IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

# CoAS-Net: Context-Aware Suction Network With a Large-Scale Domain Randomized Synthetic Dataset

Yeong Gwang Son, Tat Hieu Bui, Juyong Hong, Yong Hyeon Kim, Seung Jae Moon, Chun Soo Kim, Issac Rhee, Hansol Kang, and Hyouk Ryeol Choi, *Fellow, IEEE*

***Abstract*—Robotic grasping is one of the essential skills in robotics. From industrial to housework, robots are required to handle objects, enabling them to interact with their surroundings. Among the various tasks in robotic grasping, bin-picking is considered one of the most challenging because of the cluttered bin filled with objects. Also, for the next-level automation, they need to handle unseen objects and discriminate target objects and outliers. This letter proposes a novel dataset generation pipeline for suction-grasping in bin-picking tasks. This pipeline consists of a series of methods that progressively transit from a single object evaluation to an entire scene evaluation and lower the dimension of the labels to the image space. We trained a suction prediction FCN (Fully Convolution Network) with our dataset generated from the pipeline and conducted bin-picking experiments. Our large-scale collision-free annotation enables the network to understand the context of a bin-picking task, where collisions between the gripper and the bin or object are a concern, and distinguishing the background is crucial. The results show that our solution excels the existing methods, and the network demonstrates its context-aware grasp on objects with loosely defined RoI (Region of Interest).**

***Index Terms*—Data sets for robotic vision, deep learning in grasping and manipulation, computer vision for automation, suction grasping.**

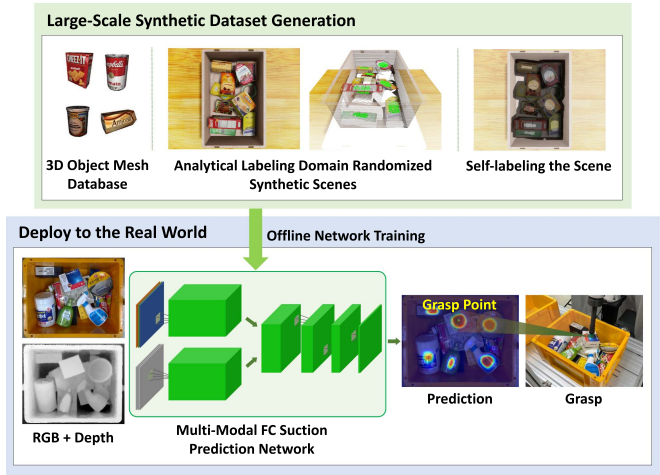


Fig. 1. **Outline of the proposed method.** Our method consists of a large-scale synthetic dataset generation pipeline and deployment of our suction prediction model to the real world using the generated dataset. The network is trained to output dense suction-based grasp prediction and send the best candidate to the robot, which executes grasping.

## I. Introduction

ROBOTIC grasping is one of the fundamental abilities in robotics. There are various applications where this skill can be beneficial, such as manufacturing, logistics, healthcare, and more. Robots with grasping skills can expand their operational capabilities and physically interact with their surroundings. Among the various tasks in robotic grasping, bin-picking is considered one of the most essential tasks, particularly due to the recent demand for advanced automation in logistics. In automated bin-picking, the system must identify objects, find the most reliable grasp pose, and stably grasp the object. This can be tackled by various challenges existing in the bin-picking task. For instance, objects inside a bin can be highly cluttered, making it difficult to accurately predict their shape, which is essential information. Additionally, the hundreds of thousands of objects in a warehouse make relying on memorization for their handling impractical.

Recently, object 6D pose estimation models, which output both translational and rotational information based on deep neural networks, have been developed to grasp objects randomly stacked in a bin [1], [2], [3]. However, most of these approaches require 3D mesh models for registration or network training, which restricts their applicability when dealing with various unseen objects.

Due to this limitation, model-free methods are getting attention as promising approaches to robotic grasping in a logistics environment due to their potential for generalization to unseen objects [8]. These methods use learning techniques to train networks that estimate grasp configuration directly from 2D images. The input to the network can be RGB [19], Depth [11], [12], [13], [14], [15], [17] or RGB-D [18], [19], [20], [21], [22]. The main challenge for the model-free approaches lies in

Manuscript received 19 September 2023; accepted 20 November 2023. Date of publication 29 November 2023; date of current version 11 December 2023. This letter was recommended for publication by Associate Editor Devesh Jha and Editor Cesar Cadena Lerma upon evaluation of the reviewers' comments. This work was supported by the Ministry of Trade, Industry and Energy (MOTIE, Korea), through the Industrial Strategic Technology Development Program under Grant 20014558. *(Corresponding author: Hyouk Ryeol Choi.)*

The authors are with the School of Mechanical Engineering, Sungkyunkwan University, Suwon 16417, South Korea (e-mail: syoungk20@g.skku.edu; buitathieu1995@gmail.com; juyong0000@skku.edu; gattswet45@g.skku.edu; msj19@skku.edu; 3309so@naver.com; issacrh@skku.edu; kanghs0822@skku.edu; choihyoukryeol@gmail.com).

Our dataset and the grasp detection model are available at https://github.com/SonYeongGwang/CoAS-Net.git.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3337692, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3337692

2377-3766 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.
Authorized licensed use limited to: Sungkyunkwan University. Downloaded on December 14,2023 at 04:36:49 UTC from IEEE Xplore. Restrictions apply.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

building reliable and large-scale training datasets. These datasets can be generated through human labor in the real world [22] or in a simulation [11], [12], [13], [14]. Considering the necessity of the high-performance model to be trained with a huge dataset, the labor-intensive dataset generation method has its limitations. Therefore, simulation-based synthetic data is an attractive alternative for creating large-scale image datasets.

This letter proposes a suction dataset generation pipeline using fully simulated images for robotic grasping. This pipeline enables us to build and dense-annotate 50,000 synthetic images without extensive labor-intensive processes. We utilized NVIDIA's newest simulator, Isaac Sim, as a scalable simulation and synthetic data generation tool that utilizes GPU devices for achieving photorealism through real-time ray tracing. This feature helps to reduce the domain discrepancy between simulation and real-world environments. In addition to that, we further reduce the domain gap by incorporating domain randomization into the dataset generation pipeline [24], [25]. Fig. 1 outlines our dataset generation pipeline and the detection model.

In this letter, we refer to context-awareness as the capability to 1) sample grasp candidates on the objects inside the bin as much as possible and 2) consider collisions without any postprocessing such as background removal or RoI settings. We hypothesize that the key to fulfilling the first requirement is a network that competitively outputs between grasp candidates and outliers for each pixel. Among the previous works, the suction detection model from [22] meets this requirement by outputting three different classes: positive grasp points, negative grasp points, and outliers, for each pixel. Based on this assumption, we adopt the annotation representation of [22], which employs three distinct labels to identify positive, negative, and ignore pixels. To address the second requirement, we systematically check for collisions between the gripper, the object, and the bin, annotating the scene to prevent collisions implicitly.

Our bin-picking experiment shows that our suction prediction network, trained with our dataset, outperforms previous methods. Additionally, its context-aware ability to understand the task of bin-picking allows for the exclusion of the background without necessitating extensive RoI settings, showing its object-oriented prediction capabilities.

Contributions of this work are threefold:

1) We introduce a novel suction dataset collection pipeline that enables the generation of a large-scale dataset without requiring labor-intensive processes.
2) We conducted ablation studies to assess the effectiveness of methods incorporated into our pipeline.
3) We demonstrated the context-aware performance of a suction prediction network trained with our synthetic dataset through a real-robot experiment and compared its results with prior works.

## II. Related Work

### A. Suction Dataset

Mahler et al. [12] developed a dataset called Dex-Net 3.0, which consists of 2.8 million synthetic point clouds with a compliant suction contact model designed for evaluating seal condition and wrench resistance. However, RGB information is not available for use in the dataset. Zeng et al. [22] built a RGB-D dataset based on real images. This has the advantage of using real images, which can minimize domain discrepancies. Despite this benefit, building large-scale datasets is challenging due to a labor-intensive labeling process. Cao et al. [21] addressed this limitation. They proposed a method to minimize human engagement during the labeling process by using 3D mesh models and their corresponding pose information within a real scene. However, human labor still requires setting up a cluttered environment and capturing each scene. Gilles et al. [23] introduced a large-scale RGB-D synthetic dataset for a bin-picking scenario. They utilized a photo-realistic simulator to build 217 k images with various annotations, including suction grasping labels. However, it does not account much for the domain gap between simulation and reality, a challenge that can be confronted in real-world deployment. Our dataset consists of large-scale, domain-randomized synthetic data generated through a labor-free annotation process. In addition, the dataset contains 6D pose of objects at each scene. We hope this can be helpful for researchers in the community.

### B. Robotic Grasping

Object 6D pose can be used to select the best grasp configuration for each object. Object 6D pose estimation models based on deep neural networks have been developed to grasp objects randomly stacked in a bin [1], [2], [3], [4]. These approaches require 3D mesh models for registration or network training, limiting their application to a broader range of fields. To avoid 3D mesh model registration, research has been done using primitive shape matching on unknown objects [5], [6] or reconstructing a 3D mesh that best fits the target unseen object [7]. Model-free methods are also attractive because they don't require 3D mesh models [8]. Object detection or instance segmentation can be done before determining the optimal grasp point with some predefined rules [9], [10]. Mahler et al. [11], [12] trained discriminative prediction model GQ-CNN with their synthesized dataset. Further, they generated a dataset from simulated cluttered scenes to fine-tune the model [13], [14]. Depth-based grasp detection methods also showed impressive results [15], [16], [17]. RGB images also can be incorporated with the depth images [18], [19], [20], [21], [22]. Features extracted from the RGB images help discriminate object boundaries and compensate for noise from the Depth data.

### C. Bridging the Domain Gap

Simulation-based dataset generation can reduce the time and cost of building large-scale datasets. However, using only a simulation-based image dataset, the domain gap affects significantly when implemented in the real world. To bridge the gap, Tobin et al. [24] introduced various randomized factors in a non-realistic simulation environment and showed that it is possible for a model to learn generalized features that can be adapted to the real world. They also found that even randomly generated, non-realistic mesh models could aid models in generalizing grasping in the real world, particularly when many mesh models

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.
Authorized licensed use limited to: Sungkyunkwan University. Downloaded on December 14,2023 at 04:36:49 UTC from IEEE Xplore. Restrictions apply.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

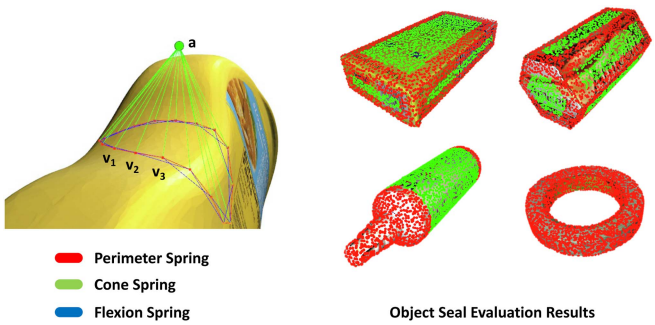


Fig. 2. **Visualization of the seal evaluation process and the evaluation results.** (Left) The compliant suction contact model is projected onto the mesh surface. (Right) The green points indicate regions where a vacuum can be maintained, while the red points show where it would fail.

were used at scale [25]. Generative adversarial network(GAN) can be used to generate non-randomized, canonical images from both simulation and the real world, which can then be fed into a vision-based grasping agent [26], [27]. In this letter, we utilized the photo-realistic simulator Issac Sim to reduce the domain gap and applied domain randomization so that the grasp detection network generalizes to real-world scenes.

## III. Domain Randomized Synesthetic Dataset

This section will provide detailed information about our dataset generation pipeline and the dataset itself.

### A. Seal Evaluation

Firstly, mesh models are evaluated to find a point where the suction cup can hold a vacuum inside the cup. We leveraged the compliant suction contact model [12] to evaluate the seal along a mesh model. The suction cup model consists of 3 different spring components: perimeter, flexion, and cone. Perimeter springs are connections of $\mathbf{v}_i$, $\mathbf{v}_{i+1}$. Flexion springs are connections of $\mathbf{v}_i$, $\mathbf{v}_{i+2}$. Cone springs connect each vertex $\mathbf{v}_i$ with the apex $\mathbf{a}$. The suction cup model is then projected to the mesh models along the approach vector calculated by the normal estimation. We uniformly sample point cloud $\boldsymbol{P} \in \mathbb{R}^{N\times 3}$ on a mesh model and measure the deformations of each spring $\boldsymbol{D}_{p,f,c}$ along the point $\boldsymbol{p} \in \boldsymbol{P}$. We set a lower threshold $T_{low}$, an upper threshold $T_{up}$, and a variance threshold $T_{var}$ to evaluate seal formulation [16]. $\boldsymbol{p}$ is considered as a valid point where the seal can be formed when all sub-springs $d_{p,f,c} \in \boldsymbol{D}_{p,f,c}$ meet the seal-formation conditions. Algorithm 1 presents the pseudo-code of the seal evaluation. Fig. 2 illustrates the compliant suction contact model projected onto the mesh surface and the results of the seal annotations based on the binary label vector $\boldsymbol{L}_{seal} \in \mathbb{R}^N$. The weight of objects is not considered during this process.

### B. Randomized Scene Generation

In this step, objects $\mathcal{O} = \{o_k | k = 1, 2, \ldots, N_o\}$ are randomly selected from our mesh model database and dropped into a bin with arbitrary poses $\mathcal{T} = \{\boldsymbol{T}_k^w | k = 1, 2, \ldots, N_o\}$ to the world frame. We use Isaac Sim to generate a synthetic environment to generate photo-realistic synthetic images, which can help reduce the domain gap in real-world data with real-time physically accurate environment rendering. The objects and poses used in each scene are also stored for subsequent tasks. To minimize the domain gap and generalize to the real world, we adopt several randomization methods as:

1) 500 different textures of the bin and ground plane
2) $[1, 8]$ variations of the number of objects inside the bin
3) Randomly initialized pose of the mesh model
4) Randomized direction and intensity of the light
5) Random camera poses

Objects are dropped inside the bin with a range of $[-0.05, 0.05]m$ and $[-0.1, 0.1]m$ for x, y-axis translation respectively, and $[0, \pi]$ for x, y, z axis rotation.

The pose of the camera is randomized by translating and rotating only with the z-axis of the world frame in the simulation, which makes a rotation along the axis perpendicular to the ground plane. We found that a top-down view image is the best as it provides a full view of the bin. Additionally, we found that our dataset is sufficient for the model to generalize to rotations along the x or y-axis. Fig. 3 shows images of the randomly generated scenes.

**Algorithm 1:** Seal Evaluation Pseudo-Code.

**Require:** Object point cloud $\boldsymbol{P}$, the number $N$ of points in $\boldsymbol{P}$, the number of verties $N_v$, and thresholds $T_{low}$, $T_{up}$, $T_{var}$.
**Ensure:** Seal label set $\boldsymbol{L}_{seal}$ of the target object.
  $\mathbf{D}_{p,f,c} \leftarrow zero^{N_v}$ ▷Initialize spring deformation
  $\mathbf{L}_{seal} \leftarrow zero^{N}$ ▷Initialize object label
  **for all** $\boldsymbol{p} \in \boldsymbol{P}$ **do**
    Align suction cup along the normal vector of $\boldsymbol{p}$.
    **for** $i \leftarrow 1, N_v$ **do**
      Calculate deformations of each spring $d^i_{p,f,c}$
      $\mathbf{append}(\boldsymbol{D}_{p,f,c}, d^i_{p,f,c})$ ▷Add $d$ to the end of $\boldsymbol{D}$
    **end for**
    **for all** $d_{p,f,c} \in \boldsymbol{D}_{p,f,c}$ **do**
      **if** $d_{p,f,c} < T_{low}$ or $(d_{p,f,c} < T_{up}$ and $var(\boldsymbol{D}_{p,f,c}) < T_{var})$
      $\mathbf{append}(\boldsymbol{L}_{seal}, 1)$
        **else**
      $\mathbf{append}(\boldsymbol{L}_{seal}, 0)$
        **end if**
      **end for**
    **end for**

### C. Wrench and Collision Evaluation

In this process, we amend the seal labeling $\boldsymbol{L}_{seal}$ of each mesh model, which is based on the seal evaluation result, to get valid grasping labels $\boldsymbol{L}_{grasp} \in \mathbb{R}^N$. A suction point that fails to resist the wrench caused by gravity can not be a suction candidate even if it can hold a vacuum. Dex-Net 3.0 [12] has defined 6 different criteria to evaluate whether the suction cup can hold an object

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.
Authorized licensed use limited to: Sungkyunkwan University. Downloaded on December 14,2023 at 04:36:49 UTC from IEEE Xplore. Restrictions apply.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Fig. 3. **RGB samples of the generated scene.** The scene is created using Isaac Sim. With each update, the randomized factors are applied to the simulation environment.

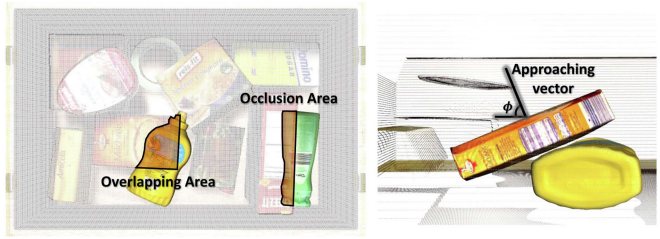


Fig. 4. **Collision and occlusion detection in the target scene.** (Left) The object on the left has partially overlapped with the object above, and the object on the right is partially occluded from the camera's viewpoint. Any positive-labeled candidate in the shaded region will be re-evaluated as a negative candidate. (Right) $\phi$ represents the angle between the table and the approaching vector of the gripper.

and resist the wrench:

$$\begin{aligned}
&\textbf{Friction}: \quad \sqrt{3}|f_x| \le \mu f_N \quad \sqrt{3}|f_y| \le \mu f_N \quad \sqrt{3}|\tau_z| \le r\mu f_N \\
&\textbf{Material}: \quad \sqrt{2}|\tau_x| \le \pi r\kappa \quad \sqrt{2}|\tau_y| \le \pi r\kappa \\
&\textbf{Suction}: \quad f_z \ge -V
\end{aligned} \tag{1}$$

$\mu$ in the Friction condition is the friction coefficient, $r$ is the radius of the contact ring, and $\kappa$ is the constant that models a material-dependent maximum stress of the suction cup when it deforms within the linear-elastic region. Cao et al. [21] has simplified these conditions by having only the 'Material' condition. They assumed that 'Friction' and 'Suction' conditions are usually satisfied while keeping the seal. Also, as the model learns to minimize the moment arm, the weight of objects does not contribute during the training. Based on this assumption, we evaluate grasp candidates on each object and filter out candidates whose pose is unsuitable for grasping. The label $l \in \boldsymbol{L}_{seal}$ of a grasp candidate remains positive if 1) the total torque $\tau_{total} = \sqrt{\tau_x^2 + \tau_y^2}$ acting on the suction cup is less than the torque threshold $\tau_{thrsh} = \pi r\kappa$, 2) the angle $\phi$ between a table and an approaching vector is larger than $\phi_{thrsh}$, and 3) there's no collision between a gripper and objects inside a bin.

The second condition removes a candidate that causes a collision between a bin and the gripper. Also, it can prevent the gripper from colliding with a bin. From the first condition, we modify $\boldsymbol{L}_{seal}$ to $\boldsymbol{L}_{wrnch}$

To check the third condition, we generate a primitive-shaped gripper model and align it with an approaching vector of a candidate. Then, assess a collision by examining whether the gripper model intersects with other objects. Candidates located on the overlapped surface of a bottom object will be eliminated through this process. In addition, we consider occlusion caused by the camera's field-of-view. This process will filter out candidates labeled as positive but occluded by other objects. After evaluating collision and occlusion, we calculate the filter ratio of each object. All the candidates of an object $o_k \in \boldsymbol{\mathcal{O}}$ will be filtered out if $\lambda_k < \lambda_{thrsh}$. Otherwise, only the candidate that satisfies the collision and occlusion conditions will remain. Here, $\lambda$ represents the filter ratio of the number of candidates before and after the evaluation. From the second and the third conditions, we refine $\boldsymbol{L}_{wrnch}$ to $\boldsymbol{L}_{grasp}$, which labels optimal grasp candidates. Binary label vector $\boldsymbol{L}_{grasp}$ contains the final annotations derived from the series of evaluations on the sampled point cloud. Fig. 4 illustrates the conditions for the collision evaluation.

Analyzing the conditions above requires locally defined seal labels $\boldsymbol{L}_{seal}$ to be globally defined to consider the surroundings, such as objects nearby or the bin. To globally define the $\boldsymbol{L}_{seal}$, we transform the object point cloud $\boldsymbol{P}$ of each object $o_k \in \boldsymbol{\mathcal{O}}$ used in the scene into the camera frame by the (2) and match to the point cloud of the whole scene.

$$\boldsymbol{P}^c = \boldsymbol{T}_w^c \boldsymbol{T}_k^w \boldsymbol{P}^k \tag{2}$$

Here, $\boldsymbol{T}_w^c$ is the world frame with respect to the camera frame, and $\boldsymbol{T}_k^w \in \boldsymbol{\mathcal{T}}$ is a posture of object $o_k$ with respect to the world frame. Now that both the point cloud of the whole scene and the point cloud of the objects $\boldsymbol{P}^c$ are defined in the camera frame, we can globally refine the label $\boldsymbol{L}_{seal}$ to get $\boldsymbol{L}_{grasp}$. Fig. 5 visualizes the filtering steps of the grasp labels.

### D. Image Conversion

In this stage, we convert $\boldsymbol{P}^c$ defined in 3D cartesian space into a 2D image plane to generate label images. To train a grasp pose detection model that outputs image-based dense prediction, we densely annotate images with three labels: 'pos' for positive grasp candidates, 'neg' for negative grasp candidates, and 'ignore' for outliers such as the background. To label the image, we first individually annotate positive labels $I_{pos}$ and negative labels $I_{neg}$ by projecting the points into the image plane with an equation:

$$u = \frac{f_x X}{Z} + c_x \,, \; v = \frac{f_y Y}{Z} + c_y \tag{3}$$

Here, $f_x$, $f_y$ are the focal length, and $c_x$, $c_y$ are the principal points which are all from the camera intrinsic parameters. We get $I_{pos}$ and $I_{neg}$ by selecting X, Y, Z from $\boldsymbol{p}^c \in \boldsymbol{P}^c$ whose label $\boldsymbol{L}_{grasp}$ is positive and negative, respectively.

As we sampled the fixed number of points along each object, $\boldsymbol{P}^c$ doesn't contain dense information in the image plane when projected. This causes a salt-pepper-like conversion result in the image. To densely label each pixel, we use morphological image processing. The process's parameters are selected, sufficient to fill the neighbors in the image plane. Especially for the negative label image, we set parameters that generate a larger region than the original. This margin helps the model avoid

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.
Authorized licensed use limited to: Sungkyunkwan University. Downloaded on December 14,2023 at 04:36:49 UTC from IEEE Xplore. Restrictions apply.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

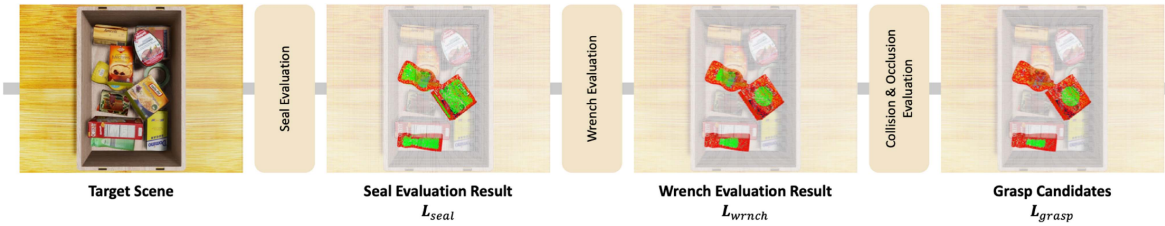


Fig. 5. **Intermidiate results visualization of our evaluation processes.** The initial grasp candidates $L_{seal}$ are progressively refined to final grasp candidates $L_{grasp}$. These processes effectively assess the fidelity of each candidate on each object and remove partial or entire candidates. All objects in the scene are evaluated, and three objects are selected to better visualize the process.

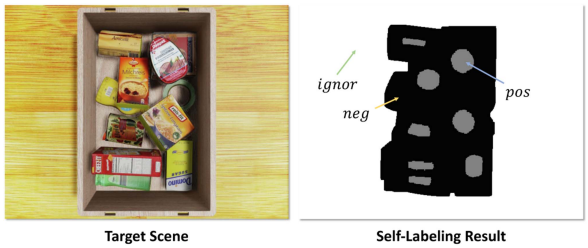


Fig. 6. **Label image for the corresponding scene.** It contains three classes which are distinguished by their color in a single-channel image.

TABLE I
DISTRIBUTION OF DATASET BASED ON OBJECT QUANTITY AND LABEL RATIOS

| Object Quantity | Ratio(%) | Label Ratio(%) | | | |
|---|---|---|---|---|---|
| | | [0, 10) | [10, 20) | [20, 30) | [30, 40] |
| 1∼2 | 26.3 | 35.1 | 53.6 | 10.8 | 0.5 |
| 3∼4 | 25.8 | 40.2 | 55.8 | 4.0 | 0.0 |
| 5∼6 | 24.8 | 51.1 | 47.5 | 1.4 | 0.0 |
| 7∼8 | 23.1 | 58.0 | 41.4 | 0.6 | 0.0 |

prediction near the edges of objects. Finally, we map each pixel of the images into label image $I_{label}$ using the mapping function $f : \{I_{pos}, I_{neg}\} \rightarrow I_{label}$. The mapping function $f$ generates an empty 3-channel image and inserts input images into the first and the second channels. Then maps the image into a single-channel image with the desired value for each label.

### E. Dataset Details

Our dataset contains 50,000 photo-realistic synthetic RGB-D images from 6,250 different scenes with corresponding label images as Fig. 6. We incorporated 150 mesh models from YCB [29], KIT [30] object and scanned mesh models [21]. The selected mesh models are suitable sizes to be placed inside a $(0.6 \times 0.4 \times 0.2)$ m bin. When evaluated with a 3 cm diameter suction cup model, these models also have enough positive grasp candidates. Additionally, our mesh model set includes 5% of small or hollow objects with no valid grasp candidates, as shown in the bottom right corner of Fig. 2 to enhance the robustness of our detection model. Table I shows the distribution of images containing varying numbers of objects. Also, it shows the distribution of images with different label ratios within the same object density category, where the label ratio is defined as the number of positive labels divided by the number of negative labels. In addition, we provide object poses within each scene. We hope this can help research communities in robotic grasping, detection, and segmentation.

## IV. SUCTION DETECTION MODEL

### A. Architecture

We developed the suction grasp detection network based on the pixel-wise dense grasp detection network from [22]. As in Fig. 7, the network takes both RGB and Depth images and extracts features of each image from the two separate ResNet101 layers. To fit the Depth image into the network, we duplicate the image to have three channels. To maintain the spatial information in output feature maps and ensure a robust prediction on a small object, we additionally incorporated Feature Pyramid Network (FPN) [28], in collaboration with the ResNet backbone. Also, we added separate $3 \times 3$ 2D convolutions with zero padding on each final feature map and combined them using concatenation. At the end of the model, a set of $1 \times 1$ 2D convolution layers reduces the channel of the combined feature map, bilinearly upsamples into the size same as the inputs, and finally applies softmax to get a pixel-wise dense prediction of suction grasping candidates as in the previous work.

### B. Training Procedure

The whole network is trained end-to-end with supervised learning based on our dataset. We use the cross entropy loss as a loss function while ignoring the third class to train the network to predict competitively among the remaining two classes. The network weights are updated based on Stochastic Gradient Descent with momentum $= 0.99$, learning rate $= 0.001$, batch size $= 32$, and a maximum number of epochs $= 500$. ResNet backbones are pre-trained with ImageNet and fine-tuned during the training with a learning rate $= 1 \times 10^{-4}$. Finally, as one of the methods for domain randomization, we apply random noise to the input depth image before being fed into the network. Training took approximately 58 hours on two Nvidia RTX 3090 GPUs and intel-10940 k CPU.

## V. EXPERIMENT

We conducted experiments based on a bin-picking task to compare with other methods. Then, we evaluated the effects of various settings using an ablation study.

### A. Experiment Setup

We used Intel Realsens L515 RGB-D camera. The camera is fixed on the top of the test bed to take an image with no blind spot in the bin. The camera is placed 1 m high from the bed in

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.
Authorized licensed use limited to: Sungkyunkwan University. Downloaded on December 14,2023 at 04:36:49 UTC from IEEE Xplore. Restrictions apply.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

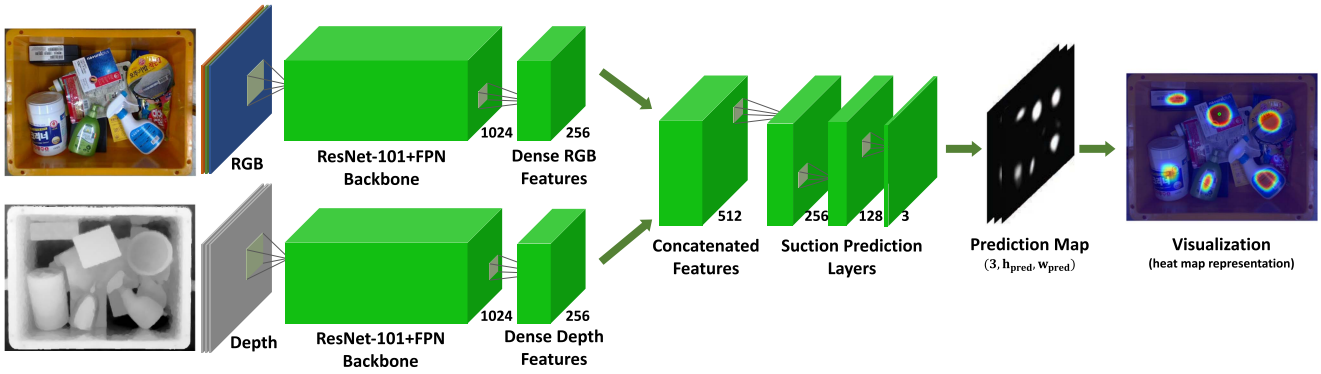


Fig. 7. **Architecture of our suction prediction network.** It extracts RGB and Depth features separately from the separately fine-tuned ResNet+FPN backbones. We apply $3 \times 3$ 2D convolution to each feature map to reduce a dimension before merging the two features. We then apply a series of $1 \times 1$ 2D convolutions to the concatenated feature map to reduce dimension to have 3 labels as individual channels. The output is then softmaxed along the channels to produce a probability of grasping.

this setup. We have prepared 14 novel objects commonly found in daily life and randomly stacked them in the bin. We have repeated this 8 times for each algorithm. We used a Rainbow Robotics RB10 6-DoF manipulator with a 3D printed suction gripper to pick the object.

### B. Experiment Details

We compared methods for suction grasp detection from previous work and utilized the best checkpoints they provided without making any additional adjustments or fine-tuning. We selected the best grasp candidate from the output for conducting the grasping task. Some of the methods leverage additional information to optimize the prediction. Segment image [12], [14] masks only the objects or background images subtracted with input images [22]. In this experiment, however, we did not give direct information about the object's location in the image. We roughly set a RoI around a bin, including the background. In this condition, the algorithms should find the best grasp and discriminate object/non-object area. We assume this can promote context-aware methods for tasks in unseen environments. The best grasp candidate is converted into 3D using the camera intrinsic parameters, and the normal from the point cloud of the scene is used for the approach direction of the robot arm.

### C. Experiment Result

We used three metrics to evaluate algorithms. **Success Rate** is defined as a ratio of the success grasps to the total grasp trials. **Clear Rate** defined as a ratio of the success grasps to the number of objects used for the experiment. **Self-Pick Rate** tells how a network can discriminate only a target object from the background. We define Self-Pick Rate as $1 - \frac{H}{N_{obj} \cdot i}$, where $N_{obj}$ represents the number of objects used, $H$ is the number of human interventions during the experiment and $i$ is the number of tests that have been done for each method. We counted grasp trials if the network predicted grasp on an object. When the network failed to grasp an object or collided with the bin 3 consecutive times, we removed the object from the bin. If the network failed to detect a grasp on an object and outputs candidates in the background 3 consecutive times, we manually rearranged objects and counted human intervention. Finally, if the network failed to detect a grasp on an object after 3 rearrangements, we manually removed a single object from the bin. We have conducted over 100 picks per method and compared the results in Table II.

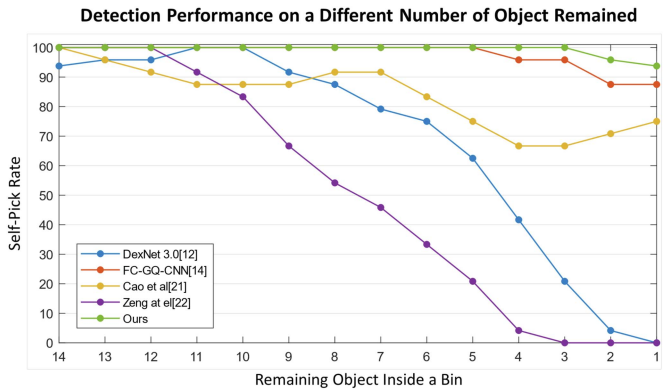


Fig. 8. **Detection performance on a different number of objects remained.** As a robot picks an object in a bin, the number of objects changes. This result shows that the number of objects affects the network's performance. This can be crucial in real-robot applications where human intervention can be expensive.

Our method outperforms Dex-Net 3.0 [12] and FC-GQ-CNN [14] in our experiment. These methods are trained with synthetic depth images and require a high-precision depth camera to bridge the gap between the simulated depth and the real-world depth, resulting in a performance drop when used with a low-cost depth camera. However, they recorded self-pick rates of more than 90%. They didn't fall sharply during the experiment as Fig. 8, which means that even if they failed to grasp an object, reliable subsequent grasp detection could be conducted, thus clear objects. Furthermore, as the results demonstrate, our method outperforms both the methods proposed by Zeng et al. [22], and Cao et al. [21]. In addition, our method shows stable and constant results during the experiment as in Fig. 8 while there are continuous decreases in performance from the two methods as the robot clears objects, which causes the background region of the bin to get larger. This implies that our

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.
Authorized licensed use limited to: Sungkyunkwan University. Downloaded on December 14,2023 at 04:36:49 UTC from IEEE Xplore. Restrictions apply.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE II
RESULT OF A REAL-WORLD NOVEL OBJECT BIN-PICKING EXPERIMENT

| Method | Total Grasp | Success Grasp | Human Intervention | Success Rate(%) | Clear Rate(%) | Self-Pick Rate(%) |
|---|---|---|---|---|---|---|
| Cao *et al.* [21] | 134 | 60 | 51 | 44.8 | 53.6 | 54.5 |
| FC-GQ-CNN [14] | 173 | 92 | 10 | 53.2 | 82.1 | 91.1 |
| Dex-Net 3.0 [12] | 178 | 96 | 3 | 53.9 | 85.7 | 97.3 |
| Zeng *et al.* [22] | 122 | 102 | 35 | 83.6 | 91.1 | 67.9 |
| **CoAS-Net (Ours)** | 119 | 110 | 1 | **92.4** | **98.2** | **99.1** |

The bold values highlight the best results for each metric.

TABLE III
COMPARISON OF SUCTION GRASP DETECTION MODELS TRAINED WITH DIFFERENT DATASET BASED ON THE NETWORK FROM [22]

| Dataset | Success Rate(%) | Clear Rate(%) | Self-Pick Rate(%) |
|---|---|---|---|
| Real [22] | 83.6 | 91.1 | 67.9 |
| Ours-Subset | 47.5 | 95.0 | 79.6 |
| **Ours-Full** | **85.2** | **95.3** | **97.9** |

The bold values highlight the best results for each metric.

TABLE IV
ABLATION STUDY ON OUR DATASET

| Evaluation Target | Sim / Real Precision | Recall | F-measure |
|---|---|---|---|
| w/o camera randomization | - | - | - |
| w/o collision model | 74.1/83.4 | 64.8/34.9 | 67.4/47.1 |
| w/o random noise | **85.1/90.2** | 59.1/24.2 | 68.0/35.5 |
| Full method | 82.6/84.2 | **68.9/53.2** | **75.5/62.6** |

The bold values highlight the best results for each metric.

network is capable of context-aware prediction in bin-picking tasks, which is robust to outliers.

To demonstrate the effectiveness of our dataset, we trained the same suction network from the previous work [22] with our dataset. Table III displays the comparison results. **Real** refers to a network trained with the real-world dataset from [22], which is identical to the one in Table II. **Ours-Subset** represents the same model trained with our dataset, which has the same limited number of images as [22]. **Ours-Full** represents our complete dataset. The model trained with the subset of our dataset performed worse than the model trained with the real-world dataset, indicating that the domain gap affects significantly with the limited amount of data. As a result, when using the full amount of our dataset, the model still performs well in the real world with context-awareness, demonstrating the effectiveness of our dataset.

### D. Ablation Study

Firstly, we evaluate how each element in the pipeline affects the performance. We evaluated three elements: camera randomization, collision model, and random noise in both the simulation and the real world. We trained our network with the same dataset except for the target element, then evaluated the networks using a metric based on [31]. Let $P_{ij}$ be the precision, $R_{ij}$ be the recall, and $F_{ij}$ be the F-measure, and the metrics are defined as follows:

$$P_{ij} = \frac{|c_i \cap g_j|}{|c_i|}, R_{ij} = \frac{|c_i \cap g_j|}{|g_j|}, F_{ij} = \frac{2P_{ij}R_{ij}}{P_{ij} + R_{ij}}$$

here $g_j$ is the ground-truth pixels for positive candidates, and $c_i$ is the predicted pixels by a network. As shown in Table IV, we can achieve ideal performance by training with the dataset

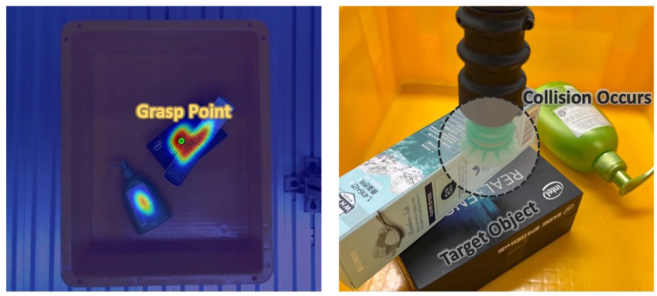


Fig. 9. **Failure case of the model trained without collision evalution.** (Left) The grasp candidate has been sampled near the edge of the object. (Right) Collision occurred between the gripper and the object when grasping the target object.

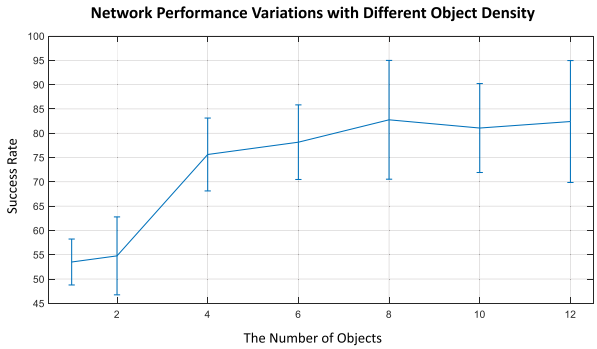


Fig. 10. **Ablation on the number of objects in the dataset.** The results show that dataset density affects network training. We selected data by density and collected it to train with constant density.

that incorporates all our adopted methods. The network trained without camera randomization has difficulty finding affordable grasp candidates in a bin. The absence of camera randomization results in a lack of ability to be generalized to the camera's location. The network trained without random noise performs better than the full method-based network in precision. The absence of random noise leads to exclusively predicting networks on real data. This negatively impacts the model, leading to low recall and, consequently, a low F-measure. The model trained without collision evaluation is observed to struggle with considering a collision during grasping, as shown in Fig. 9.

Next, we examined how the number of objects used for a single scene influences the performance. We trained the network with a discrete number of objects to achieve this. The result in Fig. 10 shows that the number of objects that comprise a scene has a pronounced effect on the network capability. Interestingly, when the number of objects is larger than 8, its impact on the network performance becomes less significant. This result justifies our selection to set 8 as the maximum number of objects.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.
Authorized licensed use limited to: Sungkyunkwan University. Downloaded on December 14,2023 at 04:36:49 UTC from IEEE Xplore. Restrictions apply.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE V
COMPARISON OF DIFFERENT DETECTION APPROACHES

| Model | Success Rate(%) | Clear Rate(%) | Self-Pick Rate(%) |
|---|---|---|---|
| CoAS-Net-Reg | 88.6 | 81.3 | 41.7 |
| CoAS-Net | **92.4** | **98.2** | **99.1** |

The bold values highlight the best results for each metric.

Finally, we assess our assumption that competitive classification among grasp candidates and the background facilitates context-aware grasping. To evaluate this, we modified our model to output grasp candidates based on regression (CoAS-Net-Reg) and compared it with our classification-based model (CoAS-Net). Table V shows our comparison results. It demonstrates that the regression-based model has achieved an affordable Success Rate and Clear Rate. However, the Self-Pick Rate was notably low, indicating that the model struggles to discriminate between target objects and the background.

## VI. CONCLUSION AND FUTURE WORK

We proposed a domain-randomized synthetic suction grasp dataset for robotic applications. To annotate large-scale data, we developed a pipeline that analytically finds suction candidates considering a bin-picking context and converts data formatting from cartesian space to image plane. Also, we further improved the suction prediction network from the previous research. We validated our method with real-world robot experiments and showed context-aware capability in bin-picking scenarios. However, a simulation-based dataset has a corner case when dealing with transparent objects. The depth information of those objects is often inaccurate in the real world. Most simulators, including Isaac Sim, struggle to replicate this phenomenon, so our dataset might lack reality. Possible future directions may include [32]: simulation of an active stereo depth camera by mimicking the IR stereo patterns and computing the depth as introduced in recent work.

## REFERENCES


[1] M. Sundermeyer, Z. C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 699–715.

[2] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1521–1529.

[3] K. Kleeberger, C. Landgraf, and M. F. Huber, "Large-scale 6D object pose estimation dataset for industrial bin-picking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 2573–2578.

[4] S. D'Avella, A. M. Sundaram, W. Friedl, P. Tripicchio, and M. A. Roa, "Multimodal grasp planner for hybrid grippers in cluttered scenes," *IEEE Robot. Automat. Lett.*, vol. 8, no. 4, pp. 2030–2037, Apr. 2023.

[5] R. Hachiuma and H. Saito, "Pose estimation of primitive-shaped objects from a depth image using superquadric representation," *Appl. Sci.*, vol. 10, no. 16, pp. 5442, Aug. 2020, doi: 10.3390/app10165442.

[6] N. Somani, C. Cai, A. Perzylo, M. Rickert, and A. Knoll, "Object recognition using constraints from primitive shape matching," in *Proc. Int. Symp. Vis. Comput.*, 2014, pp. 783–792.

[7] A. Remus, S. D'Avella, F. Di Felice, P. Tripicchio, and C. A. Avizzano, "i2c-net: Using instance-level neural networks for monocular category-level 6D pose estimation," *IEEE Robot. Automat. Lett.*, vol. 8, no. 3, pp. 1515–1522, Mar. 2023.

[8] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A survey on learning-based robotic grasping," *Curr. Robot. Rep.*, vol. 1, pp. 239–249, 2020.

[9] C. Hernandez et al., "Team Delft's robot winner of the Amazon Picking Challenge 2016," in *Proc. Robot World Cup*, Springer, 2016, pp. 613–624.

[10] D. Morrison et al., "Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7757–7764.

[11] J. Mahler et al., "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robot. Sci. Syst.*, 2017, pp. 1–8.

[12] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1–8.

[13] J. Mahler et al., "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaau4984.

[14] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1357–1364, Apr. 2019.

[15] T. H. Bui et al., "Deep learning based 6-DoF antipodal grasp planning from point cloud in random bin-picking task using single-view," *IEEE Robot. Automat. Lett.*, vol. 8, no. 8, pp. 5196–5203, Aug. 2023.

[16] K. Tung, J. Su, J. Cai, Z. Wan, and H. Cheng, "Uncertainty-based exploring strategy in densely cluttered scenes for vacuum cup grasping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 3483–3489.

[17] P. Jiang et al., "Learning suction graspability considering grasp quality and robot reachability for bin-picking," *Front. Neurorobot.*, vol. 16, 2022, Art. no. 806898.

[18] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4/5, pp. 705–724, 2015.

[19] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 769–776.

[20] H. S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11441–11450.

[21] H. Cao, H. S. Fang, W. Liu, and C. Lu, "SuctionNet-1Billion: A large-scale benchmark for suction grasping," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8718–8725, Oct. 2021.

[22] A. Zeng et al., "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *Int. J. Robot. Res.*, vol. 41, no. 7, pp. 690–705, 2022.

[23] M. Gilles, Y. Chen, T. R. Winter, E. Z. Zeng, and A. Wong, "Metagraspnet: A large-scale benchmark dataset for scene-aware ambidextrous bin picking via physics-based metaverse synthesis," in *Proc. IEEE 18th Int. Conf. Automat. Sci. Eng.*, 2022, pp. 220–227.

[24] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.

[25] J. Tobin et al., "Domain randomization and generative models for robotic grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3482–3489.

[26] K. Bousmalis et al., "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4243–4250.

[27] S. James et al., "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12627–12637.

[28] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[29] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Proc. IEEE Int. Conf. Adv. Robot.*, 2015, pp. 510–517.

[30] A. Kasper, Z. Xue, and R. Dillmann, "The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics," *Int. J. Robot. Res.*, vol. 31, no. 8, pp. 927–934, May 2022.

[31] A. Dave, P. Tokmakov, and D. Ramanan, "Towards segmenting anything that moves," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 1493–1502.

[32] Q. Dai et al., "Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 374–391.


IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.
Authorized licensed use limited to: Sungkyunkwan University. Downloaded on December 14,2023 at 04:36:49 UTC from IEEE Xplore. Restrictions apply.