Deep Learning-based 6-DoF Grasp Estimation for Industrial Bin-Picking

Adam Purnomo, Shogo Arai, *Member, IEEE,* Fuyuki Tokuda, Kazuhiro Kosuge, *Fellow, IEEE*

***Abstract*—We consider the bin-picking problem where a robot picks an object with a parallel gripper from randomly piled-up objects inside a bin until the bin is empty. The learning-based methods have been the most prominent methods to tackle this challenge. Many of the previous works focus on 4-DoF grasp where the robot can only approach the object vertically. While these methods work well for small objects, their performance suffers dramatically when dealing with wide objects as shown by our experiments. Several works utilize 6D object's pose estimation algorithms to achieve 6-DoF grasp. However, obtaining a precise 6D object's pose under cluttered scenes is computationally expensive, and many of these methods are limited to simple-shaped objects. We propose an efficient method to estimate 6-DoF grasp utilizing a single convolutional neural network (CNN). Our proposed method does not estimate the 6D object's pose. Instead, we introduce a novel way to represent the grasp approaching direction in the form of 2D images. Our network takes inputs of depth images and outputs the score for each grasp candidate and the corresponding 2D representation of the grasp approaching direction. We trained our network with a combination of synthetic and real data sets for better results. Finally, we conducted bin-picking experiments showing how our 6-DoF grasp estimation performs better than 4-DoF grasp estimation and takes less than a few seconds to compute.**

***Index Terms*—bin-picking, grasp estimation, deep learning**

I. Introduction

ONE of many laborious and repetitive tasks in the factory that are still largely done manually by humans is the grasping task. Grasping objects seems to be a trivial task for humans, and yet it is a daunting task for robots. Robotic grasping has been usually treated as a perception problem. The problem requires the robot to recognize the target object from a scene and determine how it should approach the object while avoiding collisions.

Early studies on robotic grasping exploit the geometrical shape and physical characteristics force closure [1], or grasp wrench space metric [2]. These analytical methods work under the assumption that the model and the pose of the object are perfectly known. The technological advancement of 3D measurement has made obtaining 3D point cloud data from a real scene possible. Thanks to that, there have been many 3D points registration algorithms proposed that match the obtained 3D point cloud data to the 3D model of the target object to obtain the object's pose such as Iterative Closest Point (ICP) [3], Robust Point Matching [4], kernel correlation [5], and coherent point drift [6]. However, not only these methods are prone to error under uncertainty, an actual implementation of these methods can be computationally slow during execution.

Adam Purnomo, Shogo Arai, Fuyuki Tokuda, and Kazuhiro Kosuge are with Department of Robotics, Tohoku University.

With the development of hardware technology and the affordable price of computing power, learning-based methods have found their way into many robotics applications. The learning-based methods, which rely heavily on the quantity and the quality of the data set to train a model, are usually more robust under uncertainty compared to analytical methods. Previous works suggest that learning-based methods for robotic grasping have superior performance compared to the analytical counterparts [7]–[10].

While robotic grasping for a single object can be considered a solved problem, grasping an object among many objects that are stacked at random poses inside a bin, or is often referred to as the bin-picking task, is still largely an unsolved problem. Bin-picking is still considered a challenging problem due to the complexity of the scene compounded with the inherent uncertainty in sensing.

In the last couple of years, remarkable progress has been made by several groups to tackle the bin-picking problem utilizing deep learning. Many of the previous works simplify the problem to 4-DoF grasp or vertical grasp where the robot can only approach the object from the vertical direction [11]–[14]. This is mainly because obtaining the 6D pose of many objects from a bin-picking scene becomes much more difficult and involves an additional step such as 3D point cloud instance segmentation [15], [16]. It has been shown that even with 4-DoF grasp, the automated bin-picking system can have a good performance [14], [17], [18]. However, 4-DoF grasp limits the flexibility of the robot arm to deal with various objects. For example, we have found that 4-DoF grasp does not perform well when dealing with wide objects.

Several works tried to address this problem by utilizing 6D pose estimation algorithms. While these approaches can certainly address the limitation of the 4-DoF grasp, they usually involve a very large model of neural network which of course requires an expensive computational resource. On top of that, many of these works have only been tested with simple-shaped objects which might not necessarily translate to an industrial setting where the target objects often have a complex shape.

In this paper, we explore a method to estimate 6-DoF grasp from bin-picking scenes in an industrial setting using a convolutional neural network (CNN) without having to estimate the 6D pose of the target object. We gained inspiration from the idea that the robot does not need to know exactly the precise 6D pose of the target object to get a reasonably good grasp. A rough estimate on how to approach the object is often more than sufficient to grasp the target object. We use what we call grasp approaching pose vector which determines

from which direction the robot gripper should approach the target object in Cartesian space. Our network evaluates the grasp candidates represented as grasp rectangle [19] taken from a single depth image and outputs the 2D projection of grasp approaching pose vector at once. This 2D projection can later be converted back to a 3D vector with the knowledge of the camera intrinsic matrix. We tested our method on real experiments, and we found out that our method can estimate 6D grasp sufficiently well with an average grasp success rate of 85.74%. In short, the main contribution of this work is as follows:

- A novel method to represent a 6-DoF grasp in the form of 2D images, allowing fast 6-DoF grasp estimation.
- A network architecture that can evaluate both the robustness of grasp candidates (the grasping score) and the 6D grasping pose (the grasp approaching pose vector).
- Bin-Picking experiments evaluating the performance of the proposed method using 4 types of industrial objects.

II. Related Works

The revolution of deep learning begins when LeCunn et al. [20] and Krizhevsky et al. [21] generalized the back-propagation algorithm for training multi-layer networks. Deep learning has become popular in the robotics field due to its efficacy in solving complex robotic tasks. Early efforts to utilize deep learning on grasping problem is done by training neural networks to detect which part of the object yields the most stable grasps represented as grasp rectangles [7]–[10]. The result of the experiments showed that the deep learning-based methods outperform classical grasp planning algorithms. Since gathering manually-labeled data by humans is expensive, Mahler et al. [22] showed that it is possible to train a network using synthetic depth images and use the network on real depth images.

Previously mentioned methods only examined grasping objects in isolation. Since then, many researchers have tried to tackle the grasping problem under a cluttered scene where many objects are stacked together, or often referred to as bin-picking. In general, there are 2 types of bin-picking problems; the first one is when only one type of object is stacked together (homogeneous heap), and the second one is when several types of objects are present in the bin (heterogeneous heap). Grasping objects in the bin-picking problem can also be done with either parallel-jaw grasp or suction grasp depending on the types of objects.

There have been several methods introduced to formulate the bin-picking problem including supervised learning with simulated data [13], [14], [23], [24], unsupervised learning where actual robots were trained to directly grasp objects given the object scene from camera [25], [26], Partial Markov Decision Process (PMODP) [11], and learning closed-loop visual controller [12], [27]. These methods use a parallel-jaw gripper and only consider 4-DoF grasp where the gripper approaches the target object vertically. For many bin-picking scenarios, 4-DoF parallel-jaw grasp is more than sufficient especially if the target object is small. This is because even though the target object is tilted, it won't result in a huge vertical gap between two contact points, hence the gripper could still close and grasp the object. However, 4-DoF grasp will likely be insufficient when dealing with wide objects as we found from our experiment results.

Others consider using suction gripper [28], [29], or multi-affordance grasp where both parallel-jaw gripper and suction gripper are used [17], [30]–[32]. With a suction gripper, 6-DoF grasp can be achieved without having to estimate the 6D pose of the target object. One can use the normal vector taken from the contour of the target grasp to determine the appropriate approaching direction of the gripper [17], [28]–[30]. These methods were able to estimate the 6-DoF grasp for suction grasp without having to estimate the 6D pose of the target object. However, a suction grasp works best if the target point has a smooth surface and might not be suitable for some types of objects such as objects with small surfaces and rich texture, or porous objects. In the case of multi-affordance grasp in the aforementioned works, when suction grasp is deemed unsuitable, the grasping task is still done with 4-DoF parallel-jaw grasp [17], [31], [32].

A common way to achieve 6-DoF grasp is a two-step method where at first object segmentation from the cluttered scene is done to isolate the target object, and a pose estimation algorithm is used. The pose estimation can be done by using iterative point cloud (ICP) [33]–[35], a neural network to regress to a quaternion describing the orientation of the target object [32], point pair feature matching algorithm (PPF) [16], [36], or voting-based matching scheme [37]. Once the 6D pose of the target object is obtained, the grasp candidate can be evaluated subsequently. While these methods have been proven to work, but they require additional steps to segment the objects and estimate the 6D pose of the object before evaluating the grasp candidates. Additional steps would mean additional computational time in the pipeline.

A full 6D pose information of the target object is not necessary to achieve a reasonably good 6-DoF grasp. This can be achieved by sampling 6-DoF grasp candidates from point clouds and treat it as a classification problem [38], or directly regressing to a 6-DoF grasp configuration from a point cloud [39], [40]. However, these methods were tested using household or warehouse objects which relatively have simpler shape and texture, and some of them were not tested in a bin-picking scene where the objects are stacked on top of each other. Unlike household objects, the objects found in an industrial setting usually have more complex shapes and are harder to grasp by robot gripper. We have yet to find a 6-DoF grasp estimation method that is tested in an industrial setting where the bin-picking scene is heavily cluttered.

In this work, we focus on the problem of the 6-DoF parallel-jaw grasp for bin-picking in an industrial setting. The remainder of this paper is structured as follows. In section III, the overview of the whole system and the representation of grasp candidates will be discussed. Section IV describes the architecture of the network that we used, the training scheme, and training data set generation. Section V presents the experiment results we conducted and the discussion about the results. Finally, section VI summarizes the conclusion of this work and discuss the possible future work for improvements.

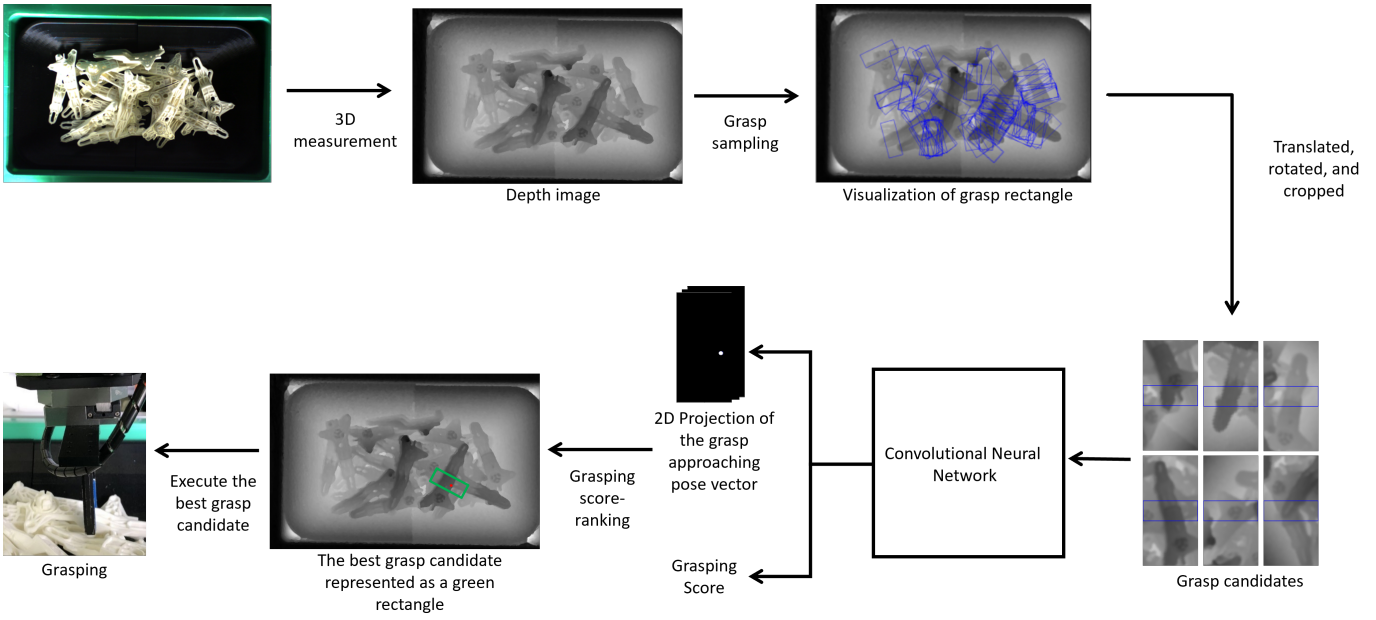


Fig. 1: Overview of the proposed method.

III. Methods

A. System Overview

In this work, we consider the case where the heap inside the bin is homogeneous, or in another word, it only contains one type of object. The overview of the proposed method is shown in Fig. 1. Depth images of the bin-picking scene are obtained with a 3D sensor. We extract edges with Canny edge detection [41] and compute normal vectors of each pixel of the edges from the depth images to sample grasp candidates. A valid grasp candidate on 2D images is defined as a pair of pixels that belong to the outer edges of the target object. The grasp sampling is done by taking many pairs of pixels that belong to the outer edges and pruning the ones outside the gripper width's limit range or the ones without sufficient condition for force closure, mathematically described by

$$w_{\min} < \|\boldsymbol{c_2} - \boldsymbol{c_1}\| < w_{\max}, \tag{1}$$

$$(\boldsymbol{c_2} - \boldsymbol{c_1}) \cdot \boldsymbol{n_2} < \cos(\arctan(\mu)), \tag{2}$$

$$(\boldsymbol{c_1} - \boldsymbol{c_2}) \cdot \boldsymbol{n_1} < \cos(\arctan(\mu)), \tag{3}$$

where, $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are the pixel coordinates of contact points, $w_{\max}$ and $w_{\min}$ are the maximum and minimum width of the gripper on 2D pixel coordinate system, $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ are the surface normal vectors of the contact points, and μ is the friction coefficient. The eq. (2) and eq. (3) not only ensures that a pair of pixels satisfy the force closure condition, but it also guarantees that a pair of pixels represent a tangible grasp candidate (antipodal grasp candidates) rather than an empty space created by two adjacent objects. The visualization of the sampled grasp candidates from the edges is shown in Fig .2

We represent each pair of pixels that qualifies to be a grasp candidate with grasp rectangle [19]. The depth image is then translated, rotated, and cropped creating individual grasp images. In each grasp image, the rectangle is at the center of the image, and its orientation is aligned with the horizontal axis, making the learning and inference process easier [22]. The images are then fed to the CNN, and the CNN will output the grasping score and the 2D projection of the grasp approaching pose vector. The grasping score, in the range of 0 to 1, represents how robust a particular grasp candidate is. The 2D projection of the grasp approaching pose vector is later converted to a 3D vector and used to determine the approaching direction of the gripper. The grasp candidate with the highest grasping score is chosen, and a grasping action can be executed.

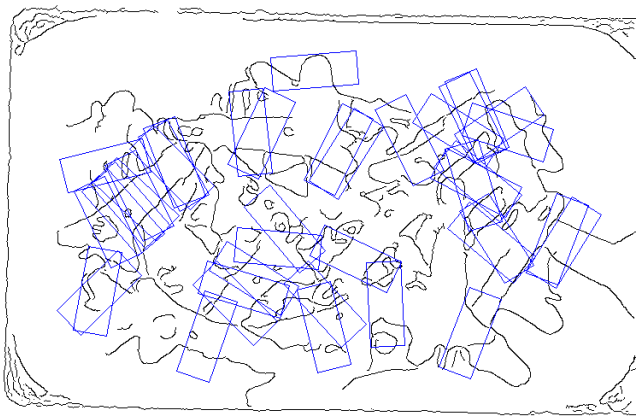


Fig. 2: A visualization of sampled grasp candidates. The black lines represent the extracted edges from the scene and the blue lines represent sampled grasp candidates. The grasp candidates are sampled by taking many pairs of pixels that belong to the outer edges of target objects and pruning the ones outside the gripper width's limit range or the ones without sufficient condition for force closure.

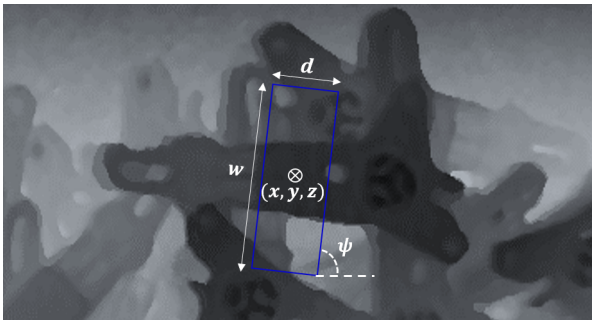


Fig. 3: A Grasp rectangle as a representation of a grasp candidate.

B. Grasp Representation

We use the grasp rectangle, which was first introduced by Jiang et al. [19], to represent a grasp candidate. A Grasp rectangle on a depth image encodes 6 variables about the grasp candidate. Those are the 3D position of the center of the grasp, x, y, and z, the rotation around the vertical axis ψ, the gripper width on 2D image w, and the finger's dimension d, as shown in Fig. 3. Since we only consider a homogeneous heap inside the bin, the variables w and d are set to be constant. With the grasp rectangle, we have information about 4 out of a total 6-DoF of the grasp candidate in the Cartesian space which allows the robot to grasp the target object only from the vertical direction.

We use what we call grasp approaching pose vector $\boldsymbol{v}$ that determines the approaching direction of the gripper to the center of the grasp in the target object. This vector starts from the center of the grasp and points to the opposite direction of the grasp approaching direction as shown in Fig. 4. This vector should always be perpendicular to the grasping stroke direction, and the magnitude of the vector is set to be constant. Provided the knowledge of the object's pose in the bin, we can determine the grasp approaching pose vector in three following steps. First, we sample many vectors along the perpendicular plane to the grasp stroking direction. We then prune the vectors correspond to the grasping action that results in a collision with the surrounding object. Finally, we take the one that minimizes the dot product with the gravitational force since the vector is in the opposite direction of the grasp approaching direction.

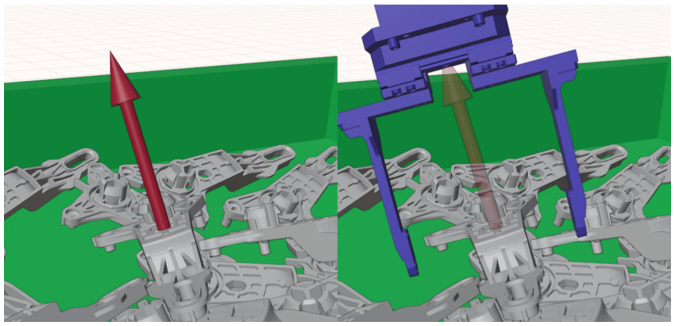


Fig. 4: An illustration of a grasp approaching pose vector that starts from the center of the grasp and points to the opposite direction of the approaching direction in Cartesian space.

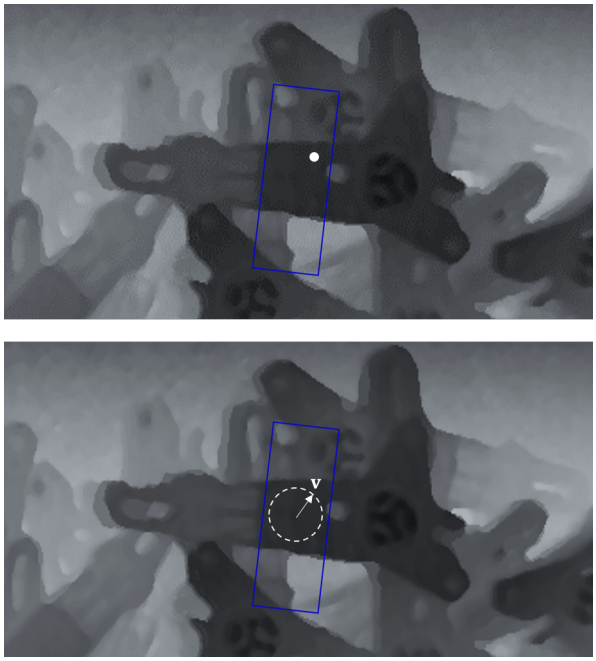


Fig. 5: (Top) Predicted pixel location representing the 2D projection of a grasp approaching pose vector by CNN. (Bottom) Visualized 2D projection of a grasp approaching pose vector.

However, since we do not know the object's 6D pose in the bin, we train the CNN to predict the 2D projection of the grasp approaching pose vector represented as a dot as shown in Fig. 5. The location of the dot relative to the center of the rectangle is the 2D projection of the grasp approaching pose vector relative to the center of the grasp. With the knowledge of the camera intrinsic matrix, the 2D vector can be converted to 3D vector. Once the 3D vector is obtained, we can obtain the Euler rotation angles by treating the vector as if it has been rotated from an initial vector that is pointing to the vertical axis. We extract the Euler angles from the equations given by,

$$\boldsymbol{v}_{\text{init}} = \begin{bmatrix} 0 \\ 0 \\ \|\mathbf{v}\| \end{bmatrix}, \tag{4}$$

$$\boldsymbol{v} = Z(\psi)Y(\theta)X(\phi)\boldsymbol{v}_{\text{init}}, \tag{5}$$

$$Z(\psi)^{-1}\boldsymbol{v} = Y(\theta)X(\phi)\boldsymbol{v}_{\text{init}}, \tag{6}$$

$$\boldsymbol{v'} = Y(\theta)X(\phi)\boldsymbol{v}_{\text{init}}, \tag{7}$$

where $\boldsymbol{v}$ is the grasp approaching pose vector, $Z(\psi)$, $Y(\theta)$, and $X(\phi)$ are the rotation matrices, ψ, θ, and ϕ are the Euler angles around z, y, and x-axis respectively. The value of ψ can be obtained from the orientation of the grasp rectangle. We can analyze eq. 6 element by element, and the value of θ and ϕ are obtained by,

$$\phi = \frac{1}{\|\boldsymbol{v}\|} \arcsin\left(-v'_2\right), \tag{8}$$

$$\theta = \frac{1}{\|\boldsymbol{v}\|} \arcsin \frac{v'_1}{\cos\phi}, \tag{9}$$

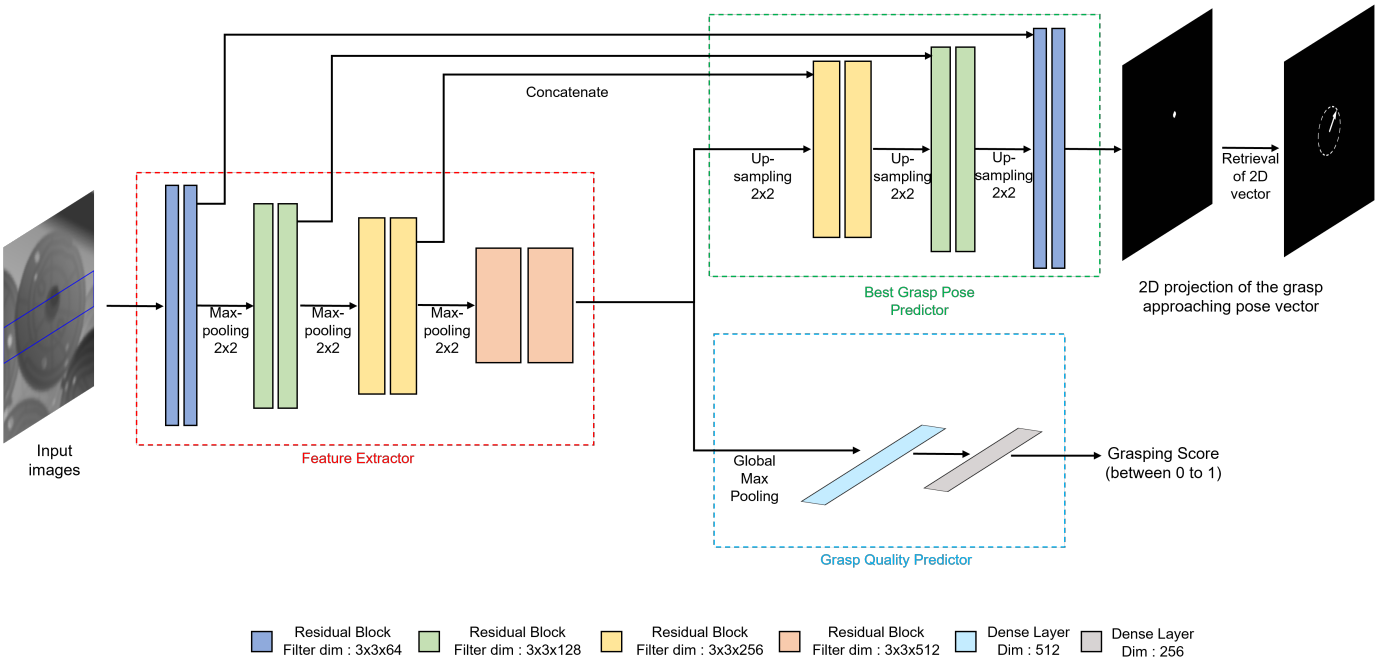


Fig. 6: Architecture of the proposed network.

where v_1' and v_2' are the first and the second entry element of the vector $\boldsymbol{v'}$.

IV. Learning 6-DoF Grasp Estimation

A. Network Architecture

The proposed network architecture is shown in Fig. 6. As mentioned before, the network takes inputs of individual grasp image and outputs the grasping score and the 2D projection of the grasp approaching pose vector. The network is divided into three parts: feature extractor, grasp pose estimator and grasp quality estimator. The feature extractor consists of 4 stages of convolution processes where each stage is comprised of 2 units of residual blocks [42]. The grasp pose estimator consists of 3 stages of deconvolution processes, and each stage is also comprised of 2 units of residual blocks.

Inspired from the architecture of U-Net [43], the output of the first three stages of convolution processes in the feature extractor part are forwarded and concatenated to the input of corresponding deconvolution processes with the same size as shown in Fig. 6. The last part is the grasp quality estimator which consists of 2 dense layers with 512 and 256 nodes each. Instead of using max-pooling with fixed sliding window size, we use global max-pooling for the input of the first dense layer in the grasp quality estimator to make sure that the network can take an arbitrary size of input images.

We treat the output layer of the grasp pose estimator as a pixel-wise binary classification problem with the same size as the input images. The bright part, shown as a white dot in the picture, is the estimated location of the 2D projection of the grasp approaching pose vector relative to the center of the grasp candidate. On the other hand, the output layer of the grasp quality estimator is treated as a normal classification problem which outputs a number between 0 to 1 where the higher the number is, the better the grasp quality is. The activation function of all hidden layers is a ReLu unit function [44], while the activation function of the last output layers of the grasp pose estimator and the grasp quality estimates is a sigmoid function.

B. Training Process

We divide the training process into two sessions as shown in Fig. 7. In the first session, we train the feature extractor and the grasp quality estimator. In this process, the network is trained to classify the individual grasp images into feasible grasp (positive class) and unfeasible grasp (negative class). Since there is a class imbalance between the feasible and unfeasible grasp in the training data set, we use a focal loss function which was introduced by Lin et al. [45]. Focal loss introduces two hyper-parameters, α, and γ. We chose $\gamma = 2$ as what is mentioned in the paper, and $\alpha = 0.75$. We trained the network with 30 epochs, batch size equal to 2, SGD optimizer, and the learning rate is equal to 0.001. The first training session took 18 hours in total with NVIDIA GPU GTX 1080.

In the second session, we reused the weight of the feature extractor part, froze the weight, and combine it with the grasp pose estimator. In other words, we only trained the grasp pose estimator part in this session. As mentioned previously, we treat this problem as a pixel-wise binary classification. Therefore, the network is trained to predict which pixels belong to the positive class. In the labeled training data set, out of all pixels in the output layers, only one pixel belongs to the positive class, indicating the location of the 2D projection of the grasp approaching pose vector. Hence, there is a huge class imbalance in this problem. To tackle this problem, aside from using the focal loss function, we set the bias of the last layer so that the initial output of the networks are all zero-

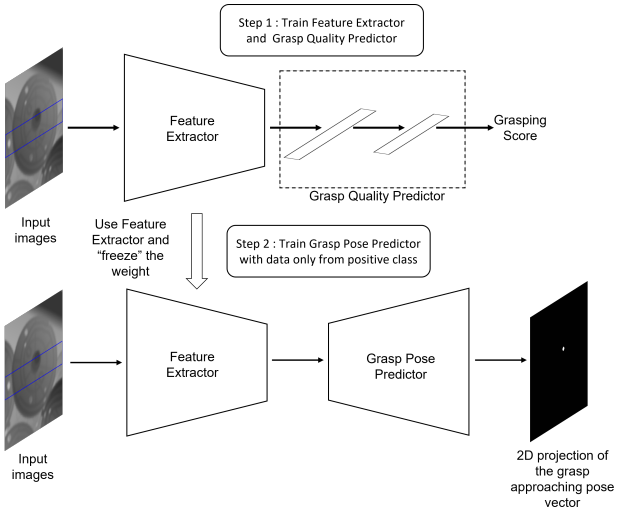


Fig. 7: Two steps of the training process.

valued pixels. We set the value of $\gamma = 2$, and $\alpha = 0.75$. We decided to scale up the loss function by a factor of 1000 since the original loss function is too small and the training process did not go well. We trained the network with 40 epochs, batch size equal to 1, SGD optimizer, and the learning rate is equal to 0.001. The Second training session took 23 hours in total with Nvidia GPU GTX 1080.

C. Training Data Set Collection

We created two separate data sets for each training session. For the first training session, we gather a data set that consists of 24,000 manually annotated individual grasp images from 60 real bin-picking scenes. We complement it with 46,000 synthetic individual grasp images from computer simulation. In labeling the synthetic individual grasp images, we take into account several factors that determine the feasibility of a grasp candidate, including the occlusion rate from the camera perspective, collision check between the surrounding objects, and grasp wrench space metrics [2].

For the second training session, we created 30,000 synthetic individual grasp images from computer simulation and its corresponding 2D projection of the grasp approaching pose vector. The second data set only contains synthetic grasp images of feasible grasp candidates. For one, the grasp approaching pose vector is only meaningful if the grasp candidate itself is feasible to execute. Secondly, the creation of the 2D projection of the grasp approaching pose vector requires information about the object's 6D pose in the bin, which will be explained shortly. Since obtaining the object's 6D pose from an actual scene is rather challenging, we decided to only use synthetic data instead.

To create the 2D projection of the grasp approaching pose vector, we first sample antipodal grasp candidate on a 3D object data in its local coordinate, sample many vectors on the plane perpendicular to the grasp stroking direction for each grasp candidate, and prune the one that collides with the object itself in the object's local coordinate as what is done by Wan et al. [46]. When we perform a simulation of stacking objects inside a bin, we transform all the sampled vectors for each grasp candidate from the local coordinate to the global coordinate depending on the object's pose inside the bin and prune again all the grasp approaching pose vector that results in a collision with the surrounding object. Out of all the remaining vectors for each grasp candidate, we choose the one that minimizes the dot product with the gravitational force, and we project it into the pixel coordinate from the camera perspective. We then put label 1 to the location of the projection of the vector and 0 for the rest of the pixels.

V. Results and Discussion

A. Experiments

In this section, we evaluate the proposed method by conducting real bin-picking experiments. The experimental setup of the experiment is shown in Fig.8. We used a 6-DoF Denso robotic arm as the manipulator, a pneumatic parallel gripper, and an Ensenso N-30 depth camera with 1280 x 1024 resolution for the 3D sensor. As mentioned previously, we only focus on bin-picking with a homogeneous heap where only one type of object is present inside the bin. So, we evaluated our method with 4 types of objects as shown in Fig.9 separately. We stacked around 20-25 objects inside a box depending on the object's size. For each object, we performed both 4-DoF and 6-DoF bin-picking experiments to create a baseline comparison of how well our proposed method performs. The 4-DoF bin-picking experiments were based on the method proposed by [18]. For each experiment, we performed around 79 to 106 grasp attempts, and we classify each grasp attempt into successful attempts and failed attempts. A successful grasp attempt is an attempt when the

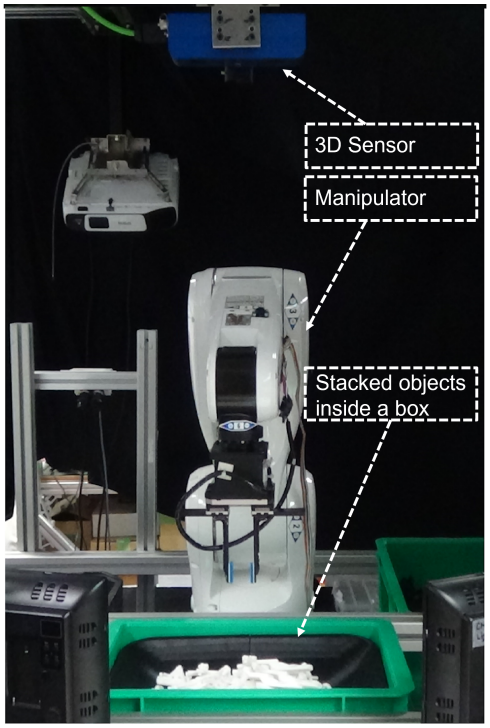
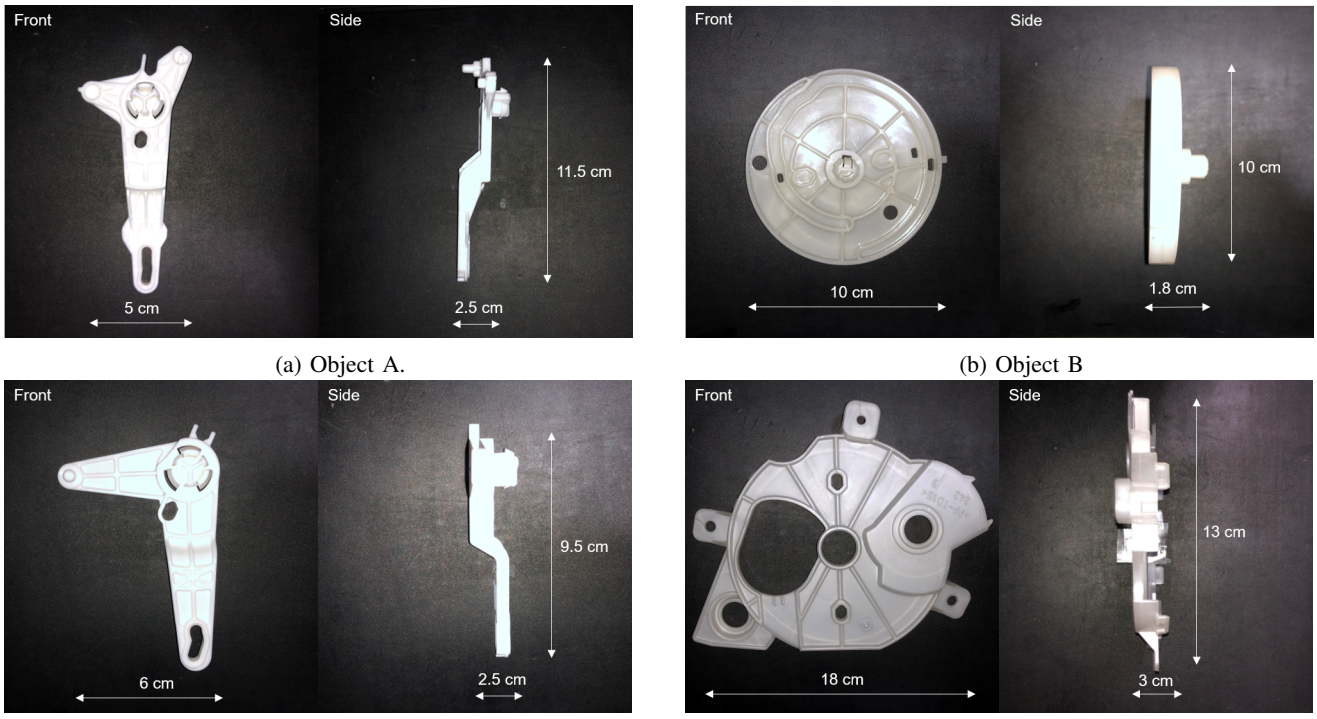


Fig. 8: Experimental setup for bin-picking experiments.



(a) Object A. (b) Object B

(c) Object C (d) Object D

Fig. 9: Objects used to conduct bin-picking experiments.

robot gripper can successfully grasp the object with a grasping pose close to the ideal grasping pose and remove it from the box. Otherwise, we considered it as a failed attempt. The success rate of each experiment is simply the ratio between successful attempts and the total attempts that the robot made.

The computational time for each grasp attempt largely depends on the number of grasp candidates sampled from the scene. We took 50 samples each time to ensure that there is at least one feasible grasp candidate. It took on average around 5 seconds to compute the inference with NVIDIA GPU GTX 1080. The bottleneck computation during the inference process is the memory of the GPU which did not allow us to use batch size more than 1.We later also tested our model with NVIDIA Tesla K-80 and the computational time can be reduced to 2.6 seconds on average by doubling the batch size.

There are several cases where the robot could not find any feasible grasp candidates from the scene. Usually, this happens because the objects are closely located to one and another leading to a collision if a grasp attempt is performed. This problem can be mitigated by simply shaking the stack to change the configuration of the objects. During our experiment, this was done manually. However, to fully automate this process, an additional system can be installed to shake the box whenever no feasible grasp candidate is found. The result of the experiments is summarised in table I and table II, and several examples of estimated grasps with its corresponding grasp attempt is shown in Fig.10 and Fig.11.

B. Discussion

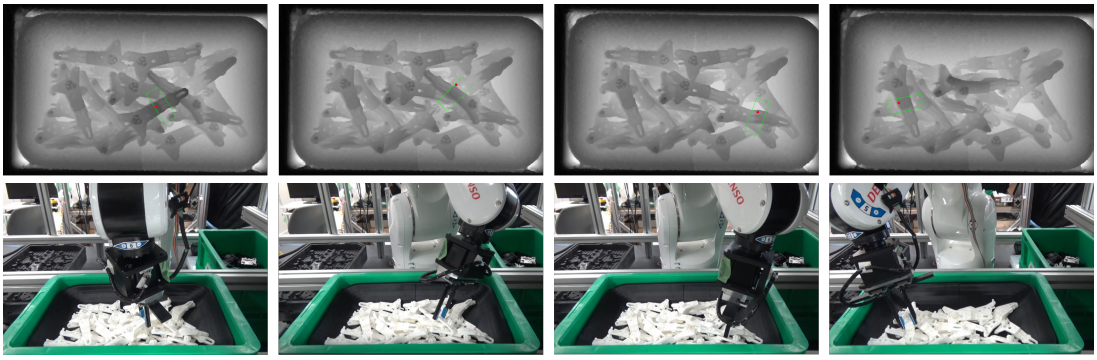
From our experiments summarized in table I and table II, we found that our proposed method performed reasonably well in all experiments with a grasp success rate ranging from 84.21% to 89.62%. As a comparison, without the 6-DoF grasp estimation, the grasp performance yields subpar grasp success rates especially for object B and object D. The considerable performance degradation of 4-DoF grasp estimation on object B and object D can be explained intuitively. Both objects have a wide shape which means a slight tilt of the orientation of the target object results in a bigger vertical gap between both grasp contact points as illustrated in Fig.12. When the robot arm tries to reach the grasp candidate vertically, it tends to fail to grasp the target object. This problem does not become a huge concern with small object types since a slight tilt of the orientation does not result in a huge vertical gap.

TABLE I: Experiment results of bin-picking with 4-DoF grasp estimation.

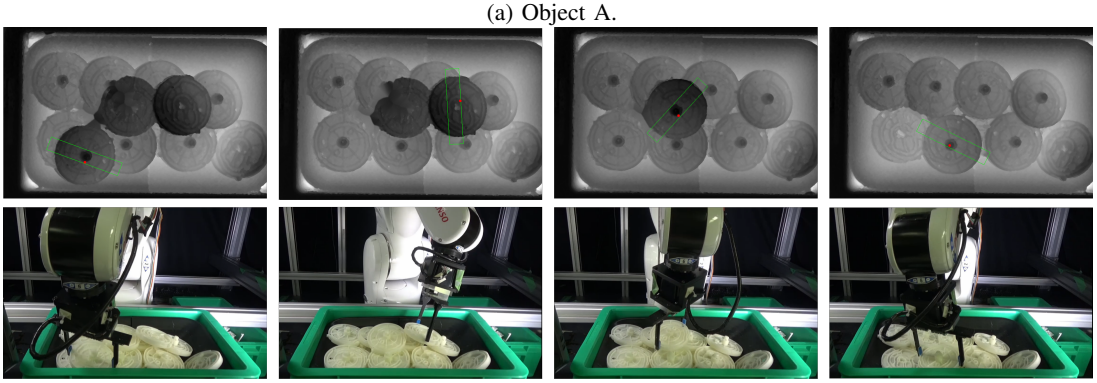
4-DoF Grasp Estimation	Successful Attempts	Failed Attempts	Total Attempts	Success Rate
Object A	60	23	83	72.29%
Object B	43	37	80	53.75%
Object C	79	19	98	80.61%
Object D	49	39	88	55.68%

TABLE II: Experiment results of bin-picking with 6-DoF grasp estimation.

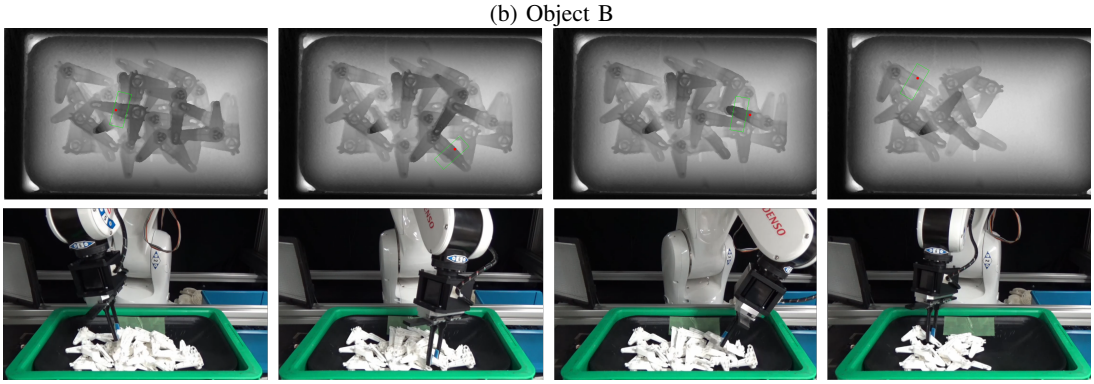
6-DoF Grasp Estimation	Successful Attempts	Failed Attempts	Total Attempts	Success Rate
Object A	66	10	76	86.84%
Object B	95	11	106	89.62%
Object C	80	15	95	84.21%
Object D	66	10	76	86.84%



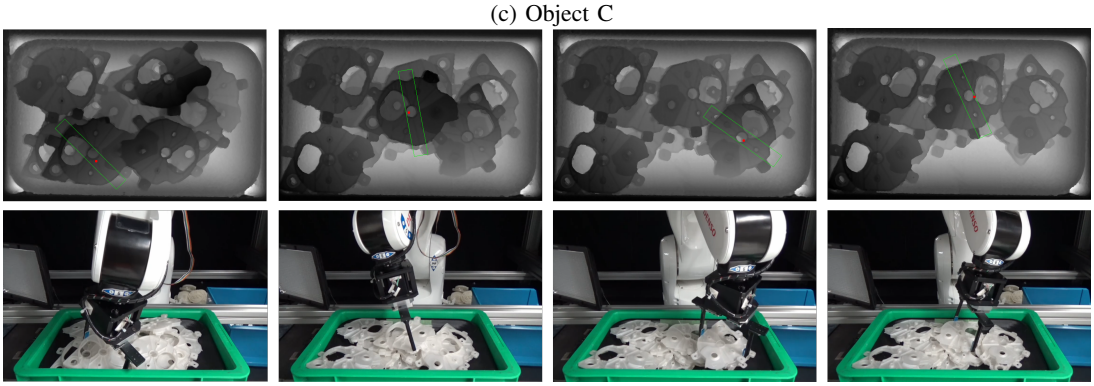
(a) Object A.



(b) Object B

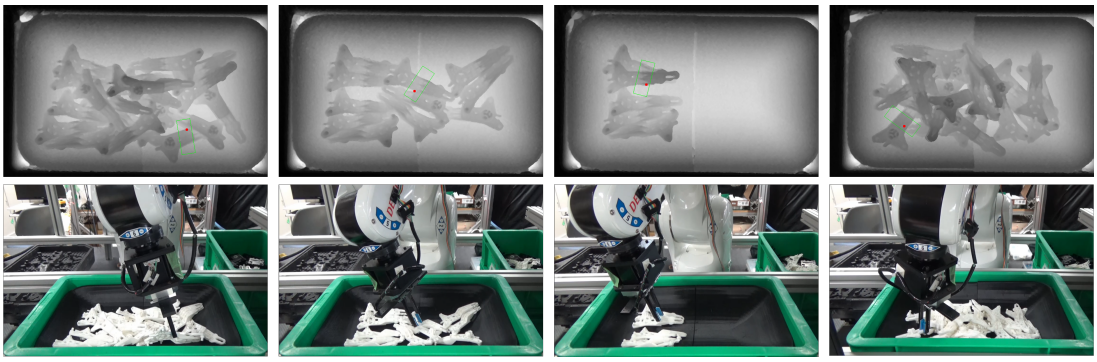


(c) Object C

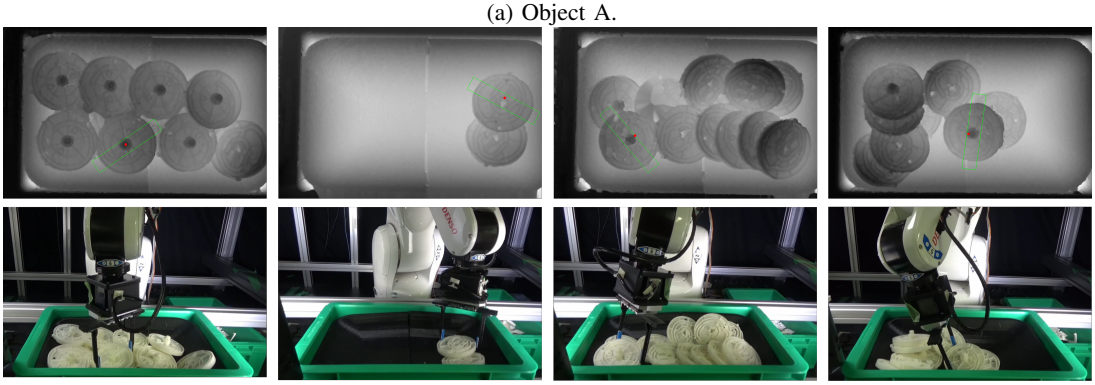


(d) Object D

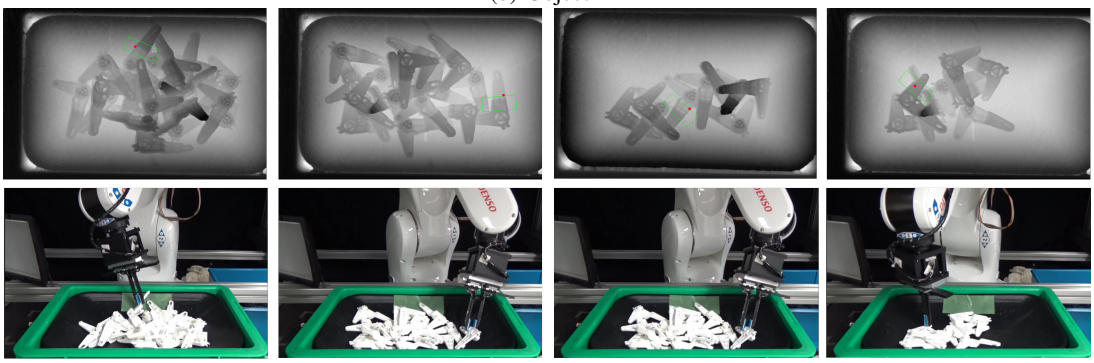
Fig. 10: Examples of success attempts. The estimated best grasp candidates with its 2D projection of approaching pose vector on depth images (Top row). The corresponding success grasp attempts (Bottom row).



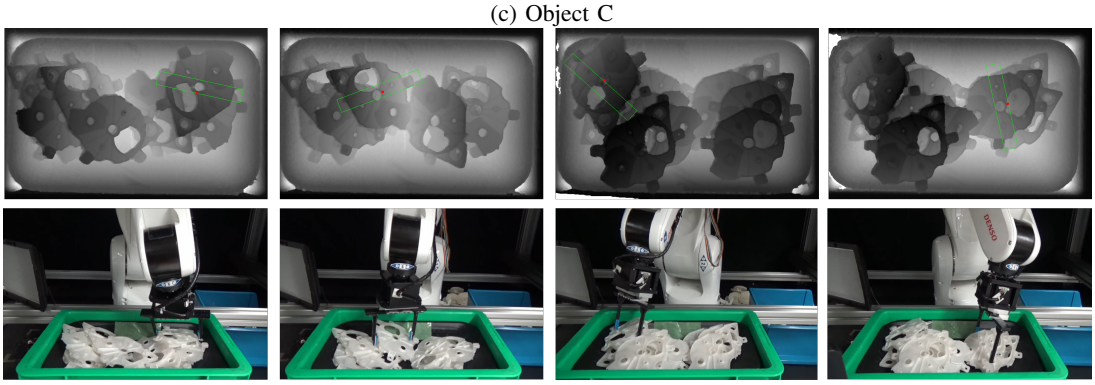
(a) Object A.



(b) Object B



(c) Object C



(d) Object D

Fig. 11: Examples of failed attempts. The estimated best grasp candidates with its 2D projection of approaching pose vector on depth images (Top row). The corresponding failed grasp attempt (Bottom row).

TABLE III: Comparison with other 6-DoF bin-picking methods.

	[28]	[29]	[31]	[39]	[38]	**Ours**
Type of Training Data	Synthetic	Synthetic	Real	Real	Real	Synthetic + Real
Input Data	Depth Image	Depth Image	RGB-D Image	Point Cloud	Point Cloud	Depth Image
Type of Target Objects	Adversarial	Textureles/Planar	Household	Household	Household	Industrial
Gripper Type	Suction	Suction	Suction	Parallel	Parallel	Parallel
Average Inference Time (s)	-	0.034	0.06	0.0126	-	2.6
Average Success Rate	81%	97.5%	94.55%	77.10%	77.76%	86.88%

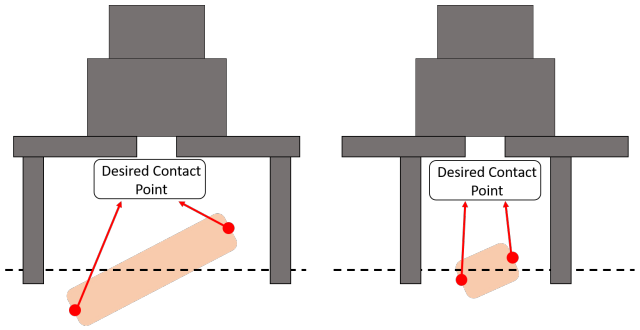


Fig. 12: An illustration of how 4-DoF grasp on wide object will likely fail in a picking scenario.

There is a discrepancy between results shown in I and our previous results in [18]. The discrepancy can be explained by the way how the experiments were set. In this paper, we placed our objects inside a bin with a convex base, while in [18], experiments were conducted on a planar surface. Placing objects in a box on a convex base made them concentrated in the middle with higher stacks which leads to more variation on orientation. In contrast, placing objects on a planar surface tends to make many objects at a stable pose, thus making them easier to grasp.

Failed grasp attempts in our method can be attributed to several factors such as inaccurate grasp pose estimation or grasp scoring and collision with surrounding objects. Inaccurate grasp pose estimation makes the most of failed grasp attempts which can be seen from figure 11. Sometimes, the location of the red dot which represents the 2D projection of the grasp approaching vector drifted too far from the center of the grasp rectangles, resulting in excessively tilted grasp approaching pose vector. This phenomenon is even more pronounce in wide objects because of the reason mentioned in the above paragraph. The inaccurate prediction of grasp pose estimator is mainly because the grasp pose estimator was only trained using synthetic depth images which do not have the same quality as real depth images. Utilizing real depth images to train the grasp pose estimator might not be a practical way since we need to know the pose of the target object to create the label for the training data. A possible extension of our work is to incorporate a Generative Adversarial Network (GAN) which is trained to augment the synthetic depth images to look closer to real depth images. In this way, we can gather the training data cheaply while improving the performance of the grasp pose estimator on a real bin-picking scene.

Our method is not aimed to compete with 6D pose estimation algorithms since the goal of our method is not to estimate the 6D pose of the object, but to estimate how to approach the target object with 6-DoF grasp. We aimed to show that 6-DoF grasp can be achieved even without having to estimate the 6D pose of the object. With 6D pose estimation algorithms, the estimated grasp might be more precise, but grasp planning requires more computational complexity at the same time. Our method trades precision with simplicity, and for most of the time, the estimated 6-DoF grasp is more than sufficient enough to grasp the target object as shown from our experiment results. We instead try to compare our methods with other bin-picking methods that employ 6-DoF grasp in their experiments. The comparison is summarized in table III.

From table III, it can be immediately noticed that methods that utilized suction grippers have high average grasp success rate than methods that utilized parallel grippers. As mentioned in Section 2, suctions gripper works best if the target objects have a large and smooth surface that makes suction grasp possible. The method proposed by [29] is only tested with textureless or planar objects, while the method proposed by [33] is tested with household objects that have smooth and approximately planar surface provided by Amazon Robotics Challenge. In [28], their method is tested with several kinds of objects. Their method achieves a grasp success rate of 97% when tested against prismatic objects. However, when their method is tested against adversarial objects, objects with few available suction-grasp points, the grasp success rate drops to 81% as shown in table III. We believe that the objects we used during our experiments would qualify as adversarial objects considering how uneven the surface of the objects is. Compared to other methods that utilized parallel gripper [38], [39], our method has better grasp success rate, albeit with a longer inference time. However, we believe that we can further reduce the inference time of our method by increasing the batch size during the inference process by using GPUs with higher memory capacity.

VI. Conclusion

This paper proposed a new deep learning-based method to tackle 6-DoF grasp estimation without having to estimate the 6D pose of the target object. Our method utilized a novel way to represent 6-DoF grasps utilizing what we call grasp approaching pose vector which determine the approaching direction of the gripper to the target object. A convolutional neural network is trained to evaluate the robustness of grasp

candidates and estimate the 2D projection of the grasp approaching pose vector. With the knowledge of the camera intrinsic matrix, the 2D projection can be converted to 3D vector which encodes information about 6-DoF grasps. Our experiment results show that the proposed method performed reasonably well with grasp success rate ranging from 84.21% to 89.62%. Most failed grasp attempts can be attributed to inaccurate grasping pose estimation due to the fact that the grasp pose estimator part is only trained with synthetic depth images. A possible future work to improve the performance of our method is to incorporate a Generative Adversarial Network (GAN) trained to augment synthetic depth images to look closer to real depth images.

REFERENCES

[1] V.-D. Nguyen, "Constructing stable force-closure grasps," in *Proceedings of 1986 ACM Fall joint computer conference*, 1986, pp. 129–137.

[2] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 3, 1992.

[3] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, 1992.

[4] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2d and 3d point matching: pose estimation and correspondence," *Pattern Recognition*, vol. 31, no. 8, pp. 1019–1031, 1998.

[5] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 558–569.

[6] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

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

[8] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1316–1322.

[9] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 769–776.

[10] J. Watson, J. Hughes, and F. Iida, "Real-world, real-time robotic grasping with convolutional neural networks," in *Towards Autonomous Robotic Systems*, Y. Gao, S. Fallah, Y. Jin, and C. Lekakou, Eds. Cham: Springer International Publishing, 2017, pp. 617–626.

[11] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 515–524.

[12] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 291–300.

[13] R. Matsumura, Y. Domae, W. Wan, and K. Harada, "Learning Based Robotic Bin-picking for Potentially Tangled Objects," in *IEEE International Conference on Intelligent Robots and Systems*, 2019.

[14] Y. Domae, A. Noda, T. Nagatani, and W. Wan, "Robotic General Parts Feeder: Bin-picking, Regrasping, and Kitting," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., may 2020, pp. 5004–5010.

[15] Y. Xu, S. Arai, D. Liu, F. Lin, and K. Kosuge, "FPCC: Fast Point Cloud Clustering for Instance Segmentation," *arXiv preprint arXiv:2012.14618*, 2021.

[16] C. Zhuang, Z. Wang, H. Zhao, and H. Ding, "Semantic part segmentation method based 3D object pose estimation with RGB-D images for bin-picking," *Robotics and Computer-Integrated Manufacturing*, vol. 68, apr 2021.

[17] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, 2019.

[18] S. Arai, Z. Feng, F. Tokuda, A. Purnomo, and K. Kosuge, "Deep Learning-based Fast Grasp Planning for Robotic Bin-picking by Small Data Set without GPU," *TechRxiv preprint*, 2021.

[19] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, 1998.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, 2017.

[22] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-Net 2.0: Deep learning to plan Robust grasps with synthetic point clouds and analytic grasp metrics," in *Robotics: Science and Systems*, vol. 13, 2017.

[23] R. Matsumura, K. Harada, Y. Domae, and W. Wan, "Learning based industrial bin-picking trained with approximate physics simulator," in *Intelligent Autonomous Systems 15*, M. Strand, R. Dillmann, E. Menegatti, and S. Ghidoni, Eds. Cham: Springer International Publishing, 2019, pp. 786–798.

[24] H. Tachikake and W. Watanabe, "A learning-based robotic bin-picking with flexibly customizable grasping conditions," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9040–9047.

[25] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, 2016.

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

[27] D. Morrison, P. Corke, and J. Leitner, "Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[28] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning," *arXiv preprint arXiv:1709.06670*, 2017.

[29] P. Jiang, Y. Ishihara, N. Sugiyama, J. Oaki, S. Tokura, A. Sugahara, and A. Ogawa, "Depth image–based deep learning of grasp planning for textureless planar-faced objects in vision-guided robotic bin-picking," *Sensors*, vol. 20, no. 3, 2020.

[30] A. Iriondo, E. Lazkano, and A. Ansuategi, "Affordance-based grasping point detection using graph convolutional networks for industrial bin-picking applications," *Sensors (Switzerland)*, vol. 21, no. 3, 2021.

[31] A. Zeng, S. Song, K. T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. Chavan Dafle, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *International Journal of Robotics Research*, 2019.

[32] M. Schwarz, C. Lenz, G. M. Garcia, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke, "Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2018.

[33] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1386–1383.

[34] A. T. Iglesias, I. Pastor-López, B. S. Urquijo, and P. García-Bringas, "Effective bin picking approach by combining deep learning and point cloud processing techniques," in *Hybrid Artificial Intelligent Systems*, E. A. de la Cal, J. R. Villar Flecha, H. Quintián, and E. Corchado, Eds. Cham: Springer International Publishing, 2020, pp. 534–545.

[35] S. Lee and Y. Lee, "Real-time industrial bin-picking with a hybrid deep learning-engineering approach," in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2020, pp. 584–588.

[36] D. Liu, S. Arai, J. Miao, J. Kinugawa, Z. Wang, and K. Kosuge, "Point pair feature-based pose estimation with multiple edge appearance models (ppf-meam) for robotic bin picking," *Sensors*, vol. 18, no. 8, 2018.

[37] W. Yan, Z. Xu, X. Zhou, Q. Su, S. Li, and H. Wu, "Fast Object Pose Estimation Using Adaptive Threshold for Bin-Picking," *IEEE Access*, vol. 8, pp. 63 055–63 064, 2020.
[38] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3629–3635.
[39] Y. Qin, R. Chen, H. Zhu, M. Song, J. Xu, and H. Su, "S4g: Amodal single-view single-shot se(3) grasp detection in cluttered scenes," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 53–65.
[40] M. Sundermeyer, A. Mousavian, R. Triebel, and F. Dieter, "Contact-graspnet: Efficient 6-dof grasp generation in clutteredscenes," *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
[41] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, 2016.
[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
[44] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.
[45] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, 2020.
[46] W. Wan, K. Harada, and F. Kanehiro, "Planning Grasps for Assembly Tasks," *arXiv preprint arXiv:1903.01631*, 2019.