IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 1, 2021, accepted August 30, 2021, date of publication September 7, 2021, date of current version September 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110795

Robotic Waste Sorter With Agile Manipulation and Quickly Trainable Detector

TAKUYA KIYOKAWA, (Member, IEEE), HIROKI KATAYAMA, YUYA TATSUTA, JUN TAKAMATSU, (Member, IEEE), AND TSUKASA OGASAWARA, (Member, IEEE)
Nara Institute of Science and Technology (NAIST), Ikoma, Nara 630-0192, Japan

Corresponding author: Takuya Kiyokawa (kiyokawa.takuya@is.naist.jp)

This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP14004 and Project JPNP20012.

ABSTRACT Owing to human labor shortages, the automation of labor-intensive manual waste-sorting is needed. The goal of automating waste-sorting is to replace the human role of robust detection and agile manipulation of waste items with robots. To achieve this, we propose three methods. First, we provide a combined manipulation method using graspless push-and-drop and pick-and-release manipulation. Second, we provide a robotic system that can automatically collect object images to quickly train a deep neural–network model. Third, we provide a method to mitigate the differences in the appearance of target objects from two scenes: one for dataset collection and the other for waste sorting in a recycling factory. If differences exist, the performance of a trained waste detector may decrease. We address differences in illumination and background by applying object scaling, histogram matching with histogram equalization, and background synthesis to the source target-object images. Via experiments in an indoor experimental workplace for waste-sorting, we confirm that the proposed methods enable quick collection of the training image sets for three classes of waste items (*i.e.,* aluminum can, glass bottle, and plastic bottle) and detection with higher performance than the methods that do not consider the differences. We also confirm that the proposed method enables the robot quickly manipulate the objects.

INDEX TERMS Robotics and automation, robot vision systems, computer vision, recycling, machine learning, object detection.

I. INTRODUCTION

In the context of long-standing human-labor shortages, the automation of various tasks by robots is ever more in demand. The automation of sorting container and packaging waste is an urgent example, and several related studies have been conducted worldwide [1]–[4]. Among the general waste articles produced by society, container and packaging wastes are dominant. Thus, many companies have been tackling this issue [5], [6].

Normally, vast amounts of unsorted recyclable waste are gathered at a collection site and manually sorted into designated boxes or transport lanes according to categories (*e.g.,* aluminum can, glass bottle, or plastic bottle). The goal of automating this process is to replace the human role of detection and manipulation of the waste items with robots.

The associate editor coordinating the review of this manuscript and approving it for publication was Ze Ji.

A key difficulty is agility, because conveyor transportation speeds should be as high as possible, owing to the large volumes of waste to be sorted. Another challenge is to robustly detect short lifecycle objects that are dirty on the surface or deformed and/or damaged.

With this in mind, we construct a robotic waste-sorting system (see Fig. 1) with the robust detection and agile manipulation needed for recycling factories. In this study, to achieve agile waste-sorting manipulation, we first propose a combined manipulation method using graspless *push-and-drop* and *pick-and-release* manipulation. Second, we propose a robotic training dataset collection system to automatically capture images and annotate them for training a deep–learning (DL)-based waste detector. We attempt to improve the robustness by applying a domain adaptation method to the collected dataset.

DL-based object detectors [7]–[13] can infer the location and category of objects having a variety of appearances

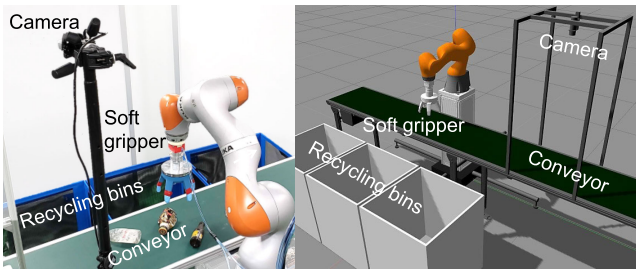


FIGURE 1. Configuration of proposed automated waste-sorting system.

in images. However, massive training datasets [14]–[17] are required, owing to the many parameters to be optimized [18]. With recent decreases in product lifecycles, unknown waste items frequently appear at the sorting factories. Thus, we must quickly update the training dataset with new waste images for fine-tuning.

To quickly create an object-image dataset using our system, a target object is placed on an automatic rotating stage and imaged from multiple viewpoints using a hand–eye robot arm shown, as in Fig. 2. The robot arm and rotating stage are automatically controlled while capturing images. Our previously proposed automatic annotation method [19] using augmented-reality (AR) marker detection [20] is applied to captured images. To train the DL-based waste detector, we place the collection-target object on the rotating table for image capture. However, we do not have to manually annotate the images. Using automatic annotation methods of this nature, prior experiments have achieved a six-class object detection [21].

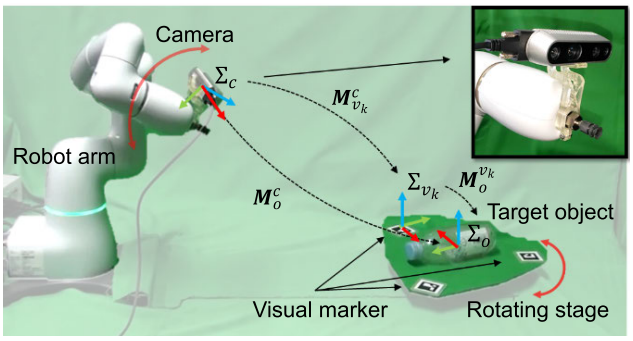


FIGURE 2. Robotic training dataset collection system that facilitates image capturing and automatically annotates labels and bounding boxes.

Although object images in the real-world can be easily provided, they often appear differently from items found in the working environment. Thus, detection performance can decrease when collecting images without consideration for adaptation methods.

The waste-sorting workplace exists in an indoor environment for this study. Thus, it can be fixed in terms of illumination and background. We propose methods to reduce the differences easily and effectively for such conditions.

This study focuses on two domain differences in terms of illumination and background between the dataset collection environment in Fig. 2 and the waste-sorting environment in Fig. 1. First, we adjust the object size in the image to be as close as possible to the real one in the waste-sorting scene. Subsequently, we apply histogram matching (HM) to images using a red–green–blue (RGB) color space to reduce illumination differences. Based on our qualitative observations for RGB histograms of the object images captured in the waste-sorting environment, we apply histogram smoothing for the collected images to further make the RGB histogram resemble the destination images. Furthermore, to reduce the differences of background conditions, we use background-synthesized and histogram-matched images as the training images.

The contributions of this study are threefold.

1) In the proposed sorting manipulation method by push-and-drop, the time required for the manipulation of one object is about 1.9 s faster than pick-and-release.
2) The proposed robotic training dataset collection system composed of a hand–eye robot arm, a rotating stage, and visual markers enables agile object-image capturing from multiple viewpoints. The time required for the proposed automatic collection is 12.3 s: 99.1% faster than prior methods.
3) As a benefit of proposed object-image dataset adaptation method, we achieve improved waste-detection accuracy. We further propose the addition of a small real-world dataset captured in the waste-sorting scene to the domain-adapted dataset. Training with this dataset achieves a detection accuracy of 79%, which is 39% higher than using the original one that lacks domain adaptation and real-world images.

II. RELATED WORK

A. ROBOTIC WASTE SORTER

To achieve an agile robotic sorter for a huge volume of waste, previous studies sorted items transported on a conveyor using suction grippers for quick grasping and manipulation [6], [22]. Graspless [23], [24], prehensile pushing [25], and non-prehensile manipulation [26], [27] methods, like our push-and-drop technique, have not been applied thus far. Therefore, the feasibility of push-and-drop has remained untested until now, notwithstanding that such manipulations using robotic hands are reasonable methods of agile manipulation.

Conventional automatic sorting systems are based on different types of sensors (*e.g.,* optical [28]–[30] and thermal techniques [31], [32]). Mao *et al.* [33] proposed a classifier using a convolutional neural network to classify an RGB object image that included one waste item. Furthermore, DL-based algorithms using RGB and RGB-depth (RGBD) sensors have been used to detect and segment individual waste items from a densely cluttered pile [6], [22], [34]–[36].

B. GENERATING A TRAINING DATASET FOR A DL-BASED DETECTOR

Deep convolutional neural networks can automatically discover the needed representations for object detection and classification from large datasets in a manner similar to that of the human visual cortex [37]. Although larger datasets enable robust detection and classification of waste items having diverse appearances, the construction of such datasets demands an enormous amount of time and effort. Binyan *et al*. [34] used 47,988 images of recyclable waste on a conveyor for training and testing a deep neural–network model. 3,999 images were originally collected, and additional ones were augmented via flipping and scaling the collected images. Bai *et al*. [38] achieved garbage recognition with small errors using training datasets comprising 40,000 training and 7,000 testing images grouped into six classes: five garbage and one non-garbage. Zhihong *et al*. [39] used 1,480 images only for the detection of a glass bottle on a conveyor transporting various waste items. These are distinguished from automatic collection methods like ours.

DL-based vision systems are fast and can detect vast categories of objects. However, as mentioned, the cost of manual image annotation remains very high. To tackle this, two major efforts to easily collect large datasets are under way. One approach includes (1) data augmentation to enrich image datasets for improving the generalizability of DL models, and the other deals with (2) the simplification of labor-intensive annotation processes to increase the number of datasets with reduced human intervention. This study applies both types.

In the research of (1), Takahashi *et al*. [40] applied random-image cropping and patching to improve classification accuracy. Zhong *et al*. [41] applied random erasing to reduce the risk of over-fitting and made the model robust to occlusion. They randomly changed pixel intensities within the selected region of an arbitrary size. Cubuk *et al*. [42] proposed a method of automatically searching for data augmentation policies directly from a dataset (*AutoAugment*). Each policy expresses several choices and orders of possible augmentation operations, wherein each operation is an image–processing function (*e.g.,* translation, rotation, or color normalization). Lim *et al*. [43] proposed *FastAutoAugment*, an improved policy extraction method that is significantly faster than the original AutoAugment, which requires thousands of graphical-processing-unit hours, even for small datasets.

In the research of (2), to make human annotation easier, effective and easy-to-use annotation tools [44], [45] were proposed. However, with these, humans still spent too much time on annotation. For example, polygonal annotations for instance segmentation were conducted via interactive image-region mouse clicks by human annotators. Ling *et al*. [46] proposed a graph-convolutional network, *Curve-GCN*, to automatically predict the vertices of instances in the images. An annotator can choose any wrong control points and move them onto the correct object boundary. Only its immediate neighbors will be re-predicted based on manual annotation. Benenson *et al*. [47] designed software capable of correcting wrong annotations by clicking on images. Based on the corrective clicks, the segmentation mask for the annotation was automatically updated. These human-in-the-loop polygonal annotations take only a few seconds for each image, but they also require corrective clicks for the vertices, owing to the need for annotation quality assurance.

Another interesting approach is the use of an RGBD sensor [48] and visual markers [49], [50] to automatically segment objects from the background. These approaches are like ours. However, in the previous approaches, the automatic collection of multi-view object images and their domain adaptations were out-of-scope. Our robotic training dataset collection system of multi-view images gives the dataset variety and quantity and is useful when training the garbage detector to handle various appearances. Image adaptation methods of reducing the differences of domains are necessary to enable faster image collection.

C. DOMAIN ADAPTATION FOR DL-BASED VISION SYSTEM

Despite the many ideas explored, the predominant datasets were built by humans using bounding boxes or polygonal masks [14]–[17]. Our proposed method can automatically annotate object images without human intervention. Because there are differences in object appearance between the dataset collection environment shown in Fig. 2 and the waste-sorting environment shown in Fig. 1, the collected dataset using the robotic collection system could not be directly used to train the waste detector.

Domain adaptation is a specific scenario in transfer learning that can be used to effectively remove domain differences. Domain adaptation has been shown to be effective for the transfer learning of models in different computer vision tasks, including image classification [51], object recognition [52], object detection for indoor kitchen scenes [53], outdoor scenes [54], water-colors [55], and semantic segmentation [56].

Georgakis *et al*. [53] tackled an issue like ours. To automatically generate image datasets that emulate real environments, they superimposed two-dimensional images of textured object models into images of real indoor environments reflecting a variety of locations and scales. They verified the efficacy of a seamless cloning (SC) method to mitigate the effects of changes in illumination and contrast. They also verified an object–scaling method that used the depth of the selected position of a real household environment.

In this study, we tackle the issue of domain adaptation for a collected waste-image dataset ourselves so that it can be adapted to a real waste-sorting problem. For this reason, we create a waste dataset using images of 33 aluminum cans, 33 glass bottles, and 33 plastic bottles.

We also strongly support the efficacy of domain adaptation for the waste-sorting environment. In particular, we evaluate

more methods to mitigate the changes of object-size appearance, image illumination, contrast and background.

III. AUTOMATICALLY GENERATING TRAINING DATASET

This section first describes the proposed robotic training dataset collection system using a small hand–eye robot arm and an automatic rotating stage. Next, we explain the methods for reducing the differences of the illumination and the background. The object appearances differ between dataset-collection and waste-sorting environments.

For domain adaptation, we consider how to match the original domain of the generated training dataset to that of the target domain of the waste-sorting environment.

A. MULTI-VIEWPOINT OBJECT IMAGE ACQUISITION

Fig. 2 shows our robotic training dataset collection system that includes a small hand–eye robot arm and a controllable rotating stage. Using the small hand–eye robot arm equipped with an RGB camera, we collect images from multiple viewpoints by moving the robot arm to capture a target object placed on the automatic rotating stage.

An RGBD camera is used for both object-image dataset collection and the robot vision capability of the proposed robotic waste-sorting system, because we minimize the effects of the camera in the detection experiments. Depth information is not used to generate the training dataset, but the same camera as the waste-sorting environment is. The white balance and the exposure of the camera are fixed during image dataset collection and robot experiments.

Fig. 3 shows the proposed dataset collection procedure with its automatic annotation method [19]. Fig. 4 shows the process for the object region extraction shown in Fig. 3. To extract the region in consideration of the outline blur caused by anti-aliasing, alpha matting is applied to the captured image. We used *large–kernel matting*, a fast method for high quality matting [57]. We used a Python library *PyMatting* [58] for alpha matting. Trimap is used for alpha matting and is automatically generated by applying dilation processing to the image that the markers are removed.

The generated approximate object mask is according to the estimated object pose related to the camera. If coordinate systems for the hand–eye camera, k-th visual marker, and the object are Σ_c, Σ_{v_k}, and Σ_o, the transformation, $\boldsymbol{M}_o^c$, from Σ_c to Σ_o shown in Fig. 2 is calculated as

$$\boldsymbol{M}_o^c = \boldsymbol{M}_{v_k}^c(\boldsymbol{r}_{v_k}^c, \boldsymbol{\theta}_{v_k}^c)\boldsymbol{M}_o^{v_k}, \tag{1}$$

where $\boldsymbol{M}_{v_k}^c$, $\boldsymbol{M}_o^c$, and $\boldsymbol{M}_o^{v_k}$ are transformations from Σ_c to Σ_{v_k}, from Σ_c to Σ_o, and from Σ_{v_k} to Σ_o, respectively. The translation vector, $\boldsymbol{r}_{v_k}^c$, and the rotation vector, $\boldsymbol{\theta}_{v_k}^c$, are estimated from the detected visual markers.

B. OBJECT IMAGE SCALING FOR CONSISTENCY OF GEOMETRY

Object image scaling is applied to the collected images to reduce the differences in appearance caused by the varying distances between the camera and the object. To accomplish this, the size of the object placed on the automatic rotating stage is adjusted to be fitted to the size of the object placed on the conveyor in the waste-sorting scene.

As shown in Fig. 5, the visual markers on the marker board in both images are detected. For geometric consistency of the dataset images, the size of the object region in the image is adjusted according to the scaling parameter, k, estimated as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \tag{2}$$

$$k = \frac{d_t}{d_s}, \tag{3}$$

where d_s and d_t are the distances from the camera coordinate system, Σ_c, to the marker board coordinate systems, Σ_s and Σ_t, of the source and target images.

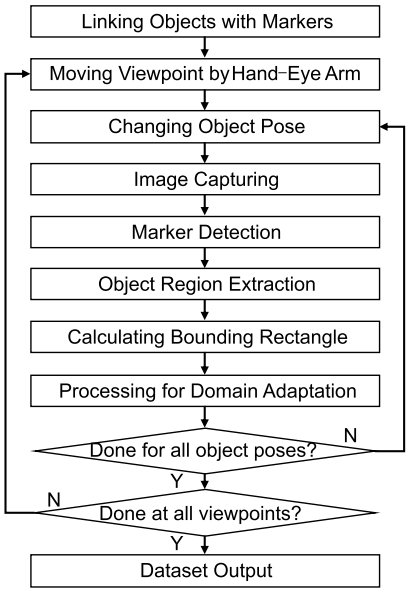


FIGURE 3. Flow of the image dataset collection by the proposed robotic training dataset collection system.

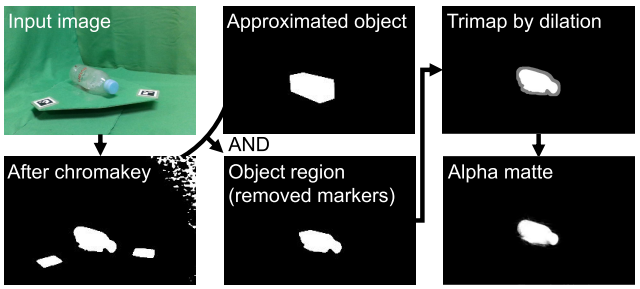


FIGURE 4. Extracted object region (bottom center) by applying AND operation with the image after chromakey (bottom left) and the image showing the approximated object (top center) in the estimated pose based on marker detection, automatically generated trimap (top right), and the generated alpha matte (bottom right) used for alpha matting.

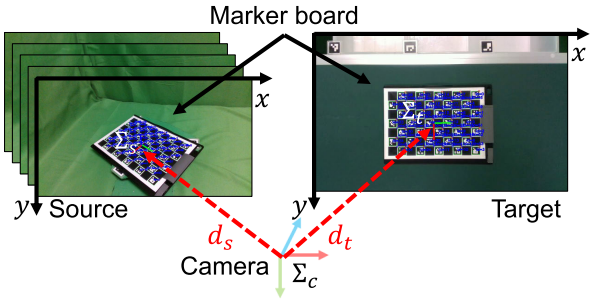


FIGURE 5. Illustration of calculating the scaling parameter, k, representing the distances from the camera to the center of the rotating stage used for dataset collection and one point of the conveyor in the waste-sorting scene.

C. COLOR MATCHING AND BACKGROUND SYNTHESIS FOR CONSISTENCY OF ILLUMINATION

For the color matching proposed in this study, histograms of pixel values in the RGB color space are calculated from an object-area image captured in the waste-sorting environment, and HM [59] is performed. The generated image has a distribution similar to the illumination in the waste-sorting environment. Thus, the difference in the illumination is reduced.

The cumulative distribution, $cdf_s(i)$ $(i = 1, 2, .., l)$, of the input image's histogram, $\boldsymbol{h}_s$, is matched to the cumulative distribution, $cdf_t(i)$, of target image's histogram, $\boldsymbol{h}_t$. Each cumulative distribution function (CDF) is calculated as

$$cdf_s(i) = \sum_{j=1}^{i} \frac{h_s(j)}{N_s}, \quad cdf_t(i) = \sum_{j=1}^{i} \frac{h_t(j)}{N_t}, \tag{4}$$

where l is the number of bins in the histogram, and N_t and N_s are the number of pixels in each image.

To extract the boundary between the object and the background, using the automatically generated trimap, we apply alpha blending [60] to the image at the time of image collection to combine it with the background image captured in the waste-sorting environment. Then, we apply HM to the image of only the area within the bounding box of the object.

Images used for applying HM to the image of the plastic bottle are shown in Fig. 6. The leftmost image shows the source image, the image to the right of the source image is a target image as the destination, the image to the right of the target image shows a result of the HM, and the rightmost image shows the image after EQ. We use *Contrast Limited Adaptive Histogram Equalization (CLAHE)* [61] to smooth jaggy histogram distributions by the EQ. Finally, background-synthesized and histogram-matched images are used to train the waste detector.

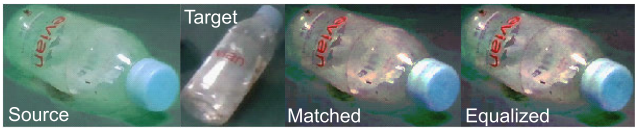


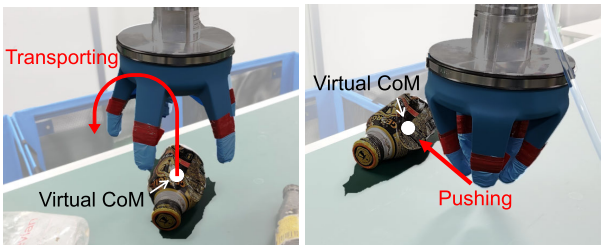
FIGURE 6. HM applied to a plastic bottle image. "Source" and "Target" indicate the input image and the image with the target histogram to match. "Matched" and "Equalized" are the images after application of HM and after application of the EQ of Matched, respectively.

IV. AGILE HANDLING OF CONVEYED OBJECTS

A. TWO TYPES OF SORTING MANIPULATION

In this study, one sorting task is designated to move a waste item from the conveyor to an adjacent recycling box. Regarding the waste-sorting robot, quickness is required alongside sorting accuracy. Therefore, two types of sorting manipulation are performed according to the desired waste detection results.

As shown in Fig. 7, the two types are (1) manipulation by picking and releasing and (2) manipulation by pushing and dropping. A gripper with one degree of freedom can perform these manipulations.



(a) Pick-and-release (b) Push-and-drop

FIGURE 7. Illustration of key scenes in the two proposed types of manipulation (*i.e.,*(a) pick-and-release and (b) push-and-drop) to sort the waste (*i.e.,* aluminum can, glass bottle, and plastic bottle) on a conveyor belt.

In manipulation (1), the object is grasped by the five fingers of the soft gripper so that the estimated point near the object center (*virtual CoM*) becomes to the grasping center. The gripper pose is adjusted to enable grasping along the straight line on the estimated object silhouette passing through the virtual CoM, which is illustrated in Fig. 7(a). Then, the robot arm trajectory is planned and generated so that it approaches the target object and departs from it in its fixed grasping pose.

In manipulation (2), the soft gripper pushes the object around the virtual CoM using a straight-line trajectory and drops the object into the target recycling bin (Fig. 7(b)). The trajectory of a robot arm is generated to push the object in a direction that connects the virtual CoM to the front center of the recycling bin.

In contrast to pick-and-release, push-and-drop does not require grasping. The average time in the 10 trials to finish the push-and-drop operation was 3.3 s, although the time in the case of the pick-and-release operation took 5.2 s. Thus, using the push-and-drop operation as much as possible shortens the combined manipulation time.

B. SELECTIVE EXECUTION AND IMPLEMENTATION

Algorithm 1 describes the entire algorithm used to select a feasible manipulation from the two available types. The algorithm is based on the following policy considering the

time constraints of feasible manipulation to handle the waste items conveyed.

1) We adapt a first-in-first-out strategy to determine how to manipulate the frontmost waste item on the conveyor.
2) Push-and-drop is primary performed if possible because of its quickness.
3) If both types of manipulation are determined to be infeasible based on the time constraints, the target waste item is ignored (shown as "**continue**" in Algorithm 1).

The positions indicated by the parameters are drawn in Fig. 8. We define the width and height of the object silhouette in the image as s_x and s_y, respectively. We define the x- and y-axial distances from the center of the target object's silhouette to the recycling bin's center line as l_{bx} and l_{by}, respectively. These are calculated from the object's silhouette mask image and the results of a detected marker attached to the recycling bins. l_e is the x-axial distance from the object to the image right end.

t_{pd} and t_{pp} are the time variables representing the times required for push-and-drop and pick-and-release manipulation, respectively. These are determined in preliminary experiments to measure the manipulation time in all points on the conveyor and the premeasured gripper open–close time.

v_{pd} and v_c are the speed variables for the push-and-drop manipulation and transportation of conveyor. These are preset parameters (*i.e.,* the speed of the push-and-drop manipulation and the transportation speed of the conveyor are constant for the waste-sorting).

Here, we consider following three time constraints to select the manipulation type in Algorithm 1.

1) The inequality, $(s_x/2)/v_c > l_{by}/v_{pd}$, holds in the cases where target waste item is far from bins and too small to push. This indicates that it is impossible to execute the push-and-drop operation.
2) The inequality, $t_{pp} < l_e/v_c$, holds in the state that the target waste item cannot be manipulated in the pick-and-release manipulation time. This indicates that it is too late to start the pick-and-release.
3) The inequality, $t_{pd} < l_{bx}/v_c$, holds in the state that the target waste item is conveyed to a position where the robot cannot push it into the target bin. This indicates that it is too late to start the push-and-drop.

Detected objects are assigned silhouettes extracted using the input-depth image. Using the known distance from the RGBD sensor to the conveyor, we create the silhouette mask as the object regions on the conveyor that are closer to the camera than the conveyor. The centroids of the silhouette pixel areas are estimated for each object as the virtual CoM.

All parameters are estimated from RGB and depth images of one frame to maintain fast computations for the waste detector. The virtual CoM from the 2.5-dimensional RGBD image reflect an ill-posed problem. We assume that an object's shape can be approximated as a solid revolution with a uniformly distributed mass. The underlying assumption

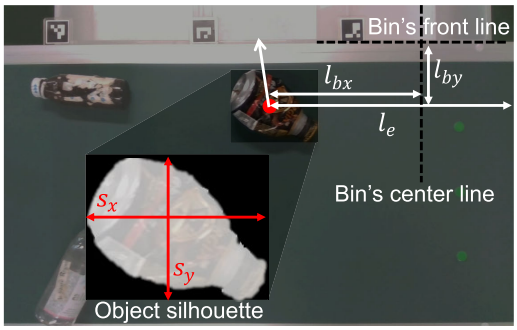


FIGURE 8. Parameterization for the sorting manipulation selection algorithm.

Algorithm 1 Sorting Manipulation Selection

Input: An image of one place on the conveyor

procedure SELECT-MANIPULATION-TYPE
  Recognize waste items and markers in the image
  Calculate s_x and s_y from the object silhouette mask
  Calculate l_{bx}, l_{by} and l_e from the recognition results
  if $(s_x/2)/v_c > l_{by}/v_{pd}$ **then**
    if $t_{pp} < l_e/v_c$ **then**
      continue
    Execute pick-and-release on robot
    continue
  if $t_{pd} < l_{bx}/v_c$ **then**
    continue
  Execute push-and-drop on robot

enables us to estimate the virtual CoM using the object silhouette extracted from the RGB and depth image. The virtual CoM is calculated as the centroid of a grayscale image.

Using the estimated common parameters, the unique parameters (*i.e.,* grasp position and pushing direction) are calculated based on the methods mentioned in Section IV-A. The arm motions for picking, releasing, pushing, and dropping and their connecting trajectories are planned and generated using *MoveIt!* [62].

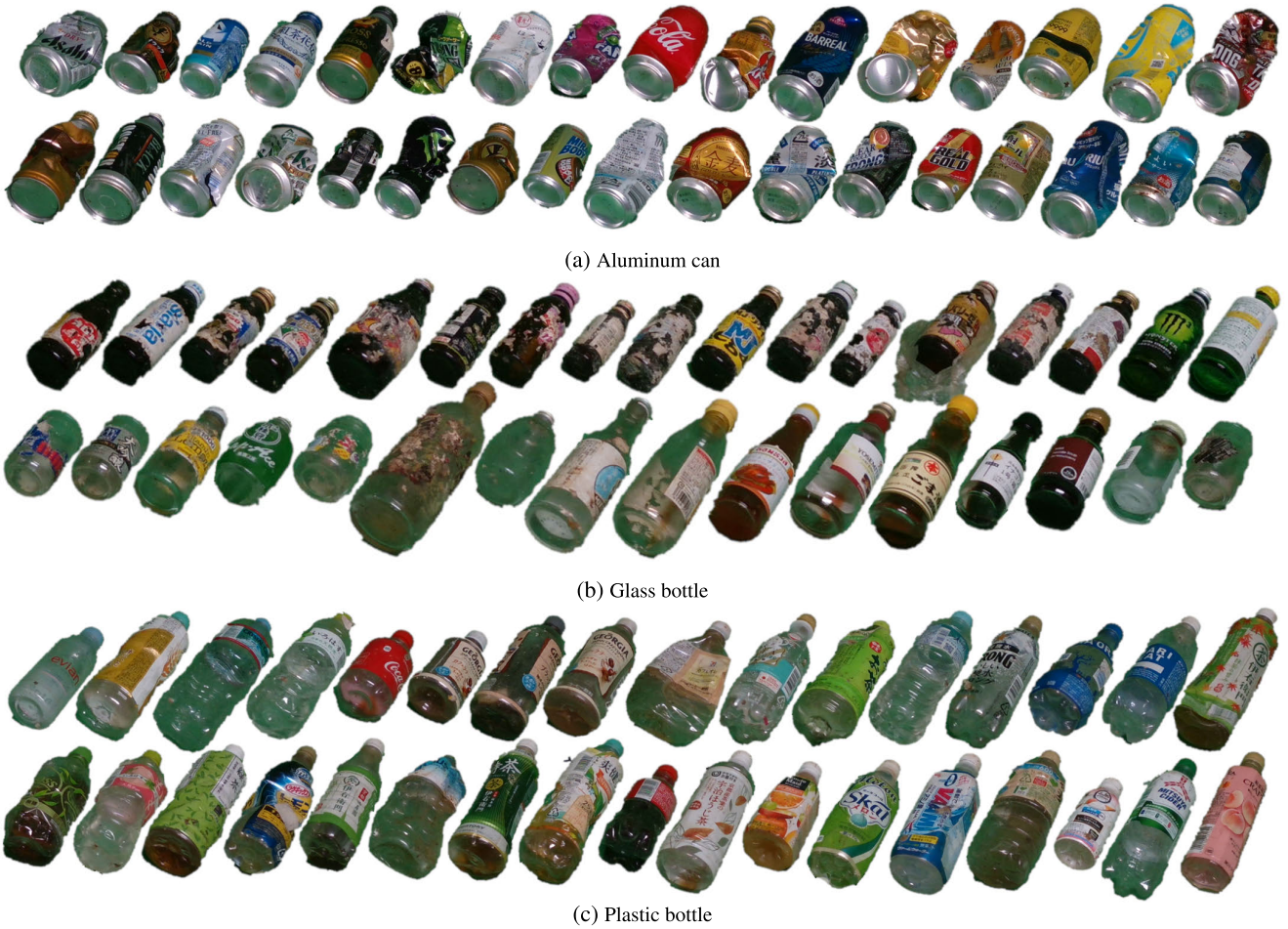
V. EXPERIMENTS

A. OUTLINE OF EXPERIMENTS

First, to evaluate the quickness of the proposed robotic training dataset collection system, we compare the collection time by the proposed dataset generation with the collection time by the manual dataset generation (Section V-C). Furthermore, we show the accuracy of the annotation results (Section V-D).

Second, we show the similarity of the images applied adaptation methods with those captured from the real scene (Section V-E). To evaluate the performance of the waste detector trained with the image dataset that applied the proposed adaptation method having the highest similarity, we show the detection results of the sorting-target waste item by the detector (Section V-F).

Third, to evaluate the feasibility of the proposed manipulation methods, we discuss the success rate of the sorting



(a) Aluminum can

(b) Glass bottle

(c) Plastic bottle

FIGURE 9. The waste samples of (a) aluminum cans; (b) glass bottles; and (c) plastic bottles used in the experiments.

manipulation and the average time required by each manipulation method (*i.e.,* pick-and-release and push-and-drop) (Section V-G).

B. EXPERIMENTAL SETUP

As shown in Fig. 2, we used *COBOTTA* (DENSO WAVE INCORPORATED) with *RealSense D435* (Intel Inc.) as the small hand–eye robot and used *OSMS-60YAW* (SIGMAKOKI CO.,LTD.) as the rotating stage. We used *ArUco*, an AR library [63], [64] to detect AR markers for registering the object pose of each object image collected using the proposed robotic training dataset collection system. This object poses were used to generate an approximate object mask. ArUco was used to specify the positions of the recycling bins in the waste-sorting experiments.

An evaluation experiment of the waste-sorting was performed using the robotic waste-sorting system shown in Fig. 1. In this paper, we experimented with the minimum configuration of one camera and one manipulator.

We used a robot arm, *LBR iiwa 14 R820* (KUKA), and a soft gripper, *SOFTmatics* (Nitta Corporation), whose five fingers were covered with a soft material to handle the many sharp objects present in a recycling facility. We used an RGBD camera employing active infrared stereo, the same RealSense camera as the camera used in the dataset collection. The camera can measure depth information with high sensitivity, even for translucent objects and those having complex shapes and opacity, which are common in container and packaging waste. The target waste samples contained 33 different aluminum cans, 33 glass bottles, and 33 plastic bottles, as shown in Fig. 9. The target objects were sampled from the waste samples in a recycling factory for industrial waste items.

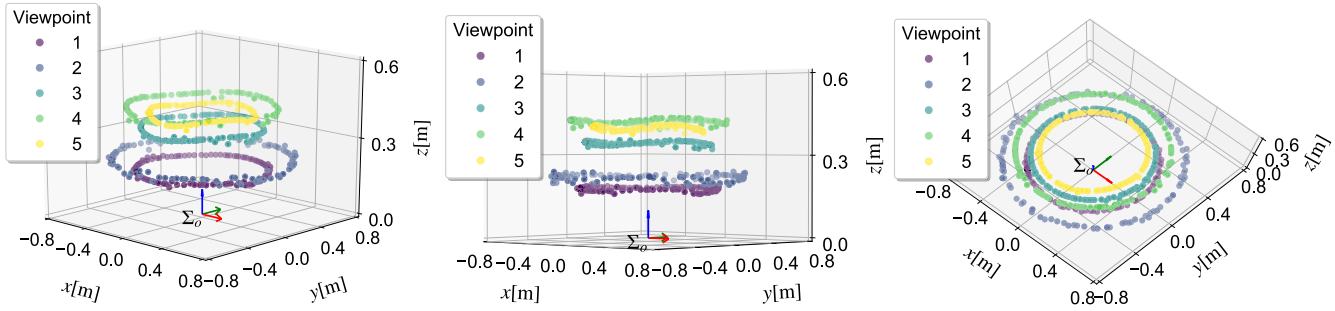
C. IMAGE DATASET COLLECTION TIME

To demonstrate the effectivity of the robotic training dataset collection system compared with the collection methods previously proposed in [19], [21], this section describes the results of the comparison of times needed to collect image datasets.

Table 1 shows the average time needed to collect 100 images and the method (automatic or manual) for three processes: object replacement, image acquisition, and annotation.

TABLE 1. Average time to collect 100 image datasets. Automatic or manual is shown next to the time measured.

	Type of automatic dataset collection					
	Single marker [19]		Multiple markers [21]		Proposed	
Necessary process	Time [s]	Automatic / Manual	Time [s]	Automatic / Manual	Time [s]	Automatic / Manual
Object replacement	900	Manual	5232	Manual	2.05	Manual
Image acquisition		Manual		Manual	10.2	Automatic
Annotation	444	Automatic	-	Automatic	-	Automatic
Total	1344	-	5232	-	12.3	-



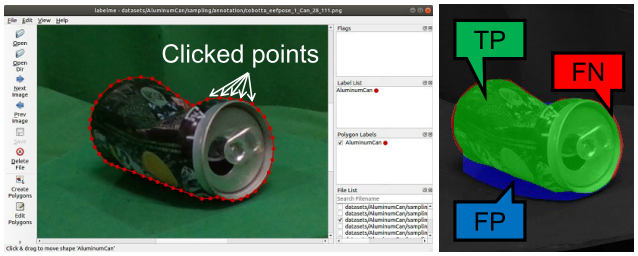
(a) Bird's-eye view (b) Top view (c) Side view

FIGURE 10. Variations of viewpoints taken by the proposed robotic training dataset collection system. Σ_o shows the object coordinate system shown in Fig. 2. Viewpoint IDs from 1 to 5 represent the five viewpoint patterns adjusted by changing the joint pose of the small robot arm.

In the first proposed method using a single marker [19], an object with a marker attached directly was actually used in a real-work environment. In this method, humans manually change the object types and the poses of the objects. Therefore, it took a relatively long time of 900 seconds. In addition, the annotation was automatically performed by image processing after all the image capturing was completed. As the result, it was 444 seconds for 100 images. In the extended method using multiple markers [21], object replacement and image acquisition were performed manually as in the single marker method. Combined with the time required for these manual operations and the time required for automatic annotation that was being processed in parallel, it took the longest time of 5232 seconds.

The proposed dataset collection was completed in 12.3 s on average for 100 images. The results indicate that the time required for collecting the training set was incredibly shortened compared with the other methods. The viewpoints taken by the proposed robotic training dataset collection system are widely scattered as shown in Fig. 10, suggesting that a dataset having large variations can be collected in a short period.

The total time required to collect the training set comprising 59,400 (120 object-orientation patterns × 5 viewpoint patterns × 99 objects) images captured with a green screen was about 111 min. Such a short collection time enables us to easily increase the number of training sets when the target waste increases or changes.



(a) Clicked points in annotation tool (b) Evaluation

FIGURE 11. Visualization of manual annotations needed to generate ground truth to evaluate the proposed automatic object region extraction. Left image shows the window of the annotation tool (labelme) and several annotated image points. Right image shows the parameterization of the evaluation results of the automatic object region extraction.

D. QUANTITATIVE EVALUATION OF ANNOTATIONS

To evaluate the annotation results, the automatically object-extracted image is compared with the manually annotated image, as shown in Fig. 11. Using a manual annotation tool named *labelme*[1] and by clicking several points on the object contour in images, the images are annotated by humans for evaluation.

Based on true-positive (TP), false-positive (FP), and false-negative (FN) results, as shown in Fig. 11, we calculated the intersection over union (IoU), precision, recall, and

[1]https://github.com/wkentaro/labelme

F-score [65] as

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{5}$$

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8}$$

Table 2 shows the results of the object region extraction in the training set.

TABLE 2. Results of our object region extraction in our automatic dataset generation. Each element shows mean±standard deviation of IoU [%], Precision [%], Recall [%], and F-score [%]. The mean values are calculated from randomly selected 33 images of each object category in the three categories.

Object	Metric [%]			
	IoU	Precision	Recall	F-score
Aluminum can	71±16	71±17	98±1.7	81±11
Glass bottle	67±17	69±18	96±4.4	79±13
Plastic bottle	77±14	78±15	97±2.2	86±9.8

In all trials and categories, the mean values of precision rated around 70%. The mean values of recall were rated higher than 95% and with smaller standard deviations than those of precision. These results suggest that there were some false predictions. However, there were few missed pixels in the ground truth. As a result, the calculation provides a low IoU with a mean of F-score of 80%.

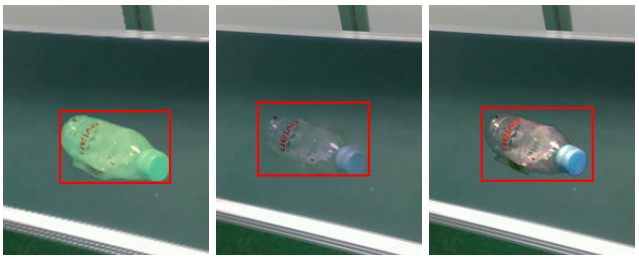
E. EFFECT OF REDUCING DIFFERENCES FROM WASTE-SORTING SCENE

In this section, we discuss the effect of the proposed method of reducing the differences from the waste-sorting scene. To evaluate the performance of the proposed color adjustment, we compare it with two other methods.

The first unifies color reproducibility by applying *color correction* (CC) using *ColorChecker Passport Photo* (X-Rite, Inc.), which has a panel of 24 industry-standard color-reference chips. The CC in this study is based on a color-transfer method that can adjust the colors in an image to match a target-image color profile [66]. The goal is to create a transform so that, when it is applied to the values of every pixel in a source image (the left of Fig. 14), it returns values mapped to a target image (the right of Fig. 14) profile [67].

The other is an easy-to-use image-rendering SC method [68] used in the fields of computer graphics [69] and computer vision [70]–[72]. SC was once used to create a photomontage by pasting an image region onto a new background using Poisson image editing [68]. Fig. 12 shows the results of CC, SC, and HM. The parameters needed in the methods described in this section are organized in the Table 3.

Fig. 13 shows histograms in the RGB color space of the images in Fig. 6. The histogram distributions in the RGB color space of the target image (Target) and the converted image (Matched) are visually similar after applying HM.



(a) Color correction (b) Seamless cloning (c) Histogram matching

FIGURE 12. Comparison of appearances of synthesized images: (a) synthesized images with CC applied; (b) SC applied; and (c) HM applied.

TABLE 3. Necessary images for adaptation methods.

Method	Necessary images
Image scaling	One image pair including a calibration board
Background synthesis (BS)	One background image
Color correction (CC)	One image pair including a color checker
Seamless cloning (SC)	One background image
Histogram matching (HM)	One object image captured in a real scene

To conduct a quantitative evaluation, the distance between two histogram distributions were evaluated using *earth–mover's distance* (EMD) [73] and *Bhattacharyya distance* (BD) [74].

To evaluate the image similarity with the object image captured in the real scene, we calculated the histogram distributions of the four types, which include the original, BS, BS+CC, SC, and BS+HM.

The effects of the proposed method, BS+HM+EQ, were compared to those of BS+HM, HM, and BS, which are derivatives of the proposed method. We also compared the comparative methods BS+CC and SC as other color adjustment methods.

The calculated values of the EMD and BD in the RGB color space are shown in Table 4 and Table 5. To compare the images to the object images captured in the real scene, we used those cropped by the bounding boxes as shown in Fig. 12 in red boxes.

The result of the CC shows that the EMD and BD are larger compared with the result of HM. In the case of the CC, the homography transformation matrix in the RGB color space must be calculated using source and target images, including the color checker shown in Fig. 14. On the other hand, because the source shown in Fig. 6 is converted to become similar to the target shown in Fig. 6, for HM, a higher similarity was achieved.

The calculated values of the EMD and BD suggests that the similarity of the image was largely improved by applying HM, including the area translucent to the back of the object or the plastic bottle's cap. This is because the appearance as

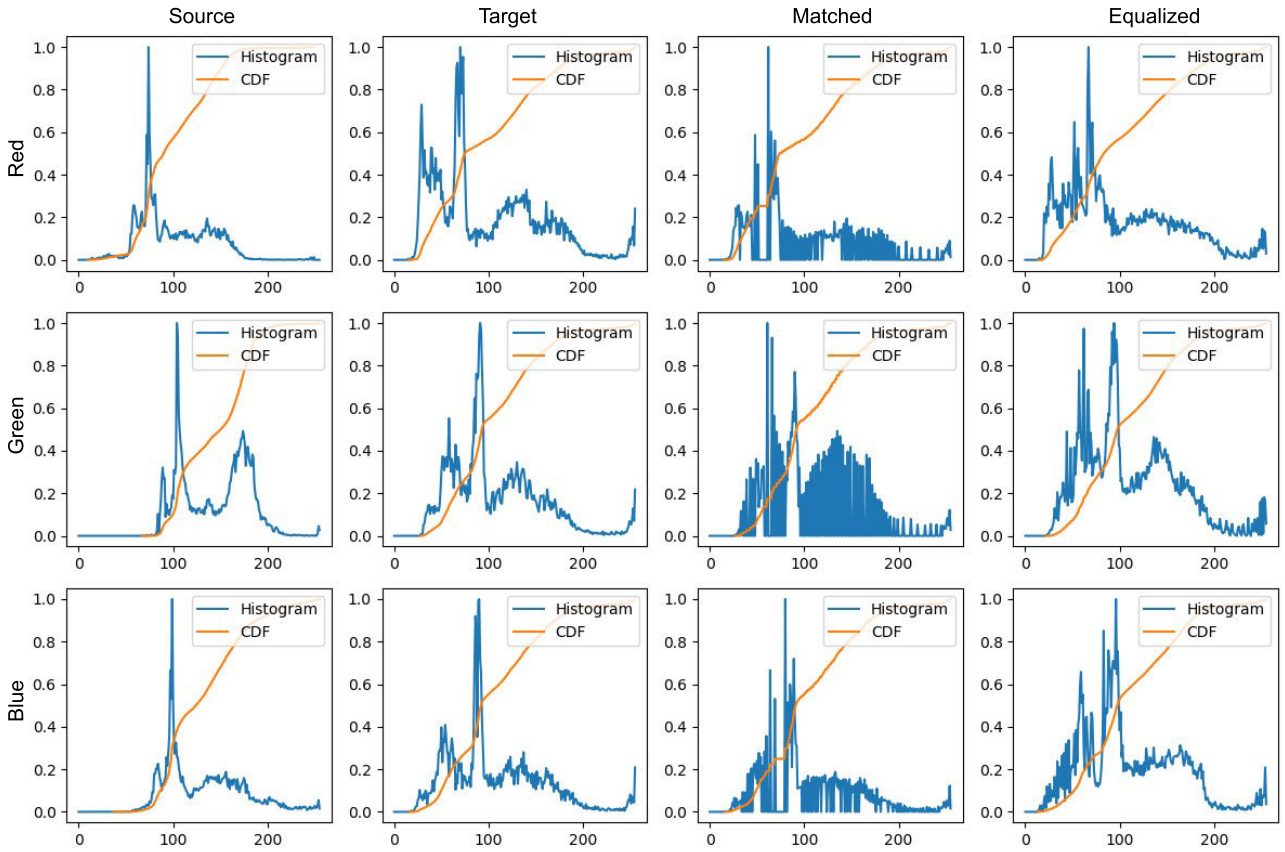


FIGURE 13. RGB histograms and CDFs of the image applied with HM. The graphs are histograms of the RGB color space of the four images on Fig. 6. The title names correspond to the names displayed in each image shown in Fig. 6.

TABLE 4. Calculated values of EMD between the reference image (captured in the real scene) and processed images in the training sets. The histogram comparison was conducted in the RGB color space. The values that indicate the highest similarity are shown in bold.

Training set	Object category		
	Aluminum can	Glass bottle	Plastic bottle
Original	5.86e-1	8.45e-1	1.59e0
BS	7.55e-1	9.41e-1	1.88e0
BS+CC	8.65e-1	6.50e-1	1.77e0
SC	2.62e0	2.10e0	4.82e0
BS+HM	7.36e-3	5.04e-3	5.25e-3
BS+HM+EQ†	**6.27e-3**	**4.67e-3**	**3.96e-3**

† Proposed method in this study.

TABLE 5. Calculated values of BD between the reference image (captured in the real scene) and processed images in the training sets. The histogram comparison was conducted in the RGB color space. The values that indicate the highest similarity are shown in bold.

Training set	Object category		
	Aluminum can	Glass bottle	Plastic bottle
Original	0.381	0.425	0.400
BS	0.436	0.476	0.445
BS+CC	0.403	0.419	0.428
SC	0.454	0.493	0.493
BS+HM	0.430	0.445	0.467
BS+HM+EQ†	**0.220**	**0.245**	**0.193**

† Proposed method in this study.

improved to approximate the target image. It also suggests that the BS+HM+EQ provided the highest similarity.

F. DETECTION ACCURACY

Table 6 shows mean values of detection accuracy for the three target-object categories. As an accuracy metric, we calculated the mean F-score when the IoU threshold was set to 0.5. We also calculated the F-score using detection results with a confidence value higher than 0.5. Using a training dataset automatically generated by the proposed method, detection was performed using a waste detector with a trained model

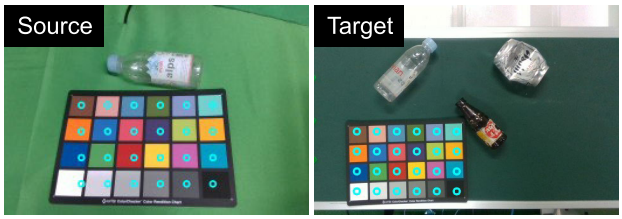
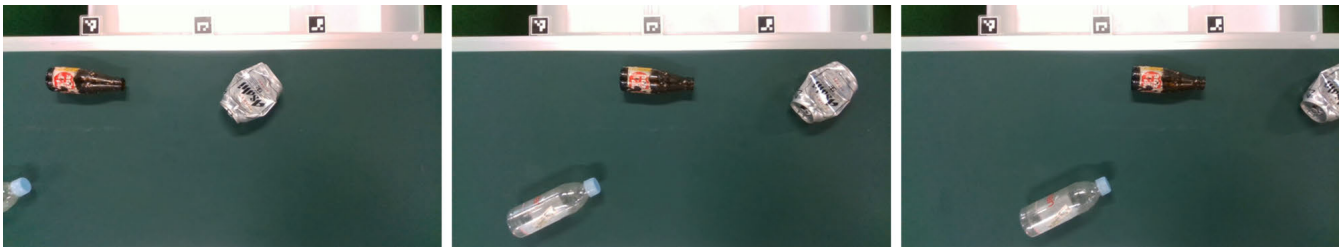
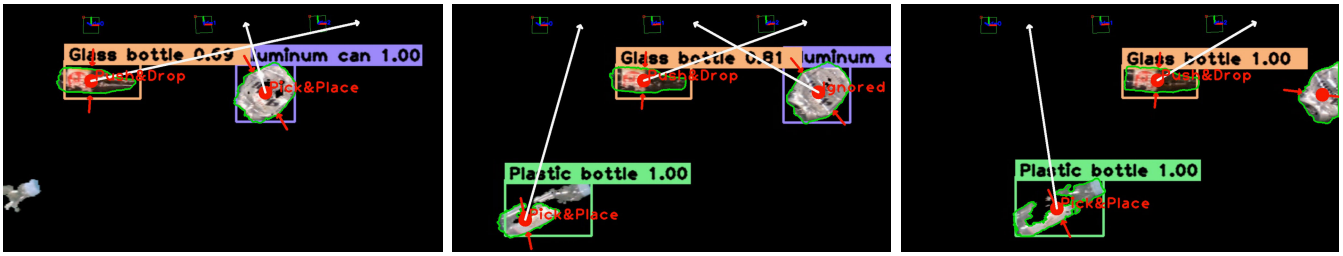


FIGURE 14. Images used for estimating the color homography transformation matrix for CC.

of the *single shot multibox detector* (SSD) [9]. SSD is a general object detector with a convolutional neural–network



(a) Image sequence recorded in the waste-sorting environment



(b) Visualization of detection results of our waste detector

FIGURE 15. (a) image captured in the waste-sorting scene and (b) the image drawn from the detection results of the waste detector.

TABLE 6. F-scores of waste category detection using DL-based waste detector trained using each training set [%]. Mean indicates the mean values of F-score in the three object categories.

Training set	Object category			Mean
	AC*[a]	GB*[b]	PB*[c]	
1. Original	43	76	2.0	40
2. BS	57	45	34	45
3. BS+CC	19	51	23	31
4. SC	14	51	10	25
5. BS+HM	17	59	13	30
6. BS+HM+EQ	22	64	28	38
7. Mixed (1,2,6)	54	53	31	46
8. Real with 7†	72	89	75	79

*[a] AC is the abbreviation of aluminum can.
*[b] GB is the abbreviation of glass bottle.
*[c] PB is the abbreviation of plastic bottle.
† Proposed method in this study.

architecture that learns different anchor boxes. Fig. 15 shows the detection results.

The original shows the result of using 59,400 (120 object-orientation patterns × 5 viewpoint patterns × 99 objects) images captured with a green screen shown in Fig. 2. BS, BS+CC, SC, BS+HM, and BS+HM+EQ show image training sets subjected to BS only, BS and CC, SC, BS and HM; and BS+HM with EQ, respectively. Mixed show the training set that we randomly collected images from the three sets of Original, BS, and BS+HM+EQ. All the training sets include 59,400 images.

The last set (Real with 7) is a mixed training set that includes the Mixed and 80 images recorded in the real scene, as shown in Fig. 16. The conveyor moves at a constant speed in one direction. Thus, if the image acquisition frequencies of the camera are aligned, the object positions in the images can be shifted at a constant interval. Therefore, if we apply manual annotation to only the images of the first frames appearing in the video, we can obtain the image sequence annotated by moving the bounding boxes. We collected the 80 images from two videos in the waste-sorting scene in this manner. To improve the quickness of video annotation, in a future work, we plan to use automatic video annotation methods [75], [76].

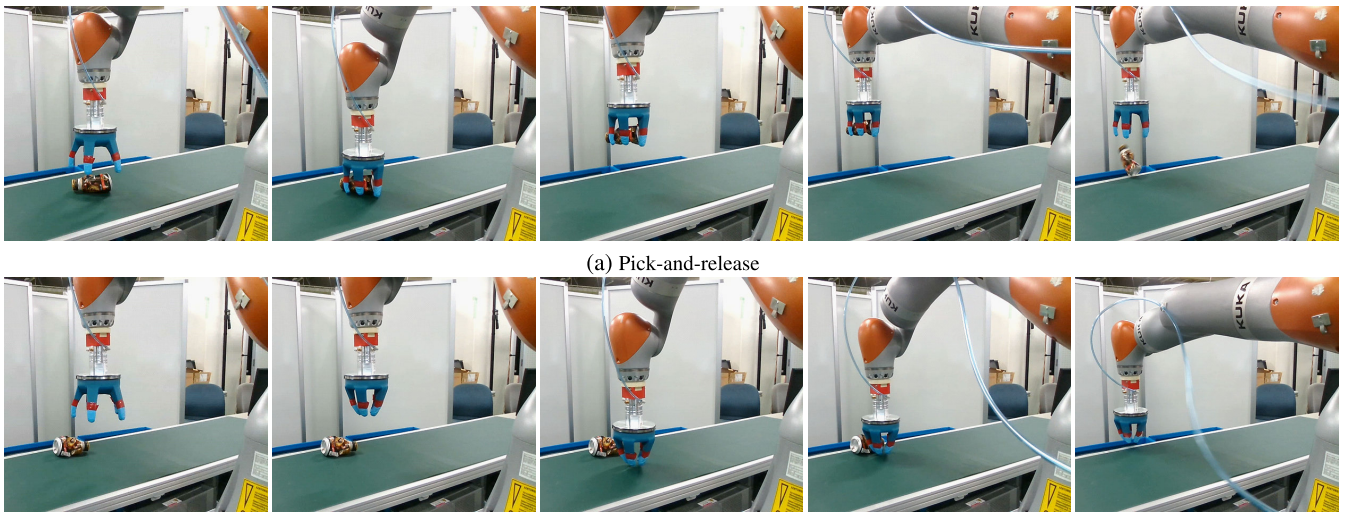
The detection results shown in Table 6 suggest that Mixed provided the highest accuracy of training without images recorded in the waste-sorting environment in the training sets except Real with 7. Therefore, our experimental results demonstrate that the accuracy of the waste detector can be improved by applying the aforementioned object scaling, HM with EQ and BS to reduce the differences from the waste-sorting environment. Surprisingly, the detector with the BS-only dataset showed the almost same accuracy as did Mixed. The comparison for these detection accuracies should be done in the future using the backgrounds of various waste-sorting environments.

By adding the small real-world image dataset including the 80 images, we achieved the highest accuracies of detection, even when the number of items in the dataset was small. The small real-world image dataset not only significantly outperformed the other in terms of accuracy, but the images were also quickly collected. The time needed to capture a video was about 1 min, and the time needed to annotate only six objects in the six images was about 2 min. This was about 3 min total.

G. FEASIBILITY OF ROBOTIC WASTE SORTER

Fig. 17 shows the process by the sorting robot. In this study, the virtual CoM was calculated as the centroid of the

FIGURE 16. Real-world image sequences annotated by humans. *Manual indicates manually annotated bounding boxes. We conducted manual annotation to the video frame in which a new object first appeared. The other images were automatically annotated based on the constant speed of the conveyor and the camera framerate.



(a) Pick-and-release

(b) Push-and-drop

FIGURE 17. Two types of manipulation implemented to a waste-sorting robot.

object silhouette extracted from the depth image when the object was viewed from directly above (red dots shown in Fig. 15(b)). The grasp positions during pick-and-release were determined as a straight line on the object silhouette passing through the center of mass perpendicular to the principal axis, as drawn by the red arrows in Fig. 15(b).

While sorting manipulation of the waste items by a robot, we evaluated whether the robot succeeded in sorting the waste detected on the conveyor. The success rates of 10 trials of each sorting manipulation for each object category are shown in Table 7. The results indicate that the pick-and-release operation provided a highly accurate sorting manipulation compared with push-and-drop. The average time taken in the 10 trials to finish the push-and-drop operation was 3.3 s, although the time in the case of the pick-and-release operation took 5.2 s. Our algorithm reduced the time required for manipulation by simplifying the manipulation process.

TABLE 7. Results of the sorting manipulation. Each element shows success rate [%] in each 10 trials.

Method	Object category			Mean
	AC^{*a}	GB^{*b}	PB^{*c}	
Pick-and-release	80	70	60	70
Push-and-drop	60	50	50	57

^{*a} AC is the abbreviation of aluminum can.
^{*b} GB is the abbreviation of glass bottle.
^{*c} PB is the abbreviation of plastic bottle.

As examples of failures in the pick-and-release, we confirmed cases where a large object did not fit in the grasping area, cases where the grasping failed due to an error of the estimated virtual CoM, and cases where the released object by placing motion did reach the target bin.

First, we must consider another grasping method based on the gripper's grasping area and target object size. In the case of container and packaging waste, there are many large slender objects. Thus, we need another grasping method in which the thinnest part can be sandwiched between two of the five fingers. Second, we require object segmentation [77], [78] or foreground extraction [79]–[81] methods that use color information, because the silhouette sometimes cannot be generated, owing object–region extraction errors by the depth image. Third, the target garbage item was not put into the target bin, because the acceleration of the robot arm sent it flying over top. We should not slow the robot arm motion even for this case, owing to the low agility of manipulation. We instead require a particular a release motion by a robot arm that accounts for acceleration. The pick-and-place for dynamic objects [82] could also achieve highly accurate sorting.

As an example of failures in the push-and-drop, we first confirmed cases where the gripper's fingers could not make good contact with the sides of the target objects. To ensure reliable contact for pushing, we must consider the waste-item shape and the orientation of the gripper.

Second, we confirmed a case in which the target object overshot the bin and another where the target object was too heavy to exit the conveyor. There was also a case in which the target object only rotated after pushing. Therefore, we need to generate a pushing motion based on the target object weight and shape [83].

VI. DISCUSSION ON FUTURE ISSUES

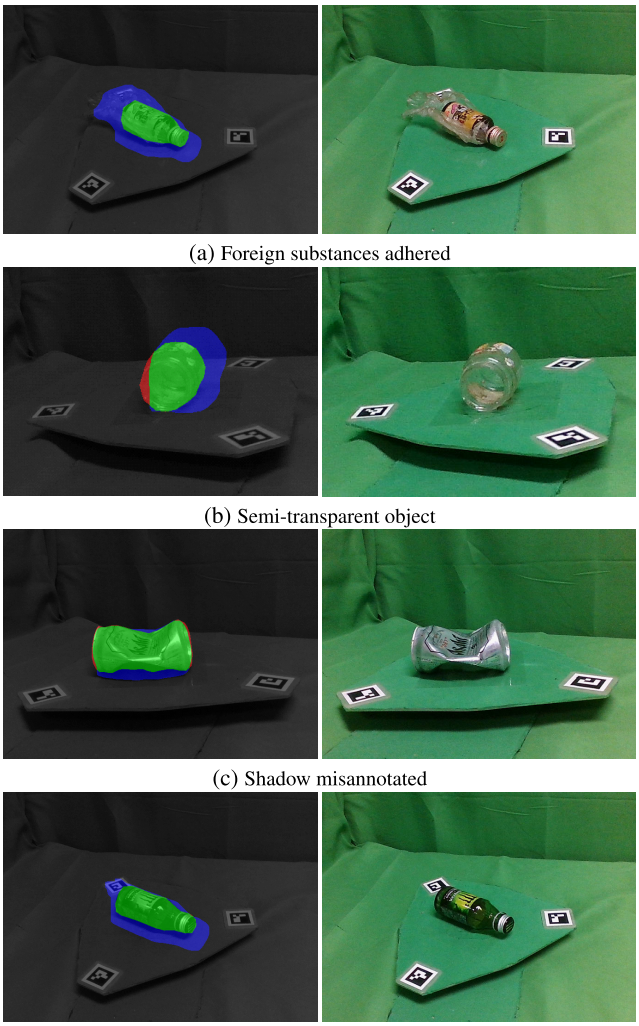
A. ENSURING HIGH CONSISTENCIES OF ILLUMINATION AND GEOMETRY

The purpose of this study, apart from reducing the time required for dataset collection, was to achieve a highly accurate detector. Within this context, for the consistency of illumination, we proposed a method that matches only the luminance distribution information of the image without considering a camera-response function [84] and the distribution of the light source [85], [86] in the different environments. In reality, these optical models must be considered when obtaining more realistic images that are similar to real-world ones. However, estimation methods requiring less labor are needed.

In terms of geometric consistency, in this study, only the distance from the camera to the object was considered. However, a 3D model is needed to transform the geometry more precisely. One idea for generating realistic images via a 3D model requires free viewpoint image synthesis based on 3D shape reconstruction methods, such as *Space carving* [87], and a geometric registration and an alignment using an RGBD video [88].

B. PRECISE ANNOTATION

Fig. 18 shows the four cases that had difficulty annotating collected images, especially for cases of difficult object-region extraction. The problematic images shown in Fig. 18 include an object adhered to foreign substances, a semi-transparent object, a shadow under the object, and a green object.



(a) Foreign substances adhered

(b) Semi-transparent object

(c) Shadow misannotated

(d) Green object

FIGURE 18. Problematic images difficult to annotate. The coloring in each left image is the same as that of Fig. 11.

The foreign substances shown in Fig. 18(a) needs to be removed from the target object, because the waste detector is not designed to recognize this part. Consequently, the waste-sorting robot cannot grasp and push the part. Fig. 18(b) shows a misannotated semi-transparent object. For the automatic annotation, we could in the future use another method that does not rely exclusively on optical information. As shown in Fig. 18(c), because it may be difficult to distinguish a boundary from a shadow, object region extraction may fail. In a future work, it will be necessary to improve the algorithm so that it is robust to shading by referring to illumination estimation methods [89], [90] and

DL-based shadow detection and removal methods [91], [92]. To avoid difficulty of region extraction caused by similar colors, as shown in Fig. 18(d), background coloring should be considered.

VII. CONCLUSION

In this study, to achieve an agile waste-sorting method, we first proposed two types of manipulation and a selection algorithm based on time constraints of the conveyed waste.

Second, to reduce the time required for capturing object images and annotations, we developed a robotic training dataset collection system using a small hand–eye robot and a rotating stage.

Third, to fill the gap between the generated image set and the one captured from a waste-sorting scene, we provided an image adaptation method.

In our experiment, we successfully automatically generated a training set using the proposed robotic training dataset collection system. To train the waste detector, we applied the proposed adaptation method, including histogram matching with histogram equalization, background synthesis, and object scaling of the collected dataset. Finally, the waste detector performed waste detection, and the robotic waste-sorting system successfully performed pick-and-release and push-and-drop in a real work environment.

The dataset collection time was reduced to at least 1% or less of the previously proposed automatic dataset collection method. We verified that the waste detector could detect target waste items (*i.e.,* aluminum cans, glass bottles, and plastic bottles) in a waste-sorting environment. As a result, the mean F-score for all objects was about 46%, and the accuracy was higher than the method lacking adaptation methods. We achieved a highly accurate detector trained with the training set, including the proposed dataset and a small dataset captured in a real scene. The mean value of the F-score in the three object categories was about 79%.

The robot successfully demonstrated the two types of manipulation at a success rate greater than 61%. The push-and-drop of the graspless manipulation more quickly performed the sorting manipulation for one object than did the pick-and-release method by 1.9 s. The average time taken in the 10 trials to finish the push-and-drop operation was 3.3 s, although the time in the case of the pick-and-release operation took 5.2 s.

As our future works, we consider other system configurations: the one system using multiple cameras to more accurately detect the waste items and the one system using other flexible endeffectors like brush-shaped gripper to more robustly manipulate the irregular-shaped waste items.

REFERENCES

[1] S. P. Gundupalli, S. Hait, and A. Thakur, "A review on automated sorting of source-separated municipal solid waste for recycling," *Waste Manage.*, vol. 60, pp. 56–74, Feb. 2017.

[2] K. Chahine and B. Ghazal, "Automatic sorting of solid wastes using sensor fusion," *Int. J. Eng. Technol.*, vol. 9, no. 6, pp. 4408–4414, Dec. 2017.

[3] N. Barrero, D. Galvis, and C. Martinez, "Industrial robots for waste separation tasks: An approach to industry 4.0 in Colombia," in *Proc. 9th Int. Conf. Prod. Res.-Americas*, 2018.

[4] R. Sarc, A. Curtis, L. Kandlbauer, K. Khodier, K. E. Lorber, and R. Pomberger, "Digitalisation and intelligent robotics in value chain of circular economy oriented waste management—A review," *Waste Manage.*, vol. 95, pp. 476–492, Jul. 2019.

[5] T. Gibson, "Recycling robots," *Mech. Eng.*, vol. 142, no. 1, pp. 32–37, Jan. 2020.

[6] T. J. Lukka, T. Tossavainen, J. V. Kujala, and R. Tapani, "ZenRobotics recycler-robotic sorting using machine learning," in *Proc. SBS*, 2014, pp. 1–8.

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[10] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *CoRR*, vol. abs/1804.02767, Apr. 2018.

[11] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2Det: A single-shot object detector based on multi-level feature pyramid network," in *Proc. AAAI*, 2019, pp. 9259–9266.

[12] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9626–9635.

[13] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[15] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.

[16] T.-Y. Lin, M. Maire, B. Serge, H. James, P. Perona, D. Ramanan, P. Dollár, and L. C. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[18] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[19] T. Kiyokawa, K. Tomochika, J. Takamatsu, and T. Ogasawara, "Fully automated annotation with noise-masked visual markers for deep-learning-based object detection," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1972–1977, Apr. 2019.

[20] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. IWAR*, 1999, pp. 85–94.

[21] T. Kiyokawa, K. Tomochika, J. Takamatsu, and T. Ogasawara, "Efficient collection and automatic annotation of real-world object images by taking advantage of post-diminished multiple visual markers," *Adv. Robot.*, vol. 33, no. 24, pp. 1264–1280, Dec. 2019.

[22] J. V. Kujala, T. J. Lukka, and H. Holopainen, "Classifying and sorting cluttered piles of unknown objects with robots: A learning approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 971–978.

[23] Y. Aiyama, M. Inaba, and H. Inoue, "Pivoting: A new method of graspless manipulation of object by robot fingers," in *Proc. IROS*, 1993, pp. 136–143.

[24] Y. Maeda and T. Arai, "Planning of graspless manipulation by a multifingered robot hand," *Adv. Robot.*, vol. 19, no. 5, pp. 501–521, Jan. 2005.

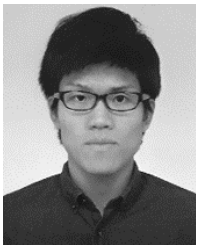
[25] N. Chavan-Dafle and A. Rodriguez, "Prehensile pushing: In-hand manipulation with push-primitives," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 6215–6222.

[26] M. T. Mason, "Progress in nonprehensile manipulation," *Int. J. Robot. Res.*, vol. 18, no. 11, pp. 1129–1141, 1999.

[27] K. M. Lynch and M. T. Mason, "Dynamic nonprehensile manipulation: Controllability, planning, and experiments," *Int. J. Robot. Res.*, vol. 18, no. 1, pp. 64–92, 1999.

[28] S. R. Ahmad, "A new technology for automatic identification and sorting of plastics for recycling," *Environ. Technol.*, vol. 25, no. 10, pp. 1143–1149, Oct. 2004.
[29] J. Huang, T. Pretz, and Z. Bian, "Intelligent solid waste processing using optical sensor based sorting technology," in *Proc. CISP*, 2010, pp. 1657–1661.
[30] H. Jull, J. Bier, R. Künnemeyer, and P. Schaare, "Classification of recyclables using laser-induced breakdown spectroscopy for waste management," *Spectrosc. Lett.*, vol. 51, no. 6, pp. 257–265, Jul. 2018.
[31] S. P. Gundupalli, S. Hait, A. Thakur, and A. Trivedi, "Classification of recyclables from e-waste stream using thermal imaging-based technique," in *Urbanization Challenges in Emerging Economies: Energy and Water Infrastructure*. American Society of Civil Engineers, 2018, pp. 67–78.
[32] S. P. Gundupalli, S. Hait, and A. Thakur, "Classification of metallic and non-metallic fractions of e-waste using thermal imaging-based technique," *Process Saf. Environ. Protection*, vol. 118, pp. 32–39, Aug. 2018.
[33] W.-L. Mao, W.-C. Chen, C.-T. Wang, and Y.-H. Lin, "Recycling waste classification using optimized convolutional neural network," *Resour., Conservation Recycling*, vol. 164, Jan. 2021, Art. no. 105132.
[34] L. BinYan, W. YanBo, C. ZhiHong, L. JiaYu, and L. JunQin, "Object detection and robotic sorting system in complex industrial environment," in *Proc. Chin. Automat. Congr. (CAC)*, Oct. 2017, pp. 7277–7281.
[35] H. Karbasi, A. Sanderson, A. Sharifi, and C. Pop, "Robotic sorting of used button cell batteries: Utilizing deep learning," in *Proc. IEEE Conf. Technol. Sustainability (SusTech)*, Nov. 2018, pp. 1–6.
[36] Z. Zhang, H. Wang, H. Song, S. Zhang, and J. Zhang, "Industrial robot sorting system for municipal solid waste," in *Intelligent Robotics and Applications*. Springer, 2019, pp. 342–353.
[37] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 44–436, May 2015.
[38] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Deep learning based robot for automatically picking up garbage on the grass," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 382–389, Aug. 2018.
[39] C. Zhihong, Z. Hebin, W. Yan, W. Yanbo, and L. Binyan, "Multi-task detection system for garbage sorting base on high-order fusion of convolutional feature hierarchical representation," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 5426–5430.
[40] R. Takahashi, T. Matsubara, and K. Uehara, "RICAP: Random image cropping and patching data augmentation for deep CNNs," in *Proc. ACML*, 2018, pp. 786–798.
[41] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI*, 2020, pp. 13001–13008.
[42] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 113–123.
[43] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast AutoAugment," in *Proc. NIPS*, 2019, pp. 6665–6675.
[44] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4930–4939.
[45] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 616–625.
[46] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, "Fast interactive object annotation with curve-GCN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5257–5266.
[47] R. Benenson, S. Popov, and V. Ferrari, "Large-scale interactive object segmentation with human annotators," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11700–11709.
[48] M. Suchi, T. Patten, D. Fischinger, and M. Vincze, "EasyLabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 6678–6684.
[49] D. De Gregorio, A. Tonioni, G. Palli, and L. Di Stefano, "Semiautomatic labeling for deep learning in robotics," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 611–620, Apr. 2020.
[50] S. Akizuki and M. Hashimoto, "Semi-automatic training data generation for semantic segmentation using 6DoF pose estimation," in *Proc. VISAPP*, 2019, pp. 607–613.
[51] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.
[52] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 999–1006.
[53] G. Georgakis, A. Mousavian, A. C. Berg, and J. Košecká, "Synthesizing training data for object detection in indoor scenes," in *Proc. RSS*, 2017, pp. 1–9.
[54] H.-K. Hsu, W.-C. Hung, H.-Y. Tseng, C.-H. Yao, Y.-H. Tsai, M. Singh, and M.-H. Yang, "Progressive domain adaptation for object detection," in *Proc. CVPR*, 2019, pp. 1–5.
[55] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5001–5009.
[56] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2507–2516.
[57] K. He, J. Sun, and X. Tang, "Fast matting using large kernel matting Laplacian matrices," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2165–2172.
[58] T. Germer, T. Uelwer, S. Conrad, and S. Harmeling, "PyMatting: A Python library for alpha matting," *J. Open Source Softw.*, vol. 5, no. 54, p. 2481, Oct. 2020.
[59] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Reading, MA, USA: Addison-Wesley, 2001, ch. 3, pp. 94–102.
[60] R. Szeliski, *Computer Vision: Algorithms and Applications*. London, U.K.: Springer-Verlag, 2011, ch. 2.
[61] K. Zuiderveld, "Contrast limited adaptive histograph equalization," in *Graphic Gems IV*. San Diego, CA, USA: Academic, 1994, pp. 474–485.
[62] S. Chitta, I. Sucan, and S. Cousins, "Moveit!" *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 18–19, Apr. 2012.
[63] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognit.*, vol. 51, pp. 481–491, Mar. 2016.
[64] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image Vis. Comput.*, vol. 76, pp. 38–47, Aug. 2018.
[65] Z. Wang, E. Wang, and Y. Zhu, "Image segmentation evaluation: A survey of methods," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5637–5674, Dec. 2020.
[66] J. C. Berry, N. Fahlgren, A. A. Pokorny, R. S. Bart, and K. M. Veley, "An automated, high-throughput method for standardizing image color profiles to improve image-based plant phenotyping," *PeerJ*, vol. 6, p. e5727, Oct. 2018.
[67] H. Gong, G. Finlayson, and R. Fisher, "Recoding color transfer as a color homography," in *Proc. BMVC*, 2016, pp. 17.1–17.11.
[68] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003.
[69] T. Kakuta, T. Oishi, and K. Ikeuchi, "Real-time soft shadows in mixed reality using shadowing planes," in *Proc. MVA*, 2007, pp. 195–198.
[70] Y. Mukaigawa, H. Miyaki, S. Mihashi, and T. Shakunaga, "Photometric image-based rendering for image generation in arbitrary illumination," in *Proc. ICCV*, 2001, pp. 652–659.
[71] I. Sato, M. Hayashida, F. Kai, Y. Sato, and K. Ikeuchi, "Fast image synthesis of virtual objects in a real scene with natural shadings," *Syst. Comput. Jpn.*, vol. 36, no. 14, pp. 102–111, 2005.
[72] F. Okura, M. Kanbara, and N. Yokoya, "Mixed-reality world exploration using image-based rendering," *J. Comput. Cultural Heritage*, vol. 8, no. 2, pp. 1–26, Mar. 2015.
[73] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, Nov. 2000.
[74] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.
[75] C. Vondrick and D. Ramanan, "Video annotation and tracking with active learning," in *Proc. NIPS*, 2011, pp. 28–36.
[76] I. Kavasidis, S. Palazzo, R. D. Salvo, D. Giordano, and C. Spampinato, "An innovative web-based collaborative platform for video annotation," *Multimedia Tools Appl.*, vol. 70, no. 1, pp. 413–432, May 2014.
[77] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[78] B. Xiong, S. D. Jain, and K. Grauman, "Pixel objectness: Learning to segment generic objects automatically in images and videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2677–2692, Nov. 2019.
[79] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut' interactive foreground extraction using iterated graph cuts," in *Proc. ACM SIGGRAPH*, 2004, pp. 309–314.
[80] B. Schölkopf, J. Platt, and T. Hofmann, "Dynamic foreground/background extraction from images and videos using random patches," in *Proc. NIPS*, 2006, pp. 929–936.
[81] H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure, "Robust foreground extraction technique using Gaussian family model and multiple thresholds," in *Proc. ACCV*, 2007, pp. 758–768.
[82] A. Cowley, B. Cohen, W. Marshall, C. J. Taylor, and M. Likhachev, "Perception and motion planning for pick-and-place of dynamic objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 816–823.
[83] J. Zhou, R. Paolini, J. A. Bagnell, and M. T. Mason, "A convex polynomial force-motion model for planar sliding: Identification and application," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 372–377.
[84] J. Takamatsu and Y. Matsushita, "Estimating camera response functions using probabilistic intensity similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[85] I. Sato, Y. Sato, and K. Ikeuchi, "Illumination from shadows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 3, pp. 290–300, Mar. 2003.
[86] K. Hara, K. Nishino, and K. Ikeuchi, "Multiple light sources and reflectance property estimation based on a mixture of spherical distributions," in *Proc. ICCV*, 2005, pp. 1627–1634.
[87] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *Int. J. Comput. Vis.*, vol. 38, no. 3, pp. 199–218, Jul. 2000.
[88] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5556–5565.
[89] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, "On the removal of shadows from images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 59–68, Jan. 2006.
[90] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios, "Illumination estimation and cast shadow detection through a higher-order graphical model," in *Proc. CVPR*, Jun. 2011, pp. 673–680.
[91] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, and D. Samaras, "Shadow detection with conditional generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4520–4528.
[92] L. Qu, J. Tian, S. He, Y. Tang, and R. W. H. Lau, "DeshadowNet: A multi-context embedding deep network for shadow removal," in *Proc. ICCV*, 2017, pp. 2308–2316.



TAKUYA KIYOKAWA (Member, IEEE) received the B.E. degree from the National Institute of Technology, Kumamoto College, Japan, and the M.E. and Ph.D. degrees in engineering from Nara Institute of Science and Technology, Japan, in 2018 and 2021, respectively. Since 2021, he has been with Osaka University, Japan, as a Specially-Appointed Assistant Professor, and with Nara Institute of Science and Technology, as a Specially-Appointed Assistant Professor. His current research interests include robot manipulation and robot vision for agile reconfigurable robotic systems toward agile manufacturing. He is a member of RSJ, JSME, SICE, and JSAI.

HIROKI KATAYAMA received the B.E. degree from Kumamoto University, Japan, and the M.E. degree from Nara Institute of Science and Technology, Japan, in 2020. He is currently working as a Chief Technology Officer with dTosh Inc. His research interest includes sound signal processing for robot audition.

YUYA TATSUTA received the B.E. degree from Saitama University, Japan, and the M.E. degree from Nara Institute of Science and Technology, Japan, in 2021. He is currently working with OKI Electric Industry Company Ltd. His research interests include biomimetics and soft robotics for dexterous manipulations. He is a member of RSJ.

JUN TAKAMATSU (Member, IEEE) received the Ph.D. degree in computer science from The University of Tokyo, Japan, in 2004. From 2004 to 2008, he was with the Institute of Industrial Science, The University of Tokyo. He was a Visiting Researcher with Microsoft Research Asia, in 2007. From 2008 to 2020, he was an Associate Professor with Nara Institute of Science and Technology, Japan. He was a Visitor at Carnegie Mellon University, in 2012 and 2013, and a Visiting Scientist at Microsoft, in 2018. He is currently working as a Senior Researcher with Applied Robotics, Microsoft. His research interests include robotics, including learning-from-observation, task and motion planning, feasible motion analysis, 3D-shape modeling and analysis, and physics-based vision. He is a member of RSJ.

TSUKASA OGASAWARA (Member, IEEE) received the Ph.D. degree from The University of Tokyo, Japan, in 1983. From 1983 to 1998, he was with the Electrotechnical Laboratory, Ministry of International Trade and Industry, Japan. From 1993 to 1994, he was a Humboldt Research Fellow with the Institute for Real-Time Computer Systems and Robotics, University of Karlsruhe, Germany. He joined Nara Institute of Science and Technology, Japan, in 1998, where he was a Professor with the Division of Information Science, from 1998 to 2020. He is currently an Executive Director and the Vice President of Nara Institute of Science and Technology. His research interests include human–robot interaction, dexterous manipulation, human modeling, and bio-inspired robotics. He is a member of RSJ.