Sensors and Materials, Vol. 36, No. 8 (2024) 3519–3537
MYU Tokyo

S & M 3748

Integration of Stacked Object Recognition and Robotic Arm Gripping System Utilizing Mask Region-based Convolutional Neural Network and Red–Green–Blue Depth Images

Shu-Yin Chiang* and Yu-Kai Zhuo

Department of Information and Telecommunications Engineering, Ming Chuan University,
No. 5 De Ming Rd., Gui Shan District, Taoyuan City 333, Taiwan

(Received October 26, 2023; accepted May 14, 2024)

Keywords: RGBD, Mask R-CNN, stacked object recognition, gripping angle, robotic arm gripping

In this study, we explore the challenge of object recognition by robots in scenarios where individual objects cannot be identified due to stacking. We harness the capabilities of the light detection and ranging (LiDAR) camera as a sensor for object detection, leveraging its advanced sensing technology to simultaneously capture both color and depth images of objects. To address the issue of image overlap caused by stacking, the Mask region-based convolutional neural network (R-CNN) is employed for object recognition and segmentation. Additionally, through image mapping transformation, the positions of individual objects in the red, green, and blue (RGB) image are projected onto the depth image to extract their corresponding depth information. Given the disparity in camera positions between the color and depth cameras, occlusions and variations in depth mapping can result in missing depth values for certain objects. To mitigate this, various statistical methods are utilized to fill in these missing values and enhance the accuracy of the extracted depth information. Furthermore, by calculating the width and length of the rectangle of the rotated image, the angle with the smallest value is selected as the gripping angle of the object. Finally, leveraging the transformed coordinates and planned trajectory, the robotic arm executes the gripping task on stacked objects. This execution process validates the accuracy of the three-dimensional spatial information and showcases the functionality of an intelligent robotic arm.

1. Introduction

The development of robotic arms represents a significant milestone in industrial automation. Robotic arms excel at executing repetitive, hazardous, or confidential tasks, thereby reducing the depletion of human resources. Achieving the desired outcome of having robotic arms replace human labor in tasks involving object manipulation necessitates a combination of elements: the robotic arm's visual sensing system, motion control, and gripper design. Currently, most robotic arms are programmed to move and come equipped with visual systems for functions such as object recognition(1,2) and color detection, enabling them to grip objects. Watanabe *et al.*(3) even

*Corresponding author: e-mail: sychiang@mail.mcu.edu.tw
https://doi.org/10.18494/SAM4711

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

asserted that object recognition forms the foundational component of new technologies in robotic manipulations. However, challenges arise when dealing with objects that are interconnected, possess similar colors, or are stacked, making it difficult for the robotic arm's visual sensing system to distinguish between them. Even when the visual system can differentiate between different objects, the gripper design may not permit the simultaneous gripping of multiple objects. Consequently, the aim of this research is to explore how robotic arms can learn to identify and tackle the challenges presented by interconnected, similarly colored, and stacked objects.

In typical robot vision systems, object recognition often relies on identifying color blocks in images. This method has been employed in some research; however, when environmental noise or interference from similarly colored images is present, it becomes essential to process the images to reduce noise and facilitate independent object detection. Thus, when dealing with nonconnected objects, object recognition can be resolved by considering the object's shape.(4) Nonetheless, in most real-world industrial applications, objects are similar and not easily distinguished by shape. For planar image data, the You Only Look Once (YOLO) algorithm,(5) known for its speed and efficiency in 2D image recognition, is commonly utilized. YOLO processes each image through a convolutional neural network (CNN) classifier once, leading to significantly improved processing speed.

In the realm of object recognition, the utilization of partial image descriptors for object identification(6–8) has contributed to the advancement of robotic applications(9). The scale-invariant feature transform (SIFT)(7) is a method that offers features with invariant image scale and rotation, but it requires precise object segmentation. Several object segmentation algorithms(10,11) have been proposed to extract object regions from the background. However, these algorithms may falter in complex and cluttered environments where defining clear boundaries between the object and background regions is challenging. Another approach to object segmentation involves background subtraction combined with optical flow processing, which addresses the ambiguity of object regions within the background.(12,13) However, even after separating object features from cluttered backgrounds, numerous challenges persist in utilizing robotic systems for perceiving object features.

To address these challenges, Kim *et al.*(14) introduced the use of feature-extraction-based neural networks and object classifiers for dynamic object recognition. Bin picking currently stands as one of the most substantial challenges in robotic automation involving visual systems, as objects often share similarities and frequently are stacked. Harada *et al.*(15) proposed a robotic bin-picking approach that utilizes feature extraction to identify objects and determine suitable gripping orientations. Nonetheless, when confronted with a diverse array of objects and complex stacking scenarios, object recognition remains a focal point of research.

In scenarios dealing with nonplanar 3D objects, object recognition systems can leverage the 3D shapes of these objects. Tang *et al.*(16) proposed a method of using Kinect to capture object images and create feature points for each object. SIFT is then employed to match the feature points with objects in a database, enabling the detection and identification of arranged objects. Lenz *et al.*(17) introduced a deep learning approach for object recognition and determining optimal grasping positions. This is achieved through a two-stage learning process, where the sparse auto-encoder (SAE) algorithm(18) is utilized to find the best weights corresponding to various objects, ultimately leading to the identification of the optimal gripping locations.

Jiang *et al.*(19) introduced a fusion approach of incorporating red, green, and blue (RGB) and depth images, leveraging sensor technology to enhance everyday object recognition. The recognition results are used in a two-step approach to execute the gripping action of a robotic arm. This approach has also been used to enable the recognition of hand gestures from a user seated in a wheelchair, allowing for control of a robotic arm's gripping behavior. Matsuno *et al.*(20) and Wang *et al.*(21) proposed the use of depth information for stacked object recognition and object pose analysis. By calculating the plane from the depth data, they obtain the plane's vector and computed relevant angles on the basis of distances from the plane. However, in those studies, prior knowledge of the object's shape (e.g., rectangular) was assumed, which enabled object recognition through matching of the distances to the object's plane.

When confronted with stacked objects, employing YOLO for object recognition leads to overlapping bounding boxes, resulting in inaccuracies in the actual positions of the objects and the complication of precise gripping control of the robotic arm. To tackle the challenge of overlapping bounding boxes in object recognition with YOLO, we can leverage semantic segmentation techniques. He *et al.* introduced the Mask region-based CNN (R-CNN) architecture,(22) which conducts semantic segmentation within the region of interest (ROI). By implementing ROI alignment and pixel-level classification, Mask R-CNN adeptly segments objects within an image and assigns distinctive colors to them. The Mask R-CNN architecture effectively distinguishes overlapped and intertwined objects, enabling pixel-level coloring and recognition. This approach substantially enhances the accuracy of stacked object recognition. As seen in Fig. 1, the teddy bear is not recognized by YOLO but is recognized by Mask R-CNN. In this study, we conducted a comparative analysis of the results obtained using YOLO and Mask R-CNN in scenarios involving overlapped and stacked objects. Our findings revealed that Mask R-CNN achieves a superior recognition rate. As a result, we utilize Mask R-CNN as the foundation for object recognition in our research.

Siradjuddin *et al.*(23) proposed the utilization of a Kinect vision system for controlling a 7-axis robotic arm. Given that the target object for grasping was a ball, they implemented a Kalman filter to track the object (ball) and utilized the linear velocity and angular velocity derived from the arm's Jacobian matrix to track the end-effector, thereby achieving object

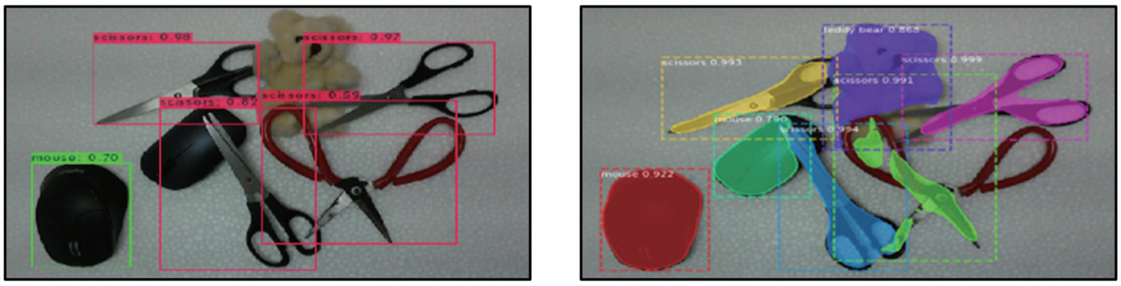


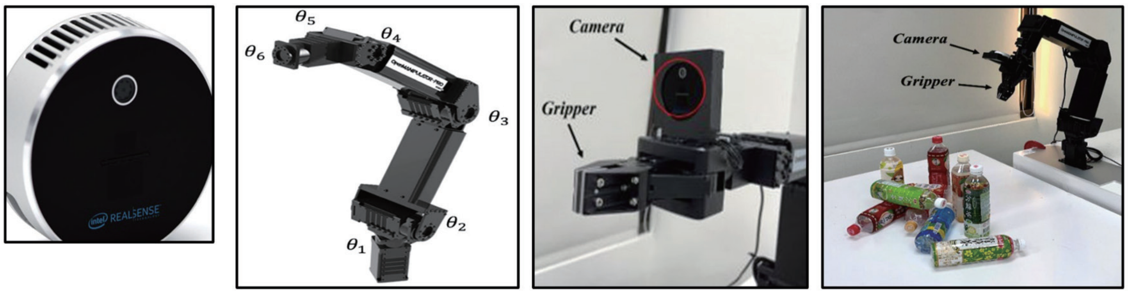
Fig. 1. (Color online) Object recognition by (a) YOLO and (b) Mask R-CNN.

grasping functionality. In a separate study, Song *et al.*(24) focused on vision-based adaptive arm grasping. They employed Kinect to capture the object and applied speeded up robust features for the rapid extraction of object features. These features were then compared with those in a database system for object recognition. The grasp position of the object was determined on the basis of the information in the database, and its position in the image was calculated. By automatically adjusting the arm's movement taking into account the error between the target object's position and the arm's grasp position, they successfully achieved object grasping. Huang *et al.*(25) employed dual-camera lenses to calculate the three-dimensional coordinates of the target object. By implementing visual integration and an error compensation system for position control, the robotic arm reached the target with an error of approximately 2 cm.

Building upon the literature discussed earlier, we propose image-processing methodologies tailored for managing stacked objects using a robotic arm, integrating sensor technology to acquire RGB and depth information. Additionally, we outline angle calculations pertinent to the gripping technique, ensuring alignment with the research goals. The system's key innovation lies in its ability to utilize sensor data to adjust the robotic arm, enabling precise determination of the optimal gripping angle within a stack of items. The remainder of this paper is structured as follows. In Sect. 2, we describe the materials and methods employed. The experimental results and conclusions are presented in Sects. 3 and 4, respectively

2. Materials and Methods

In this study, an integration of cutting-edge technologies is employed, including the utilization of an Intel RealSense light detection and ranging (LiDAR) camera L515 for depth sensing capabilities. It is coupled with the OpenManipulator-P 6-axis robotic arm sourced from a reputable robotics company and powered by a laptop computer equipped with an NVIDIA GeForce RTX 2080 graphics card. In this research, we aim to realize the potential of intelligent robotic arms in gripping stacked objects. By leveraging the sensing capabilities of the LiDAR camera, the system has enhanced object recognition and manipulation, contributing to advancements in robotic gripping techniques for complex scenarios involving stacked items. The hardware configuration is illustrated in Fig. 2. The L515 LiDAR camera is positioned above



(a) (b) (c) (d)

Fig. 2. (Color online) System hardware architecture. (a) LiDAR camera, (b) robotic arm, (c) vision system, and (d) experimental setup.

the robotic arm to capture both color and depth information of the objects. The OpenManipulator-P robotic arm is equipped with six degrees of freedom and features an open-close gripper. It boasts a payload capacity of approximately 3 kg and a reach radius of around 64 cm. The arm ensures precise positioning with a repeatability of 0.05 mm. The specifications of the six-axis angle range are given in Table 1.

The system flowchart for this study is depicted in Fig. 3. The Intel RealSense LiDAR camera L515 is employed in the system to capture both color and depth images of the scene. Python serves as the programming language for implementing the object recognition system. The robot operating system (ROS) framework is utilized to interface with the robotic arm and acquire the endpoint coordinates of the objects. The Mask R-CNN processes the captured images, performing semantic segmentation to isolate and label individual objects within the scene. By integrating depth information with color object data, the system accurately determines the depth of each detected object. The object with the shortest distance to the robotic arm becomes the target object. On the basis of the identified objects and their respective depth information, the system determines the optimal sequence for gripping the stacked objects, taking their distances into account. This information is then employed to control the movement of the robotic arm using inverse kinematics. The intricate implementation and functionality of the object recognition system is elaborated upon in the subsequent sections.

Table 1
OpenManipulator-P specifications.

		OpenManipulator-P
Degree of freedom		6
Payload (kg)		3
Repeatability (mm)		±0.05
Speed (Each joint, °/s)		180
Weight (kg)		5.5
Reach (mm)		645
Working range (°)	θ_1	±180
	θ_2	±90
	θ_3	−90–+90
	θ_4	±180
	θ_5	±90
	θ_6	±180

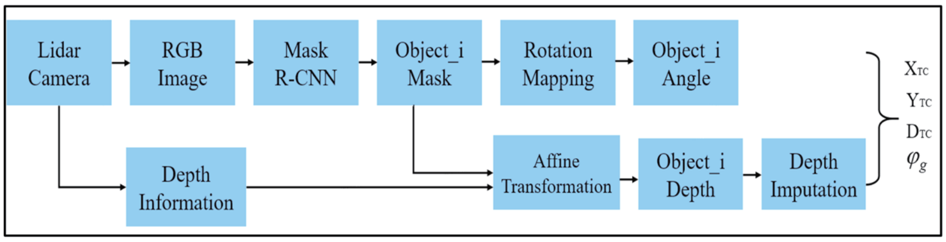


Fig. 3. (Color online) System flow chart.

2.1 Object recognition system

Initially, the color image is processed for object recognition using the Mask R-CNN algorithm, as depicted in Fig. 4. Mask R-CNN conducts semantic segmentation by categorizing pixels into different object classes. In addition to object classification and bounding box determination, a pivotal aspect of Mask R-CNN is the generation of masks for distinct objects, which are depicted with unique colors. Mask R-CNN employs the ROI alignment technique for mapping regions of interest, and it utilizes bilinear interpolation to enhance the accuracy of object bounding boxes. Ultimately, the algorithm colors all pixels within the object's boundaries in the image, thereby achieving masking functionality.

In the experimental setup of this study, various beverage bottles are stacked and positioned on the laboratory table. The camera mounted on the robotic arm captures the color image of the tabletop, as shown in Fig. 5(a). The captured image is subsequently subjected to Mask R-CNN processing for object recognition, resulting in individual objects being represented in different colors, as displayed in Fig. 5(b). The object recognition accuracy, summarized in Table 2, ranges from 0.994 to 0.721. However, it should be noted that, owing to object occlusion, two objects are not correctly identified.

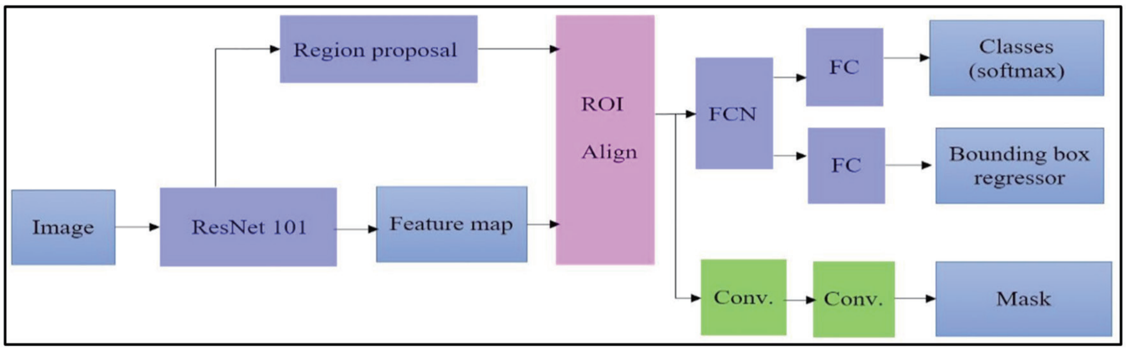


Fig. 4. (Color online) Mask R-CNN structure.

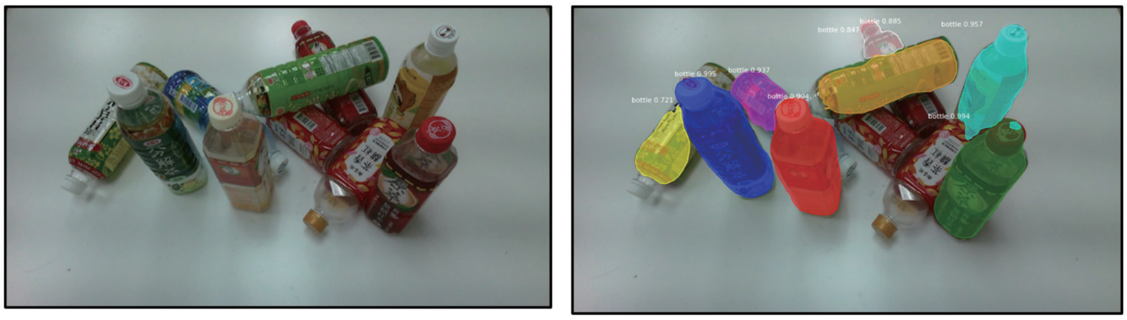


Fig. 5. (Color online) Individual object recognition using Mask R-CNN.

Table 2
Object recognition rate.

Object color	Object recognition rate
Green	0.994
Red	0.994
Blue	0.995
Orange	0.885
Light blue	0.957
Yellow	0.721
Purple	0.937
White	0.847

2.2 Integration of depth information

Upon recognizing individual objects within the image of stacked objects, the color and depth information is amalgamated to acquire the depth of each object. The position of each object is defined by Eq. (1), where O_i signifies the *i*-th object among *N* objects, while I_X and I_Y denote the coordinates of the *i*-th object. We employ an affine transformation to map the coordinates (I_X, I_Y) from the color image to the coordinates (d_x, d_y) in the depth image, as depicted by Eq. (2). The parameters T_X and T_Y represent the translation amounts along the *X* and *Y* axes between the two images, while R_{00}, R_{01}, R_{10}, and R_{11} are the coefficients of the affine transformation, which can be determined through experimentation. The depth coordinates, d_x and d_y, convey the depth information of object *i*, as expressed in Eq. (3), with d_i denoting the depth of the *i*-th object. By merging the color image [Fig. 6(a)] with the depth image [Fig. 6(b)], we can derive the depth of each object, enabling the robotic arm to calculate relative distances. The target object T_o for gripping is designated as the object with the shortest distance to the robotic arm, as stated in Eq. (4). The bar indicates the average value.

$$O_i(I_X, I_Y),\ i = 1, 2, \cdots, N,\ (I_X, I_Y) \in O_i \tag{1}$$

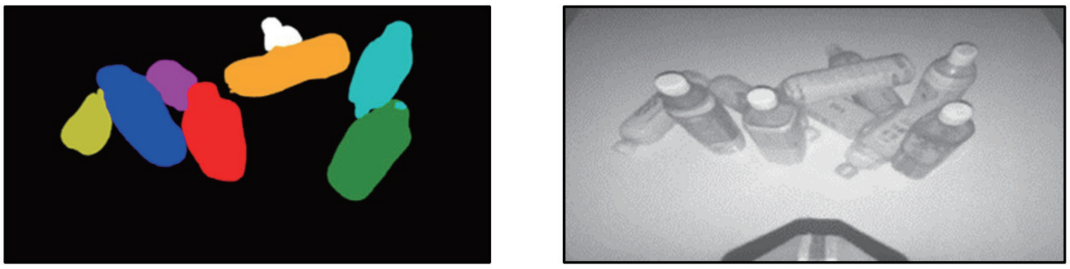
$$\begin{bmatrix} R_{00} & R_{01} & T_X \\ R_{10} & R_{11} & T_Y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_X \\ I_Y \\ 1 \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \\ 1 \end{bmatrix} \tag{2}$$

$$d_i(d_x, d_y),\ i = 1, 2, \cdots, N,\ (d_x, d_y) \in O_i \tag{3}$$

$$T_o = \underset{i}{\operatorname{argmin}}(\overline{d_i(d_x, d_y)}) \big| (d_x, d_y) \in O_i,\ i = 1, 2, \cdots, N \tag{4}$$

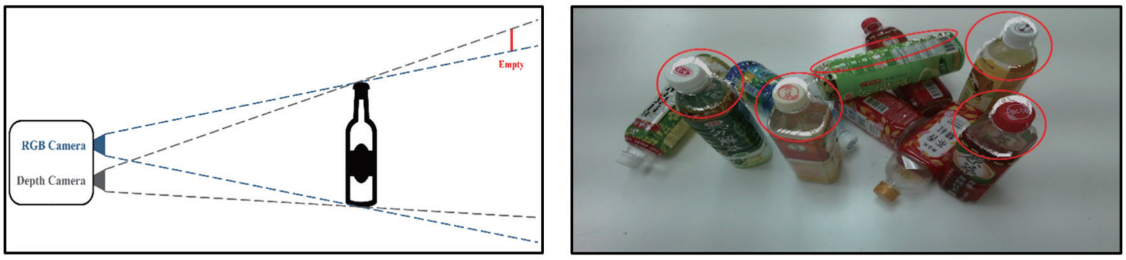
2.3 Depth value imputation

Because of slight differences in the image coverage between the color camera and the depth camera, as shown in Fig. 7(a), it was observed during the experiments that there are instances



(a) (b)

Fig. 6. (Color online) Integration of color and depth images. (a) Masks and (b) depth of the color image.



(a) (b)

Fig. 7. (Color online) Empty values in depth image. (a) Empty value due to different coverage and (b) depth empty value regions.

where the depth values in the captured images are missing, as illustrated in Fig. 7(b). The white regions along the edges in Fig. 7(b) represent the areas with missing depth values in the depth image.

Therefore, after applying Mask R-CNN object segmentation to the objects in Fig. 8, they are labeled in order of their original depth distances, from the nearest to the farthest, as indicated by the numbers 1 to 8 in Fig. 8. For the objects in Fig. 8 with missing depth values, statistical methods such as skip, delete, mean, median, and mode are employed to handle these missing values, and the modified depth value (dm_i) is given by Eqs. (5)–(8). Equation (5) represents the skip and delete method, and Eqs. (6)–(8) indicate the mean, median, and mode of the depth of the object, respectively. The skip method simply unprocesses the missing values, and the delete method directly removes the missing values without considering them in the calculation of the mean. Therefore, we define the variable n_i in Eq. (9) as the regions with depth not equal to 0, and z_i in Eq. (10) as the missing value regions with depth equal to 0. Since the skip, mean, median, and mode methods require the calculation of the overall average depth of the objects, while the delete method deletes the missing regions, let C represent the classification of skip, mean, median, and mode, and D represent the classification of delete. Thus, we determine the average depth of an object by C and D, respectively, as shown in Eqs. (11) and (12). Hence, after depth value determidation, we recalculate the target object for gripping with the shortest distance from the robotic arm, as stated in Eq. (13). Then, we obtain the depth of the target object, D_{TC}, by Eq. (14).

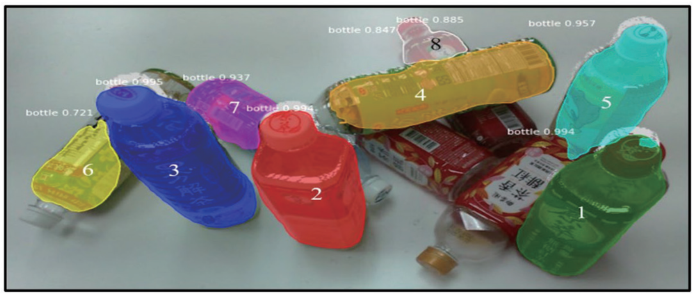


Fig. 8. (Color online) Dealing with object empty values in depth image.

$$dm_{skip_i}(d_x,d_y)=\begin{cases} d_i(d_x,d_y), & \text{if } d_i(d_x,d_y)\neq 0 \\ 0, & \text{if } d_i(d_x,d_y)=0 \end{cases},\ i=1,2,\cdots,N,\ (d_x,d_y)\in O_i \tag{5}$$

$$dm_{mean_i}(d_x,d_y)=\begin{cases} d_i(d_x,d_y), & \text{if } d_i(d_x,d_y)\neq 0 \\ \text{mean}(d_i(d_x,d_y)), & \text{if } d_i(d_x,d_y)=0 \end{cases},\ i=1,2,\cdots,N,\ (d_x,d_y)\in O_i \tag{6}$$

$$dm_{median_i}(d_x,d_y)=\begin{cases} d_i(d_x,d_y), & \text{if } d_i(d_x,d_y)\neq 0 \\ \text{median}(d_i(d_x,d_y)), & \text{if } d_i(d_x,d_y)=0 \end{cases},\ i=1,2,\cdots,N,\ (d_x,d_y)\in O_i \tag{7}$$

$$dm_{mode_i}(d_x,d_y)=\begin{cases} d_i(d_x,d_y), & \text{if } d_i(d_x,d_y)\neq 0 \\ \text{mode}(d_i(d_x,d_y)), & \text{if } d_i(d_x,d_y)=0 \end{cases},\ i=1,2,\cdots,N,\ (d_x,d_y)\in O_i \tag{8}$$

$$n_i(d_x,d_y)=n_i(d_x,d_y)+1,\ i=1,2,\cdots,N,\ (d_x,d_y)\in O_i \tag{9}$$

$$z_i(d_x,d_y)=\begin{cases} 0, & \text{if } d_i(d_x,d_y)\neq 0 \\ 1, & \text{if } d_i(d_x,d_y)=0 \end{cases},\ i=1,2,\cdots,N,\ (d_x,d_y)\in O_i \tag{10}$$

$$\overline{dm_{C_i}(d_x,d_y)}=\frac{\sum_{d_x}\sum_{d_y} dm_{C_i}(d_x,d_y)}{\sum_{d_x}\sum_{d_y}[n_i(d_x,d_y)+z_i(d_x,d_y)]},\ i=1,2,\cdots,N,\ (d_x,d_y)\in O_i, \tag{11}$$

where C = {skip, mean, median, and mode}.

$$\overline{dm_{D_i}(d_x,d_y)}=\frac{\sum_{d_x}\sum_{d_y} dm_{skip_i}(d_x,d_y)}{\sum_{d_x}\sum_{d_y} n_i(d_x,d_y)},\ i=1,2,\cdots,N,\ (d_x,d_y)\in O_i, \tag{12}$$

where D = {delete}.

$$T_o = \begin{cases} \underset{i}{\operatorname{argmin}}[\overline{dm_{C_i}(d_x,d_y)}] \big| (d_x,d_y) \in O_i, C = \{\text{skip,mean,median,mode}\} \\ \underset{i}{\operatorname{argmin}}[\overline{dm_{D_i}(d_x,d_y)}] \big| (d_x,d_y) \in O_i, D = \{\text{delete}\} \end{cases}, \ i = 1,2,\cdots,N \tag{13}$$

$$D_{TC} = \begin{cases} \overline{dm_{\mathrm{C}T_o}(d_x,d_y)}, \ C = \{\text{skip,mean,median,mode}\} \\ \overline{dm_{\mathrm{D}T_o}(d_x,d_y)}, \ D = \{\text{delete}\} \end{cases} \tag{14}$$

2.4 Object angle calculation

The experiment is conducted with stacked objects of different colors and stacked objects of the same color for recognition. To determine the angles of the objects, the object images are first binarized. The binarization process converts the image into a binary representation $B(I_X, I_Y)$, where the value of 1 represents the target object and 0 represents nontarget objects. The centroid of each object is then computed using the centroid formula, Eqs. (15) and (16) for the X and Y coordinates of the centroid, respectively.

$$X_{TC} = \frac{\sum_{I_Y}\sum_{I_X}(B_{T_o}(I_X,I_Y) \times I_X)}{\sum_{I_Y}\sum_{I_X} B_{T_o}(I_X,I_Y)}, \ (I_X,I_Y) \in O_i \tag{15}$$

$$Y_{TC} = \frac{\sum_{I_Y}\sum_{I_X}(B_{T_o}(I_X,I_Y) \times I_Y)}{\sum_{I_Y}\sum_{I_X} B_{T_o}(I_\mathrm{X},I_Y)}, \ (I_X,I_Y) \in O_i \tag{16}$$

Next, we calculate the length and width of the object's bounding rectangle, as shown in Fig. 9. L_x and L_y represent the width and length of the rectangular box, respectively, and their formulas are given by Eqs. (17) and (18). When calculating the image rotation angle, denoted as ij, for each object, the lengths $L_x(j)$ and $L_y(j)$ are determined using Eqs. (19) and (20). We record the smaller of $L_x(j)$ and $L_y(j)$ for each angle, denoted as $L_{min}(j)$ in Eq. (21). Finally, after rotating the image

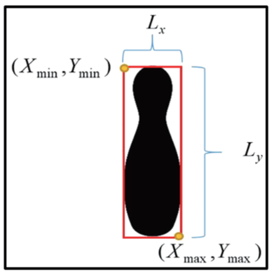


Fig. 9. (Color online) Calculating the object's bounding rectangle.

by 89°, we identify the angle φ_g that corresponds to the smallest value as the optimal gripping angle for the object, as shown in Eq. (22).

$$L_x = \left| X_{max} - X_{min} \right| \tag{17}$$

$$L_y = \left| Y_{max} - Y_{min} \right| \tag{18}$$

$$L_x(j) = \left| X_{max}^{j} - X_{min}^{j} \right|, \ j = 0,1,\cdots,89 \tag{19}$$

$$L_y(j) = \left| Y_{max}^{j} - Y_{min}^{j} \right|, \ j = 0,1,\cdots,89 \tag{20}$$

$$L_{min}(j) = \min(L_x(j), L_y(j)), \ j = 0,1,\cdots,89 \tag{21}$$

$$\phi_g = \underset{j}{\operatorname{argmin}}(L_{min}(j)), \ j = 0,1,\cdots,89 \tag{22}$$

To provide a detailed explanation of how the robotic arm determines the angle for gripping an object, we use Fig. 10 for illustration. In Fig. 10(a), the object is initially positioned at an angle of 30°, but the machine vision system interprets it as 0°. Subsequently, the image of the object is rotated to 20°, 25°, 60°, 75°, and 85°, respectively, as shown in Figs. 10(b) to 10(f). Finally, the angle of 60° results in the smallest rectangular bounding box length, i.e., ϕ_g = 60°. Therefore, the robotic arm rotates by 60° to grip the object.

2.5 Robotic arm control

The Denavit–Hartenberg (DH) parameters for the six-axis robotic arm used in this study are given in Table 3. In this table, *a* indicates the distance traveled along the *X*-axis from the current coordinate axis to reach the subsequent axis, *α* represents the angle through which the current coordinate axis rotates around the *X*-axis to align with the next axis, *d* is the distance traveled along the *Z*-axis from the current coordinate axis to reach the following axis, *θ* denotes the angle by which the current coordinate axis rotates around the *Z*-axis to align with the subsequent axis,

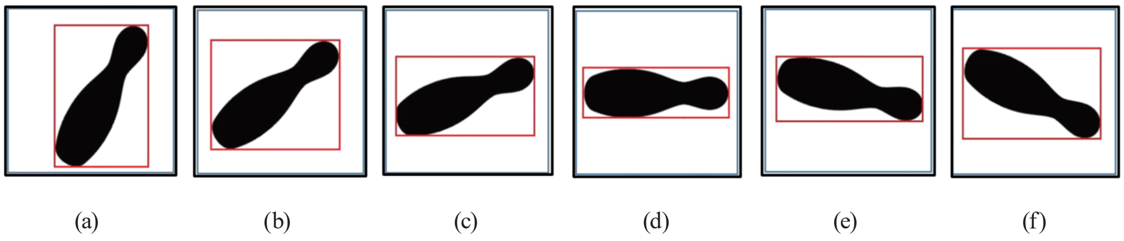


Fig. 10. (Color online) Width and height of the rectangle frame after image rotation. (a) 0, (b) 20, (c) 25, (d) 60, (e) 75, and (f) 85°.

Table 3
D–H parameters.

link k	a_{k-1} (mm)	α_{k-1} (rad)	d_k (mm)	θ_k (rad)
1	0	0	0	θ_1
2	0	$-\pi/2$	0	θ_2
3	a_2 (265.69)	0	0	θ_3
4	a_3 (30)	$-\pi/2$	d_4 (258)	θ_4
5	0	$\pi/2$	0	θ_5
6	0	$-\pi/2$	0	θ_6

and k corresponds to the motor number. Then, using inverse kinematics to obtain the rotation degree, the robotic arm is moved to the desired position.

After obtaining the position of the target object in image coordinates through machine vision, it is necessary to determine the position of the object in the base coordinate system of the robotic arm for gripping. This transformation process involves converting the image coordinate system to the world coordinate system, and then from the world coordinate system to the arm base coordinate system, as illustrated in Fig. 11. Figure 12 depicts the eye-in-hand arm coordinate transformation, where ${}^{c}\boldsymbol{T}_w$ represents the transformation matrix from the world coordinate system to the camera coordinate system, ${}^{g}\boldsymbol{T}_c$ represents the transformation matrix from the camera coordinate system to the gripper coordinate system, ${}^{b}\boldsymbol{T}_g$ represents the transformation matrix from the gripper coordinate system to the arm base coordinate system, and finally, ${}^{b}\boldsymbol{T}_w$ represents the transformation matrix from the world point coordinate system to the arm base coordinate system.

First, we establish the following definitions: (u, v) represents the pixel coordinates relative to the image center, with (X_w, Y_w, Z_w) for world coordinates, (X_g, Y_g, Z_g) for the gripper coordinate system, (X_c, Y_c, Z_c) for the camera coordinate system, and (X_b Y_b, Z_b) for the arm base coordinate system. As shown in Eq. (23), z represents the depth of the camera with respect to the world coordinate system, K represents the camera's intrinsic parameters, and $\boldsymbol{R}$ and $\boldsymbol{T}$ are the transformation matrices characterizing the camera's extrinsic parameters with respect to the world coordinate system. The process of transforming pixel coordinates into world coordinates is grounded in the principles of the pinhole camera. Once the pixel coordinates have been converted into world coordinates, they are subsequently mapped onto the mechanical arm's base coordinate system using the transformation matrix ${}^{b}\boldsymbol{T}_w$, which is a composite of three transformation matrices, as depicted in Eq. (24). ${}^{c}\boldsymbol{T}_w$ embodies the transformation matrix for converting from the world coordinate system to the camera coordinate system, as described by Eq. (25). ${}^{g}\boldsymbol{T}_c$ represents the transformation matrix for converting from the camera coordinate system to the gripper coordinate system, as delineated in Eq. (26). ${}^{b}\boldsymbol{T}_g$ stands for the transformation matrix of transitioning from the gripper coordinate system to the arm base coordinate system, as outlined by Eq. (27). Finally, by integrating Eqs. (23) through (27), we can compute the transformation of image coordinates into the arm base coordinate system.

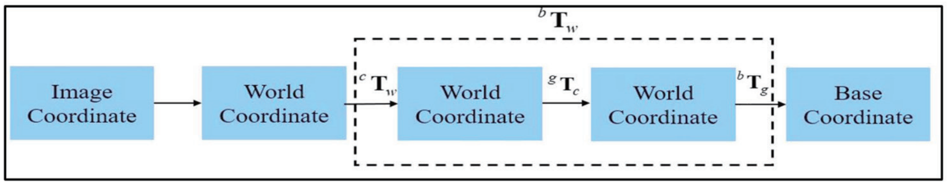


Fig. 11. (Color online) Coordinate transformation process.

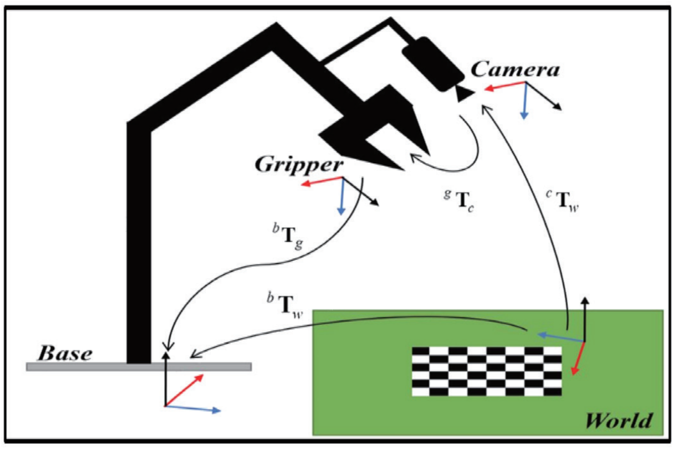


Fig. 12. (Color online) Schematic of eye-in-hand arm coordinate transformation.

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{23}$$

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} = {}^{b}\boldsymbol{T}_w \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = {}^{b}\boldsymbol{T}_g \cdot {}^{g}\boldsymbol{T}_c \cdot {}^{c}\boldsymbol{T}_w \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{c}\boldsymbol{R}_w & {}^{c}\boldsymbol{t}_w \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = {}^{c}\boldsymbol{T}_w \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{25}$$

$$\begin{bmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{g}\boldsymbol{R}_c & {}^{g}\boldsymbol{t}_c \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = {}^{g}\boldsymbol{T}_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{26}$$

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} = \begin{bmatrix} {}^b\boldsymbol{R}_g & {}^b\boldsymbol{t}_g \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{bmatrix} = {}^b\boldsymbol{T}_g \begin{bmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{bmatrix} \tag{27}$$

3. Results and Discussion

The experiment is conducted with stacked objects of both different colors and similar colors for recognition. Using the results of object recognition, the robotic arm performs gripping actions by moving and rotating around the objects accordingly.

3.1 Experiments with differently colored stacked objects

First, to illustrate the superior recognition capabilities of Mask R-CNN in the context of stacked objects, we conducted training and compared the differences between Mask R-CNN and YOLOv7 using the Microsoft COCO dataset. As shown in Fig. 13, Mask R-CNN can identify objects with higher accuracy than YOLOv7. The results for the two methods are summarized in Table 4, wherein it can be seen that YOLOv7 exhibits lower recognition rates, some as low as 0.35, or fails to recognize objects when they are occluded.

Following the research methodology outlined in Sect. 2, we perform object recognition on the colored images using Mask R-CNN. After identifying objects, we obtain the center points of individual objects and then acquire depth information by matching images. For depth value determination, the first column in Table 5 represents the object color, the second column indicates the true distance, the third column shows the unprocessed values (i.e., depth values equal to 0) for the regions with missing values, and the fourth column represents the values with the missing values directly removed, without being considered in the calculation of the mean. The fifth column fills the missing values with the mean, the sixth column uses the median to replace in the missing values, and in the seventh column, the missing values are replaced by the mode. The closest true value is shown in bold font. The bold font entries in Table 5 show that the median and mode are closest to the actual results in many cases. Also, it is observed that for closer distances, replacing the missing values with the median yields results closer to the true

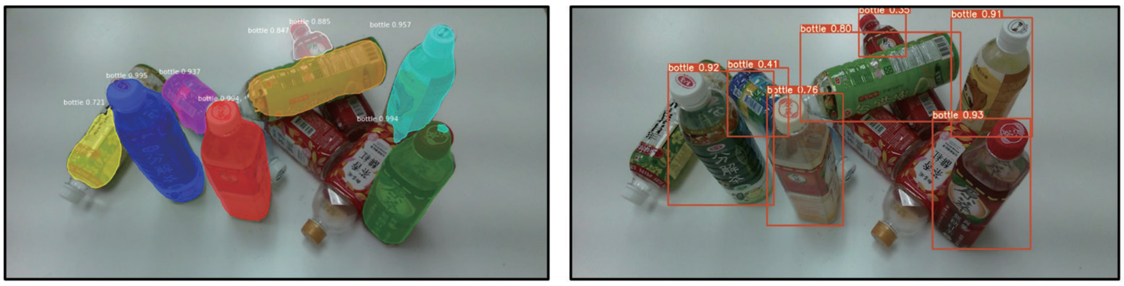


Fig. 13. (Color online) Object recognition by Mask RCNN and YOLO.

Table 4
Object recognition rate.

Object color	Recognition rate (Mask R-CNN)	Recognition rate (Yolo v7)
Green	0.994	0.930
Red	0.994	**0.760**
Blue	0.995	0.920
Orange	0.885	0.800
Light blue	0.957	0.910
Yellow	0.721	**X**
Purple	0.937	**0.410**
White	0.847	**0.350**

Table 5
Distance calculation (cm) by various methods.

Colored object	Real	Skip		Delete		Mean		Median		Mode	
		Mean	Error %	Mean	Error %	Mean	Error %	Mean	Error %	Mean	Error %
Green	48.0	47.38	1.29	47.97	0.06	47.97	0.06	**47.99**	**0.02**	47.82	0.37
Red	50.3	49.12	2.35	50.78	0.95	50.78	0.95	50.85	1.09	**50.35**	**0.10**
Blue	51.5	50.94	1.09	51.47	0.06	51.47	0.06	**51.48**	**0.04**	51.34	0.31
Orange	53.6	**53.36**	0.45	**53.36**	**0.45**	**53.36**	**0.45**	**53.36**	**0.45**	**53.36**	**0.45**
Light blue	55.5	53.3	3.96	55.21	0.52	55.21	0.52	**55.23**	**0.49**	54.95	0.99
Yellow	58.6	**58.04**	**0.96**	**58.04**	**0.96**	**58.04**	**0.96**	**58.04**	**0.96**	**58.04**	**0.96**
Purple	60.8	58.05	4.52	60.57	0.38	60.57	0.38	60.56	0.39	**60.59**	**0.35**
White	64.2	63.2	1.56	64.14	0.09	64.14	0.09	64.13	0.11	**64.19**	**0.02**
Avg. error %	—	—	2.02	—	0.43	—	0.43	—	0.44	—	0.44

distances. Conversely, for objects at greater distances, replacing the missing values with the mode provides results closer to the true distances. However, when calculating the percentage error and taking the average, it can be found that the average errors obtained with the delete, mean, median, and mode methods are all significantly smaller than that of the skip method. Therefore, if the goal is to accelerate experimental results and reduce the amount of computations, using the delete method can yield results the fastest with errors similar to those of the mean, median, and mode methods.

3.2 Experiments with similarly colored stacked objects

For stacked objects of the same color, we use Mask R-CNN for recognition. First, we place objects such as screws, nuts, hex keys, and wrenches of the same color on the table. Because of the similarity in the colors of these objects, special handling is required during annotation. We augment the dataset of the colored images using data augmentation techniques to a total of 1000 images. Additionally, we relabel the dataset to prevent cases where a single object is segmented into two separate semantic entities. The training results for object recognition with object recognition accuracy in the range of 0.918 to 0.995 are shown in Fig. 14.

When objects are placed on a white background, the presence of reflections in the camera image can lead to difficulties in recognition. Therefore, we change the background to black. As shown in Fig. 15, it can be observed that the resulting object recognition rate improves and

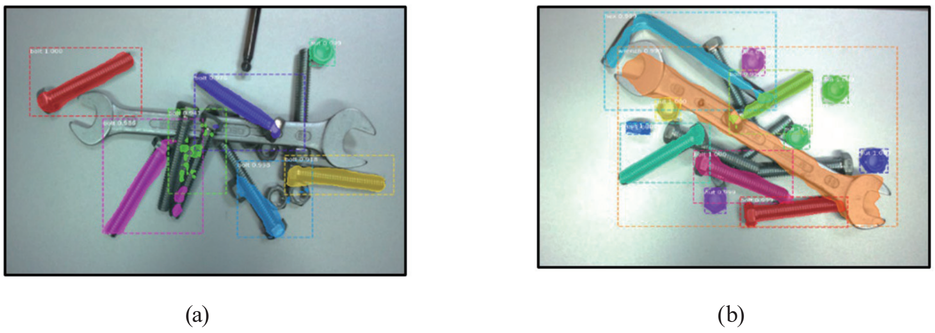


Fig. 14. (Color online) Recognition results using Mask R-CNN for objects of the same color. Recognition rates are (a) 0.918 and (b) 0.995, respectively.

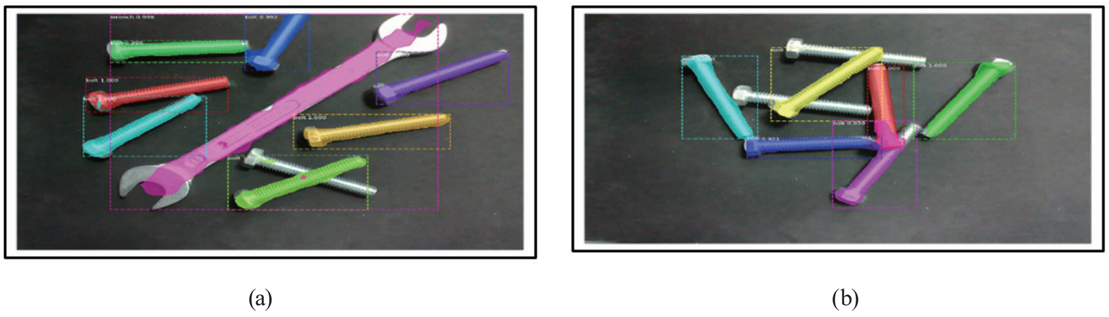


Fig. 15. (Color online) Object recognition of same-color items on low-reflection background. Recognition rates are (a) 0.992 and (b) 0.939, respectively.

objects that are not occluded are now clearly distinguishable. The mean average precision of multiple experimental results is 0.938. When there is a clear distinction between the object and background colors, the object segmentation performance is optimal.

3.3 Experiments on robotic arm control and gripping tasks

After obtaining the target, the system applies the image rotation method to find the most suitable gripping angle. Figure 16 shows the results of the experiment, with Fig. 16(a) displaying the overhead view captured by the camera mounted on the robotic arm and Fig. 16(b) showing the color information obtained after applying Mask R-CNN to the image. Through affine transformation using the color and depth information, the bottle masked green is determined to be the closest. Therefore, the robotic arm follows a plan to move slowly from points 1 to 5 above the object. The coordinates of each point along the arm's movement path are shown in Figs. 16(c) and 16(d) for the top and side views, respectively. Figure 16(e) illustrates the arm opening its gripper above the object. In Fig. 16(f), a 4° rotation of the arm is depicted. In Fig. 16(g), the gripping action of the arm is shown. Finally, Fig. 16(h) shows the completion of the gripping task.

Figure 17 illustrates the rotation of the robotic arm and object angle. In Fig. 17(a), image recognition identifies the nearest object as the horizontally positioned green plastic bottle. The

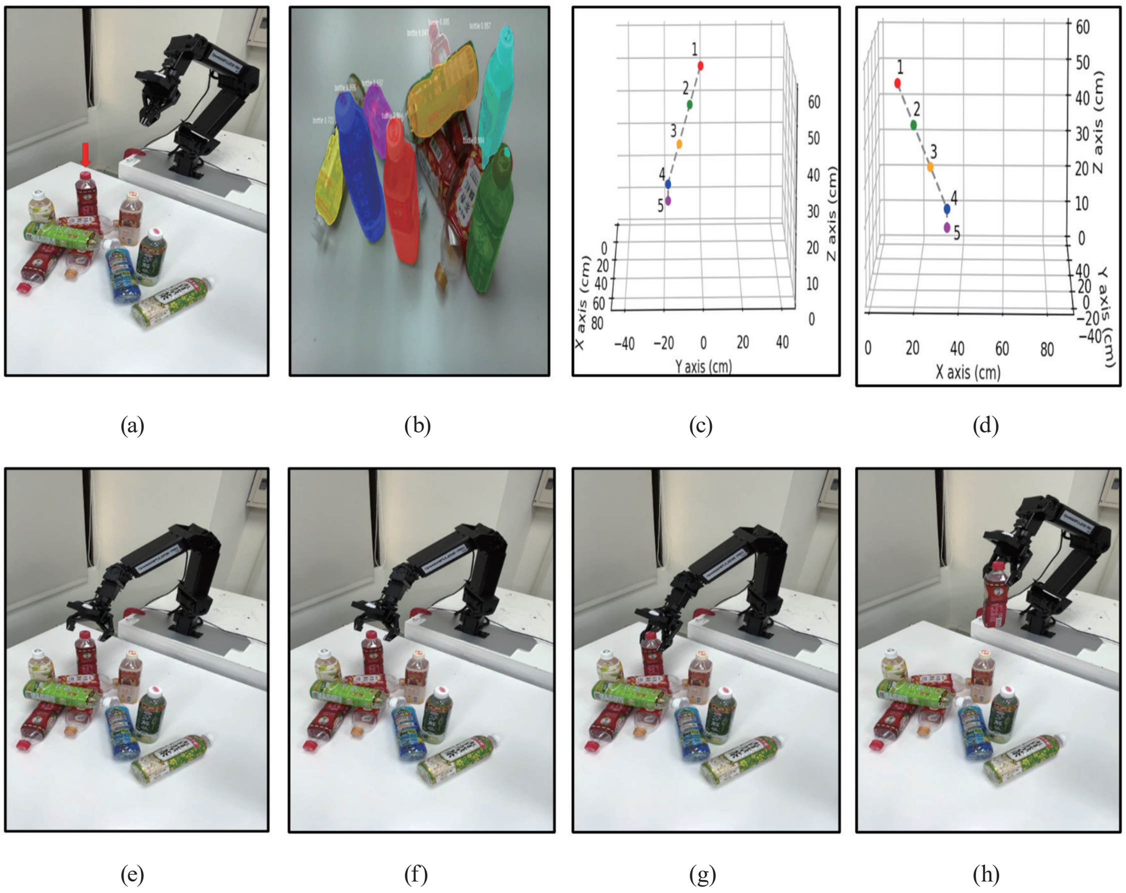


Fig. 16. (Color online) Experiment with objects of different colors: arm movement path planning and gripping. (a) Target object, (b) Mask RCNN, (c) arm trajectory front view, (d) arm trajectory side view, (e) arm movement, (f) arm rotation by 4°, (g) arm gripping, and (h) task completion.

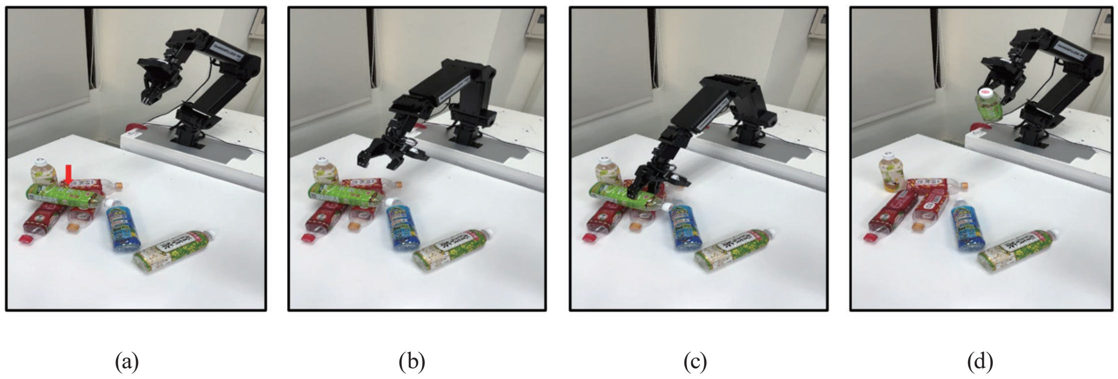


Fig. 17. (Color online) Experiment with objects of different colors: arm rotation and gripping. (a) Arm movement, (b) arm rotation by 82°, (c) arm gripping, and (d) task completion.

robotic arm moves above it, calculates the object's angle, and then rotates the arm by 82° [Fig. 17(b)] to perform the gripping action [Fig. 17(c)], ultimately completing the gripping task [Fig. 17(d)].

4. Conclusions

In this study, the successful object segmentation of stacked items was achieved through the utilization of RGB-D images captured from above the robotic arm, facilitated by the integration of sensor technology, specifically the RGB-D imaging capabilities of the Intel RealSense LiDAR camera L515. The Mask R-CNN algorithm was then employed to perform the segmentation process effectively. Additionally, by mapping coordinates between depth and color images, the system was able to extract comprehensive 3D information about the objects, showcasing the capabilities of the integrated sensing technology in facilitating precise spatial understanding and manipulation tasks. Some color images had missing depth values owing to camera mapping; they were compensated by replacing them with median values. Through image rotation calculations, the robotic arm accurately determined the optimal gripping angles for the objects. Finally, the robotic arm employed the computed path from the visual system to control the rotation angle of the gripper above the objects and effectively grip them, achieving the objective of handling stacked objects using 3D object information. The study encompassed the handling of objects of various colors as well as objects of the same color. Differently colored bottles were employed as objects of different colors, as they were larger and stacking scenarios were less complex, resulting in the successful completion of machine vision and robotic gripping tasks. To introduce greater complexity, we also intentionally utilized similarly colored objects such as screws, nuts, and wrenches for object recognition. Experimental results consistently demonstrated an average recognition rate of 0.938, even in scenarios with low-reflection image backgrounds. Consequently, the results of this research confirm the efficacy of the proposed method for handling stacked objects.

Moving forward, our research will focus on the utilization of grippers or suction cups for end-effector manipulation of various types of object. This will enable the robotic arm to adeptly manage objects with higher levels of variability.

Acknowledgments

This research was supported by the National Science and Technology Council of Taiwan under grants MOST 110-2221-E-130-011 and NSCT 112-2221-E-130-005.

References

1 X. Li, R. Cao, Y. Feng, K. Chen, B. Yang, C. W. Fu, Y. Li, Q. Dou, Y. H. Liu, and P. A. Heng: IEEE Rob. Autom. Lett. **7** (2022) 3961. https://doi.org/10.1109/LRA.2022.3149026
2 C. Kaymak and U. Aysegul: 2018 Int. Conf. Artificial Intelligence and Data Processing (IEEE, 2018) 1. https://doi.org/10.1109/IDAP.2018.8620916
3 T. Watanabe, K. Yamazaki, and Y. Yokokohji: Adv. Rob. **31** (2017) 1114. https://doi.org/10.1080/01691864.2017.1365010
4 A. H. Wei and B. Y. Chen: Int. J. Adv. Rob. Syst. **17** (2020). https://doi.org/10.1177/1729881420921102
5 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2016) 779. https://doi.org/10.48550/arXiv.1506.02640
6 K. Mikolajczyk and C. Schmid: IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 1615. https://doi.org/10.1109/TPAMI.2005.188

7 D. Lowe: Int. J. Comput. Vision **60** (2004) 91. https://doi.org/10.1023/B:VISI.0000029664.99615.94
8 A. Collet, D. Berenson, S. Srinivasa, and D. Ferguson: Proc. IEEE Conf. Robotics and Automation (IEEE, 2009) 48–55. https://doi.org/10.1109/ROBOT.2009.5152739
9 C. Kemp, A. Edsinger, and E. Torres-Jara: IEEE Robotics and Automation Magazine **14** (2007) 20. https://doi.org/10.1109/MRA.2007.339604
10 R. Beveridge, J. Griffith, R. Kohler, A. Hanson, and E. Riseman: Int. J. Comput. Vision **2** (1989) 311. https://doi.org/10.1007/BF00158168
11 J. Shi and J. Malik: IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 888. https://doi.org/10.1109/34.868688
12 C. Stauffer and W. Grimson: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 1999) 246–252. https://doi.org/10.1109/CVPR.1999.784637
13 B. Horn and B. Schunck: Artif. Intell. **17** (1981) 185. https://doi.org/10.1016/0004-3702(81)90024-2
14 K. Kim, J. Cho, J. Pyo, S. Kang, and J. Kim: Int. Conf. Control, Artificial Intelligence, Robotics and Optimization (IEEE, 2017) 237. https://doi.org/10.1109/ICCAIRO.2017.52
15 K. Harada, W. Wan, T. Tsuji, K. Kikuchi, K. Nagata, and H. Onda: Int. Symp. Experimental Robotics (2016) 646. https://doi.org/10.1007/978-3-319-50115-4_56
16 J. Tang, S. Miller, A. Singh, and P. Abbeel: 2012 IEEE Int. Conf. Robotics and Automation (IEEE, 2012) 3467. https://doi.org/10.1109/ICRA.2012.6224891
17 I. Lenz, H. Lee, and A. Saxena: Int. J. Rob. Res. **34** (2015) 705. https://doi.org/10.1177/0278364914549607
18 I. Goodfellow, Q. Le, A. Saxe, H. Lee, and A. Y. Ng: Advances in Neural Information Processing Systems **22** (2009).
19 H. Jiang, J. P. Wachs, and B. S. Duerstock: 2013 IEEE Int. Conf. Rehabilitation Robotics (IEEE, 2013) 1. https://doi.org/10.1109/ICORR.2013.6650447
20 D. Matsuno, R. Hachiuma, H. Saito, J. Sugano, and H. Adachi: 2020 IEEE 29th Int. Symp. Industrial Electronics (IEEE, 2020) 1409. https://doi.org/10.1109/ISIE45063.2020.9152510
21 C. Wang, R. Wang, and J. Zheng: 2019 IEEE 5th Int. Conf. Mechatronics System and Robots (IEEE, 2019) 129. https://doi.org/10.1109/ICMSR.2019.8835467
22 K. He, G. Gkioxari, P. Dollár, and R. Girshick: Proc. IEEE Int. Conf. Computer Visio (IEEE, 2017) 2980–2988.
23 I. Siradjuddin, L. Behera, T. M. McGinnity, and S. Coleman: 2012 IEEE World Cong. Computational Intelligence (IEEE, 2012) 1. https://doi.org/10.1109/IJCNN.2012.6252597
24 K.-T. Song and S.-C. Tsai: Proc. IEEE Int. Conf. Automation and Logistics (IEEE, 2012) 155–160. https://doi.org/10.1109/ICAL.2012.6308189
25 C.-H. Huang, C.-S. Hsu, P.-C. Tsai, R.-J. Wang, and W.-J. Wang: 2011 IEEE Int. Conf. Systems, Man, and Cybernetics (IEEE, 2011) 1699. https://doi.org/10.1109/ICSMC.2011.6083916

About the Authors

Shu-Yin Chiang received her M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1994 and 1999, respectively, both in electrical engineering. She is a professor in the Department of Information and Telecommunications Engineering, Ming Chuan University, Taiwan, R.O.C. Her research interests include intelligent control, artificial intelligence, robot applications, and wireless sensor networks. (sychiang@mail.mcu.edu.tw)

Yu-Kai Zhuo received his B.S. and M.S. degrees from Ming Chuan University, Taiwan, R.O.C., in 2021 and 2022, respectively, both in information and telecommunications engineering. His research interests include artificial intelligence and robot applications. (kevin510610@gmail.com)