

Received 10 July 2024, accepted 26 July 2024, date of publication 1 August 2024, date of current version 12 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3436849

RESEARCH ARTICLE

Developing a Cooking Robot System for Raw Food Processing Based on Instance Segmentation

KYUNGHOO JANG¹, JAEIL PARK¹, AND HYEUN JEONG MIN², (Member, IEEE)

¹Department of Industrial Engineering, Ajou University, Yeongtong-gu, Suwon-si, Gyeonggi-do 16499, Republic of Korea

²Department of Integrative Systems Engineering, Ajou University, Yeongtong-gu, Suwon-si, Gyeonggi-do 16499, Republic of Korea

Corresponding author: Hyeun Jeong Min (solusea@ajou.ac.kr)


This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by Korean Government through Ministry of Science and ICT (MSIT) under Grant 2021R1F1A1062194 and Grant 2021R1F1A1051242, and in part by the Ministry of Small and Medium-Sized Enterprises (SMEs) and Startups under Grant S3305062.

ABSTRACT This study presents an autonomous cooking robot system developed to improve culinary tasks through the classification and individual grasping of primary food materials. Our focus is on the recognition and manipulation of fried chicken parts and raw shrimp, which are essential in various culinary preparations, particularly frying. To distinguish and segment a specific target from a mix of similar objects, we utilize a Mask Region-based Convolutional Neural Network (Mask R-CNN) algorithm. Moreover, our robotic system incorporates a pose estimation technique to handle food materials of varying shapes. This system addresses the use of a direction vector transformed to determine 3D poses in real world, enabling a two-finger cooking robot to accurately grasp soft food materials. We have performed real robot experiments to demonstrate the system's ability to handle both fried chicken pieces and raw shrimp, verifying that our proposed method is effective. Additionally, we have confirmed the accuracy of our image segmentation approach.

INDEX TERMS Cooking robot, R-CNN, soft objects, food technology, hand robot, vision sensor.

I. INTRODUCTION

Robots are currently employed in various fields. For example, in addition to industrial robots in factories, robots serve as guides, and are present in household appliances and autonomous vehicles. With the rapid growth of technology, it is expected that the utilization of robots in our daily lives will continue to increase. The participation of robots in real-world environments is challenging, but much research has been performed on robots under dynamic and uncertain circumstances, such as cooking robots [1], [2]. This work presents a cooking robot system with a two-finger collaborative robot and a vision system along with kitchen appliances, as shown in Figure 1.

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng H. Zhu .

Pick-and-place tasks involving various types of food materials are essential for cooking robots [3]. This work considers two robotic tasks of cooking and serving a food recipe. We deal with food materials in the form of fried chicken pieces for serving, and in the form of raw shrimp for frying. The cooking robot faces two challenges in its operational tasks: first, the food materials might be stacked or piled up on each other, complicating the robot to discern and select the specific item to be picked up. Second, identifying an appropriate grasping point is challenging when dealing with food items that are soft and have irregular shapes and sizes.

To address these challenges, our approach involves segmenting the target object using a camera mounted on a robotic arm, focusing on identifying its mask while excluding the background. For determining an appropriate grasping point, our study addresses locating a thin area on the food material, which is crucial for handling delicate items like raw shrimp

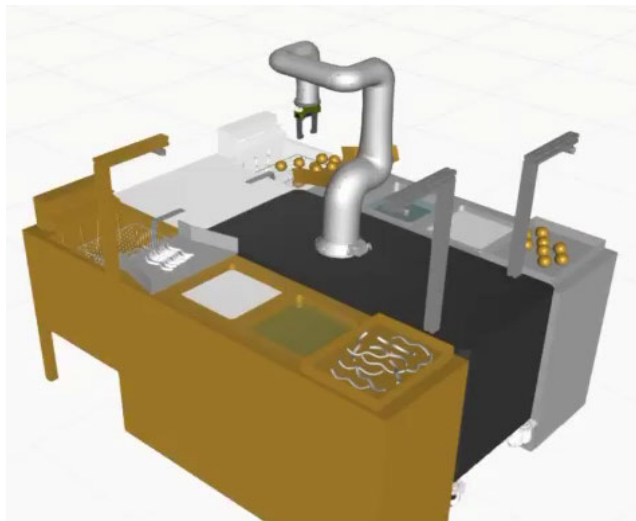


FIGURE 1. 3D image of our cooking robot system.

during the frying process. In this study, the robot hand is designed to pick up each shrimp individually and to dip it in a bowl containing a buttermilk mixture and then into a second bowl containing a flour mixture.

For the segmentation task, we employ the Mask R-CNN [4] framework, which has shown effectiveness in object segmentation. This method allows for precise delineation of the target object [5], [6]. Our segmentation integrates a deep residual network as the backbone of our training network. We leverage a pre-trained model from the COCO instance segmentation dataset [25]. The segmentation process determines the target object by calculating probabilities, which are obtained from the integrated losses across mask generation, object classification, and bounding box precision. To ensure the robot securely grasps the object without dropping it and to minimize any areas that remain uncoated with batter, we identify the object's thinner sections. To do this, we utilize the k-means clustering algorithm. This work determines the most suitable number of clusters (optimal K), which in turn identifies the direction vector for the thin area of the food material for grasping. For the experiments, we selected a recipe for buttermilk crispy fried chicken and raw shrimp as shown in Figure 2.



FIGURE 2. Typical material preparation for buttermilk crispy fried chicken, and chicken parts in a deep fryer.

The main contribution is the development of a two-finger robot system equipped with an onboard camera, designed

for tasks involving frying and serving. This system enables the robot to adapt its grasping strategy based on the depth variations in the captured images. Additionally, this study contributes to estimate the poses of a soft object having various shapes. By deriving the direction vector from pose estimation, the robot can determine a stable grasping point, thereby enhancing its speed and efficiency in tasks. Finally, the system has been tested and implemented on a real robotic platform, picking up and placing chicken parts and raw shrimp.

II. LITERATURE REVIEW

Research on food classification in images based on image segmentation has been conducted [8]. Additionally, studies have explored the selection of various materials by robots with robot control abilities [9], [10], [11], [12], [13], as well as real-time robotic application involving image segmentation [14], [15], [16]. In [14], real-time images are captured for a fast-moving objects in a robotic application, while [15] presents a CNN-based algorithm for effective robot grasping. Our framework handles real-time images acquired from a 3D camera. The robotic hand is controlled using pose estimation achieved through image transformation. This process entails extracting the center position and orientation of the selected object with the highest probability, followed by transforming the grabbing positions from image coordinates to real-world 3D coordinates. Our segmentation is based on deep residual learning [17], which shows better than previous algorithms [18], [19]. Depth estimation combined with RGB data is addressed in [20] in order to recognize the 3D shape of an object.

The recognition and picking of items with irregular shapes is considered difficult to automate in the food industry although there have been several related studies on food handling tasks. Sakamoto et al. discussed the handling of objects with varying visco-elasticity and adhesiveness properties and non-homogeneity such as sushi rolls by a robot hand [21]. Pettersson et al. designed a magnetorheological fluid gripper to handle a mixture of products such as vegetables and fruits arriving on a conveyor [22]. Likewise, the Korea Institute of Machinery & Materials has developed a robot hand capable of handling everyday jobs as delicate as holding soft tofu. In this robot hand, the shape of the distal end of a suction-type gripper is changed to fit the surface of the object being grabbed [23]. In this case, a large contact area is required to generate sufficient friction, and it is typically difficult to grasp thin objects. Conversely, robots constructed using soft materials, which are lightweight, can handle deformities and individual differences in the target objects. The use of vacuum suction pads is one solution for picking up fresh or packed food products. Wang et al. proposed a dual-mode soft gripper made of a rubber material that can grasp and suck different types of food materials with large variations in shapes and properties [12]. Suction pads located on the individual fingers of the gripper perform

vacuuming separately to maximize the success rate of suction. To grasp larger objects, Hawkes et al. proposed a gripper that utilizes shearing forces derived from controllable fibrillar gecko-inspired adhesives cast directly onto a thin film [24].

These papers proposed a robotic grasping system for automatically sorting objects based on machine vision. This system achieves the identification and positioning of target objects in complex background before using a manipulator to automatically grab the sorted objects [30], [31]. However, to the best of our knowledge, almost all the previous studies have not considered the location on a prepared food item for picking up and placing the food item. To solve this problem, in this paper, the picking up of food materials using machine learning is proposed and evaluated in grasping experiments with chicken parts.

III. COOKING ROBOT SYSTEM

In this section, we describe the object segmentation and the estimation of locations for grasping food materials. We elaborate on our framework in Section III-A, providing a comprehensive overview of how perception and action interplay within our robotic system. Next, we describe our segmentation algorithm and the process of annotating the dataset in Section III-B. We discuss the image transformation process along with finding its direction vector in Section III-C. Finally, we detail the two-finger grasping methodology employed by the robot hand in Section III-D.

A. FRAMEWORK

Our framework includes a deep learning recognition system and a robot grasping system that interact with each other. For the recognition system, we use a 3D camera. Our robotic hand has a two-finger gripper, and the pose is estimated by the image transformation process. Figure 3 shows the relationship between image training, inference, and pose estimation for a robotic hand in our framework. We use TCP/IP socket communications to transfer information such as the grasping points, orientation, and depth of an object from the camera system to the robot. A 3D camera is mounted on the robot hand as shown in the figure. Due to the movement of the robot hand, objects are differently represented in their angles or sizes on images. We discuss the dataset preparation process and the learning model in Section III-B and the pose estimation finding a thinner area along with a direction vector in Section III-C.

As depicted in the figure, a camera is mounted at the end of our robot arm. Each image is captured in real-time and used for target segmentation. We captured images at a rate of 2 frames per second (fps) to estimate a target object for grasping. The term ‘Inference food materials segmentation’ in the figure denotes the process of target segmentation using Mask R-CNN, as discussed in Section III-B. The selected target also requires estimation of the direction vector, covered in Section III-C, as well as control of the robot hand for grasping, detailed in Section III-D. The term ‘Pose estimation

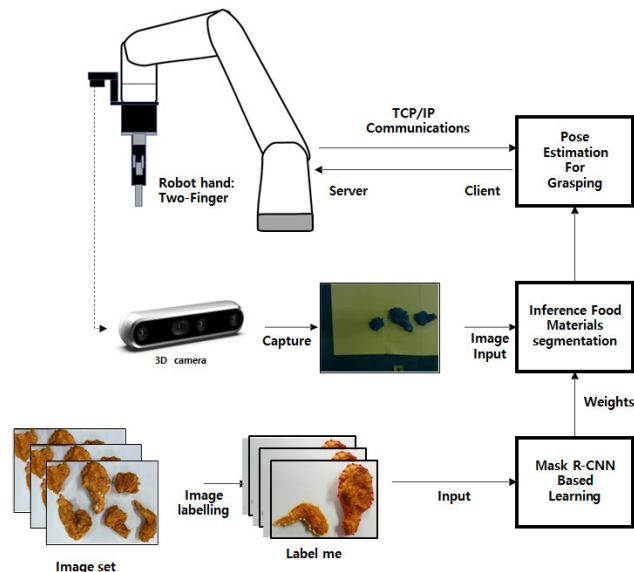


FIGURE 3. System framework.

for grasping’ in the figure denotes the estimation of the direction vector and the location of the grasping point.

B. TARGET SEGMENTATION AND DATASET

This section describes the dataset preparation process and the employed target segmentation method. Our approach capitalizes on the Mask R-CNN architecture for precise instance segmentation. The learning model, as illustrated in Figure 4, encompasses distinct components including a Feature Pyramid Network (FPN), Region Proposal Network (RPN), and Region of Interest (ROI) dedicated to both a box head and a mask head functionalities. The backbone network is structured by integrating a residual network (ResNet) with FPN. The residual network comprises either 50 or 101 hidden layers, denoted as ResNet-50 and ResNet-101, respectively. Subsequently, ROIs are aligned to facilitate classification and bounding box regression tasks. A fully connect layer (FC layer) is employed for the box head, whereas a fully convolutional network (FCN) is deployed for mask generation. During the training phase, each candidate region undergoes evaluation based on its total loss, computed as a composite of classification (I_{class}), bounding box regression (I_{bbox}) and mask (I_{mask}) losses, represented as $E = I_{class} + I_{bbox} + I_{mask}$. This target mask serves as a pose estimation for a robotic hand to find a stable location to grasp it.

Preparing an appropriate dataset for training and testing is crucial for the success of the learning model. Initially, we adapted the pre-trained COCO segmentation model from ModelZoo [12]. To align our labeling data with the COCO format, we developed our customized dataset focused on fried chicken parts and raw shrimp. Given the diverse shapes of chicken parts, we categorized them into distinct classes: drumstick, wing, thigh, back, wingette,

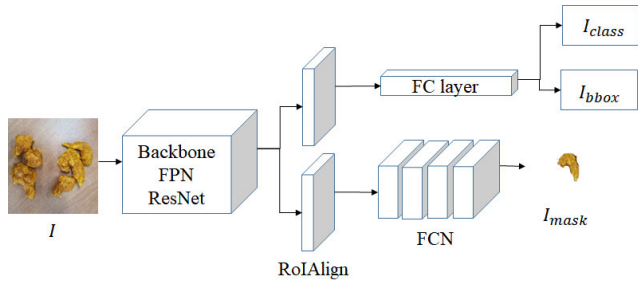


FIGURE 4. The architecture of mask R-CNN.

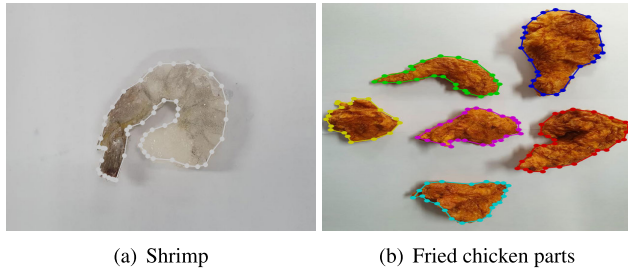


FIGURE 5. Examples of data annotation and labeling.

and breast. We also classified raw shrimp into one class: the shrimp itself. Figure 5 illustrates examples of data annotations used for training, where shrimp and chicken parts represented in distinct colors to differentiate among the classes.

For data augmentation, we utilized techniques to enhance the robustness of our training dataset. In our robotic system, the depth camera is positioned on the robot, necessitating that the data segmentation be adaptable to changes in camera perspective and motion. Moreover, the objects' scale in the images can vary due to factors such as the distance between the camera and the objects and the objects' differing inherent sizes. To address these challenges, we generated augmented data featuring scale-invariant representations of objects in a variety of sizes [29]. Our data augmentation strategy includes adjusting the scale of objects, encompassing both reductions and enlargements.

For original N dataset, each object is augmented according to the scale factor γ , which is chosen by the expert knowledge. Let N_{aug} be the number of augmented data. Let $D = \{(I_k, m_k, b_k)\}_{k=1}^{N+N_{aug}}$ be the dataset, where $I_k, m_k,$ and b_k are the k -th image, a set of annotated objects, and its bounding box, respectively. The k -th image is represented as $I_k = I_{img} \cup \gamma_j b_k$, where I_{img} is an empty image with the same size of the width and height as its original image or a randomly chosen image. Then data augmentation is varied depending on the chosen scale factors $\gamma = \{\gamma_i | i \in \mathbb{N}, -1 < \gamma_i < 1\}$. In the context, negative numbers indicate a reduction in the object's sizes, while positive numbers used to denote enlargements. In this work, we augmented dataset with the scale factors of $\gamma = \{0.2, 0.1, 0, -0.1, -0.2, -0.3, -0.4, -0.5, -0.6\}$.

C. ESTIMATION OF DIRECTION VECTOR

After the target object is selected, we need to estimate its numerical attributes to facilitate grasping. Given that our robot hand is equipped with a two-jointed gripper, we need to compute the object's position, orientation, and direction to ensure effective grasping. In addition, we assume that food materials are typically grouped together in a disorganized manner. Therefore, the robot must adept at singling out and picking up the designated target from a cluster of potential objects. We utilize the principal component analysis (PCA) algorithm to determine the orientation of the target and to find the best direction for grasping the target with a two-finger gripper [26].

The PCA method is based on an analysis of the eigenvalues of the covariance matrix and their corresponding eigenvectors. Because the target data on the image have x-and y-axis coordinates, we construct the covariance matrix as $\Sigma_{XY} = \begin{bmatrix} E_{XX} & E_{XY} \\ E_{YX} & E_{YY} \end{bmatrix}$, where E_{XY} denotes the covariance of X and Y . The orientation (θ) of the target is computed as

$$\theta = \arctan\left(\frac{v_{max}(y)}{v_{max}(x)}\right), \tag{1}$$

where $v_{max} = \arg \max_{\lambda} \{v \in V | \Sigma_{XY} v = \lambda v\}$. The coordinates of the target in an image along the x- and y-axes are estimated by the image segmentation. We then estimate the orientation θ of the target as that of its longest axis along which the largest amount of data is scattered by using the eigenvalues and eigenvectors of the covariance matrix regarding the data distribution.

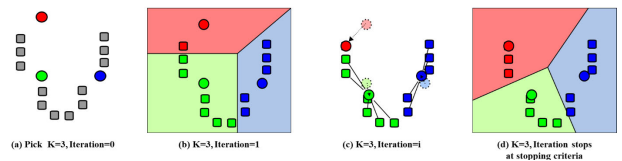


FIGURE 6. Iterative clustering to find the center of an object.

The direction in which the two-finger robot hand needs to grasp is then estimated to determine the optimal grasping location. In this work, the optimal grasp is identified at a thinner section of the object, as human chefs often prefer to pick up and hold slender portions of chicken parts when transferring individual pieces from batter mix to dry ingredients. It not only helps to reduce the areas of uncoated batter mix but also assists in preventing the chicken parts from being dropped by the robot's hand. To facilitate this, we employ the K-means clustering algorithm to identify the object's center on the image plane. In K-means clustering, data clusters with K distributions are established and the centroid is iteratively identified as shown in Figure 6. The objects' perceived points are used as input data for the K-means clustering, and the number of clusters within the selected object is adjusted to pinpoint the optimal grasping area. The center of mass is iteratively conducted for all points

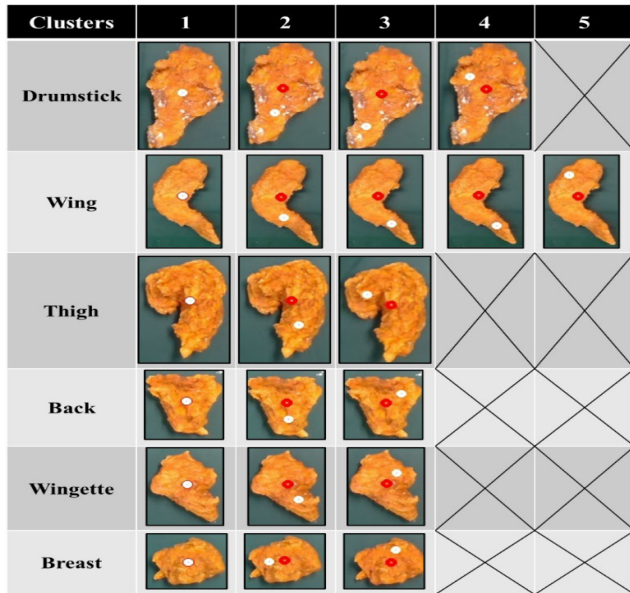


FIGURE 7. Centroid computation results for chicken parts with different K s.

within the target as follows:

$$\arg \min \sum_{x \in T} \|x - \vec{m}\|^2, \quad (2)$$

where $\vec{m} = (m_x, m_y)$ and T is the pixel coordinates in the target. The step of computing the centroid based on the distance of the points in each cluster from the cluster centroid is iterated until the following stopping criteria are met: (i) the centroids for all the clusters do not change, (ii) the points remain in the same cluster, and (iii) the maximum number of iterations is reached.

One common challenge when using the K-means clustering is the varying size of the clusters. Our solution is to implement K-means clustering incrementally from $K = 1$ and compare the results with those computed using the previous value of K [27]. This calculation is continued until the centroid of one cluster is located within a thinner area than those obtained from the previous clustering with $K - 1$ clusters. Figure 7 shows that this incremental K-means clustering is an appropriate strategy for selecting the correct number of clusters and finding a thin area. In the figure, the White circle and the red circle represent the cluster centroid with $K = 1$ and the centroid of the thin cluster with different K s, respectively. In the case of the drumstick, we stop this process at $K = 3$ because the centroid of one cluster has reached a thin area whereas the thin area found at $K = 4$ is not a suitable area for grasping. The centroids of objects located in the thin areas of chicken parts can hence be found using this procedure.

The aim of this clustering is to find the centroids of an object such as its center point (i.e., the centroid of the entire object) and its grasping point (i.e., the centroid within a thin area). Assuming that the centroid of an object is the center of

mass defined by its image points, the (arithmetic) mean of the point positions are computed separately for each dimension as follows:

$$\text{Centroid} = (c_x, c_y) = \left(\frac{\sum x_i}{N}, \frac{\sum y_i}{N} \right), \quad (3)$$

where (x_i, y_i) are the points in a cluster consisting of the object and N is the number of points. We assume that the objects are dense and flat to simply find the center of mass. We utilized a drumstick object to illustrate how the K-means clustering works. For the drumstick shown in Figure 9, the centroids are located at

$$(c_x, c_y) = (327, 226) \text{ at } K = 1, \quad (4)$$

$$(g_x, g_y) = (321, 280) \text{ at } K = 3, \quad (5)$$

using Eq. (3). As a result, the grasping point for the drumstick with different K is chosen to (g_x, g_y) . The centroids of various chicken parts with different K s are summarized in Figure 8.

Next, we estimate the direction vector in terms of the object's thinner section. For ease of understanding, the direction vectors are shown as in Figure 10, illustrating that the grasping points are positioned along a circle surrounding the object's centroid. Our direction vector is obtained as $\vec{v}_{dir} = (g_x, g_y) - (c_x, c_y)$ from the center of the object to an estimated grasping point. For instance, the direction vector of the drumstick is LEFT_DOWN since $\vec{v}_{dir} = (-6, +54)$ which represents $(-, +)$ as shown in the figure.

Finally, we measure the height of the grasping point in the thin area. This is achieved using an Intel® RealSense™ D435 depth camera in order to measure the distance from the target within the image frame to the camera. The depth information is realized by the transition from 2D detection to 3D detection when the camera is mounted on the side of the robot hand as shown in Figure 12. However, the depth information is prone to noise because of the low camera resolution. Because the RGB image of the RealSense camera has a one-to-one correspondence with the points in the depth map of interest, we filter out zero-value points in the depth map. By averaging the values of the remaining points, we determine the depth at the grasping point.

D. ROBOT HAND CONTROL FOR GRASPING

For the robot hand to grasp a target object, we utilize the n -point calibration method [32]. This method involves collecting n desired robot hand positions relative to an object, starting from a random position, to then estimate calibration coefficients. An instance of 3-point calibration is shown in Figure 11, where the object is positioned away from the robot hand. Let $\{(x_1, y_1), \dots, (x_n, y_n)\}$ be the set of target coordinates on the image plane, and $\{(x_{1d}, y_{1d}), \dots, (x_{nd}, y_{nd})\}$ denote the set of corresponding robot hand coordinates. The task is to estimate the calibration coefficients for both the x - and y -axes. Define $\vec{c}_x = [c_{x1}, c_{x2}, c_{x3}]^T$ as the vector of calibration coefficients for the x -axis and $\vec{c}_y = [c_{y1}, c_{y2}, c_{y3}]^T$ as the vector for the

Objects	Drumstick	Wing	Thigh	Back	Wingette	Breast
K-means Clustering						
Red: Center of object White: Grasping point						
K (Number of Clusters)	3	4	2	2	2	1
(c_x, c_y)	(327, 226)	(300, 253)	(314, 241)	(305, 201)	(305, 225)	(305, 227)
(g_x, g_y)	(321, 280)	(326, 298)	(326, 276)	(307, 221)	(312, 249)	(305, 227)

FIGURE 8. Grasping points of various objects.

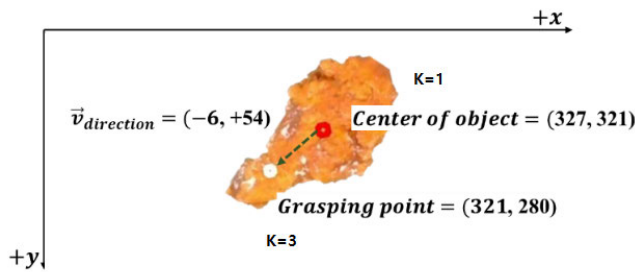


FIGURE 9. Example of a drumstick with its estimated center points.

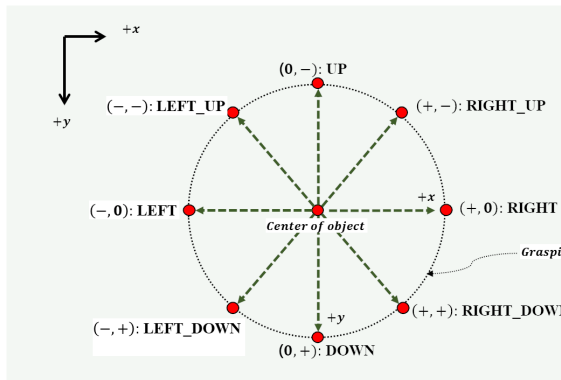


FIGURE 10. Estimation of the direction vector from the object's center to a grasping point.

y-axis. If $Z = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}$, $\vec{c}_x = (Z^T Z)^{-1} (Z^T \vec{x})$ and $\vec{c}_y = (Z Z^T)^{-1} (Z^T \vec{y})$, where $\vec{x} = [x_1 - x_{d1}, \dots, x_n - x_{dn}]^T$ and $\vec{y} = [y_1 - y_{d1}, \dots, y_n - y_{dn}]^T$. The moving distance of a robot hand are defined as follows:

$$d_x = g_x c_{x1} + g_y c_{x2} + c_{x3}, \quad (6)$$

$$d_y = g_x c_{y1} + g_y c_{y2} + c_{y3}, \quad (7)$$

where g_x and g_y are the image measurements representing the grasping point of a target object.

The grasping point (g_x, g_y) , direction (\vec{v}_{dir}) , and orientation (θ) of the object required for the robot grasping are obtained using the above procedures. We consider the example of a drumstick with the orientation $\theta = 32.1^\circ$. Its grasping point is $(321, 280)$, and its direction is LEFT_DOWN. To grasp the thin area of the drumstick, the two-finger gripper of the robot is moved to $(321, 280)$ and rotated by 57.9° if the object is aligned upward, as shown in Figure 13(a). The robot finger is then rotated backward by -57.9° . If the direction of the drumstick is RIGHT_UP, the two-finger gripper is rotated by 57.9° to grasp the drumstick, and then further rotated by 122.1° to align the drumstick upward, as shown in Figure 13(b).

IV. EXPERIMENTAL RESULTS

For the experiments, we first evaluate the accuracy of the object segmentation, then proceed with real-robot experiments. We capture and present snapshots of the robot in action, demonstrating its ability to manipulate objects based on the segmentation data. For the experimental environment, we prepared our own dataset with real food materials, and utilized a real robot. It is a collaborative robot named Indy7, recently made by Nuromeka in Korea.¹ To capture food materials' images, we positioned a depth camera directly above the robot hand, ensuring it pointed downward to obtain a clear and unobstructed view.

In the training phase, we constructed our dataset using actual food items to evaluate our robotic system's grasping capabilities, as discussed in Section III. For data collection, we initially captured a set of images in advance; however, the test dataset comprises real-time images obtained directly from the camera mounted on the robotic hand. In the real-time experiments with the robot hand, we attached a camera to the end of the robot arm. The camera captures two images every second, and one of them is used for segmenting the target object. Figure 14(a) shows an example of a real-time experiment with raw shrimps, including an image in the upper right corner where the target is marked with a red circle.

¹See more details at <https://en.neuromeka.com/cobot>

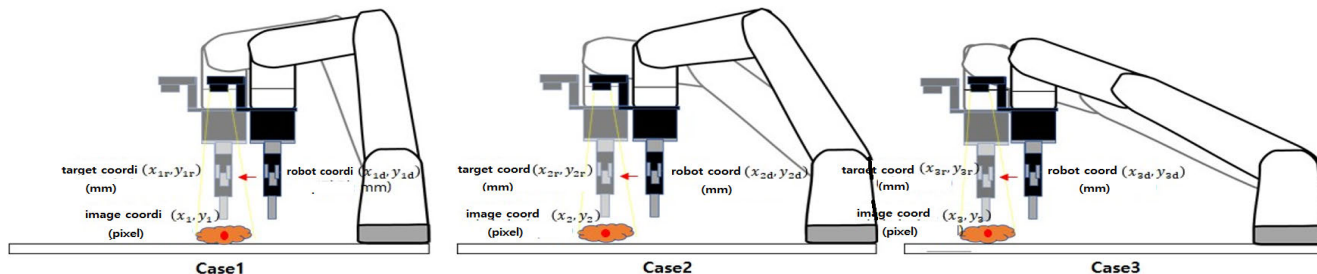


FIGURE 11. 3-point calibration.

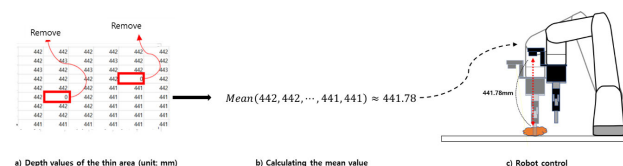


FIGURE 12. Target depth estimation flow.

TABLE 1. Dataset used in the experiment.

Category	# of instances	Category	# of instances
Breast	64	Thigh	64
Wing	64	Wingette	64
Drumstick	64	Back	40
Shrimp	144		
Total	504		

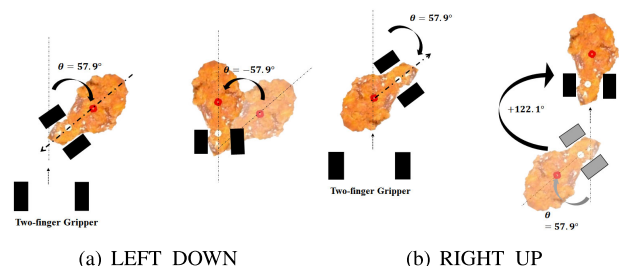


FIGURE 13. Examples of aligning a drumstick with the two-finger gripper.



FIGURE 14. Snapshots of identifying a target shrimp (marked with a red circle) from the camera mounted on the robot hand (a) and grabbing the target shrimp (b).

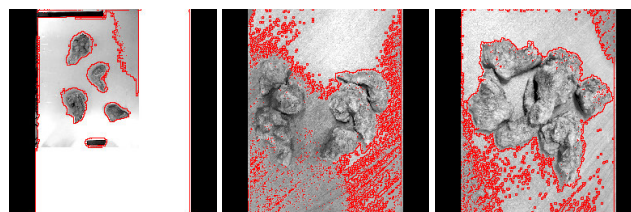


FIGURE 15. Results of segmentation for chicken parts by using U-Net.

The robot waits until the target is properly segmented and estimated, as discussed in Section III. Note that the scale of objects in each image may vary due to the different heights at which images are captured by the moving robot hand.

The food materials chosen for the experiments were primarily fried chicken parts. In an effort to diversify our dataset, we also incorporated raw shrimp. We utilized the annotation tool *labelme* [28] to annotate and label our training dataset. We categorized six different chicken parts: the chicken wing, wingette, thigh, drumstick, breast, and back. These categories are detailed in Table 1. For training

data, we used a total of 504 chicken parts and raw shrimp from the 7 different categories with the instances from each category. The dataset in the table includes data augmentation as described in Section III-B. We conduct a detailed analysis of the grasping points on these objects, examining the whole process.

For the learning model outlined in Section III-B, our computing system was equipped with an NVIDIA GeForce RTX 2070 Super GPU, utilizing CUDA 10.2 and pytorch 1.7 on a Linux platform. The dataset used for training consisted of images with various resolutions: 960×720 and 640×480 . In our experiment, we compared the total loss observed during the training phase when using a FPN combined with two different ResNet backbone network: ResNet-50 and with ResNet-101. For each training session, we conducted 4000 iterations in the learning process. As a result, the backbone network incorporating the ResNet-101 model exhibited slightly better performance compared to when ResNet-50 was used.

We compared the segmentation results with U-Net [7], and the results are shown in Figure 15. For the experiment, we used 15 training data images. The segmentation results are represented in red lines. As shown in the figure, the segmentation could distinguish isolated objects. However, when dealing with a group of objects, the segmentation

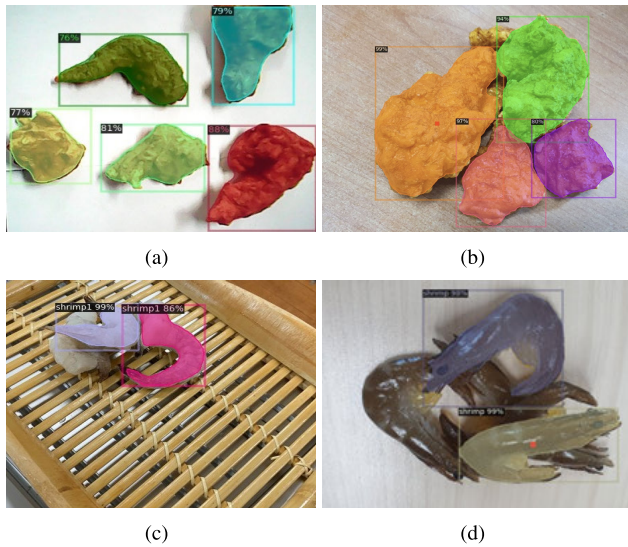


FIGURE 16. Object segmentation results. Each chicken part was correctly recognized when the parts were separated (left up) and when they over-lapped (right up). Each shrimp was identified (down).



FIGURE 17. Comparison of segmented objects from the camera placed with different heights of 35 cm (left) and 55 cm (right).

treated the entire objects as one unit, failing to differentiate individual items.

The trained model successfully segmented the chicken parts and raw shrimp as shown in Figure 16. The segmentation mask and its corresponding bounding box are illustrated using distinct colors. Figure 16(a) shows the segmentation effectiveness when the food items are spaced apart, whereas Figure 16(b) shows the results for when the items are stacked together. Furthermore, the chicken part highlighted within a small red box is shown as the target for the robotic hand’s grasping action. Specifically, the thigh and drumstick are selected for this purpose in the figure.

To compare the scale of training objects, we placed a fixed camera pointing down toward an object. We attempted to acquire segmented results at different heights from the camera. Figure 17 shows the compared results when the heights of the camera are 35 cm (left) and 55 cm (right), respectively. The black square marker at the bottom of the images is placed only to show the scale differences for comparison. Through our augmented dataset, the objects are correctly segmented, while the probabilities are compatible in those cases: 99% and 82% on the heights 35 cm (left) and 55 cm (right), respectively.

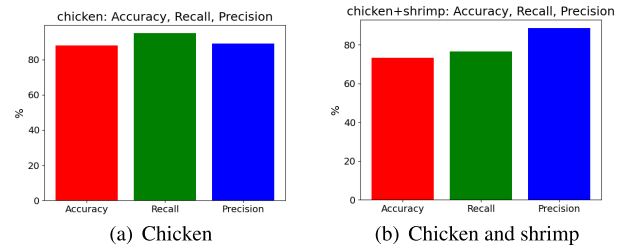


FIGURE 18. The results of the accuracy, recall, and precision.

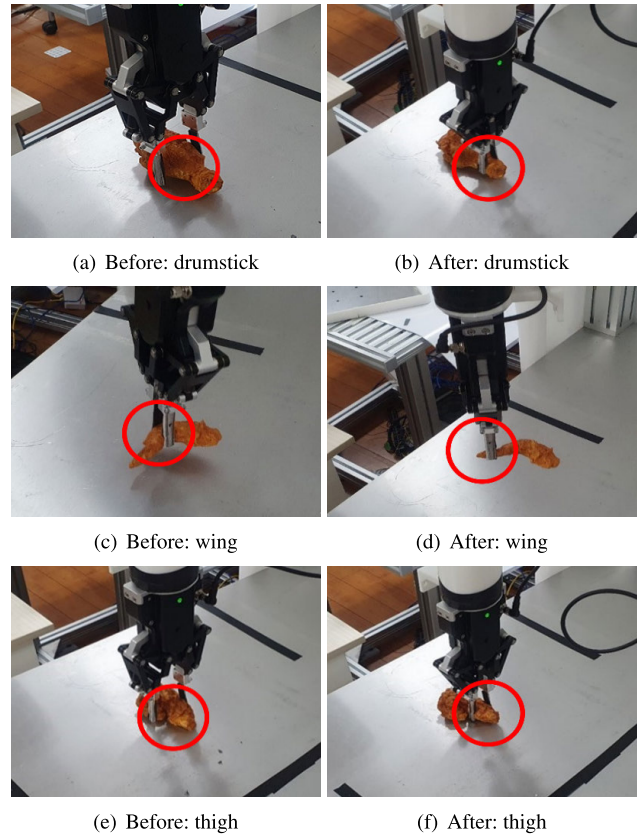


FIGURE 19. Before/After transforming to grasp the thin areas of chicken parts with a robot hand: Drumstick (up), Wing (middle), and Thigh (down).

We analyzed the accuracy of the learning model for the selected region among the segmented objects. For quantitative evaluation, we analyzed the accuracy, precision, and recall. The accuracy indicates how many objects were correctly segmented to provide information for grasping. The precision decreases if the learning model chooses an incorrect object, and the recall decreases if the learning model misses the selection of a correct object. We also analyzed the accuracy, recall, and precision when we apply the mixed dataset of fried chicken parts and raw shrimp. Figure 18 shows the results; the use of training data with chicken parts in Table 1 is shown in Figure 18(a) when using the 101-layer model, and Figure 18(b) shows the result when using augmented dataset including shrimps. Here, the test images are 34 while the number of objects on the entire



FIGURE 20. A sequence of a cooking robot system for picking up the thin areas of shrimp.

images are 101 along with the 81 true objects and the 20 false objects.

Next, we analyzed the grasping coordinates for the selected soft object. In the experiments, we tried to segment and grasp the chicken parts. The coordinates and dimensions observed in the image were adjusted to align with the robot hand's coordinate system. In Figure 19, the grasping coordinates for the chosen object are indicated by a red circle, showing the specific point where the robot is designed to grasp the object.

Finally, Figure 20 shows the robot system's process of cooking raw shrimp. The robot hand handles each shrimp, immersing it sequentially into two different mixtures: first into a bowl of buttermilk and then into a bowl of flour. This step-by-step procedure ensures each shrimp is adequately coated. The points at which the robot grasps the shrimp were determined using the approach discussed in Section III-C. The process of coating a single shrimp, which includes real-time segmentation and the subsequent handling by our robot hand, took approximately 19 seconds, dipping into water, then coating with the flour mixture, and finally positioning the shrimp for frying.

During the experiment, the camera mounted on the robot arm captures two images every second, as depicted in the figure. The robot continues to capture images until the target shrimp is identified, and then the grabbing location is estimated. Once the robot hand has grasped a shrimp, it is immersed in a bowl of buttermilk. We repeat the dipping process twice to ensure thorough coating, as illustrated in the third figure in Figure 20. The dipping process in buttermilk takes approximately 5 seconds. Subsequently, the robot hand dips the shrimp into a bowl of flour, employing two half-circle motions and two straight line motions from left to right and up and down to ensure even coating, as shown in the fourth figure in Figure 20. The process in flour takes about 8 seconds.

V. CONCLUSION

In this study, we performed object detection, selection, and segmentation through deep learning based on Mask R-CNN in which instance segmentation of an object is combined with the recognition of food ingredients. In addition, we applied PCA and the K-means clustering algorithm to the basic object information to determine the orientation and center of the object and to perform pose estimation through point conversion for grasping a thin area. The challenges of recognizing and grasping food materials, such as chicken

parts and shrimp, were solved in the cooking robot. We will perform additional deep learning to train the system to handle more varieties of fried foods and continue to enhance the postural estimation for grasping food materials in the future.

REFERENCES

- [1] Z. Wei, N. Huang, and S. Fan, "Design and implementation mechanism of cooking robot based on digital kitchen," in *Proc. Asia-Europe Conf. Electron., Data Process. Informat. (ACEDPI)*, Apr. 2023, pp. 482–488.
- [2] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen, "Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5159–5166, Apr. 2022.
- [3] E. Misimi, A. Olofsson, A. Eilertsen, E. R. Øye, and J. R. Mathiassen, "Robotic handling of compliant food objects by robust learning from demonstration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6972–6979.
- [4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
- [5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [6] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 350–359.
- [7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
- [8] S. Aslan, G. Ciocca, D. Mazzini, and R. Schettini, "Benchmarking algorithms for food localization and semantic segmentation," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 12, pp. 2827–2847, Dec. 2020, doi: 10.1007/s13042-020-01153-z.
- [9] A. Zeng et al., "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *Int. J. Robot. Res.*, vol. 41, no. 7, pp. 690–705, Jun. 2022, doi: 10.1177/0278364919868017.
- [10] N. Kimura, R. Sakai, S. Katsumata, and N. Chihara, "Simultaneously determining target object and transport velocity for manipulator and moving vehicle in piece-picking operation," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 1066–1073.
- [11] J. H. Park, J. Kim, and H. J. Kim, "Spatio-semantic task recognition: Unsupervised learning of task-discriminative features for segmentation and imitation," *Int. J. Control, Autom. Syst.*, vol. 19, no. 10, pp. 3409–3418, Oct. 2021.
- [12] Z. Wang, K. Or, and S. Hirai, "A dual-mode soft gripper for food packaging," *Robot. Auto. Syst.*, vol. 125, Mar. 2020, Art. no. 103427.
- [13] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6D object pose estimation for robot manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3665–3671.
- [14] B. Liu and Y. Zheng, "Intelligent real-time image processing technology of badminton robot via machine vision and Internet of Things," *IEEE Access*, vol. 11, pp. 126748–126761, 2023.
- [15] Y. Xu, L. Wang, A. Yang, and L. Chen, "GraspCNN: Real-time grasp detection using a new oriented diameter circle representation," *IEEE Access*, vol. 7, pp. 159322–159331, 2019.

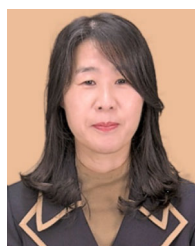
- [16] P. Rosenberger, A. Cosgun, R. Newbury, J. Kwan, V. Ortenzi, P. Corke, and M. Grafinger, "Object-independent human-to-robot handovers using real time robotic vision," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 17–23, Jan. 2021.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [20] Y. Song, J. Wen, D. Liu, and C. Yu, "Deep robotic grasping prediction with hierarchical RGB-D fusion," *Int. J. Control, Autom. Syst.*, vol. 20, no. 1, pp. 243–254, Jan. 2022.
- [21] N. Sakamoto, M. Higashimori, T. Tsuji, and M. Kaneko, "An optimum design of robotic hand for handling a visco-elastic object based on Maxwell model," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 1219–1225.
- [22] A. Pettersson, S. Davis, J. O. Gray, T. J. Dodd, and T. Ohlsson, "Design of a magnetorheological robot gripper for handling of delicate food products with varying shapes," *J. Food Eng.*, vol. 98, no. 3, pp. 332–338, Jun. 2010.
- [23] (2020). *Korean Researchers Develop New Robotic Gripper Almost Like Human Hand*. [Online]. Available: <https://pulseneews.co.kr/view.php?year=2020&no=1057688>
- [24] E. W. Hawkes, D. L. Christensen, A. Kyungwon Han, H. Jiang, and M. R. Cutkosky, "Grasping without squeezing: Shear adhesion gripper with fibrillar thin film," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2305–2312.
- [25] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: <https://github.com/facebookresearch/detectron2>
- [26] I. T. Jolliffe, *Principal Component Analysis* (Springer Series in Statistics), 2nd ed., Springer, 2002.
- [27] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Springer, 2006.
- [28] K. Wada. (2016). *Labelme: Image Polygonal Annotation With Python*. [Online]. Available: <https://github.com/wkentaro/labelme>
- [29] H. J. Min, "Generating synthetic dataset for scale-invariant instance segmentation of food materials based upon mask R-CNN," *J. Inst. Control, Robot. Syst.*, vol. 27, no. 8, pp. 502–509, Aug. 2021.
- [30] W. T. Abbood, O. I. Abdullah, and E. A. Khalid, "A real-time automated sorting of robotic vision system based on the interactive design approach," *Int. J. Interact. Design Manuf.*, vol. 14, no. 1, pp. 201–209, Mar. 2020.
- [31] M. H. Ali, K. A. Kizat, K. Yerkhan, T. Zhandos, and O. Anuar, "Vision-based robot manipulator for industrial applications," *Proc. Comput. Sci.*, vol. 133, pp. 205–212, Jan. 2018.
- [32] P.-Y. Li and Z.-W. Li, "Study on calibration algorithm of embedded touch screen," *J. Multimedia*, vol. 9, no. 4, pp. 605–610, Apr. 2014.



KYUNGHOO JANG received the B.S. and M.S. degrees in industrial engineering from Ajou University, South Korea. As the CEO of Cobotsys, he leads the company in pioneering innovations in developing sequence control for robots and various industrial communication devices. Recognized as an Expert in industrial communication, he has overseen the development of the robot execution system (RES) solution, an advanced system designed to integrate robotics into the food and beverage (F&B) industry. Currently, he is involved in research and development to integrate robotics into the food and beverage (F&B) sector, focusing on creating innovative solutions that blend AI with robotics to streamline processes, enhance efficiency, and redefine the role of robotics in food preparation, service, and quality control.



JAEL PARK received the Ph.D. degree from Pennsylvania State University, in 2005. He is currently a Professor with the Department of Industrial Engineering, Ajou University, South Korea. He is also the CEO and the Co-Founder of Cobotsys, where he leads a team in the design and implementation of unmanned processes using collaborative robots. He also serves as an Advisor to the government agency, South Korea, consulting on the adoption of robots in small and medium-sized manufacturing enterprises. His research interests include addressing the intricate challenges and crafting innovative solutions in advancing automation and robotics across various sectors, including manufacturing, agriculture, and food preparation



HYEUN JEONG MIN (Member, IEEE) received the Ph.D. degree from the Department of Computer Science, University of Minnesota, in 2013. She was a Post-Doctoral Researcher and a Research Professor with Yonsei University, from 2014 to 2019. She has been an Assistant Professor with the Department of Integrative Systems Engineering, Ajou University, South Korea, since 2020. Her research interests include robot vision, multi-robot planning, visual tracking, and multimodal analysis.

• • •