Journal of Intelligent & Robotic Systems manuscript No.
(will be inserted by the editor)

A CNN-Based Grasp Planning Method for Random Picking of Unknown Objects with a Vacuum Gripper

Hui Zhang [*, 1, 2] · **Jef Peeters** [3] · **Eric Demeester** [1] · **Karel Kellens** [1, 2]

Received: date / Accepted: date

Abstract Robotic grasping is still challenging due to limitations in perception and control, especially when the CAD models of objects are unknown. Although some grasp planning approaches using computer vision have been proposed, these methods can be seen as open-loop grasp planning methods and are often not robust enough. In this paper, a novel grasp planning method combining CNN-based quality prediction and closed-loop control (CNNB-CL) is proposed for a vacuum gripper. A large-scale dataset is generated for CNN training, which contains more than 2.3 million synthetic grasps and their grasp qualities evaluated by grasp simulations with 3D models. Unlike other neural networks which predict grasp success by assigning a binary value or grasp quality level by assigning an integer value, the proposed CNN predicts the grasp quality via a linear regression architecture. Additionally, the method adjusts the grasp strategies and detects the optimal grasp based on feedback from a force-torque sensor. Various simulations and physical experiments prove that the CNNB-CL method is robust for random noise disturbance in observation and compatible with different depth cameras and vacuum grippers. The proposed method finds the optimal grasp from 2,000 candidates within 300 ms and achieves a 92.18% average success rate for different vacuum grippers, which outperforms the state-of-the-art methods regarding success rate and robustness.

Keywords Random picking · Unknown object · force-torque sensor · Closed-loop grasp planning · Vacuum gripper

1 Introduction

Grasping is a core task that often needs to be performed by robotic systems. However, it is still challenging due to limitations in perception and control. Many aspects, such as collision

Hui Zhang
E-mail: hui.zhang@kuleuven.be
[*] Corresponding author.
[1] ACRO research group, Department of Mechanical Engineering, Wetenschapspark 27, 3590 Diepenbeek, Belgium.
[2] Core Lab ROB, Flanders Make @ KU Leuven, Belgium.
[3] LCE research group, Department of Mechanical Engineering, Celestijnenlaan 300, 3001 Heverlee, Belgium.

avoidance, objects mass distribution, grasp region constraints and object deformation [17], need to be considered when planning for a successful grasp.

Grasp planning of unknown objects is even more challenging than that of a predefined object. One of the commonly adopted approaches is to plan the grasping movement for unknown objects based on geometric analysis. For instance, it is possible to extract primitive shapes of unknown objects, and to consider them as simplified geometries, such as cubes, cylinders, cones, for grasp planning [34,49,13]. However, primitive shape extraction is a complex process and does not work well for more complex-shaped objects. A probabilistic-framework-based grasp planning model is proposed by Dong et al. [7], which is able to plan high-accuracy grasps for unknown objects under random perturbations. Nevertheless, a pre-segmentation of the objects is demanded, which is often not straightforward when many objects are stacked to each other.

An alternative approach is to train a neural network to evaluate grasp qualities for grasp candidates. In recent research, deep learning has been widely used to improve robotic grasp performance, especially for grasp pose detection and manipulation [29,18,41]. These methods usually adopt neural networks to detect potentially successful grasp poses. The adopted algorithms often use inputs such as depth images [29], RGB-D images [16] or 3D point clouds [37,28], and output a binary value to predict grasp success or an integer value to indicate the level of grasp quality. These methods are able to detect a set of good-quality grasp poses, but are often unable to define the optimal one. The output of neural networks is always influenced by the noise on the used input data, which can fail to find a grasp position. In this case, no grasp plan can be made for the robot, resulting in the need for human intervention. Furthermore, many state-of-the-art methods merely based on neural network evaluation are open-loop grasp planning methods, which cannot detect the real-time grasp status and adjust grasp strategies to keep a stable performance in practical applications. Thus, feedback information has to be taken into account for grasp planning.

To improve the robustness of the proposed grasp planning method with different setups, a CNN-based linear regression architecture instead of a classification architecture is used to estimate the grasp quality in this work. This CNN is trained by a large dataset consisting of synthetic point clouds, grasp poses and their grasp qualities. Considering the wide use of vacuum grippers for fast picking solutions, the method was evaluated for a vacuum gripper. In addition, a closed-loop grasp planning algorithm is presented, which detects the optimal grasp and monitors the grasp status based on the point clouds from the depth camera as well as feedback from the 6-DOF force-torque sensor.

The main contributions of the proposed method are:

1) A large-scale grasp dataset is built by grasp simulations with 3D models to train the neural network, which contains 2.3 M synthetic grasp examples and their corresponding grasp qualities. Unlike many existing datasets generated by manual labels or only working well for a specific size of gripper, the proposed dataset is extensive and contains grasp examples for different sizes of grippers. Random disturbances are implemented to simulate the use of physical depth cameras. The dataset can be utilized to train neural networks for any size of vacuum gripper with a round suction cup.
2) The CNNB-CL grasp planning method is proposed for vacuum grippers. Compared with other CNN-based methods that output an integer value to predict grasp success [29,37,28], the proposed neural network is a linear regression architecture and outputs a higher-resolution grasp quality. Furthermore, the CNNB-CL method utilizes the force-torque wrench of the gripper to build the closed-loop controlling strategies

and find the optimal grasp. In addition, the CNNB-CL method is compatible with both cameras with different noise levels as well as various sizes of grippers.

3) In order to measure the difficulty of grasp planning, the complexities of the set of objects for random picking are defined from Level 1 to Level 9 according to the objects' shapes and distributions in Section 4.2. The robustness and generalizability of the CNNB-CL method are explored at Level 1 to Level 6. It keeps a good performance under the condition that several unknown objects with "medium-complexity" shapes are stacked onto each other in a "multi-layer" clutter. A universal picking solution is developed for vacuum-gripper-based picking systems.

The remainder of the paper is organized as follows: Section 2 discusses related work. In Section 3, the CNNB-CL method is introduced, and its different substeps are discussed in detail. Section 4 presents a set of practical experiments to validate and benchmark the performance of the proposed grasp planning method. Finally, Section 5 summarizes and concludes the performed work.

2 Related Work

Recent research on robotic grasping mainly engages on the grasp quality evaluation with unified metric and grasp detection in dense clutter. Current grasp pose detection methods can be divided into three categories according to their basic frameworks: analytic methods, empirical methods and synthetic methods.

Analytic methods can be divided into two research lines. One of them is to plan grasps according to physical analyses of 3D models. The main idea of this research line is as follows: collect 3D models of objects and evaluate grasp qualities for each 3D model with physical principles in random perspectives, and then match input scenes, such as point clouds or RGB-D images, to the pre-analyzed 3D model database and plan the highest quality grasp according to the pre-analyzed instances. There are two critical steps in these analytic methods: 3D model analysis and 3D object registration. In the past decade, many 3D model analysis approaches have been proposed, including caging [42], force closure [40], Grasp Wrench Space (GWS) [12,4] and Task Wrench Space (TWS) [11]. Furthermore, some researchers developed simulators, for example, GraspIt! [33], OpenGRASP [46] or SynGrasp [32], to provide open-source platforms for dexterous grasp simulations based on physical principles. These simulators often evaluate grasp quality and select the preferred grasp by maximizing the grasp quality. As for the task of 3D object registration, geometric similarity and texture features are used to match real-world objects to predefined 3D objects in the database [5,3,44,20]. The limitations of these methods have to be noted, although a high-quality grasp can be found. Firstly, 3D registration is a time-consuming operation, and it is not easy to achieve real-time grasp planning. Secondly, physical robotic grasps rely on a pre-analyzed 3D model database, so it is not a feasible solution to deal with unknown objects that are not defined in the pre-analyzed database.

Another line of research on analytic methods attempts to extract primitive shapes of objects, and considers them as simplified geometries, like cubes, cylinders, cones, to plan grasps [34,49,13]. This approach does not rely on any database and is able to plan grasps for unknown objects. However, the high-quality grasp for a primitive shape is often not the same as that for a real object, hence this strategy cannot always execute the successful grasp. Additionally, Herzog et al. proposed a shape-template-based algorithm for unknown object grasping [15]. This algorithm is able to plan grasps for unknown objects by finding

the best matching object shape templates associated with previously demonstrated grasps and achieves a 75.3% average success rate.

Empirical methods use Deep Learning to develop a model and learn grasp principles. The model inputs robotic observation, and assigns a binary value to predict grasp success or an integer value to indicate the grasp quality level. Typically, developing a neural network to learn grasp principles is considered as a classification model and trained by plenty of grasps with a label of success or failure. Normally, grasp labels are marked manually in RGB-D images and point clouds [18, 10, 24, 8]. For example, Lenz et al. created the Cornell Grasp Dataset consisting of 1,035 RGB-D images of 280 different objects with manual labels [25]. A recent research work extended the Cornell Grasp dataset to 51 K grasp examples and trained a novel Generative Residual Convolutional Neural Network (GR-ConvNet) model [23]. Researchers also attempted to develop a model with another approach: Reinforcement Learning. This approach collects grasp examples by physical robotic trials that detect and confirm whether the grasp was successful or failed with various sensors, such as a depth camera, a multi-axis force sensor or a haptic sensor [39, 36, 26, 43, 2]. The neural network learns grasp principles from thousands of physical grasp trials and is able to detect robust grasps for unknown objects. Recent research by Dasari et al. reveals that video frames can also be used for neural network training [9]. The main disadvantage is that tedious collections of grasps are needed for empirical methods no matter what kind of framework described above is used. For instance, Levine et al. ran two months of physical trials with 14 robots to collect 800 K grasp examples [27]. Although several benchmarks of grasp datasets are available online, most of them are merely compatible with antipodal grippers or similar ones. As a result, new grasp examples have to be re-collected if a neural network is to be developed for a different gripper.

Synthetic methods can be seen as combining analytic methods and empirical methods, which develop neural network models to detect robust grasps from robotic observations directly. In comparison with empirical methods, synthetic methods reduce the time required for dataset collection. Unlike empirical methods mentioned above that collect grasp examples by physical trials or human labels, synthetic methods collect grasp examples by grasping 3D models in simulation. A suitable method to evaluate the robustness of virtual grasps is critical for synthetic methods. Typically, a virtual gripper is defined to grasp 3D models from different perspectives. The robustness of each grasp is evaluated to mark a label for the grasp example, which is a similar process in analytic methods. For each grasp example, a corresponding point cloud [37, 28] or depth image [29, 31, 35, 38] is generated by rendering the 3D model. Berk Calli et al. released YCB grasp benchmarks [6], in which reconstructed 3D meshes and real RGB-D images sampled by physical depth images from different perspectives are included. Although a robust synthetic method does not require a precise segmentation of the objects [37], the latest research indicated that pre-segmentation and task-constraints improve both the accuracy and the robustness of the grasp planning algorithm [41].

In this paper, the research work builds upon the synthetic methods by using a large-scale dataset to train a CNN-based linear regression architecture for grasp quality prediction. Various experiments were implemented to explore the robustness and generalizability of the proposed grasp planning method (Section 4). Finally, a universal picking solution is developed, which is compatible with different vacuum grippers and depth cameras.

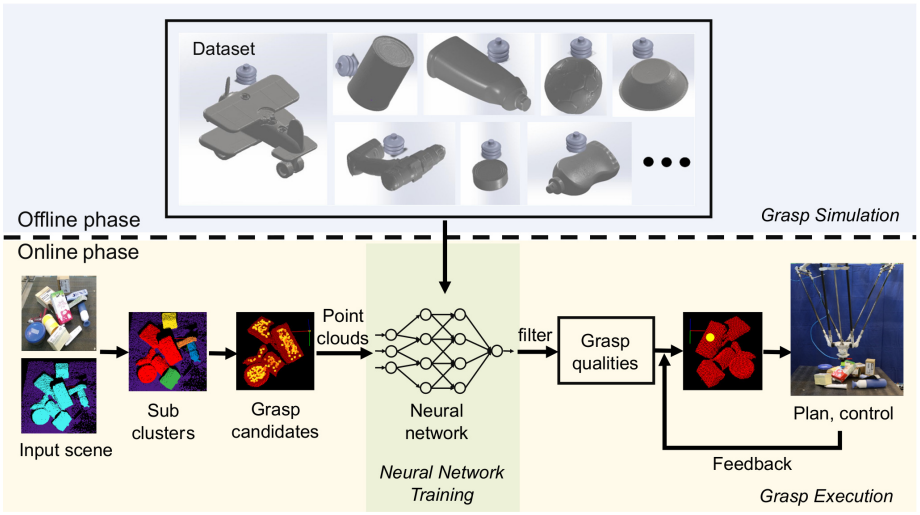


Fig. 1 Overview of the closed-loop grasp planning method.

3 Method

3.1 System Overview

Given a 3D point cloud from a depth camera as well as a gripper configuration, the grasp planning problem is to select a set of grasp candidates, then evaluate their grasp qualities and find a robust grasp to pick the object. In this paper, only vacuum-gripper-based grasping is considered, and the problem is simplified by replacing the circular model of the vacuum cup with a polygon approximation to analyze and quantify the related grasp quality. As shown in Fig. 1, the CNNB-CL grasp planning method can be divided into two phases: an offline and an online phase. Three major substeps are included: grasp simulation, neural network training and grasp execution.

Coordinate Systems. In this paper, four coordinate systems are used to evaluate the grasp quality and select the optimal grasp, named Camera Coordinate System (CCS), World Coordinate System (WCS), Robot Coordinate System (RCS) and Gripper Coordinate System (GCS) respectively. The details of these coordinate systems are shown in Fig. 2.

Grasp Simulation. The primary task of the grasp simulation is to synthesize grasp examples using a virtual gripper and 3D meshes of objects, digitize grasp qualities by numerical values and record the grasps with point clouds. The grasp simulation is mainly implemented in GCS, because any grasp trial of random picking in WCS can be converted into the grasp trial along with the vertical direction in GCS.

Given a specified 3D model $\mathcal{O}$, the random pose of the object in GCS is described as $\boldsymbol{P}_{\mathcal{O}}(x,y,z,\alpha,\beta,\gamma)$, where (x,y,z) and (α,β,γ) specify the position and rotation of the 3D model respectively.

Given a virtual gripper $\mathcal{G}$, $M_{\mathcal{G}}$ is a set of parameters of the gripper. The virtual grasp trial is implemented from the vertical direction. The grasp quality is defined by $q = Q(\boldsymbol{P}_{\mathcal{O}}, M_{\mathcal{G}}) \in \mathbb{R}$, considering the pose of object $\boldsymbol{P}_{\mathcal{O}}$ and physical properties of the gripper $M_{\mathcal{G}}$, like the coefficient of friction μ between the vacuum gripper and the object, and the maximum vacuum force $|\boldsymbol{F}_V|$.

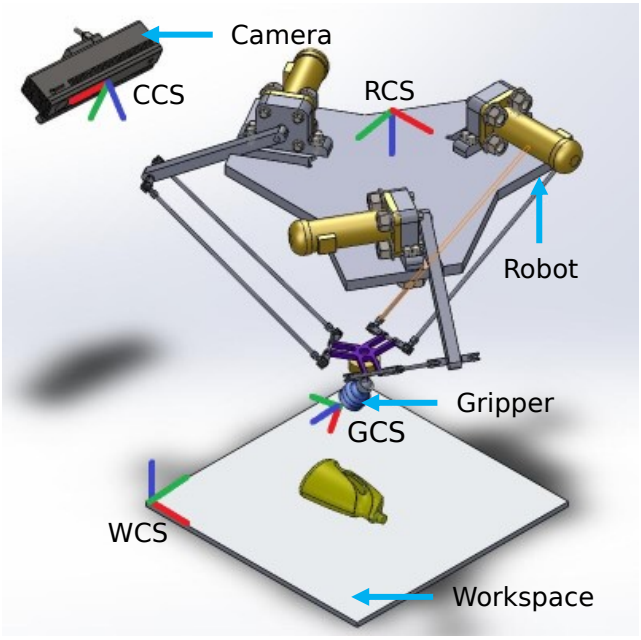


Fig. 2 Coordinate systems for grasp planning. Red, green and blue lines are the x, y, z axis of the used coordinate systems respectively. Note: Camera Coordinate System (CCS), World Coordinate System (WCS), Robot Coordinate System (RCS) and Gripper Coordinate System (GCS).

A virtual camera is deployed for robotic observation. Each grasp example is recorded by rendering a local point cloud $P \in \mathbb{R}^{3\times N}$ with N points. The details of Grasp simulation will be provided in Section 3.2.

Neural Network Training. The function of the neural network is to replace the grasp metric $Q(\boldsymbol{P}_{\mathscr{O}}, M_{\mathscr{G}})$ by $\hat{q} = Q_{\Theta}(P)$, where Θ defines the parameters of the used neural network. This replacement is necessary since the grasp quality prediction with a neural network Q_{Θ} is much faster than a traditional mathematical solution $Q(\boldsymbol{P}_{\mathscr{O}}, M_{\mathscr{G}})$ for a physical grasp trial, which will be described in Section 3.3.

Grasp Execution. According to the definitions above, the grasp quality of a real-world object cluster can be evaluated by random sampling a set of sub point clouds and predicted by the neural network Q_{Θ}. Let $M_{\mathscr{F}}$ be a set of feedback parameters from various sensors. The final grasp is defined as $\boldsymbol{g}_f \in C$, in which C is a set of grasp candidates evaluated by Q_{Θ}. The final grasp is selected with comprehensive strategies taking Q_{Θ} and $M_{\mathscr{F}}$ into account. The feedback is also utilized to optimize the grasp trail and robotic motions during the actual grasping. The details of the feedback-based optimization will be presented in Section 3.4.

3.2 Grasp Simulation: Generating a Large-Scale Dataset

The dataset is composed of a set of evaluated grasps, in which each grasp contains a local point cloud P and its grasp quality $q = Q(\boldsymbol{P}_{\mathscr{O}}, M_{\mathscr{G}})$ as shown in Fig. 3.

In the dataset, each grasp example consists of a robotic observation recording by $P_{\mathscr{O}}$ and its grasp quality q. The rest of Section 3.2 will explain how the robotic observation $P_{\mathscr{O}}$ is sampled and the corresponding grasp quality q is evaluated.

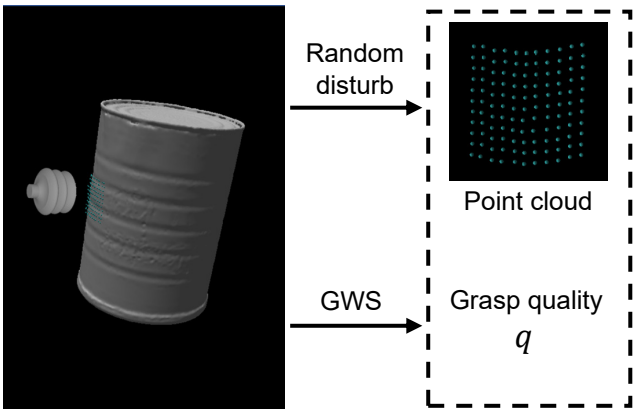


Fig. 3 A grasp candidate in the dataset.

Algorithm 1 Basic Observation Rendering

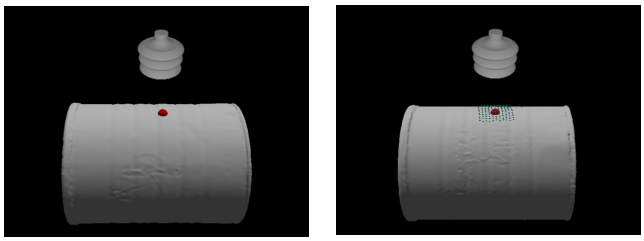
Input: 3D model $\mathscr{O}$, grasp center $\boldsymbol{c}$, virtual gripper $\mathscr{G}$, virtual camera $\mathscr{C}$
Output: observation $P_{\mathscr{O}}$
Steps:
1.1: $\boldsymbol{P}_{\mathscr{O}}$ = RandomPose($\mathscr{O}$)
1.2: $\boldsymbol{P}_{cam}$ = Set($\boldsymbol{P}_{\mathscr{O}}$,$\boldsymbol{c}$,$\boldsymbol{P}_{\mathscr{G}}$)
1.3: M_{sl} = Set($\mathscr{C}$,$\mathscr{G}$)
1.4: for $m_{sl} \in M_{sl}$ do
1.5: $\quad \boldsymbol{p}(x,y,z)$ = Intersection(m_{sl},$\mathscr{O}$)
1.6: end for
1.7: P_{org} =Desample(Collect($\boldsymbol{p}$))
1.8: $P_{\mathscr{O}}$ = Disturb(P_{org})

Observation. The observation of the object is generated based on the 3D model, which is rendered by the virtual camera with random noise.

Algorithm 1 describes the steps to generate a basic observation. Step 1.1 deploys a 3D model $\mathscr{O}$ with a random pose $\boldsymbol{P}_{\mathscr{O}}$ in the WCS and grasps the upper surface of the object with the virtual gripper $\mathscr{G}$ from the vertical direction. Step 1.2 sets the pose of the virtual depth camera $\boldsymbol{P}_{cam}$ according to the object pose $\boldsymbol{P}_{\mathscr{O}}$, the gripper pose $\boldsymbol{P}_{\mathscr{G}}$, and the grasp center $\boldsymbol{c}$. Step 1.3 defines a set of structured lights M_{sl} for the virtual camera in a grid space. The scope of the grid space is determined by both the camera resolution and the gripper size. Step 1.4 to Step 1.6 generate the point cloud by projecting the lights towards the object surface and estimating the cross points. Step 1.7 collects cross points and desamples the point cloud. The final step implements the random noises for point clouds, which simulates the case for the physical depth camera.

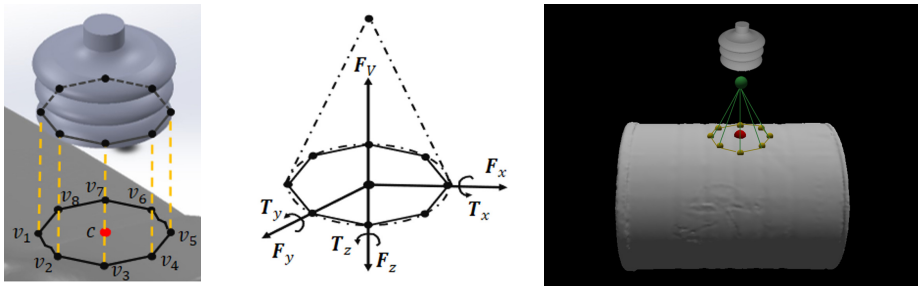
Fig. 4 presents the results of 3D model rendering and point cloud generation. To improve the neural network learning efficiency, only the desampled 11×11 point cloud around the gripper, instead of the whole observation, is taken as grasp candidate [25].

Grasp Quality. A vacuum gripper picks up an object due to an air pressure differential between the suction cup and atmosphere that sticks the object surface on the cup. A tight contact is necessary for a high-quality grasp to avoid air flowing into the suction cup and reducing the air pressure differential.



(a) (b)

Fig. 4 Example of grasp candidate. (a) A 3D object grasped by the virtual gripper. (b) The point cloud rendered by the virtual camera.



(a) (b) (c)

Fig. 5 Grasp quality evaluation based on GWS. (a) The conical spring model with an octagon bottom. (b) The forces and torques of the contact model. (c) The conical spring model for the grasp candidate.

Generally, if the objects surface is airtight and no gap exists between the perimeter of the cup and the surface, the gripper will generate the maximum vacuum force, and the grasp will be considered as a high-quality grasp. Hence, analyzing the distortion of the suction cup and the contact seal during the grasp becomes essential for grasp quality evaluation.

In this paper, the contact seal is analyzed and the grasp quality is evaluated by simplifying the suction cup into a conical spring system with an octagon bottom. The vertices of the octagon are defined as v_1, v_2, , v_8 as shown in Fig. 5. In the GCS, the forces and torques are analyzed based on GWS. The GWS is defined by a set of forces and torques, including the approaching force $\boldsymbol{F}_z$, the vacuum force $\boldsymbol{F}_V$, the frictional forces $\boldsymbol{F}_x$, $\boldsymbol{F}_y$, the torsional friction $\boldsymbol{T}_z$ and the elastic restoring torques $\boldsymbol{T}_x$, $\boldsymbol{T}_y$. The feasibility of it is evaluated under quasi-static conditions [30].

Specifically, the suction cup exerts a normal force $\boldsymbol{F}_N = \boldsymbol{F}_z + \boldsymbol{F}_V$ along the grasp direction due to the approaching force $\boldsymbol{F}_z$ and vacuum force $\boldsymbol{F}_V$. The normal force is decomposed into a set of sub normal forces and assigned to each vertex of the octagon, named $\boldsymbol{F}_N = \sum w_i \boldsymbol{f}_{N_i}$, i=1, 2, ,8 [30]. It is noteworthy that sub normal forces $\boldsymbol{f}_{N_i}$ are not always equal unless the suction cup contacts an absolutely flat surface from the vertical direction, which is not the case for a general grasp. Therefore, the weight values w_i are different and are mainly influenced by the flatness for the contact surface, coefficient of friction μ and the physical limitation of the gripper.

For the whole octagon, the GWS matrix is defined as $G \in \mathbb{R}^{6\times 8}$, in which each column is a GWS vector $[|\boldsymbol{f}_{x_i}|, |\boldsymbol{f}_{y_i}|, |\boldsymbol{f}_{z_i}|, |\boldsymbol{\tau}_{x_i}|, |\boldsymbol{\tau}_{y_i}|, |\boldsymbol{\tau}_{z_i}|]^T$ [18,30] for a vertex from the octagon computed by Equation 1.

$$\begin{cases} \boldsymbol{f}_{x_i} = w_i \boldsymbol{f}_N \cdot \boldsymbol{n}_{x_i} + \mu \boldsymbol{f}_{z_i} \cdot \boldsymbol{n}_{x_i} \\ \boldsymbol{f}_{y_i} = w_i \boldsymbol{f}_N \cdot \boldsymbol{n}_{y_i} + \mu \boldsymbol{f}_{z_i} \cdot \boldsymbol{n}_{y_i} \end{cases} \tag{1}$$

In the ideal case, the GWS vector should be $\boldsymbol{\Lambda} = [0,0,|\boldsymbol{F}_N|,0,0,0]^T$, which means that the vacuum gripper contacts with a flat surface. Definitely, it is not the case for a general grasp trial. For a general grasp candidate, the GWS vector is calculated by $G \cdot W = [|\boldsymbol{F}_x|,|\boldsymbol{F}_y|,|\boldsymbol{F}_z|,|\boldsymbol{T}_x|,|\boldsymbol{T}_y|,|\boldsymbol{T}_z|]^T$, where $W = [w_1, w_2, ..., w_i, ..., w_8]$. Then, the Euclidean Distance between $G \cdot W$ and $\boldsymbol{\Lambda}$ is defined as the robustness of the grasp. A simple method to assign the weight values $w_1, w_2, ..., w_8$ is to calculate the angles between the grasp center and vertices, and assign the weight values according to the angles. However, this method only considers the ideal case in the simulation but ignores some uncertainties and contingencies of the contact surface for a physical grasp trial. In this paper, the weight values are assigned regarding the physical limitations of the vacuum gripper, and the minimum of the Euclidean Distance is defined as the grasp quality, named $q = min||G \cdot W - \boldsymbol{\Lambda}||$ [30].

In details, the value of $G \cdot W$ is limited by several physical parameters of both the suction cup and 3D object, which include the constant of elastic limit k=0.003 for the suction cup, the coefficient of friction μ=0.5 on the contact surface, the approaching force $|\boldsymbol{F}_z|$=10.0 N and the maximum vacuum force $|\boldsymbol{F}_V|$=25.0 N. The limitations of the $G \cdot W$ are described as follows [18,30]:

$$\begin{cases} \sum w_i = 1 \\ \sqrt{3}\,|\boldsymbol{F}_x| \leq \mu |\boldsymbol{F}_N| \\ \sqrt{3}\,|\boldsymbol{F}_y| \leq \mu |\boldsymbol{F}_N| \\ |\boldsymbol{F}_z| \geq |\boldsymbol{F}_V| \\ \sqrt{2}\,|\boldsymbol{T}_x| \leq \pi r k \\ \sqrt{2}\,|\boldsymbol{T}_y| \leq \pi r k \\ \sqrt{3}\,|\boldsymbol{T}_z| \leq \mu r |\boldsymbol{F}_N| \end{cases} \tag{2}$$

All the limitations above can be converted into the limitations of w_i via the grasp matrix G. The minimization of $||G \cdot W - \boldsymbol{\Lambda}||$ subjecting to limitations of w_i is solved by Quadratic Programing (QP). This step is repeated many times with different octagon models, and the average value $\bar{q}$ is calculated as the grasp quality for each grasp.

Dataset Augmentation. To increase the diversity and robustness of the dataset, some dataset augmentation methods are implemented. On the one hand, the object is rotated several times when a grasp center is selected. Correspondingly, the grasp qualities are re-evaluated and the point clouds are re-rendered to generate more grasp examples. The object's rotation aims to simulate the uncertain pose of object in a dense clutter where many objects are randomly stacked onto each other. On the other hand, the radius of the virtual vacuum gripper is also changed from 10 mm to 30 mm (10 mm$< r <$30 mm) to improve the dataset diversity. Additionally, the coefficient of the disturbing factor σ is also set to simulate the random noise of the physical depth camera.

Several benchmarks are available for the research on robot manipulation, like 3DNet [48], KIT object database [19], YCB benchmarks [6], Cornell Grasping [25] etc, which provide various 3D mesh models, laser scans and RGB-D images. In this paper, over 2.3 M grasp candidates are generated based on 30 high-resolution 3D mesh models from YCB benchmarks with 40 hours. Specifically, the 3D models that usually cannot be grasped by vacuum gripper are excluded in YCB benchmarks, including, for example, chains, ropes,

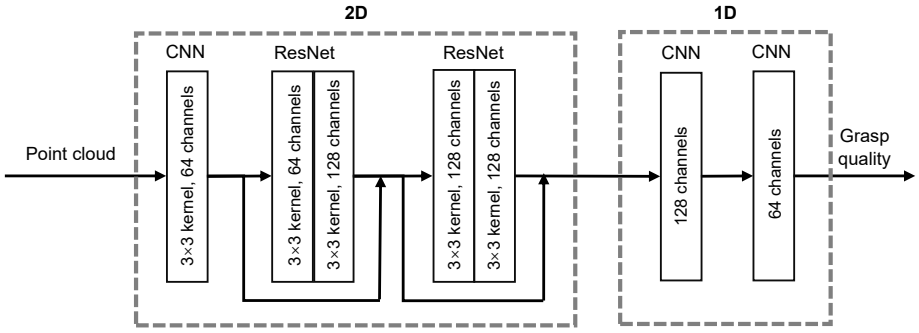


Fig. 6 Architecture of the selected neural network.

sponges, etc. The disturbing factor σ of the robotic observation is changed from 1% to 7% to improve the robustness of the trained neural network.

3.3 Neural Network Training

The task of the neural network is to define a function $Q_\Theta(P)$ to predict the grasp quality when the observation P in the GCS is given.

The grasp quality prediction is defined as a linear regression structure and solved with CNN. Over 30 networks were trained with similar architectures but different combinations of convolutional filters to find the optimal solution. The final architecture of the grasp quality prediction network is illustrated in Fig. 6, which is a 7-layer light-weighted CNN model.

As mentioned in the previous sub section, the point cloud around the coordinate (x, y, z) is encoded into three channels respectively. The CNN takes the 11×11 point cloud array, and outputs a real number to predict the grasp quality. The purpose of CNN training is to minimize the differential of the predicted value $\hat{q}$ and the real grasp quality q evaluated by GWS. The proposed CNN structure is combined with two sub structures. The first part is a 2D CNN structure. This part re-maps the input data into 64 channels and uses 2 ResNet units [14] to extract key information from the point cloud. The second part is a 1D CNN structure, which uses two layers of CNN units to descend the dimensions of the data. The performance of the network will be discussed in Section 4.1.

The Mean Squared Error (MSE) loss function is defined in Equation 3 to train the neural network. The parameters are optimized via an Adam optimizer [22].

$$L(q, Q_\Theta(P)) = \frac{1}{n} \sum_{i=1}^{n} (q_i - Q_\Theta(P_i))^2 \tag{3}$$

Adam optimizer computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. The magnitudes of Adam parameter updates are invariant to the rescaling of the gradient, and the parameters often require little tuning for the fine-tuning of a network [22].

Typically, there are six parameters to define an Adam optimizer, and three of them often need to be adjusted considering the target network and used dataset, named: a learning rate l, two exponential decay rates β_1 and β_2. The values of β_1 and β_2 should be near to 1.0. In detail, the three parameters above are determined by the following steps:

1) Define a CNN architecture, adjust the learning rate l to train the CNN until the loss function of the CNN keeps a stable status without overfitting after 100 epochs. The value of 0.0005 is provisionally selected for the learning rate l.
2) Adjust the values of β_1 and β_2 to get a better training result. Notably, the values of β_2 too close to 1 might lead to instabilities in training[22]. In this paper, the values of β_1 and β_2 are set as 0.9 and 0.999 respectively.
3) To better fit the proposed CNN architecture with different parameters for the further fine-tuning, the learning rate is initialized using multi-step gradient descent with an initial learning rate of 0.0005 and decreasing rate of 0.5.

3.4 Grasp Execution

Robotic grasp planning is a comprehensive task. Predicting the grasp quality plays an essential role in grasp planning. However, a successful grasp in the real world does not only depend on the surface flatness of the object, but also depends on the interaction of the objects with each other, especially when the distribution of objects is dense. Actually, the information acquired from observations is limited in the practical grasp. Analyzing grasp feedback from other sensors and adjusting the grasp strategies to build a closed-loop grasping method becomes necessary. For instance, the force-torque wrench of the gripper is a strong reference to detect a successful grasp. In addition, it can also be utilized to adjust the grasp strategy, approaching force and robotic speed.

The force-torque wrench feedback improves grasp performances from two aspects: adjusting the grasp strategies and robotic motion.

On the one hand, three grasp strategies for the grasp candidate selection are listed with priority below:

1) The grasp that is nearest to the cluster center.
2) The grasp that has the highest position in Z-direction.
3) A grasp that is randomly selected in the top k candidates.

The main idea of the grasp candidate selection is to find the top 10 high-quality grasps via neural network and decide the final grasp $\boldsymbol{g}_f$ with one of the three principles above according to the feedback of the previous grasp. When a failed grasp is detected, the grasp principle will be switched in the subsequent grasp trial.

On the other hand, the robotic motion is adjusted based on the real-time force-torque feedback during the grasping. First, a threshold of the approaching force $\boldsymbol{F}_N$ is defined. The robot adjusts the desired pose and avoids a badly-supported grasp when approaching to an object (Fig. 7(b)). Second, the grasp success can be detected in a similar way as described above. Once a failed grasp trial is detected, the grasp strategy will be switched in the subsequent grasp trial. Third, the robotic motion is adjusted to optimize the grasp performance. It is an effective way to avoid a failed grasp when the grasp is executed on an air-leak contact surface (Fig. 7(a)), or the object is heavy. Let $\boldsymbol{V}_{\mathscr{R}} = [v_x, v_y, v_z]$ be the desired speed of the robot along with the x, y and z axis in the RCS, and the vector $[f_{gx}, f_{gy}, f_{gz}, \tau_x, \tau_y, \tau_z]$ is the force-torque wrench of the gripper base in the RCS. The robotic speed is adjusted according to τ_{gx}, τ_{gy} and f_{gz}. Assuming that the force-torque wrench is restricted with τ_{gx}^{ref}, τ_{gy}^{ref} and f_{gz}^{ref}, the robotic speed is adjusted by Equation 4 to avoid a failed grasp during robotic moving, where k_1, k_2 and k_3 are the constants to re-scale the controlling factors for the speeds v_x, v_y and v_z. In the physical grasping, k_1, k_2, k_3, τ_{gx}^{ref}, τ_{gy}^{ref} and f_{gz}^{ref} are set as 0.3, 0.3, 0.05, 0.3 N·m, 0.3 N·m and 9.0 N respectively, and the they should be adjusted according to the

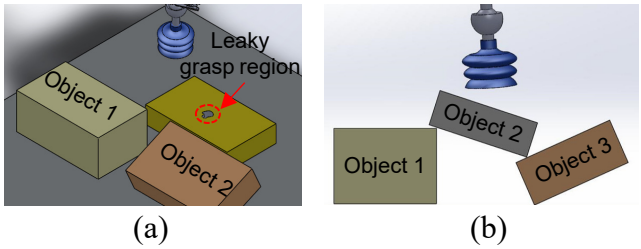


Fig. 7 Examples of bad grasps. (a) A leaky grasp region where air can flow into the vacuum gripper via a hole. (b) A badly-supported grasp. Note: the badly-supported grasp exists no matter what kind of gripper is used.

properties of the objects and setups to get a better performance. Specifically, τ_{gx}^{ref}, τ_{gy}^{ref} and f_{gz}^{ref} are adjusted considering the average weight of objects, and k_1, k_2 and k_3 are modified based on the minimum/maximum robotic speed to make sure that objects do not fall during the grasping and moving steps.

$$V'_{\mathscr{R}} = V_{\mathscr{R}} \cdot \begin{bmatrix} 1 - k_1(|\tau_{gy}|_{abs} - \tau_{gy}^{ref}) \\ 1 - k_2(|\tau_{gx}|_{abs} - \tau_{gx}^{ref}) \\ 1 - k_3(|f_{gz}|_{abs} - f_{gz}^{ref}) \end{bmatrix} \tag{4}$$

Algorithm 2 is followed for the closed-loop grasp planning. Step 2.1 implements all procedures for the point cloud $\mathbb{P}$ to reduce the disturbances for grasp planning in the subsequent steps, including background removing, voxelizing, cluster segmentation, etc. The point cloud is transferred from CCS to WCS. While a recent study reports that background removing is unnecessary to evaluate grasp quality for an antipodal gripper [35], it is still meaningful for a vacuum gripper. Otherwise, the flat table will often be believed as the most feasible grasp region. The cluster segmentation is beneficial to deal with a large cluster. Assuming 2,000 grasp candidates are randomly sampled for a large cluster with 80 cm$\times$80 cm, the average distance between each candidate is about 1.8 cm. If the input scene is segmented into two or more sub-clusters and each cluster contains several objects, it is unnecessary to evaluate the grasp quality for all sub-clusters. As a result, the average distance between each candidate will be closer, and the grasp quality can be improved. Step 2.2 samples a list of grasp candidates $\mathbb{S}$ from the point cloud. 2,000 grasp candidates are typically sampled for a physical grasp trail. Step 2.4 crops out a sub point cloud for each grasp candidate $\boldsymbol{g}_P$ and transfers it to GCS, then desamples the sub point cloud into 11×11 points and encodes data into a 3-channel matrix. In this step, the leaky grasp regions are excluded from grasp candidates as shown in Fig. 7 (a). Step 2.5 and Step 2.6 predict the quality value $\hat{q}$ for each grasp sub point cloud P via the selected neural network Q_Θ. Nevertheless, a deviation often exists between the predicted value $\hat{q}$ and the real grasp quality q. An average smooth filter is used for the grasp quality prediction to improve the accuracy, which means that the quality of each grasp is calculated by itself as well as its neighbors. Step 2.8 to Step 2.10 list the top 10 high-quality grasp candidates and select the final grasp $\boldsymbol{g}_f$ based on the 3 principles described earlier and the feedback from previous grasp trials. Step 2.11 generates the real-time robotic controlling parameters $M_{\mathscr{R}}$ in RCS, and executes the physical grasp. The feedback $M_{\mathscr{F}}$ is used to adjust the robotic motion when the gripper picks up the object.

Algorithm 2 Grasp Planning

Input: point cloud $\mathbb{P}$, feedback information $M_{\mathscr{F}}$, gripper configuration $\mathscr{G}$, neural network Q_Θ
Output: final grasp $\boldsymbol{g}_f$, robotic motion $M_{\mathscr{R}}$
Initialization: $\mathbb{Q} = \emptyset$
Steps:
2.1: $\mathbb{P}\prime$ = Preprocess($\mathbb{P}$)
2.2: $\mathbb{S}$ = Sample($\mathbb{P}\prime$)
2.3: for all $\boldsymbol{g}_P \in \mathbb{S}$ do
2.4: P = Transfer(Crop($\mathbb{P}'$,$\boldsymbol{g}_P$,$\mathscr{G}$))
2.5: $\hat{q} = Q_\Theta(P)$
2.6: $\mathbb{Q} = \mathbb{Q} \cup \hat{q}$
2.7: end for
2.8: $\mathbb{Q}'$ = Rank(Smooth($\mathbb{Q}$))
2.9: C = Top10($\mathbb{Q}'$)
2.10: $\boldsymbol{g}_f$ = Select(C,$M_{\mathscr{F}}$)
2.11: $M_{\mathscr{R}}$ = Execute($\boldsymbol{g}_f$,$M_{\mathscr{F}}$)

4 Grasping Experiments

This section describes extensive experiments to evaluate the performance of the proposed CNNB-CL grasp planning method both during the simulation and on a physical robot. A computer running on Ubuntu OS was used in the experiments, which consists of a multi-kernels 3.5 GHz Intel Core i9-9920X CPU, 64 GB of system dynamic memory (DRAM), and two Nvidia GeForce RTX 2080Ti graphics cards. The robotic system is depicted in Fig. 8. It is composed of a 4-DOF FANUC Delta robot (M-2*i*A 3SL), a 6-DOF force-torque sensor (Robotiq FT 300), a vacuum gripper (piGRIP Configurable suction cup), a depth camera (Microsoft Kinect Version 2) and a PC. The average distance between the depth camera and the work platform is 1,000 mm. The vacuum gripper is controlled via a robot controller that interacts with the PC by Socket Messaging. The depth camera and the 6-DOF force-torque sensor communicate with the PC via ROS nodes. The algorithm is programmed in Python. Due to the physical limitation of the 4-DOF robot, the vacuum gripper direction is always in the vertical direction.

4.1 Performance in Simulation

In this paper, the 2.3 M synthetic grasp examples were divided into two parts: 91% were used to train neural networks, and the remainder was utilized for tests.

Over 30 networks were trained with similar architectures but different kernel sizes and channel widths to find an optimal network structure based on the 7-layer light-weighted CNN. The channels of 7-layer CNN are defined by $c \times [16, 16, 32, 32, 32, 32, 16], c = 1, 2, 4, 8$, in which c is a constant factor to change the width of the proposed CNN. The kernel sizes of 2D CNN are set as 3×3, 5×5 and 7×7. Each CNN structure was trained over 200 epochs and tested by 200 K synthetic point clouds.

The grasp qualities of the test set changes from 2.0 to 7.5 with an average value of 3.6. The errors between the predicted quality values $\hat{q}$ and the standard quality values q were compared. Both the absolute error and the relative error were calculated for neural network assessments by the following equations:

$$absolute\ error = |\hat{q} - q|_{abs} \tag{5}$$

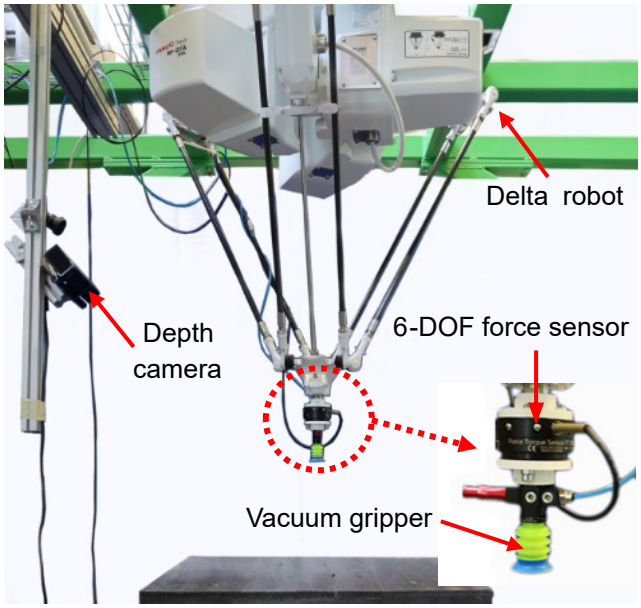


Fig. 8 Robotic setup for the experiments.

$$relative\ error = \frac{|\hat{q} - q|_{abs}}{q} \tag{6}$$

Fig. 9 presents the performances of the neural networks trained over 200 epochs. The training results indicate that the proposed CNN keeps a large average error when the kernel size is larger than 3×3 due to the low resolution of the 11×11 input point cloud. When the kernel size is 3×3, the CNN results in low relative errors (12%-18%). Hence, it is unnecessary to increase the resolution of the input point cloud with longer time cost in the real-world grasp. Furthermore, when the channel factor c is larger than four, the relative errors do not significantly decrease. The time consumption of the CNNs in Fig. 9 has no noticeable difference, as most of the time is spent on the grasp candidate sampling if the CNN structure is not very deep.

Then, the depth of the proposed CNN was increased by integrating more ResNet units. Table 1 records the absolute errors, relative errors and time consumption of the proposed CNN architectures with 2, 3, 4 and 5 ResNet units respectively. It reveals that the prediction errors of the CNNs do not dramatically decrease with over 2 ResNet units, which means that a 7-layer light-weighted CNN with 2 ResNet units is competent enough to predict the grasp quality for an 11×11 point cloud. In contrast, the time consumption is longer causing by the deeper structure, which is not good for the practical fast-picking application. In addition, a point cloud with a higher resolution often performs a better performance on the error of the grasp quality evaluation but consumes a longer time. For instance, a 7-layer CNN with 18×18 point cloud spent over 1.00 s to evaluate 2,000 grasp candidates during the tests. As a result, the final architecture of the proposed CNN is defined with c=4, a 3×3 kernel and 2 ResNet units as shown in Fig. 6.

The robustness of the proposed neural network was validated by point clouds with different noise levels. Over 6,000 synthetic point clouds with grasp qualities q ranging from 2.0 to 7.2 were selected. The grasp quality q and the predicted quality $\hat{q}$ were compared when

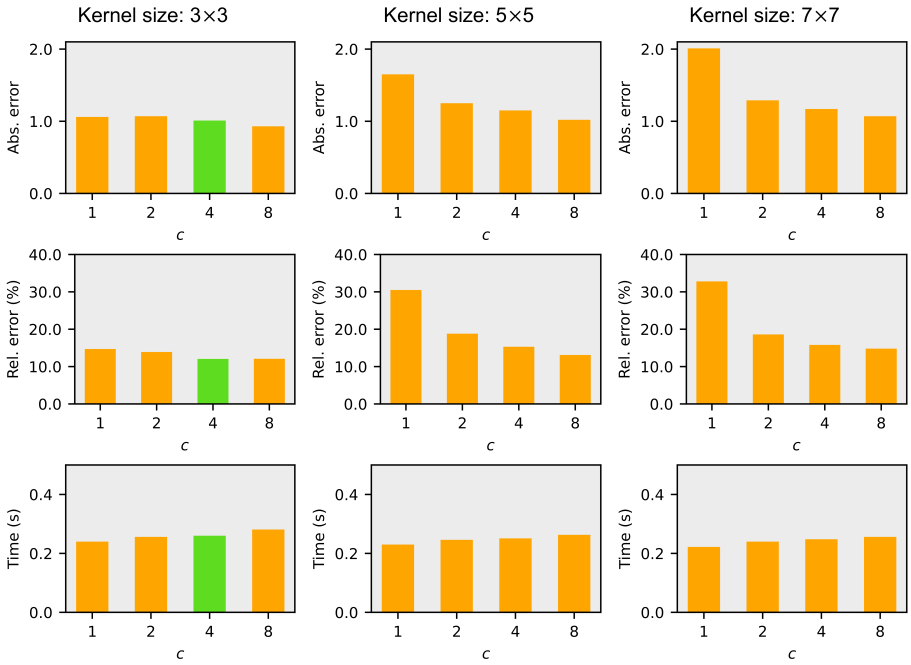


Fig. 9 Results of the neural networks training. Note: Green bars indicate the chosen architecture.

Table 1 Performances of the proposed CNNs with different ResNet units.

ResNet unit	2	3	4	5
Depth of CNN	7	9	11	13
Absolute error	1.01	0.97	0.91	0.90
Relative error(%)	12.05	11.02	10.21	10.07
Time consumption (ms)	260	270	360	970

the random noise level was changed from 1% to 9% as presented in Fig. 10. In this figure, the grasp qualities are ranked by ascending order based on their standard quality values q, then the predicted grasp quality values $\hat{q}$ under different noise levels are separately fitted by a 5-degree polynomial curve fitting. It is clear that the prediction values $\hat{q}$ increase with an increasing noise level, which reveals that the noisy disturbance has a negative influence on the grasp quality prediction. However, each fitted curve is monotonically increasing when σ is less than 9%. In other words, if the noise level is stable, the neural network always ranked the grasp qualities in the correct order, unless the noise level is quite serious. As a result, the proposed neural network has strong robustness against point cloud disturbance and is compatible with depth cameras with different noise levels. It is worth noting that the random noises were implemented on the desampled 11×11 point cloud instead of the original point cloud. If the same disturbance is implemented on the original point cloud, a better result can be expected because desampling can be considered as a smooth filter that is able to smooth the noise disturbance. The robustness and applicability of the proposed neural network will be further evaluated by experiments on a physical robot, as presented in Section 4.2.

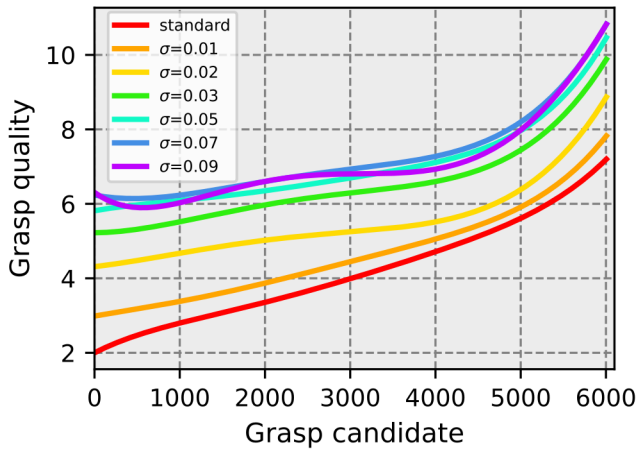


Fig. 10 Grasp quality predictions under different noise levels for synthetic grasps.

4.2 Performance on Physical Robot

The challenges of random picking mainly come from two aspects: the complexity and distribution of objects. Fig. 11(a)-(c) list some objects used in the following tests, which are separated into three categories according to the complexity of their shapes:

1) *Basic*: Objects with rigid bodies, primitive geometries (e.g. cylinders, cuboids, spheres) and smooth surfaces. No deformation happens during the grasp.
2) *Regular*: Objects with rigid bodies, varied geometries and rough surfaces. No relevant deformation happens during the grasp.
3) *Complex*: Objects with complex geometries that are difficult to find any sub simplified shape and rugged surfaces. Deformation can happen during the grasp (e.g. chain, USB cable).

Fig. 11(d)-(f) present three sets of objects in different levels of distribution. They are classified into three categories according to the average distance between the objects with each other in the clutter:

1) *Independent*: No object contacts with others in the clutter. The clutter stays on a solid and flat platform.
2) *Single-layer*: Objects contact with each other, but no overlap or only slight overlaps exist between each other. The clutter stays on a solid and flat platform.
3) *Multi-layer*: Objects are stacked onto each other in a dense clutter, and the clutter could be supported on a soft platform. Both cases could lead to a failed grasp due to the absence of the supporting force as shown in Fig. 7.

Therefore, the complexity of random picking within this paper is divided into nine levels from the easiest task to the most challenging one, according to the complexity and distribution of objects as shown in Table 2. This complexity metric is defined based on the assumption that the gripper's size and payload are suitable to grasp the objects, because the complexity metric is expected to evaluate the performance of the grasp planning method rather than that of the gripper. If the sizes and weights of objects are taken into account to define the complexity metric, it will not be fair to evaluate the performance of the grasp planning method. The following tests are implemented at Levels 1 to Level 6, because grasping

Fig. 11 Examples for the objects' complexities and distributions. (a) A set of objects with basic shapes. (b) A set of objects with regular shapes. (c) A set of objects with complex shapes. (d) A set of objects in independent distribution. (e) A set of objects in single-layer contact. (f) A set of objects in multi-layer contact.

Table 2 The complexity levels of objects' sets.

Level \ Dist. / Comp.	Independent	Single-layer	Multi-layer
Basic	1	2	3
Typical	4	5	6
Adversarial	7	8	9

Note 1: Comp. is the abbreviation of *Complexity of object*.
Note 2: Dist. is the abbreviation of *Distribution of object(s)*.

an object with complex shape (at Levels 7 to Level 9) is rather difficult, and it is not the use case for a vacuum gripper.

The used dense clutter is illustrated in Fig. 12, in which objects are randomly selected and poured on the table [37].

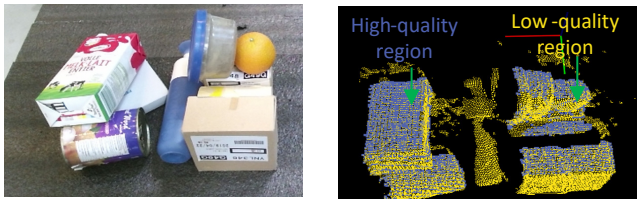
An example of the grasp quality prediction for physical grasp is shown in Fig. 13. The yellow points mark low-quality grasp regions, and the blue points show the objects surfaces with high grasp quality that are often located in flat regions.

For the physical grasp, the approaching force $\boldsymbol{F}_z$ of the vacuum gripper is set as 9.00 N. Fig. 14 shows a physical grasp example and the data from the 6-DOF force-torque sensor. The progress of the grasp is divided into four phases, and the changes of forces and torques are obvious, which give real-time feedback to improve both the grasp efficiency and success rate. In addition, the feedback is also utilized to detect the real grasp quality and optimize robotic parameters, such as moving speed, moving route and acceleration etc.

The grasp quality for a vacuum gripper mainly depends on the flatness of the object surface. However, the flatness of the grasp candidate does not only depend on the flatness of the object surface, but also depends on the performance of the depth camera during the physical grasp. One important criterion to evaluate the depth camera performance is the Root-Mean-

(a) (b)

Fig. 12 Dense clutter for experiments. (a) Part of objects used for the following tests. (b) Pouring the box contents on the working platform.



(a) (b)

Fig. 13 An example of grasp quality prediction for a real-world dense clutter. (a) The RGB image of the dense clutter. (b) Grasp qualities predicted by the proposed neural network.

Square Deviation (RMSD) at a certain distance. Previous research work concluded that the RMSD of Kinect V2 at 1,000 mm is about 1.5 mm [47]. To further evaluate the applicability and robustness of the proposed neural network, 5,000 grasp candidates were selected in physical robotic grasps. The grasp qualities are predicted with variable RMSDs to simulate point clouds from different depth cameras. Fig. 15 illustrates the results of the grasp quality predictions as RMSD ε is increasing. Considering that the standard grasp quality q in physical grasp cannot be computed as that in simulation, the grasp candidates are ranked by ascending order based on the grasp qualities when RMSD is 1.5 mm, which are the most similar predicted value $\hat{q}$ with the standard grasp quality q. In this figure, the performances of grasp quality prediction are visualized via a 4-degree polynomial curve fitting separately. It is clear that each fitted curve is monotonically increasing when $\varepsilon \leq$6.5 mm, which proves that the proposed neural network is robust enough to predict grasp quality for physical trials. Actually, plenty of research reveals that the RMSDs of common structured-light-based depth cameras, like Kinect V1/V2 [47], ASUS Xtion Pro [45] and Intel RealSense [21], are lower than 6.0 mm at 1,000 mm and much lower at a closer distance, which implies that the proposed neural network is high-generalizability and compatible with different real-world depth cameras.

Fig. 16 demonstrates the success rates of physical grasping at Level 6 when grippers sizes are changed. Each gripper was mounted on the robot and tried to grasp 10 dense clutters (100 objects totally) as shown in Fig. 12. Then the success rate r was counted by the following criterion, where N_{sg} is the number of successful grasps and N_t is the total number of trials:

$$Success\ rate = \frac{N_{sg}}{N_t} \times 100\% \tag{7}$$

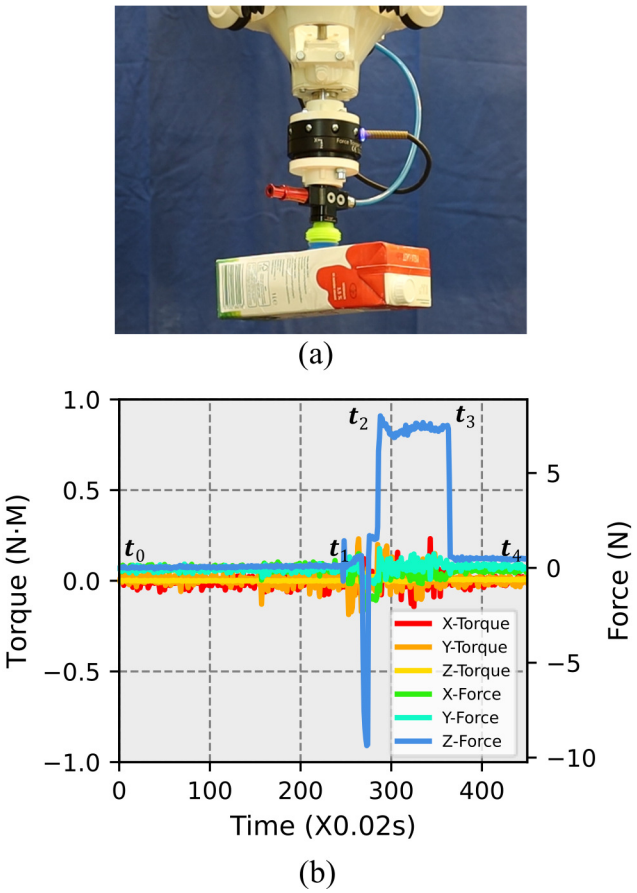


Fig. 14 A physical grasp example and the feedback from the 6-DOF force-torque sensor. (a) A physical grasp. (b) Forces and torques during the grasping. Note: Grasp is divided into 4 phases as show in the sub figure(b), the robot moves towards the object in the time slot t_0-t_1, approaches the optimal grasp candidate in t_1-t_2, lifts up the object in t_2-t_3, and then puts down it in t_3-t_4.

For the grippers with a radius of 18 mm to 27 mm, the same set of objects were used to implement the tests. As shown in Fig. 16, the success rates stay at a good level with a 92.18% average success rate. Consequently, the CNNB-CL grasp planning method can be applied with different sizes of grippers and keep a stable performance. In most cases, a failed grasp typically could be followed by a successful one and the work platform could be emptied, because the system can detect the failed grasp by the feedback and adjust the grasp priority as mentioned in Section 3.4.

4.3 Performance on Benchmark Experiments

In this section, the benchmark experiments were implemented aiming to evaluate the efficiency of the CNNB-CL grasp planning method.

Table 3 lists some basic information of four state-of-the-art methods [29,37,28,30] and the CNNB-CL method. Considering the similarities between these algorithms and the pro-

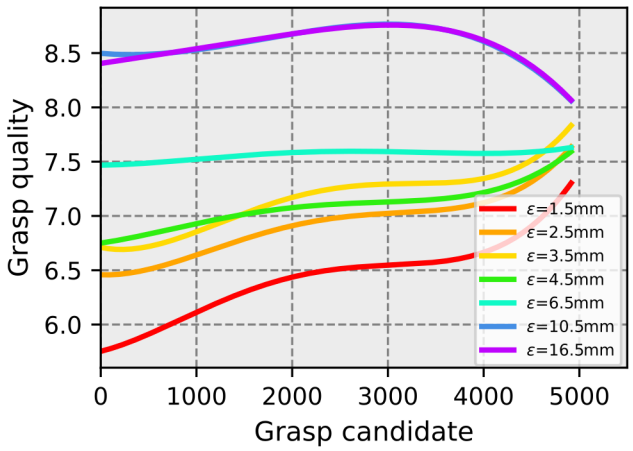


Fig. 15 Grasp quality predictions under different 3D camera noise (RMSD) levels for physical grasps.

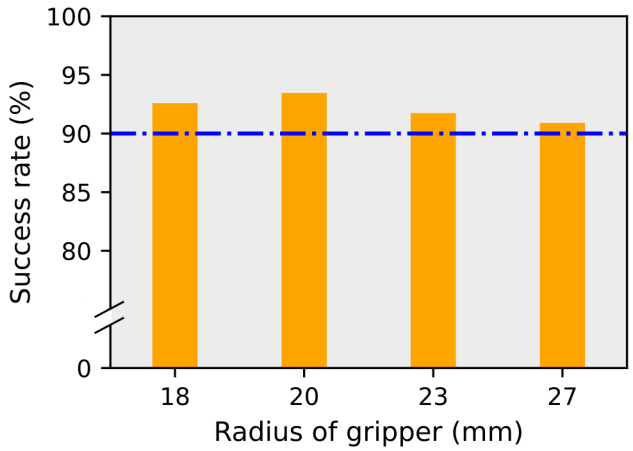


Fig. 16 Average success rates for physical grasps with different sizes of grippers and multiple clutters at Level 6.

posed method in this paper, the PointNet-GPD and Dex-Net 3.0 were selected for the comparison experiments. In addition, the commercial Pickit system (Fig. 17) was also used for the subsequent tests, which can provide picking solutions for random picking of unknown objects with jaw grippers and vacuum grippers. Notably, the objects were manually selected to make sure they can be grasped successfully by the vacuum gripper and jaw gripper used in the tests.

The success rate is a fundamental criterion to evaluate the performances of grasp planning methods in the benchmark experiments. The Dex-Net 3.0 and Pickit [1] were selected for the tests with object(s) at Level 1 to Level 6.

Table 4 presents the success rates and time consumption of the Dex-Net 3.0, Pickit, 7-layer CNN (CNNB-CL without feedback loop) and full-functional CNNB-CL method with a radius of 20 mm vacuum gripper for the random picking at Level 1 to Level 6. It

Table 3 Basic information of 4 state-of-the-art methods and the CNNB-CL.

Method	Dataset	Neural network	Gripper	Input	Output	Programming
Dex-Net 2.0	6.3 M	CNN, classification	Jaw gripper	Depth image	0, 1	Python, Tensorflow
Dex-Net 3.0	2.8 M	CNN, classification	Vacuum gripper	Depth image	Probability from CNN and GMM	Python, Tensorflow
GPD	340 K	CNN classification	Jaw gripper	Point cloud	0, 1	C++, Caffe
PointNet-GPD	300 K	CNN classification	Jaw gripper	Point cloud	0, 1, 2	Python, Pytorch
CNNB-CL	2.3 M	CNN, linear regression	Vacuum gripper	Point cloud	Real number	Python, Pytorch

Note: GMM is the abbreviation of Gaussian Mixture Model.

reveals that the Pickit achieves high success rates at Level 1 to Level 4. Hence, the Pickit is able to provide robust solutions for the picking tasks, like bin picking for the basic objects and random picking on the conveyor belt. However, the success rates of the Pickit show a relevant decrease at Level 5 and Level 6.

The Dex-Net 3.0 performs high success rates at Levels 1, 2, 4 and 5. But it reports lower success rates at Level 3 and Level 6, causing by the failed grasp trials for badly-supported objects. In addition, the robotic speed has to be slow during the tests, otherwise failed grasp often occurs when the object is heavy or the grasp position is far away from the mass center of the object. Because the open-loop control of Dex-Net 3.0 cannot predict and detect a failed grasp when the gripper is moving. These performances conclude that the Dex-Net 3.0 has a reliable neural network to predict the grasp quality via robotic observation. However, it is not robust enough when objects are stacked onto each other due to the limitations of the open-loop grasp strategy. Notably, the time consumption of the Dex-Net 3.0 is not recorded in Table 3, because it ran on the CUP instead of the GPU in the tests due to the compatibility of the used python packages and PC. It is unfair to compare its time consumption with the CNNB-CL running on the GPU.

Fig. 18 presents three typically failed grasp trials for the Pick-it, which can be summarized below that also exist in the Dex-Net 3.0:

1) The Pickit often fails to find graspable regions for the objects at Level 5 and Level 6, especially for the dense clutters as shown in Fig. 18 (a), (b), (d) and (e). Specifically, the Pickit system grasps an unknown object by estimating the geometric center of the object and selecting a feasible grasp region near the center of the object. This strategy works well for objects with basic shapes. However, it is less competent to find a feasible grasp region when the shape of the object becomes more complex.
2) Badly-supported grasps are unable to be detected merely by the observations and open-loop control strategy as shown in Fig. 18 (c) and (f), since the Pickit cannot detect and adjust the approaching force during grasping based on the open-loop controlling strategy.

The CNNB-CL detects the optimal grasp from 2,000 candidates within 300 ms. Furthermore, the 7-layer CNN has similar success rates with the Dex-Net 3.0 at Levels 1, 2, 4 and 5, which indicates that the proposed 7-layer light-weighted CNN can find a feasible

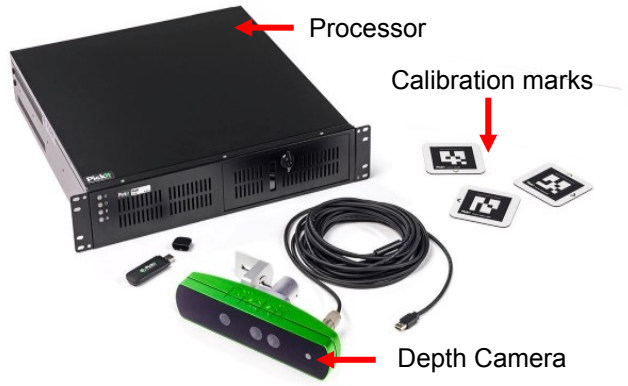


Fig. 17 Hardware of the Pickit [1].

Table 4 Performances of the Dex-Net 3.0, Pickit and CNNB-CL method at Level 1 to Level 6.

Performance \ Level		1	2	3	4	5	6
Success rate (%)	Dex-Net 3.0	100.00	**99.01**	94.34	**98.04**	92.59	87.72
	Pickit	100.00	98.04	95.24	97.09	90.91	85.47
	7-layer CNN	100.00	98.04	95.24	97.09	93.46	86.21
	CNNB-CL	**100.00**	98.04	**97.09**	**98.04**	**94.34**	**93.46**
Time (ms)	Dex-Net 3.0	-	-	-	-	-	-
	Pickit	320	330	340	330	380	450
	7-layer CNN	**236**	256	**257**	250	252	**255**
	CNNB-CL	238	**255**	261	**245**	**251**	260

grasp region based on robotic perception. Compared with the 7-layer CNN, the Dex-Net 3.0 has higher success rates owing to a more extensive training dataset and the optimization for the grasp quality prediction based on a Gaussian Mixture Model (GMM). Nevertheless, the poor performances of them at Level 3 and Level 6 report that the fully vision-based grasp planning method is not good at picking in dense clutter.

In contrast, the full-function CNNB-CL with the closed-loop controlling strategy keeps a stable performance success rate at Level 1 to Level 6. As a result, the feedback loop of the CNNB-CL significantly improves the grasp performance.

Then, the robustness of the CNNB-CL method was compared with PointNet-GPD. Grasp candidates at Level 6 were evaluated with the PointNet-GPD when the RMSD (ε) of the depth camera is changed from 1.5 mm to 4.5 mm. The outputs with ε=1.5 mm are believed as the standard reference, and the *Error* is defined as follows to evaluate the performance of PointNet-GPD:

$$Error = \hat{p} - \hat{p}_{\varepsilon=1.5} \tag{8}$$

In Equation 8, $\hat{p}$ is the output of PointNet-GPD, which is an integer value changing from 0 to 2 to indicate the grasp quality level of the candidate by ascending order. According to the equation above, the possible *Errors* are -2, -1, 0, 1 and 2. The higher *Error* means the worse prediction. Table 5 lists the results of grasp quality predictions for the 1,000 grasp candidates in the tests. In this table, the number of grasp candidate(s) under different levels of errors are counted. It shows that the output of PointNet-GPD cannot keep the same value for the same

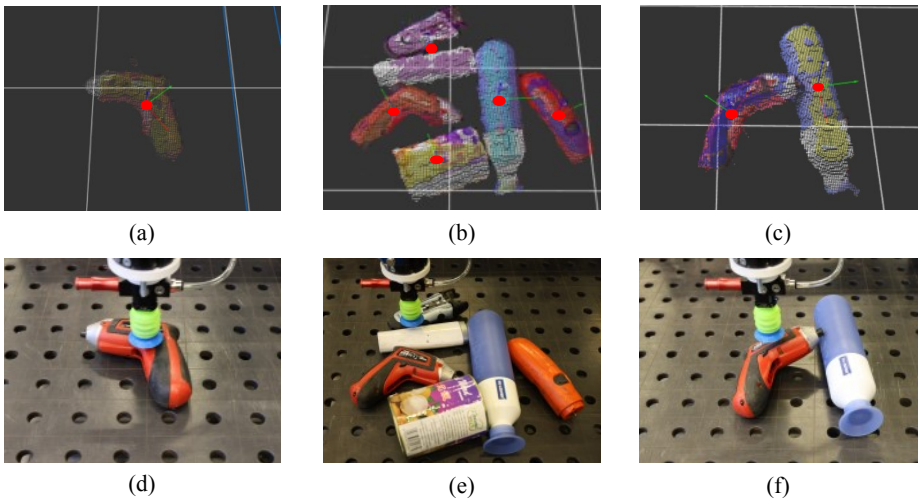


Fig. 18 Failed grasps for the Pickit. (a), (d) A failed grasp at Level 4. (b), (e) A failed grasp at Level 6. (c), (f) A badly-supported grasp at Level 5. Note: Red points in the sub figure (a)-(c) mark the grasp central points proposed by the Pickit system.

Table 5 Performances of PointNet-GPD under different noise levels.

Num. / ε (mm) / Error	1.5	2.5	3.5	4.5
-2	0	4	8	16
-1	4	15	32	50
0	1000	962	903	720
1	3	13	47	85
2	0	6	10	29

Note: Num. is the abbreviation of *Number of grasp candidate(s)*.

grasp candidate with the increasing of RMSD. Especially when $\varepsilon \geq$1.5 mm, more and more results are predicted with $Error >0$. In other words, many low-quality grasp candidates are evaluated as high-quality candidates under higher RMSD. For the grasp candidates with the same output, it is impossible to detect which one is better than others by PointNet-GPD, which implies that the low-quality grasp candidates with $Error >0$ are believed as good as the real high-quality grasp candidates and maybe lead to failed grasps. Actually, it is a common problem for the grasp pose detection algorithms based on linear classification models. Nevertheless, the CNNB-CL method is robust enough to detect the optimal grasp with $\varepsilon \leq$6.5 mm as presented in Section 4.2.

4.4 Summary and Limitations

In summary, the CNNB-CL grasp planning method is a universal and reliable solution for unknown object picking with a vacuum gripper.

In computer simulations with more than 6.0K synthetic point clouds and over 3,000 physical grasp trials, the CNNB-CL grasp planning method is robust enough to withstand different noise levels. The proposed grasp planning method is compatible with different

(a) (b) (c)

Fig. 19 The challenging objects for the CNNB-CL. (a) Complex objects at Level 7 to Level 9. (b) Light-weighted objects. (c) Black objects.

vacuum grippers and depth cameras. It achieves a success rate of over 90% for random picking tasks in dense clutters, which makes it work well in different setups. Compared with the state-of-the-art methods Dex-Net 3.0 and PointNet-GPD, the CNNB-CL grasp planning method has the advantages regarding the success rate and robustness for random picking of unknown objects at Level 3 to Level 6, especially at Level 6 when unknown objects with more complex shapes are stacked onto each other in a "multi-layer" clutter.

In the benchmark experiments, the higher success rate of the CNNB-CL method at Level 6 proves that grasp pose detection merely based on robotic observation is not reliable enough for picking in dense clutter as some additional information is helpful for grasp planning but difficult to be acquired via visual perception. In this case, increasing the complexity of neural networks makes less sense. Multi-sensor fusion and closed-loop control is a practical solution to improve the grasp performance.

Nevertheless, the limitations of the CNNB-CL have to be mentioned. Fig. 19 lists three major limitations of the CNNB-CL. First, the CNNB-CL is challenging to grasp complex objects at Level 7 to Level 9 due to the limitation of the vacuum gripper. Second, the CNNB-CL detects a successful/failed grasp and adjust the grasp strategies according to the real-time force-torque wrench of the gripper base. The detection is difficult for light-weighted objects, such as envelopes, newspapers and small carton boxes, due to the inertia of the robotic motion. Third, the proposed method does not work well for objects with a black surface in the physical grasp trails. Unlike a laser scanner or binocular depth camera, the depth camera in the tests (Kinect V2) generates depth image and point cloud based on the reflection of the near-infrared structured light, which is easy to be absorbed by a black surface. This limitation was also reported by Morrison et al. [35].

5 Conclusions

In this paper, a novel CNNB-CL grasp planning method is proposed, which integrates a CNN model to predict grasp quality and several sensors to develop the closed-loop grasp strategies. The CNN model is trained by a large-scale dataset, in which more than 2.3 M synthetic point clouds are generated, and their grasp qualities are evaluated by a unified grasp metric.

The design of the CNN model shows that some ideas from typical CNN structures for 2D image processing are also feasible to 3D point cloud processing. Over 6,000 simulations and more than 3,000 physical trials have been implemented, which reveal that the CNNB-CL grasp planning method is robust against random noise disturbance on the depth measurements and compatible with different sizes of vacuum grippers, which achieves over

90% success rates with different grippers for the random picking tasks at Level 1 to Level 6. Additionally, experiments also show that the CNNB-CL method outperforms the state-of-the-art methods regarding the robustness and success rate.

There are two main reasons that contribute to a better performance of the CNNB-CL method than other benchmark grasping methods. First, the proposed 7-layer light-weighted CNN is well-trained owing to two aspects. On the one hand, over 2.3 M synthetic grasp examples are generated with different gripper sizes and random noises, which is an exhaustive dataset to fit the cases for random physical grasp trials. On the other hand, the 7-layer CNN is robust enough to learn the grasp principles for a vacuum gripper from an 11×11 point cloud, while its architecture is relatively concise. Second, the success rate of the proposed 7-layer CNN approach has been further improved by adding a force-torque feedback loop. For example, the force-torque feedback makes it easy to deal with a badly-supported grasp existing in an over-stacked clutter and to avoid the object falling during the grasping, which often lead to failed grasps with other benchmark methods. Furthermore, three strategies are available to decide the final grasp pose for a grasp trial. Once a failed grasp trial is detected, the grasp strategy can be switched in the subsequent grasp trial until a successful grasp trial is detected.

Future research includes the application and fine-tuning of the presented approach to specific use cases, such as highly mixed post parcels and waste streams. In order to further extend the application field of the proposed method to random picking of unknown objects at clutter complexity Levels 7 to Level 9, grasp metrics for more dexterous vacuum grippers will be explored. Additionally, it is also considered to propose more measurable and objective metrics to classify objects and clusters , which will be used to define the complexities of objects' collections from Level 1 to Level 9.

Declarations

1. Funding

This work is partially supported by the China Scholarship Council (Grant NO. CSC201806090290).

2. Conflicts of interest/Competing interests

The authors declare no potential conflicts of interest with respect to the research, authorship, and publication of this article.

3. Availability of data and material

Data, graphics and video demos generated during the study are available in supplementary materials.

4. Code availability

The code and models in this paper are available via the link below. Please contact the corresponding author for any question about them.

`https://drive.google.com/file/d/1EKpnjCm_K5GOINCzXvvEWMETdXWucAJ9/view?usp=sharing`

5. Authors' contributions

Mr. Hui Zhang (main contributor and first author): concept, method and software development, design and implementation of experiments, data analysis and original draft writing, etc.

Prof. Dr. Ing. Jef Peeters: follow up, review and iteration of method, experiments, data analysis and manuscript.

Prof. Dr. Ir. Eric Demeester: follow up, review and iteration of method, experiments, data analysis and manuscript.

Prof. Dr. Ing. Karel Kellens (project promotor): concept and method optimization, data analysis, review and editing of manuscript.

6. Ethics approval

Not applicable.

7. Consent to participate

Not applicable.

8. Consent for publication

Not applicable.

References

1. Pickit. https://www.pickit3d.com/ (2020)
2. Ackermann, E.: How google wants to solve robotic grasping by letting robots learn for themselves. IEEE Spectrum, March **28** (2016)
3. Bohg, J., Johnson-Roberson, M., León, B., Felip, J., Gratal, X., Bergström, N., Kragic, D., Morales, A.: Mind the gap-robotic grasping under incomplete observation. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 686–693. IEEE (2011)
4. Borst, C., Fischer, M., Hirzinger, G.: Grasp planning: How to choose a suitable task wrench space. In: 2004 IEEE International Conference on Robotics and Automation (ICRA), pp. 319–325. IEEE (2004)
5. Brook, P., Ciocarlie, M., Hsiao, K.: Collaborative grasp planning with multiple object representations. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2851–2858. IEEE (2011)
6. Calli, B., Walsman, A., Singh, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. arXiv preprint arXiv:1502.03143 (2015)
7. Chen, D., Dietrich, V., Liu, Z., Von Wichert, G.: A probabilistic framework for uncertainty-aware high-accuracy precision grasping of unknown objects. Journal of Intelligent & Robotic Systems **90**(1), 19–43 (2018)
8. Chu, F.J., Vela, P.A.: Deep grasp: Detection and localization of grasps with deep neural networks. arXiv preprint arXiv:1802.00520 (2018)
9. Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., Finn, C.: Robonet: Large-scale multi-robot learning. arXiv preprint arXiv:1910.11215 (2019)
10. Della Santina, C., Arapi, V., Averta, G., Damiani, F., Fiore, G., Settimi, A., Catalano, M.G., Bacciu, D., Bicchi, A., Bianchi, M.: Learning from humans how to grasp: a data-driven architecture for autonomous grasping with anthropomorphic soft hands. IEEE Robotics and Automation Letters **4**(2), 1533–1540 (2019)
11. El-Khoury, S., Sahbani, A.: On computing robust n-finger force-closure grasps of 3d objects. In: 2009 IEEE International Conference on Robotics and Automation (ICRA), pp. 2480–2486. IEEE (2009)
12. Ferrari, C., Canny, J.: Planning optimal grasps. In: 1992 IEEE International Conference on Robotics and Automation (ICRA), pp. 2290–2295. IEEE (1992)
13. Fornas, D., Sales, J., Peñalver, A., Pérez, J., Fernández, J.J., Marín, R., Sanz, P.J.: Fitting primitive shapes in point clouds: a practical approach to improve autonomous underwater grasp specification of unknown objects. Journal of Experimental & Theoretical Artificial Intelligence **28**(1-2), 369–384 (2016)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE (2016)
15. Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Bohg, J., Asfour, T., Schaal, S.: Learning of grasp selection based on shape-templates. Autonomous Robots **36**(1), 51–65 (2014)

16. Jiang, Y., Moseson, S., Saxena, A.: Efficient grasping from rgbd images: Learning using a new rectangle representation. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 3304–3311. IEEE (2011)
17. Jørgensen, T.B., Jensen, S.H.N., Aanæs, H., Hansen, N.W., Krüger, N.: An adaptive robotic system for doing pick and place operations with deformable objects. Journal of Intelligent & Robotic Systems **94**(1), 81–100 (2019)
18. Kao, I., Lynch, K.M., Burdick, J.W.: Contact modeling and manipulation. In: Springer Handbook of Robotics, pp. 931–954. Springer (2016)
19. Kasper, A., Xue, Z., Dillmann, R.: The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. The International Journal of Robotics Research (IJRR) **31**(8), 927–934 (2012)
20. Kehoe, B., Matsukawa, A., Candido, S., Kuffner, J., Goldberg, K.: Cloud-based robot grasping with the google object recognition engine. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 4263–4270. IEEE (2013)
21. Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., Bhowmik, A.: Intel realsense stereoscopic depth cameras. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1267–1276. IEEE (2017)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Kumra, S., Joshi, S., Sahin, F.: Antipodal robotic grasping using generative residual convolutional neural network. arXiv preprint arXiv:1909.04810 (2019)
24. Kumra, S., Kanan, C.: Robotic grasp detection using deep convolutional neural networks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 769–776. IEEE (2017)
25. Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasps. The International Journal of Robotics Research (IJRR) **34**(4-5), 705–724 (2015)
26. Levine, S., Kumar, A., Tucker, G., Fu, J.: Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643 (2020)
27. Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. The International Journal of Robotics Research (IJRR) **37**(4-5), 421–436 (2018)
28. Liang, H., Ma, X., Li, S., Görner, M., Tang, S., Fang, B., Sun, F., Zhang, J.: Pointnetgpd: Detecting grasp configurations from point sets. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 3629–3635. IEEE (2019)
29. Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J.A., Goldberg, K.: Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. arXiv preprint arXiv:1703.09312 (2017)
30. Mahler, J., Matl, M., Liu, X., 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: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 5620–5627. IEEE (2018)
31. Mahler, J., Matl, M., Satish, V., Danielczuk, M., DeRose, B., McKinley, S., Goldberg, K.: Learning ambidextrous robot grasping policies. Science Robotics **4**(26) (2019)
32. Malvezzi, M., Gioioso, G., Salvietti, G., Prattichizzo, D.: Syngrasp: A matlab toolbox for underactuated and compliant hands. IEEE Robotics & Automation Magazine **22**(4), 52–68 (2015)
33. Miller, A.T., Allen, P.K.: Graspit! a versatile simulator for robotic grasping. IEEE Robotics & Automation Magazine **11**(4), 110–122 (2004)
34. Miller, A.T., Knoop, S., Christensen, H.I., Allen, P.K.: Automatic grasp planning using shape primitives. In: 2003 IEEE International Conference on Robotics and Automation (ICRA), pp. 1824–1829. IEEE (2003)
35. Morrison, D., Corke, P., Leitner, J.: Learning robust, real-time, reactive robotic grasping. The International Journal of Robotics Research (IJRR) **39**(2-3), 183–201 (2020)
36. Oberlin, J., Tellex, S.: Autonomously acquiring instance-based object models from experience. In: Robotics Research, pp. 73–90. Springer (2018)
37. ten Pas, A., Gualtieri, M., Saenko, K., Platt, R.: Grasp pose detection in point clouds. The International Journal of Robotics Research (IJRR) **36**(13-14), 1455–1473 (2017)
38. Patten, T., Park, K., Vincze, M.: Dgcm-net: dense geometrical correspondence matching network for incremental experience-based robotic grasping. arXiv preprint arXiv:2001.05279 (2020)
39. Pinto, L., Gupta, A.: Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3406–3413. IEEE (2016)

40. Prattichizzo, D., Trinkle, J.C.: Grasping. In: Springer handbook of robotics, pp. 955–988. Springer (2016)
41. Qian, K., Jing, X., Duan, Y., Zhou, B., Fang, F., Xia, J., Ma, X.: Grasp pose detection with affordance-based task constraint learning in single-view point clouds. Journal of Intelligent & Robotic Systems **100**, 145–163 (2020)
42. Rodriguez, A., Mason, M.T., Ferry, S.: From caging to grasping. The International Journal of Robotics Research (IJRR) **31**(7), 886–900 (2012)
43. Singh, A., Yang, L., Hartikainen, K., Finn, C., Levine, S.: End-to-end robotic reinforcement learning without reward engineering. arXiv preprint arXiv:1904.07854 (2019)
44. Song, K.T., Wu, C.H., Jiang, S.Y.: Cad-based pose estimation design for random bin picking using a rgb-d camera. Journal of Intelligent & Robotic Systems **87**(3), 455–470 (2017)
45. Swoboda, D.M.: A comprehensive characterization of the asus xtion pro depth sensor. In: I: European Conference on Educational Robotics, p. 3 (2014)
46. Ulbrich, S., Kappler, D., Asfour, T., Vahrenkamp, N., Bierbaum, A., Przybylski, M., Dillmann, R.: The opengrasp benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1761–1767. IEEE (2011)
47. Wasenmüller, O., Stricker, D.: Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In: Asian Conference on Computer Vision, pp. 34–45. Springer (2016)
48. Wohlkinger, W., Aldoma, A., Rusu, R.B., Vincze, M.: 3dnet: Large-scale object class recognition from cad models. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 5384–5391. IEEE (2012)
49. Yamanobe, N., Nagata, K.: Grasp planning for everyday objects based on primitive shape representation for parallel jaw grippers. In: 2010 IEEE International Conference on Robotics and Biomimetics, pp. 1565–1570. IEEE (2010)