2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

MultipleCupSuctionNet: Deep Neural Network for Detecting Grasp Pose of a Vacuum Gripper with Multiple Suction Cups based on YOLO Feature Map Affine Transformation

Ping Jiang[1], Komoda Kazuma[1], Haifeng Han[1] and Ooga Junichiro[1]

***Abstract*— Multiple-suction-cup grasp is preferable for picking large and heavy objects in warehouses. Deep learning methods have been widely used to predict grasp point position for single-suction-cup grasp, but few studies have examined grasp pose detection for a gripper with multiple suction cups. This study proposes MultipleCupSuctionNet, which is a deep neural network for detecting multiple cup grasp pose. To address the challenge of direct regression of poses, this neural network first infers the surface mask to compute the surface normal to obtain the direction of the gripper z-axis. The feature maps of the surfaces are then affine-transformed to surface image coordinates, based on which gripper position and rotation angle around the z-axis are predicted. Such a neural network design makes grasp pose learning easier because there is no need to consider the orientation of the surfaces so that 2D poses respective to the surface are learned. Feature map affine transformation saves computation cost because there is no need to first transform images and then extract the features to obtain surface features. Further, for each predicted grasp pose, the overlap area between cup and surface is calculated to determine which cup should be used when grasping. MultipleCupSuctionNet exhibited competitive performance (80.1% prediction accuracy) particularly in dense scenes compared with state-of-the-art planners (Dex-Net and a model-free multiple-suction-cup grasp planner). Physical picking experiments were conducted using a robot employing the proposed neural network. The experimental results showed that our robot achieved an average success rate of 94.5% for picking common objects in warehouses.**

I. INTRODUCTION

In Japan, the aging society is causing a labor shortage problem. The growth of e-commerce has intensified the problem in the field of logistics because there is high demand for labor to pick and place various items in warehouses. Automation of bin picking by robots is one method for addressing the labor shortage problem in warehouses. As the sizes of items in a warehouse may vary from small to big, the use of multiple suction cups (MSCs) is more effective than single suction cups (SSCs) because MSCs can achieve various suction contact areas to stably grasp items. An MSC gripper can use one cup to grasp small items or use more cups to increase the contact area for stably grasping items with large surface. However, determining the MSP grasp pose for the appropriate number of cups to grasp randomly posed items with various shapes and sizes is difficult.

*This work was not supported by any organization
[1]All author are with Corporate Manufacturing Engineering Center, Toshiba Corporation, 33, Shin-Isogo-Cho, Isogo-ku, Yokohama 235-0017, Japan ping2.jiang@toshiba.co.jp

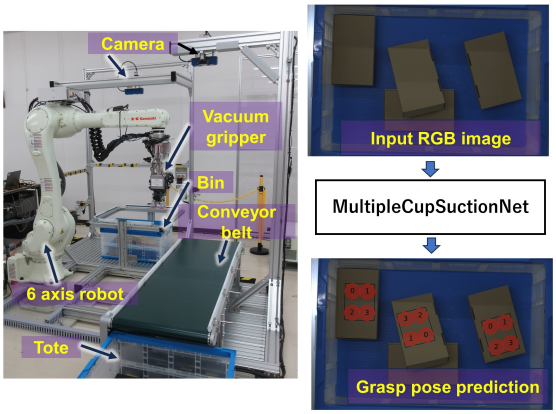

Fig. 1. MSC grasp

Deep-learning-based methods have exhibited good performance for detecting the grasp pose for SSC vacuum grippers. In previous studies, deep neural networks have been developed to directly regress the suction grasp point position or to infer the grasp quality map to find the optimal grasp point position [1], [2], [3], [4]. For SCP grasp, it is sufficient to learn only the position since the gripper orientation can be determined based on the normal direction of the grasp point. However, learning MSC grasp poses is more challenging because the orientation of the gripper also needs to be learned to make suction cups best fit the object surfaces. In addition, suction on/off cup activation control is required to avoid grasping non-target objects. One of the tasks in a warehouse is to pick an ordered number of objects. In a dense scene, if the cups are not well controlled, the gripper grasps multiple objects (a certain object and its neighboring objects [non-target objects hereafter in this paper]), leading to picking of more than the desired number of objects.

Few studies have focused on MSC grasp pose detection. To the best of our knowledge, only one recent study [5] has proposed a two-step method to detect MSC grasp pose. They used a U-net [6] to first infer the grasp quality and then performed the convolution between the grasp quality map and sampled gripper footprints to find the grasp poses and cup activation patterns. However, the computation cost greatly depended on the sampling size and scale factor of footprints, and their method may mistakenly grasp multiple objects in dense scenes.

The present study proposes MultipleCupSuctionNet for inferring the grasp pose of a vacuum gripper with MSCs as in Fig. 1. The network infers not only grasp position, but

also the gripper orientation. Furthermore, to make learning easier, we learn the grasp pose for each surface rather than all the surfaces in an image. Specifically, we extract feature maps of the input image and perform affine transformation on them to obtain surface feature maps. The grasp pose respective to each surface in surface image coordinates can then be inferred by using the affine transformed feature map. In addition, the cup activation pattern is directly determined by calculating the intersection area between suction cup and surface mask for each grasp pose without convolution of footprints.

Our contributions include the following: (1) Proposal of a novel deep neural network for detecting grasp pose and activation pattern of a vacuum gripper with MSCs. (2) Proposal of an affine-transform-based method for obtaining surface feature maps feasible for grasp pose detect. (3) Validation of proposed method by physical bin picking experiments.

II. RELATED WORK

A. SSC grasp pose detect

Data-driven methods have demonstrated effective performance for robotic picking tasks. Methods using deep neural network models have been developed to detect the grasp pose of a fingered robotic hand [7], [8], [9] and SSC vacuum gripper [10], [11]. For SSC grasp pose detection, the deep neural network predicts the position of the suction cup center. The grasp pose is then derived from the normal vector at the center. Instead of only predicting the point, other methods predict the affordance map which contains the pixel-wise grasp quality. The pixel with optimal grasp quality and its normal are output to obtain the optimal grasp pose. One representative study is Zeng et al. [1], in which fully convolutional neural networks (FCNs) were used to predict the pixel-wise suction for a two-finger and SSC vacuum gripper. They used human hand-crafted annotation data (affordance map) to train FCNs and demonstrated good performance on bin picking by four motion primitives. Later, Zeng's work was enhanced by [12], [13] to improve the generality and prediction accuracy. Another representative work is from Mahler et al. [3], where they proposed Grasp Quality Convolutional Neural Networks (GQ-CNNs) to predict the grasp quality. They defined a contact model between the cup and object to generate the grasp quality annotation data in a physical simulator. Recently, larger datasets of SSC grasp pose and affordance have been provided by [2], [14]. However, these studies have detected only the positions of suction cup center points for grasping objects. They cannot be directly applied to MSC grasp because the gripper orientation and cup activation needed to be determined.

B. MSC grasp pose detection

Most previous studies using MSCs to grasp a specific object with prior knowledge (e.g. object dimensions known in advance). Mantriota [15] used a four-cup vacuum gripper to grasp a large planar object based on the calculated necessary suction force. Kozák [16] et al. used a six-cup vacuum gripper to grasp a round part. The part shape was known and the grasp pose was derived from the estimated pose of the part from the neural network. Tanaka et al. [17] designed an 'L' shaped gripper which was equipped with MSCs on each side. The gripper was designed to simultaneously grasp two sides of a cardboard box. Those studies only dealt with one type of object and could be applied to grasping objects of various sizes.

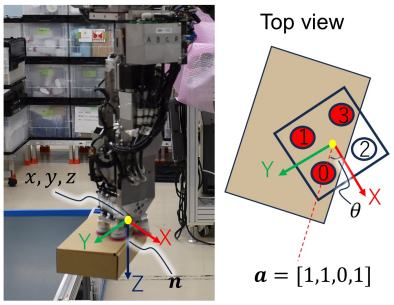

Fig. 2. Definition of grasp pose for MSC grasp

Domae et al. [18] proposed a convolution-based method that convolved gripper footprints with the binary mask of all objects without segmentation information (e.g. label of each object) to detect the MSC pose. A recent study by Schillinger et al. [5] also generated four-cup vacuum gripper footprints, but instead of convolving with a binary mask they convolved footprints with the grasp quality map estimated by U-net. However, these studies required sampling footprints. The computation cost depends on the sampling interval and scale of the footprints. Large interval and scale factor may lead to decreased accuracy of calculated grasp poses. Furthermore, since object segmentation information was not considered, these methods might output incorrect poses for grasping non-target objects.

Our method directly detects grasp poses from the RGB and depth images without sampling. Furthermore, our method detects surface segmentation masks based on which activation patterns are generated to ensure that the gripper only grasps the target object.

III. PROBLEM STATEMENT

We aim to detect the grasp pose and suction cup activation pattern given the RGB and depth images. We assume that all suction cup center positions are in the same plane and the outline of the hand is a rectangle. This assumption satisfies the layout of many common MSC vacuum grippers. In this study, we validate our method on a four-suction-cup vacuum gripper. All cups have the same specification (e.g. shape and max suction force).

As shown in Fig. 2, the MSC grasp pose (GP) is defined as $[x, y, z, \boldsymbol{n}, \theta, \boldsymbol{a}]$, where x, y, z are the world Cartesian coordinates of the grasp point position, $\boldsymbol{n}$ and θ represent the gripper orientation, $\boldsymbol{a}$ is the cup activation pattern, $\boldsymbol{n}$ is the gripper z-axis direction vector and θ is the rotation angle around $\boldsymbol{n}$. $\boldsymbol{a}$ is a one hot vector for determining which cup to activate. For example on the right side in Fig. 2, [1, 1, 0, 1] indicates that cup ids 0, 1, and 3 are to be activated.

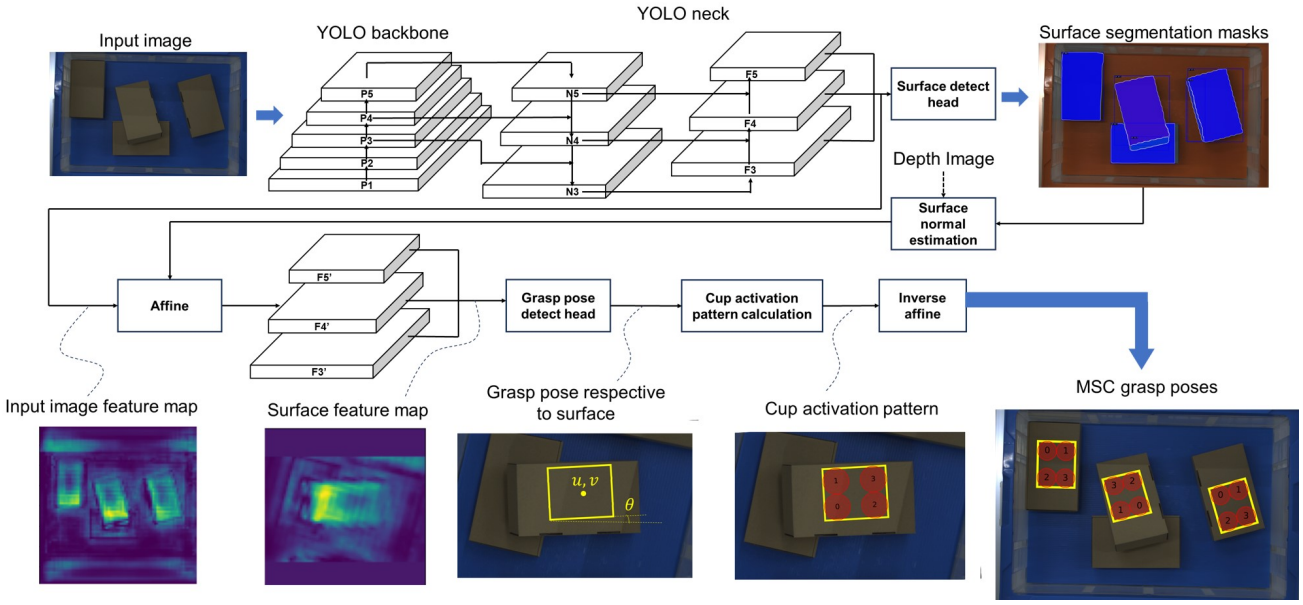

Fig. 3. MultipleCupSuctionNet

IV. MultipleCupSuctionNet

A. Overview

Figure 3 shows the framework for addressing the problem defined in section III. Directly regressing GP is difficult because of the various surface orientations, and hence $\boldsymbol{n}$ is computed first. The surface mask area is detected by YOLO [19] which is widely used for segmentation tasks. As $\boldsymbol{n}$ is always in the anti-direction of target surface normal, $\boldsymbol{n}$ can be easily obtained from surface normals which are estimated by the depth value or point cloud in the inferred surface mask area by the surface normal estimation process in Fig. 3. Detecting the remaining $x, y, z, \theta, \boldsymbol{a}$ is the core contribution of this study. Since learning x, y, z, θ for all surfaces in world coordinates is challenging, we convert x, y, z, θ to u, v, θ in surface image coordinates (grasp pose respective to surface in Fig.3) and then learned pose respective to each surface. In this case, feature maps of surfaces are required. As the features maps of the input image (input image feature map in Fig. 3) are extracted during surface mask detection, we take advantage of these to perform affine transformation on the feature maps to transform them to a viewpoint opposite to each surface (surface feature in Fig. 3). This helps to avoid additional computation for surface feature extraction during the training process. $\boldsymbol{a}$ is determined by calculating the intersection area between the cup and surface mask. Finally, we perform inverse transformation to convert u, v, θ to x, y, z, θ for all surfaces (MSC grasp poses in Fig. 3).

B. Surface mask detection

Given an RGB input image, image features P3, P4, and P5 are extracted by the YOLO backbone. The YOLO neck connects the backbone and surface mask detection head, which is composed of up-sampling and down-sampling modules. The role of it is to fuse P3, P4, and P5 and generate new input image feature maps F3, F4, and F5 that include the feature information from each of P3, P4, and P5. YOLACT++[20] was used as the surface mask detection head because it offers high detection speed for the segmentation task. Refer to [20] for more details of the mask detector.

C. Affine transformation

In order to learn GP in surface image coordinates, although the features of the surface feature map (F3', F4', F5') are needed, it is computational expensive to transform the input RGB image to a new RGB image opposite to the surface and then extract its features. Therefore, we take advantage of the input RGB image feature maps (F3, F4, F5) that are extracted by the YOLO neck, transforming them to surface image coordinates (a new viewpoint opposite to the surface) to save computational cost.

In order to achieve proper affine transformation, a 3×3 projective homography matrix G needs to be designed to ensure all necessary features are included after transformation. Let us define the camera frame of the original input image as c_1 and the new viewpoint frame (opposite to surface) after transformation as c_2. The transformation of points between the two frames is as in Eqs. (1) and (2).

$$ {}^{c_2}T_{c_1} = {}^{w}T_{c_2}^{-1}\,{}^{w}T_{c_1} \tag{1} $$

$$ {}^{w}T_{c_2} = \begin{bmatrix} {}^{w}\boldsymbol{R}_{\mathrm{sur}} & {}^{w}\boldsymbol{o}_{\mathrm{sur}} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{d}_{\mathrm{off}} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{2} $$

where ${}^{w}T_{c_1}$ is the camera pose, that is, the frame of c_1 in world coordinates; ${}^{w}T_{c_2}$ is the target viewpoint pose, that is, the frame of c_2 in world coordinates; ${}^{c_2}T_{c_1}$ is the frame transformation from c_1 to c_2; ${}^{w}\boldsymbol{R}_{\mathrm{sur}}$ and ${}^{w}\boldsymbol{o}_{\mathrm{sur}}$ are the surface orientation and center position output from the

surface normal estimation process, respectively; and $\boldsymbol{d}_{\text{off}}$ is the offset distance from c_2 to the surface.

To obtain G as in Eq. (3), Euclidean homography ${}^{c_2}H_{c_1}$ (Eq. (4)) and a scaled camera intrinsic matrix K^* (Eq. (5)) need to be calculated. Note that the camera intrinsic matrix needs to be scaled by s^* (Eq. (6)) because the input image shape (H_{im}, W_{im}) is different from feature map shape (H_F, W_F). The scale factor s^* is the smallest height and width ratio between the feature map and input image.

$$G = K^* {}^{c_2}H_{c_1} K^{*-1} \tag{3}$$

$${}^{c_2}H_{c_1} = {}^{c_2}R_{c_1} - \frac{{}^{c_2}t_{c_1} \cdot \boldsymbol{n}_{\text{sur}}}{\boldsymbol{o}_{\text{sur}} \cdot \boldsymbol{n}_{\text{sur}}} \tag{4}$$

$${}^{K^*} = \begin{bmatrix} s^* f_x & 0 & s^* c_x \\ 0 & s^* f_y & s^* c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$s^* = min(\frac{W_F}{W_{im}}, \frac{H_F}{H_{im}}) \tag{6}$$

where ${}^{c_2}R_{c_1}$ is the rotation matrix of ${}^{c_2}T_{c_1}$, f_x and f_y are the camera focal length, and c_x and c_y are the camera focus point.

In Eqs. (1)-(6), only $\boldsymbol{d}_{\text{off}}$ needs to be determined. Other parameters are constant values or can be derived from surface normal estimation results. To learn the grasp pose, the features of the surface and its surroundings are required. We set $\boldsymbol{d}_{\text{off}} = [0.0, 0.0, 1.0]$, which indicates that the new viewpoint is upright of the surface center by 1 m, to ensure that the required feature information is included after transformation.

D. Grasp pose detection

We design a grasp detection head for detecting MSC grasp poses. The grasp detection head is composed of three convolution modules. The first two are followed by batch normalization and activation function. The last one outputs the rotated box predictions (grasp pose respective to the surface in Fig. 3). Box predictions are computed based on features from different stages (F3', F4', F5'), respectively.

E. Suction cup activation pattern calculation

For each predicted grasp pose, we transformed the cup shape to surface image coordinates using G (Eq. (3)). To determine whether the ith cup is to be enabled, the intersection area between the cup shape and surface mask was computed. If the ratio of intersection area to cup shape area ($\frac{Area_{int}}{Area_{cup}}$) is larger than the threshold th, it indicates the cup can fully contact with the surface and can be activated (a_i=1 if $\frac{Area_{int}}{Area_{cup}} > th$) to suck onto the surface. We empirically set th to 0.9 because the suction cup could stably suck to the surface by the setup used in the test experiments.The cup activation pattern in Fig. 3 shows one example. The red circle indicates the enabled cup shape.

F. Loss function

The surface mask detection head and grasp pose detection head were trained separately. The surface mask detection loss L_{sur} is the sum of bounding box loss L_{bbox}, objectness loss L_{obj}, and mask loss L_{mask}. CIoU loss [21] was used for fast regression of surface bounding boxes $bbox$ ($L_{\text{sur}} = CIoU(bbox_{\text{pred}}, bbox_{\text{gt}})$). Cross entropy (CE) loss was used to compute objectness obj ($L_{obj} = CE(obj_{\text{pred}}, obj_{\text{gt}})$) and mask loss($L_{mask} = CE(mask_{\text{pred}}, mask_{\text{gt}})$).

The grasp pose detection loss L_{GP} is for direct regression of the grasp pose which is defined as rotated boxes ($rbbox$) in surface image coordinates. Rotated IoU loss L_{RIoU} was used to calculate the ratio of intersection area of predicted and ground truth rotated boxes to their union area ($L_{GP} = L_{RIoU} = RIoU(rbbox_{\text{pred}}, rbbox_{\text{gt}})$).

V. Experiment

A. Data collection and training

The surface mask detection layers were trained first. To generate a dataset of surface segmentation masks, an object segmentation deep learning model [22] was utilized to first detect the object region mask. The plane segmentation algorithm was then utilized to detect all surfaces of each object using the point cloud in the object region mask. After training the surface mask detection module, the features maps of surfaces were generated by extracting the F_3, F_4, F_5 output from YOLO neck. The ground truth grasp pose for each surface was collected by our previously proposed heuristic grasp planner [23]. The planner used a geometric analytic to evaluate the stability and safety when grasping and holding the object using MSCs in sampled candidate grasp poses. The planner finally output the most robust grasp pose in world coordinates. The feature maps and grasp poses were transformed to surface image coordinates to generate the dataset for training the grasp pose detection module. The objects included boxes, cylinders, sprayers, and detergent bottles. In certain scenes, the same types of objects were posed while the number and poses of objects were varied.

We collected a total of 3,891 data (3,343 for training, 274 for validation, and 274 for testing) for training the surface mask detection module and 2,226 data (1,787 for training, 167 for validation, and 272 for testing) for training the grasp pose detection module. Note that the size of the grasp pose dataset is smaller because not all surfaces have a grasp pose due to collisions and inverse kinematics errors.

Our MultipleCupSuctionNet was developed based on OpenMMLab [24], which is a computer vision algorithm system including commonly used deep learning frameworks. The training was conducted on an NVIDIA GeForce RTX 3090 GPU×2. Batch size was set to 48. Training epoch was set to 1000. Other training parameters such as learning rate were set to the default values.

B. Evaluation of training results

We evaluated and compared MultipleCupSuctionNet with two state-of-the-art models for SSC and MSC grasp planning, Dex-Net (FC-GQCNN-4.0-Suction) [3] and model-free

Fig. 4. Object set

MSC grasp planner [5], using the test dataset. For Dex-Net 4.0, we used the pretrained model and modified the configuration of the prediction to fit our environment. Because the source code was not provided, we implemented the model-free MSC grasp planner based on the paper and changed the footprint scale to our gripper. The evaluation metric includes planning accuracy and planning time. **Planning accuracy**: Because the final purpose is to grasp the object, a true positive prediction is defined as the gripper being able to suck onto the object surface by at least one suction cup (if any a_i in Eq (7) equals to 1) under the pose inferred by the deep neural network. The planning accuracy is defined as the number of data with true positive predictions over the entire test dataset. **Planning time**: Planning time is defined as the time cost for grasp planner to output a grasp pose.

C. Picking experiments

In order to evaluate the performance of MultipleCupSuctionNet in a real application, it was implemented on the industrial robot in Fig. 1 to perform the pick-and-place task. MutipleCupSuctionNet output multiple candidate grasp poses and a collision check was then performed to find collision free candidates. Finally, the highest candidate (highest z coordinate location) was used as the optimal grasp pose. The robot is composed of a six-degrees-of-freedom manipulator and a vacuum gripper with four suction cups. The max suction force for each cup is approximately 5 [N]. An Ensenso N36 camera is installed above the bin. The robot was required to pick randomly posed objects in the bin and then place them on the conveyor. The object set is shown in Fig. 4, including a box (0.06 [kg]), cylinder (0.09 [kg]), sprayer (0.08 [kg]), and detergent bottle (0.44 [kg]). These items are common in warehouses and have various surface sizes. A total of 50 items for each object were to be grasped. Grasp failure was defined as the robot being unable grasp any object after five attempts. If failure occured, the robot was stopped and then restarted after one object was removed manually. The performance was evaluated by success rate (number of items that can be grasped among 50 items) and average grasp attempt to pick one item. Grasp attempts were counted manually.

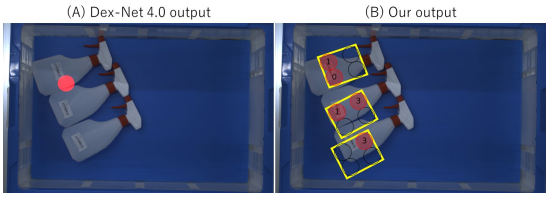

Fig. 5. Grasp planner outputs. (A) Dex-Net 4.0 output. (B) Our output.

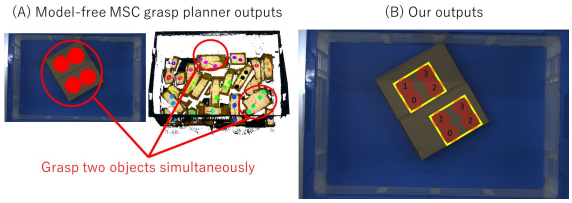

Fig. 6. Grasp planner outputs. (A) Model-free MSC grasp planner output. The middle figure is from [5]. (B) Our output.

VI. Results

A. Evaluation of training results

Table I shows the accuracy of grasp pose detection for three models. All three methods could predict the grasp poses. Our model mainly outperformed Dex-Net 4.0 and the model-free MSC grasp planner in dense scenes. Dex-Net uses only a depth image as the input so that when two objects are adjacent to each other, Dex-Net may regard them as one object and thus outputs the wrong grasp pose. As shown in the left side of Fig. 5, Dex-Net sometimes predicted the grasp point on the adjacent edge between two objects while our method could predict grasp poses for each object separately (right side of Fig. 5). The model-free MSC grasp planner sometimes grasped multiple objects as in Fig. 6. This seems to be the same planning result as in the previous study (see the middle of Fig. 6 and Fig. 9 in [5]). This may have lead to grasping non-target objects. The planning time was the fastest in Dex-Net because it does not need to calculate the gripper orientation and cup activation pattern. Our model was faster than [5] because there is no need to sample footprints and convolution computation for our method when planning the grasp pose. Note that the authors in [5] reported that their grasp planning cost 628 ms. The planning time in our study was longer than in [5] (Table I) , which might have resulted from the different size of image and footprint. To achieve optimal results, we set higher gripper footprint and input image resolutions, leading to a longer planning time.

B. Picking experiments

The picking robot achieved an average success rate of 94.5% for picking a box, cylinder, sprayer, and detergent bottle. Figure 7 shows the prediction results of MultipleCupSuctionNet. The results show that MultipleCupSuctionNet could correctly predict the grasp pose and cup activation pattern in accordance with the different surface sizes. The box and sprayer were the easier (small attempts) for the robot to pick because they have large planar surfaces. The cylinder (larger attempts) was difficult to pick because it has a curved surface rather than a planar one, so a small grasp

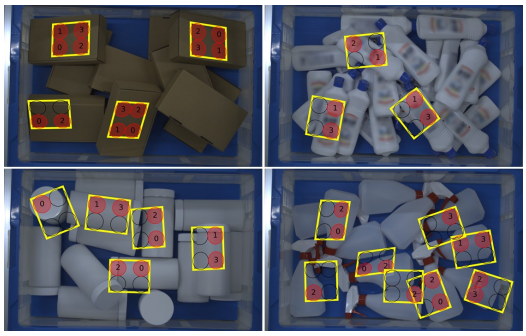

Fig. 7. MultipleCupSuctionNet prediction results. Red transparent circles are activated cups. Black circles are disabled cups. Note that only poses with a score higher than the threshold are output.

pose error may lead to grasp failure. Main failure result from no collision free path for the robot and the fact that item collided with the surroundings after being grasped.

VII. CONCLUSIONS

This study proposed a deep neural network for predicting grasp pose and determining the cup activation pattern for MSC grasp. Our network achieved a fast planning time and outperformed Dex-Net 4.0 and the model-free planner in cases where objects were located near each other. The robot achieved an average 94.5% success rate in robotic picking experiments, indicating that our proposed network model can be applied to picking task automation in warehouses.

We intend to make further improvements to this work in the future. First, the parameters (e.g., $\boldsymbol{d}_{\text{off}}$ in eq.(2)) in MultipleCupSuctionNet were determined empirically. We intend to validate the network using different parameter settings. Second, we intend to perform more experiments with different setups to further validate the network. As this study focused on validating whether the network can infer the MSC grasp poses and help the robot to successfully grasp the objects, we used only learned objects as our experimental targets. In the future, we intend to use novel objects and different types of objects posed in the bin to investigate the generalizability of the proposed network. We also intend to incorporate state-of-the-art planners into our robotics system to compare the performance in physical picking tasks in addition to the grasp pose inference performance. Experiments using a heuristic planner [23] will also be performed to study the strength of our learning based method.

TABLE I

ACCURACY OF GRASP POSE DETECTION

Model	Gripper	Input	Accuracy	Time cost
Dex-Net [3]	Single cup	Depth	60.3%(164/272)	0.19s
Model-free [5]	Four cups	RGB-D	73.9%(201/272)	5.62s
Ours	Four cups	RGB-D	80.1%(218/272)	0.47s

TABLE II

PICKING EXPERIMENT RESULTS

Object set	Success rate	Average attempts
Box	94% (47/50)	1.02 (48/47)
Cylinder	94% (47/50)	1.55 (73/47)
Sprayer	90% (45/50)	1.13 (51/45)
Detergent bottle	100% (50/50)	1.26 (63/50)
Average	94.5%	1.24

REFERENCES

[1] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *IJRR*, vol. 41, no. 7, pp. 690–705, 2022.

[2] H. Cao, H.-S. Fang, W. Liu, and C. Lu, "Suctionnet-1billion: A large-scale benchmark for suction grasping," *RA-L*, vol. 6, no. 4, pp. 8718–8725, 2021.

[3] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, p. eaau4984, 2019.

[4] P. Jiang, J. Oaki, Y. Ishihara, J. Ooga, H. Han, A. Sugahara, S. Tokura, H. Eto, K. Komoda, and A. Ogawa, "Learning suction graspability considering grasp quality and robot reachability for bin-picking," *Front. neurorobot.*, vol. 16, 2022.

[5] P. Schillinger, M. Gabriel, A. Kuss, H. Ziesche, and N. A. Vien, "Model-free grasping with multi-suction cup grippers for robotic bin picking," in *IROS 2023*. IEEE, 2023, pp. 3107–3113.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI 2015*. Springer, 2015, pp. 234–241.

[7] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *IJRR*, vol. 34, no. 4-5, pp. 705–724, 2015.

[8] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic *et al.*, "Deep learning approaches to grasp synthesis: A review," *T–RO*, 2023.

[9] R. Xu, F.-J. Chu, and P. A. Vela, "Gknet: grasp keypoint network for grasp candidates detection," *IJRR*, p. 02783649211069569, 2022.

[10] R. Araki, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Mt-dssd: multi-task deconvolutional single shot detector for object detection, segmentation, and grasping detection," *AR*, vol. 36, no. 8, pp. 373–387, 2022.

[11] L. Zhao, H. Liu, F. Li, X. Ding, Y. Sun, F. Sun, J. Shan, Q. Ye, L. Li, and B. Fang, "Implementation and optimization of grasping learning with dual-modal soft gripper," in *ICRA 2023*. IEEE, 2023, pp. 5887–5893.

[12] T. W. Utomo, A. I. Cahyadi, and I. Ardiyanto, "Suction-based grasp point estimation in cluttered environment for robotic manipulator using deep learning-based affordance map," *IJAC*, vol. 18, no. 2, pp. 277–287, 2021.

[13] S. Hasegawa, K. Wada, S. Kitagawa, Y. Uchimi, K. Okada, and M. Inaba, "Graspfusion: Realizing complex motion by learning and fusing grasp modalities with instance segmentation," in *ICRA 2019*. IEEE, 2019, pp. 7235–7241.

[14] J. Li and D. J. Cappelleri, "Sim-suction: Learning a suction grasp policy for cluttered environments using a synthetic benchmark," *T–RO*, 2023.

[15] G. Mantriota, "Optimal grasp of vacuum grippers with multiple suction cups," *MECH MACH THEORY*, vol. 42, no. 1, pp. 18–33, 2007.

[16] V. Kozák, R. Sushkov, M. Kulich, and L. Přeučil, "Data-driven object pose estimation in a practical bin-picking application," *Sensors*, vol. 21, no. 18, p. 6093, 2021.

[17] J. Tanaka and A. Ogawa, "Cardboard box depalletizing robot using two-surface suction and elastic joint mechanisms: mechanism proposal and verification," *JRM*, vol. 31, no. 3, pp. 474–492, 2019.

[18] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," in *ICRA 2014*. IEEE, 2014, pp. 1997–2004.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR 2016*, 2016, pp. 779–788.

[20] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *ICCV 2019*, 2019, pp. 9157–9166.

[21] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *AAAI 2020*, vol. 34, no. 07, 2020, pp. 12 993–13 000.

[22] V.-Q. Pham, S. Ito, and T. Kozakaya, "Biseg: Simultaneous instance segmentation and semantic segmentation with fully convolutional networks," *arXiv preprint arXiv:1706.02135*, 2017.

[23] H. Eto, S. Tokura, K. Komoda, P. Jiang, and A. Ogawa, "Development of a grasp planning algorithm using grasp robustness metric," in *JSME Conf. Robotics and Mechatronics*, 2020, pp. 1P1–B03.

[24] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.