**b**)

Figure 7. The results of point cloud clustering. (**a**) Original image; (**b**) point cloud clustering.

For the case of multiple objects, the grasping order was according to the centroid positions of objects, considering a top-down grasping approach. Based on the grasping success rate at each position, all grasping poses with a success rate larger than the threshold $\varepsilon = 0.9$ were output. Then, the distances between the successful grasping poses and the centroids of the objects were calculated using Equation (8). All grasping poses were sorted according to the value.

$$\rho_i = \sqrt{\lambda_1(x_i - x_c)^2 + \lambda_2(y_i - y_c)^2 + \lambda_3(z_i - z_c)^2}/q_i \quad (8)$$

Here, $\lambda_1 = 0.45$, $\lambda_2 = 0.45$, $\lambda_3 = 0.1$, q_i is the value of the grasping success rate for each pose, $(x_i,\ y_i,\ z_i)$ are the grasp position coordinates, and $(x_c,\ y_c,\ z_c)$ are the centroid position coordinates.

3. Experiments and Results

3.1. Experimental Platform

An experimental platform was established as shown in Figure 8. The experimental platform was based on the Kinova Gen2 robot (Boisbriand, QC, Canada), and the two-finger gripper was MicoHand (Boisbriand, QC, Canada). The visual sensor was the Intel RealSense D435i (Santa Clara, CA, USA). The configuration of the host PC was the same as that used in the CNN model training in Section 2.4. In addition, the simulation was performed synchronously based on the V-rep.

Figure 8. The experimental platform for the robot grasping.

The experimental process of each grasping task is shown in Figure 9. The experimental system mainly consisted of four parts: visual perception, grasping planning, action execution, and visualization. The depth camera was used to obtain the color and depth maps in real time, with the depth map used for scene reconstruction and the color map used to visualize the camera's perspective. For each grasping task, the first step was to perform a 3D scene reconstruction using multi-views fusion. The robot with the camera was controlled to reach the designated position in sequence, and after completing the observation tasks of three views, the voxel grids containing 3D scene information were generated. Then, the voxel grids were input into the trained CNN model. And, after optimizing the output results of the CNN model, the final grasping pose was obtained. Finally, coordinate conversion was performed, and the robot completed the subsequent grasping action.

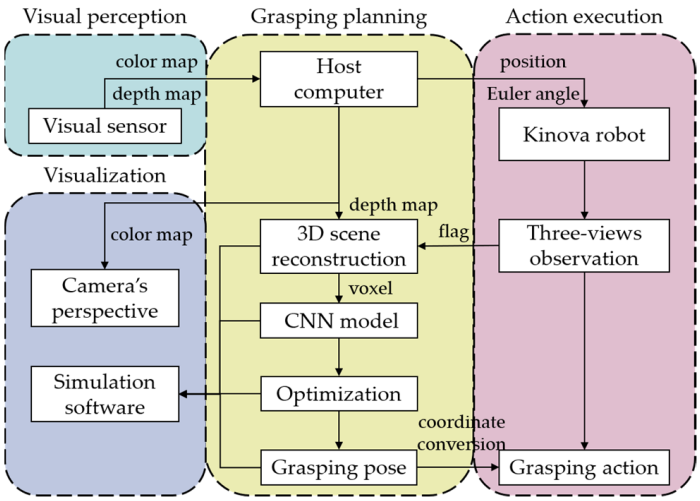

Figure 9. The experimental process of each grasping task.

3.2. Single-Object Grasping Experiments

Eleven common everyday objects were selected for grasping experiments, including objects with regular geometric shapes, such as a milk carton with a rectangular shape, a

coffee can with a cylindrical geometric shape, and some objects with special shapes such as duck toys and special parts.

For the single-object grasping experiment, the grasping success rate was used for the evaluation. The whole grasping process was as follows: the robot moved to the three-observation perspective for scene reconstruction, planned the grasping pose of the end of the robot, moved to the designated pose for grasping, and placed the object in the designated position, as shown in Figure 10. If the object was not grabbed successfully or the object fell during the movement, the grasping process failed.

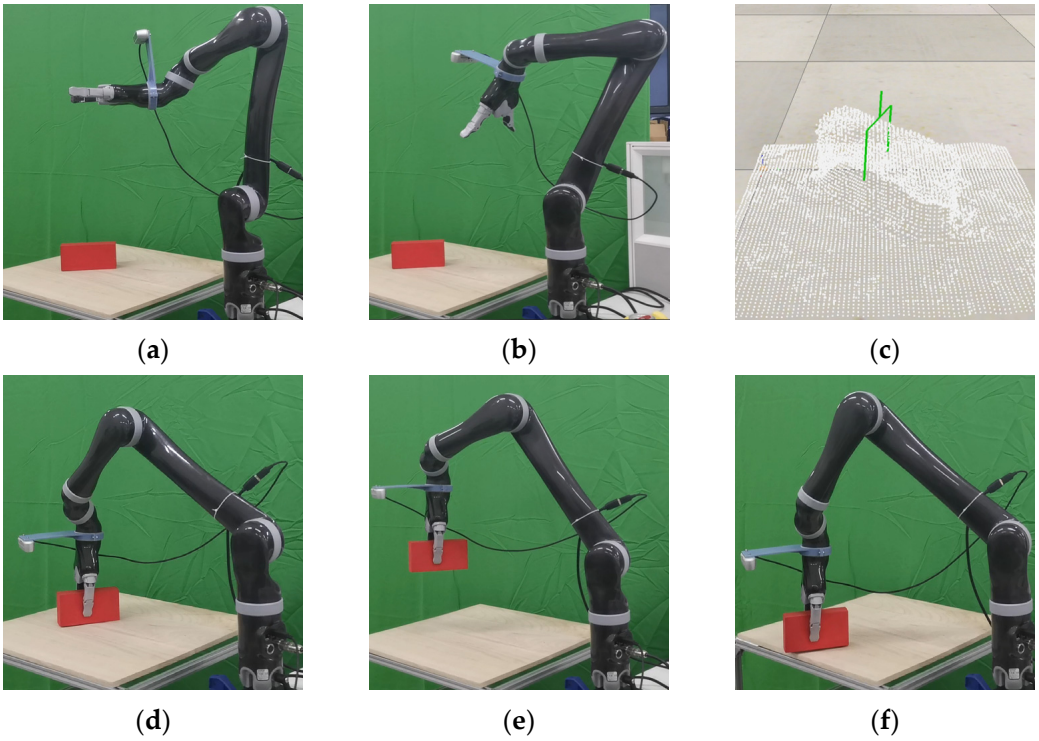

Figure 10. The whole process of the single-object grasping experiment. (**a**) Initial position; (**b**) 3D scene reconstruction; (**c**) grasping planning; (**d**) grasping action; (**e**) moving procedure; (**f**) placement.

Firstly, grasping experiments were performed on the two regular objects, cuboid and cylinder. According to the above process, grasping attempts were performed 100 times for each type of object. In these experiments, the poses of the object were randomly chosen. The side length of the cuboid exceeded the limit of the gripper opening width in the situations of the flat placements. Therefore, these cases were removed, and the number of times the grasping task was completed for each viable attempt was recorded, as shown in Table 1.

Table 1. Results of the regularly shaped object grasping experiments.

Object	Number of Grasping Attempts	Number of Viable Grasping Attempts	Number of Successes	Grasping Success Rate
Cuboid	100	90	86	95.6%
Cylinder	100	100	95	95%
Total	200	190	181	95.3%

A 95.6% and 95% grasping success rate was obtained for cuboid and cylinder, respectively. Overall, this method had a 95.3% grasping success rate for these two types of objects.

Further, grasping experiments were carried out 10 times on 11 kinds of objects, as shown in Figures 11 and 12, and Video S1. The results of each grasping experiment were counted, as shown in Table 2.

Figure 11. Successful cases in the single-object grasping experiment.

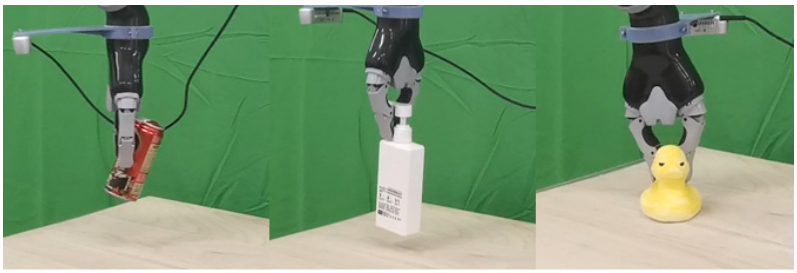

Figure 12. Failure cases in the single-object grasping experiment.

Table 2. Results of the single-object grasping experiments.

Object	Number of Grasping Attempts	Number of Successes	Success Rate
Cuboid	10	9	90%
Milk carton	10	10	100%
Motor carton	10	10	100%
Shampoo bottle	10	8	80%
Roll of paper	10	10	100%
Coffee can	10	9	90%
Medicine bottle	10	8	80%
Measuring reel	10	10	100%
Sticky tape	10	7	70%
Duck toy	10	7	70%
Printed part	10	9	90%
Total	110	97	88.2%

Of the 110 experiments, the grasping tasks were completed successfully 97 times, and the grasping success rate was 88.2%. For the failed experiments of the shampoo bottle, which was approximately cuboid, the main reason was that the bottle nozzle disturbed the motion planning. And for the failed experiments of the medicine bottle and sticky tape, which were approximate to cylinders, the main reason was the objects slipping from the two-finger gripper. The printed part and the duck toy had irregular shapes, with a 90% and 70% grasping success rate, respectively. Some successful cases are shown in Figure 11, and some of the typical failure cases are shown in Figure 12 and Video S4.

Since the 3D reconstruction process only requires depth maps, lighting conditions should not influence the grasping results in principle. To verify the adaptability of the method to lighting conditions, random experiments were performed 10 times in a dark environment without illumination. An infrared thermal camera was used to record the grasping process, as shown in Figure 13 and Video S2. The results showed that the grasping success rate was 90% in these experiments. The grasping pose estimation results of the method were barely influenced by the lighting conditions in our experiment scenes.

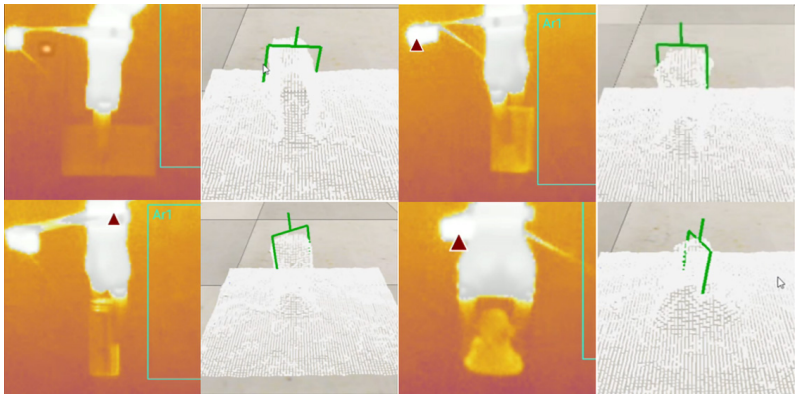

Figure 13. Experiments in a dark environment without illumination.

3.3. Multiple-Object Grasping Experiments

For the multiple-object grasping experiments, several objects were randomly selected, and four multiple-object scenes were constructed, as shown in Figure 14 and Video S3. In each scene, the robot repeated the grasping actions until all objects in the scene were cleared. The numbers of objects and grasping attempts in each scene were counted, and the grasping experiment for each scene was performed five times, as shown in Table 3.

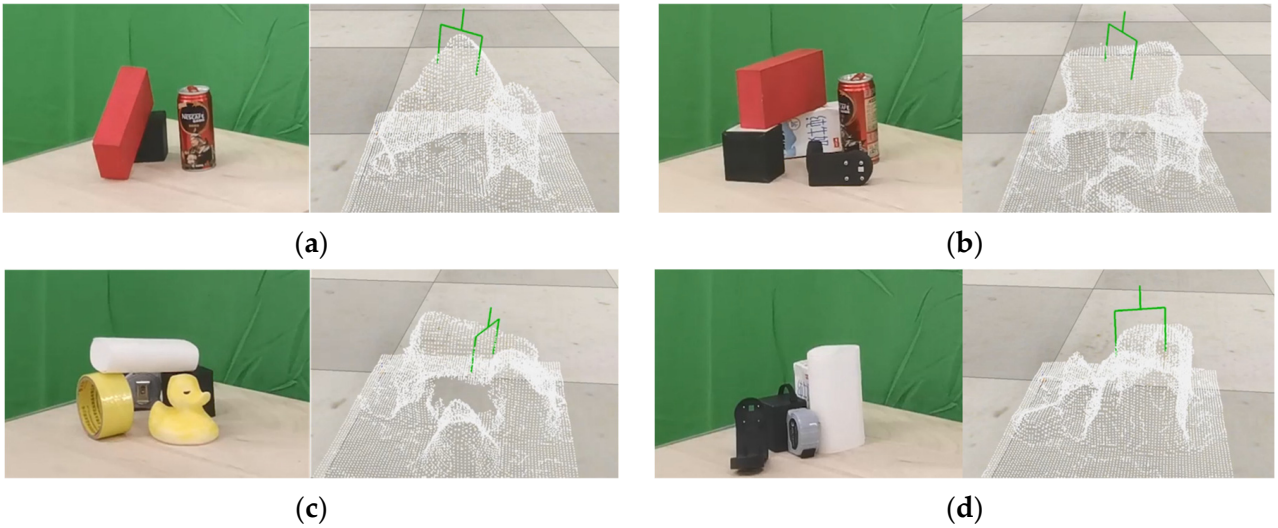

Figure 14. Stacking scenes in the multiple-object grasping experiments. (**a**) Scene I; (**b**) scene II; (**c**) scene III; (**d**) scene IV.

The overall grasping success rate of the algorithm was 81.1%, which was lower than that of the single-object environment. In particular, the grasping success rate was 71.4% in Scene IV. This is due to the mutual occlusion and stacking of multiple objects, which interfered with the planning process of the algorithm. The stacking situation has an influence on the determination of the centroid of the object, as it is difficult to segment the point cloud of each object individually. The position of the centroid of the determined

object may appear between the two objects, resulting in a grasping failure. In addition, grasping failures because of the robot's workspace limitation also occurred.

Table 3. Results of the multiple-object grasping experiments.

Scene	Number of Objects	Number of Grasping Attempts	Number of Successes	Success Rate
Scene I	3	16	15	93.8%
Scene II	5	34	25	73.5%
Scene III	5	26	25	96.2%
Scene IV	5	35	25	71.4%
Total	18	111	90	81.1%

The proposed method was compared with VGN [23] in the consistent environment of the multi-object grasping experiments. And a 2.8 percent higher overall success rate of the proposed method was achieved, as shown in Table 4.

Table 4. Comparison of success rates between the proposed method and VGN [23] in the multiple-object grasping experiments.

Scene	Number of Objects	Number of Successes	Proposed Method		VGN [23]	
			Number of Grasping Attempts	Success Rate	Number of Grasping Attempts	Success Rate
Scene I	3	15	16	93.8%	16	93.8%
Scene II	5	25	34	73.5%	33	75.7%
Scene III	5	25	26	96.2%	30	83.3%
Scene IV	5	25	35	71.4%	36	69.4%
Total	18	90	111	81.1%	115	78.3%

Based on the above experiments, the method proposed in this paper demonstrates good adaptability to grasping different geometric objects. In particular, for grasping slender axis objects, it has a high success rate. The robot can grasp this kind of objects in different positions and orientations, such as the milk carton and motor carton. Meanwhile, unknown objects can be applied to the method, and the pose for grasping can be obtained. The robot employing this method can grasp irregularly shaped objects, such as the duck toy and printed part. The grasping results show that the method is successful in planning a pose for grasping irregularly shaped objects, based on the dataset of objects with regular shapes. Furthermore, the grasping results are almost uninfluenced by lighting conditions. Even in a completely dark environment without illumination, the method can be successfully used to grasp objects.

The application of the method also has limitations, which will be the main research focus in future work. The method was unable to adapt well to the situation of pose planning for grasping flat objects in the experiments. In addition, the grasping pose may be not the optimum result in real-world cluttered environments, where accurately obtaining the centroid position of objects in occlusion and overlap scenarios is limited. Therefore, this research has the potential to improve the segmentation algorithm and appropriate metrics for ranking grasping poses. These further improvements of the adaptability of robots in various practical scenarios can support the applications in industries like manufacturing, healthcare, and logistics.

4. Conclusions

In this work, we presented a stable grasping pose estimation method for robots. In this method, the grasping success rate, approach angle, and gripper opening width can be output from the input voxel through a CNN model. Meanwhile, the grasping dataset was produced based on common sense, and the model was trained in the physical simulator. In

addition, the grasping position was optimized according to the distribution of the object centroid in order to improve the grasping success rate. Finally, grasping experiments involving 2 regularly shaped objects, 11 single objects, and multiple objects, as well as in a dark environment without illumination, were performed in the established experimental platform, and the effectiveness of the method was validated. The total grasping success rate was 90.5% for the regularly shaped objects, 88.2% for the single objects, and 81.1% for the cluttered scene of multiple objects. The results show that a robot using this method can grasp different geometric objects including irregular shapes. Meanwhile, the method has adaptability to a dark environment without illumination. This method can be optimized to accurately estimate the centroid position of objects in occlusion and overlap scenarios to further improve its adaptability in real-world cluttered environments.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/machines11100974/s1, Video S1: The single-object grasping experiments. Video S2: Experiments in a dark environment without illumination. Video S3: The multiple-object grasping experiments. Video S4: Failure cases in the grasping experiments.

Author Contributions: Conceptualization, J.Z., D.S. and Y.Z.; methodology, T.Z., C.W. and Y.W.; software, Y.W. and S.Z.; validation, T.Z. and C.W.; data curation, Y.W. and S.Z.; writing—original draft preparation, T.Z. and C.W.; writing—review and editing, Y.Z.; visualization, T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China, grant number 2022YFB4700300; the National Nature Science Foundation of China, grant numbers 52105016 and 52025054; and the fellowship of China Postdoctoral Science Foundation, grant number 2022M710957.

Data Availability Statement: Not applicable.

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

Appendix A. The Setting Rules of the Grasping Dataset

Types	Object Coordinates	Rules
Cuboid	X, Y, Z	a. Determine the z-axis, y-axis, and x-axis directions of the grasping; b. Allow a rotation r_y of the gripper around its y-axis, $r_y \sim N(0,\ \pi/18)$; c. Allow an offset distance d_z of the gripper along the z-axis, $d_z \sim U(0.015 - l_z/2,\ 0.035 - l_z/2)$; d. Allow an offset distance d_x of the gripper along the x-axis, $d_x \sim N(0,\ l_x/6)$.
Cylinder	X, Y	a. Determine the z-axis direction of the grasping; b. Allow a rotation r_x of the gripper around its x-axis, $r_x \sim U(0,\ 2\pi)$; c. Allow a rotation r_y of the gripper around its y-axis, $r_y \sim N(0,\ \pi/18)$; d. Allow an offset distance d_z of the gripper along the z-axis, $d_z \sim U(0.015 - l_z/2,\ 0.035 - l_z/2)$; e. Allow an offset distance d_x of the gripper along the x-axis, $d_x \sim N(0,\ l_x/6)$.
	Z	Same as cuboid
Tri-prism	X	a. Determine the z-axis, y-axis, and x-axis directions of the grasping; b. Allow a rotation r_y of the gripper around its y-axis, $r_y \sim N(0,\ \pi/27)$; c. Allow an offset distance d_z of the gripper along the z-axis, $d_z \sim U(0.015 - l_z/2,\ 0.035 - l_z/2)$.
	Y, Z	Same as cuboid
Hemicylinder	X	a. Determine the z-axis, y-axis, and x-axis directions of the grasping; b. Allow a rotation r_y of the gripper around its y-axis, $r_y \sim N(0,\ \pi/24)$; c. Allow an offset distance d_z of the gripper along the z-axis, $d_z \sim U(0.015 - l_z/2,\ 0.035 - l_z/2)$.
	Y, Z	a. Determine the z-axis, y-axis, and x-axis directions of the grasping; b. Allow a rotation r_y of the gripper around its y-axis, $r_y \sim U(-\pi/2,\ \pi/2)$; c. Allow an offset distance d_z of the gripper along the z-axis, $d_z \sim U(0.015 - l_z/2,\ 0.035 - l_z/2)$.

References

1. Du, G.; Wang, K.; Lian, S.; Zhao, K. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review. *Artif. Intell. Rev.* **2021**, *54*, 1677–1734. [CrossRef]
2. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Beetz, M. Persistent point feature histograms for 3D point clouds. In Proceedings of the International Conference on Intelligent Autonomous Systems, Baden, Germany, 24 July 2008; pp. 119–128.
3. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3212–3217.
4. Drost, B.; Ulrich, M.; Navab, N.; Ilic, S. Model globally, match locally: Efficient and robust 3D object recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 998–1005.
5. Hinterstoisser, S.; Holzer, S.; Cagniart, C.; Ilic, S.; Konolige, K.; Navab, N.; Lepetit, V. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 858–865.
6. Hinterstoisser, S.; Holzer, S.; Cagniart, C.; Ilic, S.; Konolige, K.; Navab, N.; Lepetit, V. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 1521–1529.
7. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In Proceedings of the Conference on Robotics—Science and Systems, Pittsburgh, PA, USA, 26–30 July 2018; pp. 1–10.
8. Tekin, B.; Sinha, S.N.; Fua, P. Real-time seamless single shot 6D object pose prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 292–301.
9. Wang, C.; Xu, D.; Zhu, Y.; Martín-Martín, R.; Lu, C.; Fei-Fei, L.; Savarese, S. DenseFusion: 6D object pose estimation by iterative dense fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3338–3347.
10. He, Y.; Sun, W.; Huang, H.; Liu, J.; Fan, H.; Sun, J. PVN3D: A deep point-wise 3D keypoints voting network for 6 DoF pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11629–11638.
11. He, Y.; Huang, H.; Fan, H.; Chen, Q.; Sun, J. FFB6D: A full flow bidirectional fusion network for 6D pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 3003–3012.
12. Jiang, Y.; Moseson, S.; Saxena, A. Grasping from rgbd images: Learning using a new rectangle representation. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3304–3311.
13. Ten Pas, A.; Gualtieri, M.; Saenko, K.; Platt, R. Grasp pose detection in point clouds. *Int. J. Robot. Res.* **2017**, *36*, 1455–1473. [CrossRef]
14. Morrison, D.; Corke, P.; Leitner, J. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In Proceedings of the 14th Conference on Robotics—Science and Systems, Pittsburgh, PA, USA, 26–30 July 2018; pp. 1–10.
15. Liang, H.; Ma, X.; Li, S.; Görner, M.; Tang, S.; Fang, B.; Sun, F.; Zhang, J. Pointnetgpd: Detecting grasp configurations from point sets. In Proceedings of the International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019; pp. 3629–3635.
16. Mousavian, A.; Eppner, C.; Fox, D. 6-DoF graspnet: Variational grasp generation for object manipulation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2901–2910.
17. Sundermeyer, M.; Mousavian, A.; Triebel, R.; Fox, D. Contact-graspnet: Efficient 6-DoF grasp generation in cluttered scenes. In Proceedings of the IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021; pp. 13438–13444.
18. Mahler, J.; Pokorny, F.T.; Hou, B.; Roderick, M.; Laskey, M.; Aubry, M.; Kohlhoff, K.; Kroger, T.; Kuffner, J.; Goldberg, K. Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1957–1964.
19. Mahler, J.; Liang, J.; Niyaz, S.; Laskey, M.; Doan, R.; Liu, X.Y.; Ojea, J.A.; Goldberg, K. Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In Proceedings of the Conference on Robotics—Science and Systems, Cambridge, MA, USA, 12–16 July 2017; pp. 1–10.
20. Mahler, J.; Matl, M.; Liu, X.Y.; Li, A.; Gealy, D.; Goldberg, K. Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018; pp. 5620–5627.
21. Mahler, J.; Matl, M.; Satish, V.; Danielczuk, M.; DeRose, B.; McKinley, S.; Goldberg, K. Learning ambidextrous robot grasping policies. *Sci. Robot.* **2019**, *4*, eaau4984. [CrossRef] [PubMed]
22. Li, S.; Keipour, A.; Jamieson, K.; Hudson, N.; Swan, C.; Bekris, K. Demonstrating large-scale package manipulation via learned metrics of pick success. In Proceedings of the Conference on Robotics—Science and Systems, Daegu, Republic of Korea, 10–14 July 2023; pp. 1–11.
23. Breyer, M.; Chung, J.J.; Ott, L.; Siegwart, R.; Nieto, J. Volumetric grasping network: Real-time 6 DoF grasp detection in clutter. In Proceedings of the Conference on Robot Learning, Cambridge, MA, USA, 16–18 November 2020; pp. 1–10.

24. Liu, M.; Li, K.; Zhang, N.; Wei, N. Effects of finger combination and center of mass for digit force control during multi-finger grasping. In Proceedings of the International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, Suzhou, China, 19–21 October 2019; pp. 1–5.
25. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
26. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd* **1996**, *96*, 34.