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

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.

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 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].

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-preshensile 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. Huan 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. Rodrigo 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. 5. 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 \( \Sigma_c, \Sigma_{vk}, \) and \( \Sigma_o, \) the transformation, \( M_o^v \), from \( \Sigma_c \) to \( \Sigma_o \) shown in Fig. 2 is calculated as

\[
M_o^v = M_{vk}^o(r_{vk}^c, \theta_{vk}^c) M_{vk}^c,
\]

where \( M_{vk}^c, M_{vk}^o, \) and \( M_{vk}^{vo} \) are transformations from \( \Sigma_c \) to \( \Sigma_{vk}, \) from \( \Sigma_c \) to \( \Sigma_o, \) and from \( \Sigma_{vk} \) to \( \Sigma_o, \) respectively. The translation vector, \( r_{vk}^c, \) and the rotation vector, \( \theta_{vk}^c, \) 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

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

(2)

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

(3)

where $d_s$ and $d_t$ are the distances from the camera coordinate system, $\Sigma_c$, to the marker board coordinate systems, $\Sigma_s$ and $\Sigma_t$, of the source and target images.

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, $cdf_s(i) \ (i = 1, 2, \ldots, l)$, of the input image’s histogram, $h_s$, is matched to the cumulative distribution, $cdf_t(i)$, of target image’s histogram, $h_t$. Each cumulative distribution function (CDF) is calculated as

$$
cdf_s(i) = \sum_{j=1}^{i} \frac{h_s(j)}{N_s},
$$

$$
cdf_t(i) = \sum_{j=1}^{i} \frac{h_t(j)}{N_t},
$$

(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.
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.

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 \( v_e \) 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, \( (s_x/2)/v_e > l_{by}/v_{pd} \), 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, \( t_{pp} < l_{e}/v_e \), 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, \( t_{pd} < l_{bx}/v_e \), 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
Algorithm 1 Sorting Manipulation Selection

Input: An image of one place on the conveyor

1: procedure SELECT-MANIPULATION-TYPE
2: Recognize waste items and markers in the image
3: Calculate $s_x$ and $s_y$ from the object silhouette mask
4: Calculate $l_{bx}$, $l_{by}$ and $l_c$ from the recognition results
5: if $(s_x/2)/v_c > l_{by}/v_{pd}$ then
6: if $t_{pp} < l_c/v_c$ then
7: continue
8: Execute pick-and-release on robot
9: continue
10: if $t_{pd} < l_{bx}/v_c$ then
11: continue
12: Execute push-and-drop on robot

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 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 V-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 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 OMS5-60YW (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 effectiveness 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 I shows the average time needed to collect 100 images and the method (automatic or manual) for three processes: object replacement, image acquisition, and annotation.

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

Table I: Average time to collect 100 image datasets. Automatic or manual is shown next to the time measured.

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

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 F-score[65] as

\[
\text{IoU} = \frac{TP}{TP + FP + FN},
\]

\[
\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},
\]

\[
\text{Precision} = \frac{TP}{TP + FP},
\]

\[
\text{Recall} = \frac{TP}{TP + FN}.
\]

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

In all trials and categories, the mean values of precision rated around 70%. The mean values of recall were rated higher.

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

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. 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
Fig. 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 III: Necessary images for adaptation methods.

| Method                | Necessary images                                      |
|-----------------------|--------------------------------------------------------|
| Image scaling         | One image pair including a calibration board           |
| Background synthesis  | One background image                                   |
| Color correction      | One image pair including a color checker                |
| Seamless cloning      | One background image                                   |
| Histogram matching    | One object image captured in a real scene              |

The calculated values of the EMD and BD are 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 IV and Table V. 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.
(a) Image sequence recorded in the waste-sorting environment

(b) Visualization of detection results of our waste detector

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

Fig. 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.

TABLE V: 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 | 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.

or the plastic bottle’s cap. This is because the appearance as improved to approximate the target image. It also suggests that the BS+HM+EQ provided the highest similarity.

F. Detection Accuracy

Table VI 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 of the single shot multibox detector (SSD) [9]. SSD is a general object detector with a convolutional neural–network architecture that learns different anchor boxes. Fig. 15 shows the detection results.
TABLE VI: 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     | AC | GB | PB | Mean |
|------------------|----|----|----|------|
| 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   |

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

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 VI 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 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 VII. The results indicate that the pick-and-release
TABLE VII: Results of the sorting manipulation. Each element shows success rate [%] in each 10 trials.

| Method          | Object category | AC \( ^{a} \) | GB \( ^{b} \) | PB \( ^{c} \) | Mean |
|-----------------|-----------------|----------------|----------------|----------------|------|
| 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.

The 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.

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, an object–region extraction caused by similar colors, as shown in Fig. \( ^{18} \). To avoid difficulty of region extraction caused by similar colors, as shown in Fig. \( ^{18} \), 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 (e.g., 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.

ACKNOWLEDGEMENT

This work was supported by the projects, JPNP14004 and JPNP20012 subsidized by the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] S. P. Gundupalli, S. Hait, and A. Thakur, “A review on automated sorting of source-separated municipal solid waste for recycling,” Waste Management, vol. 60, pp. 56–74, 2017.
[2] K. Chahine and B. Ghazal, “Automatic sorting of solid wastes using sensor fusion,” International Journal of Engineering and Technology, vol. 9, no. 6, pp. 4408–4414, 2018.
[3] N. Barrero, D. Galvis, and C. Martínez, “Industrial robots for waste separation tasks: An approach to industry 4.0 in colombia,” in Proc. 9th International Conference on Production Research-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 Management, vol. 95, pp. 476–492, 2019.
[5] T. Gibson, “Recycling Robots,” Mechanical Engineering, vol. 142, no. 01, pp. 32–37, 2020.
[6] T. J. Lukka, T. Tossavainen, J. V. Kujala, and R. Tapani, “ZenRobotics Recycler – robotic sorting using machine learning,” in SBS, pp. 1–8, 2014.
[7] R. Girshick, “Fast R-CNN,” in ICCV, pp. 1440–1448, 2015.
[8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in NIPS, pp. 91–99, 2015.
[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in ECCV, pp. 21–37, 2016.
[10] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” CoRR, abs/1804.02767, 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 AAAI, pp. 9259–9266, 2019.
[12] T. Zhi, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in ICCV, pp. 9626–9635, 2019.
[13] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in CVPR, pp. 10781–10790, 2020.
[14] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in CVPR, pp. 248–255, 2009.
[15] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, and J. Winn, “The pascal visual object classes challenge: A retrospective,” International Journal of Computer Vision, vol. 111, no. 1, pp. 98–136, 2015.
[16] T.-Y. Lin, M. Maire, B. Serge, H. James, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in ECCV, pp. 740–755, 2014.
[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 CVPR, pp. 3213–3223, 2016.
[18] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–3232, 2019.
[19] T. Kiyokawa, K. Tomochika, T. Takamatsu, and T. Ogasawara, “Fully automated annotation with noise-masked visual markers for deep-learning-based object detection,” Robotics and Automation Letters, vol. 4, no. 2, pp. 1972–1977, 2019.
[20] H. Kato and M. Billinghurst, “Marker tracking and hmd calibration for a video-based augmented reality conferencing system,” in IWAR, pp. 85–94, 1999.
[21] T. Kiyokawa, K. Tomochika, T. Takamatsu, and T. Ogasawara, “Efficient collection and automatic annotation of real-world object images by taking advantage of post-diminished multiple visual markers,” Advanced Robotics, vol. 33, no. 24, pp. 1264–1280, 2019.

Fig. 18: Problematic images difficult to annotate. The coloring in each left image is the same as that of Fig. 11.
[77] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[78] B. Xiong, S. Jain, and K. Grauman, “Pixel Objectness: Learning to segment generic objects automatically in images and videos,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2677–2692, 2019.

[79] C. Rother, V. Kolmogorov, and A. Blake, “‘GrabCut’: Interactive foreground extraction using iterated graph cuts,” in *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 *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 *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 *IROS*, 2013, pp. 816–823.

[83] Jiaji Zhou, R. Paolini, J. A. Bagnell, and M. T. Mason, “A convex polynomial force-motion model for planar sliding: Identification and application,” in *ICRA*, 2016, pp. 372–377.

[84] J. Takamatsu, Y. Matsushita, and K. Ikeuchi, “Estimating camera response functions using probabilistic intensity similarity,” in *CVPR*, pp. 1–8, 2008.

[85] S. Imari, S. Voichi, and I. Katsushi, “Illumination from shadows,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, pp. 290–300, 2003.

[86] K. Hara, K. Nishino, and K. Ikeuchi, “Multiple light sources and reflectance property estimation based on a mixture of spherical distributions,” in *ICCV*, pp. 1627–1634, 2005.

[87] K. N. Kutulakos and S. M. Seitz, “A theory of shape by space carving,” *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199—218, 2000.

[88] S. Choi, Q.-Y. Zhou, and V. Koltun, “Robust reconstruction of indoor scenes,” in *CVPR*, pp. 5556–5565, 2015.

[89] G. D. Finlayson, S. D. Hordley, Cheng Lu, and M. S. Drew, “On the removal of shadows from images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59–68, 2006.

[90] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios, “Illumination estimation and cast shadow detection through a higher-order graphical model,” in *CVPR*, pp. 673–680, 2011.

[91] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, and D. Samaras, “Shadow detection with conditional generative adversarial networks,” *IEEE International Conference on Computer Vision*, pp. 4520—4528, 2017.

[92] L. Qu, J. Tian, S. He, Y. Tan, and R. W. H. Lau, “DeshadowNet: A multi-context embedding deep network for shadow removal,” *IEEE International Conference on Computer Vision*, pp. 2308—2316, 2017.