Proceedings of the SICE Annual Conference 2020
September 23-26, 2020, Chiang Mai, Thailand

Grasping motion for small non-rigid food using instance semantic segmentation

Wiwat Tulapornpipat[1†], Alongkorn Pimpin[1], Sakmongkon Chumkamon[2], and Eiji Hayashi[2]

[1] Department of Mechanical Engineering, Faculty of Engineering,
Chulalongkorn University 254 Phayathai Road, Pathumwan, Bangkok 10330 Thailand
(E-mail: w.tulapornpipat@gmail.com)
[2] Mechanical Information Science and Technology, Kyushu Institute of Technology,
680-4 Kawazu, Iizuka-shi, Fukuoka, 820-8502, Japan

Abstract: The importance of food automation becomes a challenge in this era since food is an essential factor for humans. In this paper, we designed the robot framework to autonomously generate grasping motion for non-rigid food objects. The system could recognize and localize objects and target regions for pick-and-place rather than only the position. Assembling a lunch box with different food required advance in both hardware and software to create an efficient process. The robot platform based on a seven-axis industrial robot arm equipped with instance segmentation based on Cascade Mask R-CNN for Japanese food. A modular end effector was designed and prototyped to facilitate soft gripper and vacuum pad on the single unit, which allows the system to handle different food objects. In the experiment, we also evaluated the performance of pick-and-place process. The system can successfully pick-and-place food into a lunch box with an outcome successful rate of 90%.

Keywords: Instance semantic segmentation, Motion planning and Factory automation

1. INTRODUCTION

Technological transformation not only disrupts many businesses but also changes the standard production process. The significance of robotics in the food industry is rising since it is more affordable and more intelligent with help from Artificial Intelligence (AI). In the automated food assembling process, the system must recognize and localize both objects and goals, plan and execute motion to pick objects, then place it in the region of goal. Therefore, the image processing technologies typically integrate with the industrial robot arm. The integrated system could do many precision tasks, such as red meat processing[1], food packaging, and assembling.

There are four interesting applications in image processing. Firstly, object detection (localization and classification) identifies objects with bounding boxes and labels. Secondly, semantic segmentation classifies each pixel and separates the region of pixels with different classes. Thirdly, instance segmentation further separate objects of the same class to be an individual region of each object. Fourthly, panoptic segmentation combines instance, and semantic segmentation also recognizes background. Among these four applications, the automated food assembling process needs at least instance segmentation because each lunch box ingredients mostly on the tray, so we must successfully recognize it individually.

Another issue is the handling technique. There are three main types of the gripper, which are vacuum pad, rigid gripper, and soft gripper [2]. Three main criteria, which are workability in clean environments, capability of grasp or pick various objects, and damage to food products, are usually used to choose the technique. Table 1 compares the criteria of each handling technique.

For example, a vacuum pad can pick various objects within load capacity when the vacuum pad is well suitable with object surface conditions (flat/curve, greasy or not, ductile/brittle). However, a vacuum pad is not the right choice when considering about clean environment such as the food industry because it draws in air from the opening of pad, so any debris or contaminant could clog the system. Nevertheless, a vacuum filter can suppress this problem. Rigid gripper typically picks only particular objects that match the custom design of the finger. Nevertheless, the rigid gripper is suitable for clean environments if designed correctly.

On the contrary, soft gripper, which by principle is a multibody mechanism in the way that different sections of the finger are not equally deformed, could grasp non-rigid objects. Therefore, it becomes a gripper of choice nowadays. However, the main drawback is the size of the system that is the biggest among these three. Both vacuum pad and soft gripper inflict little bruise, tear, and deformation damage to the food [3].

Table 1 Comparison between three types of gripper

<table>
<tr><th rowspan="2"></th><th rowspan="2">Vacuum pad</th><th colspan="2">Gripper</th></tr>
<tr><th>Soft</th><th>Rigid</th></tr>
<tr><td>Variable force</td><td>Yes</td><td>Yes</td><td>No</td></tr>
<tr><td>Multi-purpose</td><td>Yes</td><td>Yes</td><td>No</td></tr>
<tr><td>Clean environment</td><td>No</td><td>Yes</td><td>Yes</td></tr>
<tr><td>Size</td><td>Small</td><td>Large</td><td>Medium</td></tr>
</table>

In our research group, we have implemented Robot Operating System I (ROS-I) because it is a common framework in the research community and industrial robotics; therefore, the scaling-up from research to industrial-scale need little to no additional work [4].Our core platform is MoveIt! because it is an open-source robotic manipulation platform for prototyping designs and benchmarking algorithms [5]. Lastly, Open Motion Planning Library (OMPL) is used because it contains state-of-the-art of sampling-based motion planning algorithm, which creates the graph as it searches and does not require prior graph structure [6]. The sampling-based methods are suitable for industrial robot arm because it would be computationally expensive to explore every possibility in the graph to find the optimal motion.

In this paper, we explained the recent development to expand the ability of our system for the food industry. The current goal is the implementation of the developed system in the Japanese food industry for objects such as karaage, onigiri, bologna, and lunchbox.

In the hardware part, we proposed a modular end effector, which was designed from the ground up to facilitate both vacuum pad and soft gripper in a single unit, and this unit has designated area for co-axis IMU. Vacuum pad should be suitable for bologna and lunchbox, while soft gripper should be suitable for karaage and onigiri. In the software part, we proposed a proprietary instance segmentation model.

2. METHODOLOGY AND DESIGN

As mentioned earlier, we proposed a modular end effector, which has designated area for co-axis IMU. Furthermore, we also add a 6-axis force-torque sensor and an RGB-D camera to achieve the task of handling objects. Figure 1 shows the hardware details of the proposed design.

2.1 Hardware Design

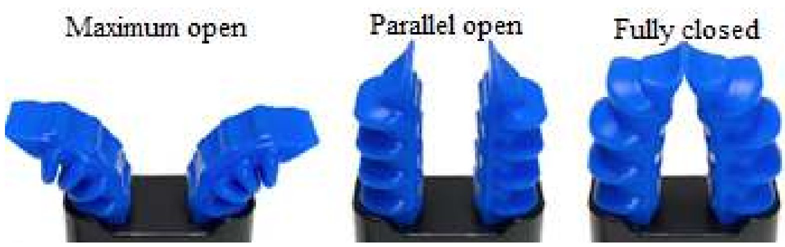
There are many shapes of vacuum pad, such as standard for flat objects, deep for round shape, sponge, and multi-bellow can pick uneven surface objects. Nevertheless, multi-bellow can handle fragile objects too, which is suitable for bologna and box. We used a multi bellow vacuum pad, which has ten mm diameter, four mm bellows stroke, and eight Newton of theoretical suction force at -100 kPa vacuum, as shown in Fig.1a.

There are several types of soft gripper, such as contact-driven compliant fingers [7], tendon-driven [8], fluidic elastomer actuators. Soft Robotics mGrip was chosen because it is a commercial fluidic elastomer soft gripper modular system. The mGrip's maximum pressure of compressed air is 14psi. There are eight on-the-fly selectable profiles, which can pre-defined pressure and opening width, as shown in Fig.1b.

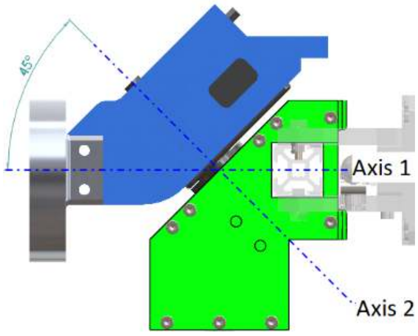
The branching point in our design is the question of how we attached both gripper in a single unit. We could attach both grippers with a rigid structure[9] and use degrees of freedom of industrial robot arm to select gripper. This approach has the advantage of low manufacturing and maintenance costs since it requires none of the active actuators, and it uses little additional space. On the contrary, this approach leads to complications in motion planning to specify additional goals as to which gripper pick objects. Moreover, if we need to measure the weight of objects, it will be challenging to vectorize gravity. So, our proposed design uses another servomotor to select the type of gripper by rotation in the auxiliary axis. This approach eliminates previously mentioned complications because the axis of both grippers was maintained as if it directly attached to the industrial robot arm, as shown in Fig.1c.

Dynamixel MX-106R servomotor was chosen because of twenty Newton of allowable axial load in a compact size and weighs only 153 grams. Additionally, we can control position within 360 degrees at a 12-bit resolution and visualize the servo motor state with ROS and RViz [10] via 12V RS485-USB converter.

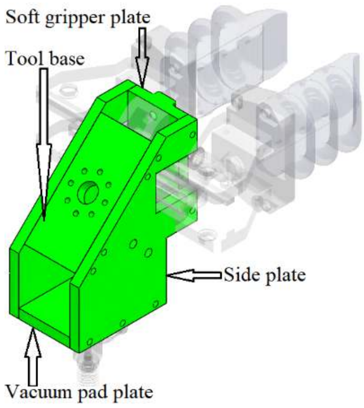
a) Theoretical suction force



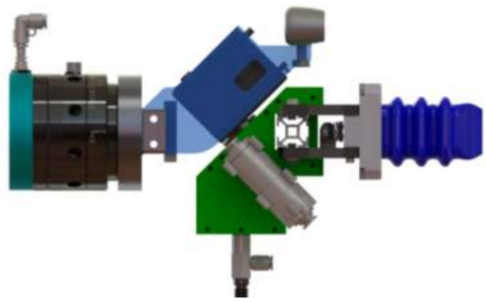
b) Difference opening width of Soft Robotics finger



c) Auxiliary axis of rotation



d) Components of tool base



e) CAD of modular end effector

Fig. 1 Detail of proposed design

We designed our end effector with modularity design for ease of manufacturing. Both grippers attached to individual plates then assembles to side plates and base plate, which attached to the servo motor, as illustrated in Fig.1d. This design also allows for a fast tool change.

Since this research cooperates with Yaskawa Japan, we used Motoman SIA5F 7-axis industrial robot arm because of its best-in-class wrist performance gives us freedom of reachable space with repeatability at ±0.05mm.

Robotiq FT300 6-axis force-torque sensor was installed for an object's weight measurement. This sensor has little signal noise when compared to a full-scale value of three hundred Newton resultant force in three axes and the minimum threshold for the static state (the smallest variation that sensor can detect reliably) as shown in Table 2. Additionally, this sensor also provides a ROS package to communicate via 24V RS485-USB converter up to 100 Hz, and the stiffness of this sensor is high enough to allow the attached tool to perform precision tasks, as shown in Table 3.

Table 2 Characteristics of Robotiq FT300

	Signal noise	Minimum threshold
Force in X, Y, Z	0.1 N	1 N
Moment in X, Y	0.005 Nm	0.02 Nm
Moment in Z	0.003 Nm	0.01 Nm

Table 3 Stiffness of Robotiq FT300

	Axis X, Y	Axis Z
Force	3.2 x 10^6 N/m	3.9 x 10^6 N/m
Moment	4.7 x 10^3 Nm/rad	4.6 x 10^3 Nm/rad

9-axis IMU with Bluetooth Low Energy (BLE) 4.1 from LP-Research was used for force-torque sensor calibration and gravity compensation. This sensor can transmit data up to 400 Hz in quaternion format (ROS native format). The battery lasts six hours with power consumption 132mW at 3.3 V.

A three-way vacuum valve (CKD VSXP-T666) was chosen because it significantly reduces the vacuum release time. Two-way valves use only vacuum break air but three-way valves also let atmospheric pressure from additional airway into the system to equalized vacuum to atmospheric pressure.

The system must monitor the vacuum pressure via a built-in analog pressure sensor of vacuum ejector and pause the operation until vacuum pressure reached the threshold because different object requires difference vacuum pressure.

A six-litre oil-less air compressor capable of 0.7 MPa with an automatic pressure switch at 0.49 MPa was chosen for soft gripper and vacuum break air supply. Two-stage oil-less vacuum pump capable of -100 kPa vacuum without reservoir was chosen for a vacuum pad.

Our system has two RGB-D cameras, Intel Realsense D435i was attached to the modular end effector as a wrist camera for instance segmentation, and Microsoft Azure Kinect DK was attached to a fixed frame for object avoidance with 3D reconstruction. In addition, Intel Realsense D435i has IMU, active IR stereo depth sensing and RGB sensor. Microsoft Azure Kinect DK has IMU, 1 MP Time-of-Flight (ToF) depth, and RGB sensor.

Computer-Aided-Design of the modular end effector is shown in Fig. 1e. Arduino UNO was used to actuated the ejector and break valves, read the pressure of vacuum ejector, actuated and select profiles of the soft gripper. Furthermore, Arduino also controlled power of the vacuum pump, as shown in Fig. 2.

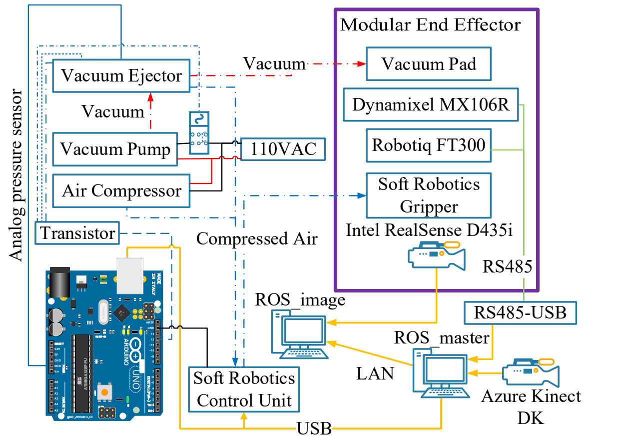


Fig. 2 Hardware diagram of modular end effector

2.2 Software Design

Firstly, the system must know where objects and goals are. Therefore, our instance segmentation algorithm is cascade mask R-CNN [11]. We used R-50-FPN backbone, which has less performance in terms of mask and bounding box average precision but more memory efficient (use lower memory but has faster inference time) than X-101-64x4d-FPN as shown in Table 4 [12], with PyTorch style and learning rate of 1x to train the model. Model for Japanese food such as piles of overlapped karaage, onigiri, box, and bologna was trained for 5,000 epochs with various sizes of images and area of segments, and a number of objects per image as shown in Table 5 with additional 42 images of workspace. The dataset has a broad range of annotation's area to strengthen the model.

Table 4 Instance segmentation on COCO test-dev

Backbone	Mem (GB)	Inf time (fps)	Mask AP	Box AP
R-50-FPN	6.0	11.2	35.9	41.2
X-101-64x4d-FPN	12.2	6.7	39.2	45.3

Table 5 Area of images and annotation in the dataset

		Quantity	Area [pixel^2]		
			Median	Std Dev.	Average
Total image		267	330,000	705,578	611,606
Annotation	Karaage	531	4,587	18,840	11,063
	Onigiri	350	10,191	114,722	38,777
	Lunchbox	105	37,983	31,584	41,957
	Bologna	213	11,214	116,519	42,774

3D centroid from each instance segment must be extracted to represent the object's coordinates in ROS. Start with a raw RGB image (Fig. 3a), then an instance segmented image (Fig. 3b). After that, OpenCV was used to extract 2D centroids from instance segments (illustrated in Fig. 3c) and deproject 2D centroids pixel coordinates (shown in Fig. 3d) to 3D distance coordinates with Eq. (1). Then, they were sorted by depth and 2D distance for coordinates with the same depth. We represent 3D coordinates with ROS markers, as shown in Fig. 4.

$$\begin{aligned} x &= (U - ppx)/fx * D * depth\_scale, \\ y &= (V - ppy)/fy * D * depth\_scale, \\ z &= D * depth\_scale \end{aligned} \tag{1}$$

where U, V, and D are horizontal, vertical and depth pixel value, ppx and ppy are the pixels value of the center of projection, fx and fy are the focal lengths of the image, and depth_scale is distance per depth pixel value.

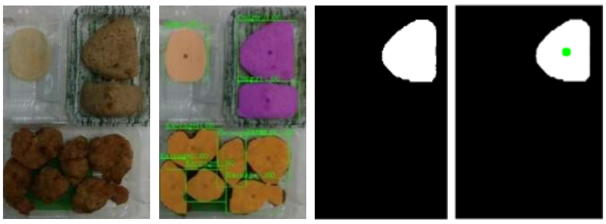

a) b) c) d)

Fig. 3 2D centroid pixel coordinates extraction process

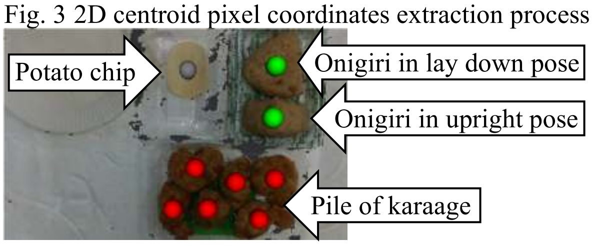


Fig. 4 Markers of 3D centroid visualized in RViz

ROS topics, which actuated both grippers with vacuum pressure monitoring and soft gripper's profile selection, was created to bridge the communication between ROS_master and Arduino UNO.

Graph search algorithms assume the deterministic system, so the majority of graph search found an optimal path because of exhaustive search. Sampling-based rapidly-exploring random trees (RRT*) [13] was used to generate the pick-and-place motion from object to the goal location.

2.3 System Evaluation

We conducted our experiment to evaluate pick and place tasks. The task started from ready to pick pose where our wrist camera sees most of the workspace, karaage, which is an object in the elliptical area and lunchbox, which is a goal in the rectangular area as shown in Fig.5. Then, 3D coordinates of objects and goals were fed into ROS network from ROS_image node, which is image processing PC (Intel i9-9900k with Nvidia Geforce RTX 2080Ti and 32GB of RAM). Then, master node (Dell Inspiron 7590) executed the procedure to pick karaage and place it in the lunchbox. The karaage was randomly placed within the area, which occasionally varied within ± 10cm related to the original location.

We deliberately to not vary orientation of objects because of two reason. Firstly, our system doesn't have object pose estimator. Secondly, orientation of non-rigid food doesn't matter because size of karaage is about the same in any orientation.

We also experiment food assembling process with karaage, onigiri in upright pose (camera see as rectangular from above), and stacked bologna, as illustrated in Fig.6.Firstly, the robot arm picks and places both karaage and onigiri with soft gripper in succession. Secondly, it changes the tool to vacuum pad. Then, it picks and places bologna.

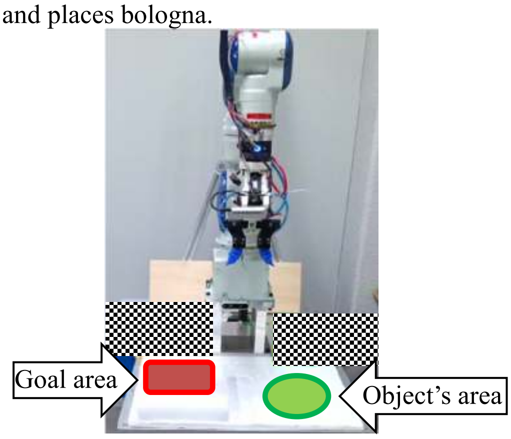


Fig. 5 Objects and goal area of karaage pick-and-place experiment

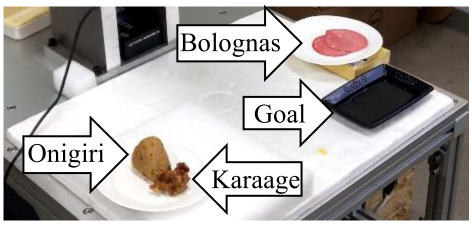


Fig. 6 *Environment setup for food assembling process*

3. EXPERIMENTAL RESULTS

From a total of 22 experiments, there are twenty of a normal success task which soft gripper fully grasp karaage and place it in the box, as shown in Fig.7. Only one abnormally success task, which soft gripper partially grasps karaage because the left finger of the gripper is caught the edge of paper plate but still successfully places it in the box as shown in Fig. 8. Only one failed task, which is soft gripper cannot grasp karaage because the gripper push karaage out while grasping it, as shown in Fig. 9.

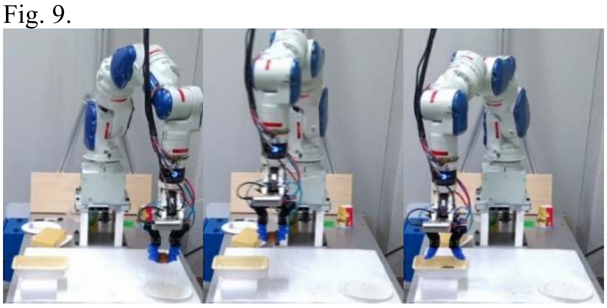


Fig. 7 Normal success task

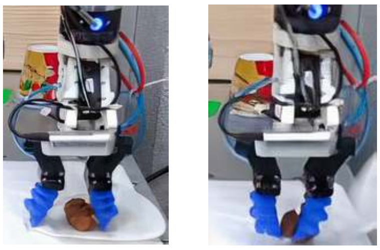
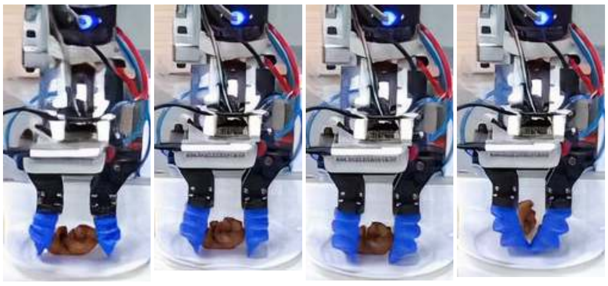


Fig. 8 Abnormally success task



a) Gripper approach karaage and align main axis with centroid of karaage

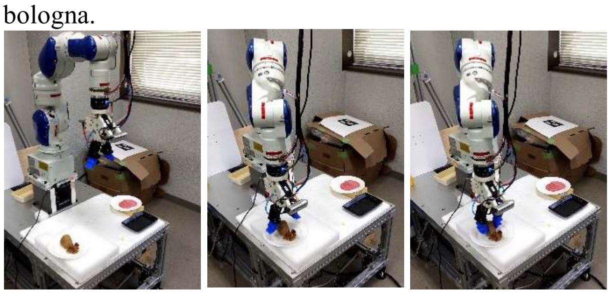
b) Right and left fingers have unequal displacement due to centroid lean toward right.

c) Karaage started to be push back.

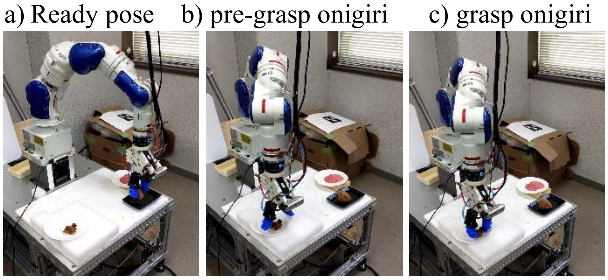
d) Karaage is fully out of grasp.

Fig. 9 Failed task

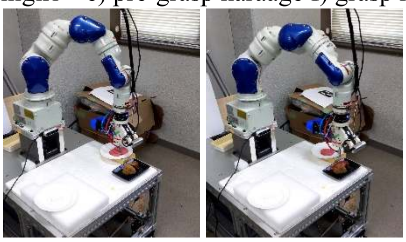
Results of assembling process experiment show success rate at 100% with no grasping problem from karaage to onigiri. Then, change to vacuum pad to picks bologna.



a) Ready pose b) pre-grasp onigiri c) grasp onigiri



d) Place onigiri e) pre-grasp karaage f) grasp karaage



g) Pre-pick bologna h) place bologna

Fig. 10 Food assembling process experiment

Table 6 Summary of experiment result

	Success		Fail
	Normal	Abnormal	
Karaage	20	1	1
Assembling	5	0	0

4. DISCUSSION

Firstly, our instance segmentation model must be more robust to the variance of lighting conditions because sometimes our model labels non-related objects with similar colors in poor lighting workspace as targets. We also considered expanding instance segmentation model with more lunch box ingredients such as rice, vegetables, and fruits to create a public dataset and increase the feasibility of research.

Secondly, the system must optimally grasp the objects because our failed task has a root cause from grasping. Grasp pose optimizer takes point cloud as input and give output as an optimal grasping pose.

Thirdly, dynamic gravity compensation for weight measurement, which provides redundant proof with wrist camera that either gripper still grasps object during motion, is required to stop and re-initiate task from the beginning. Additionally, we could extend our research to a nutrition label for each box if we can measure the weight of food objects.

5. CONCLUSION

In this paper, we proposed the robot framework for the food assembly process. The robot could locate the region of non-rigid food objects and lunchboxes with our proprietary instance segmentation model, but there is some flaw with lighting. Moreover, the robot could grasp the variety of food with our proposed design of the modular end effector, which has been prototyped and redesign to improve the performance. Then, we use the sampling-based rapidly-exploring random tree (RRT*) of OMPL in ROS for generating the pick-and-place motion as well as the evaluation of the performance. Our results showed that the robot could archive a successful rate of 90 percent with characteristic information.

REFERENCES

[1] Kompal Joshi, Tomas Joseph Norton, Jesús M Frías, Tiwari Brijesh kumar, "Robotics in meat processing: Production, Processing and Technology", *Emerging Technologies in Meat Processing*, pp.211-2292017, DOI: 10.1002/9781118350676.ch8

[2] Milane Haboon. (2020, March 19),*3 End-of-Arm Tools You Must Consider When Automating Food*, Retrieved from Soft Robotics Blog: https://www.softroboticsinc.com/resource/3-end-of-arm-tools-you-must-consider-when-automating-food/

[3] Carlos Blanes, Martin Mellado, Coral Ortiz, Ángel Valera, "Review. Technologies for robot grippers in pick and place operations for fresh fruits and vegetables", 2011, DOI: 10.5424/sjar/20110904-501-10

[4] Quigley, Morgan & Conley, Ken & Gerkey, Brian &

Faust, Josh & Foote, Tully & Leibs, Jeremy & Wheeler, Rob & Ng, Andrew., “ROS: an open-source Robot Operating System”, *International Conference on Robotics and Automation*, 2009, Corpus ID: 6324125
[5] David Coleman, Ioan Sucan, Sachin Chitta, Nikolaus Correll, “Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study”, 2014, DOI:10.6092/JOSER_2014_05_01_p3
[6] Ioan A. Şucan, Mark Moll, Lydia E. Kavraki, “The Open Motion Planning Library”, *IEEE Robotics & Automation Magazine*, 19(4), pp. 72–82, 2012, DOI:10.1109/MRA.2012.2205651
[7] Chih-Hsing Liu, Chen-Hua Chiu, Ta-Lun Chen, Tzu-Yang Pai, Yang Chen, and Mao-Cheng Hsu, “A soft Robotic Gripper Module with 3D Printed Compliant Fingers for Grasping Fruits”, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2018, DOI:10.1109/AIM.2018.8452420
[8] Meng, Jiawei & Gerez, Lucas & Chapman, Jayden & Liarokapis, Minas, “A Tendon-Driven, Preloaded, Pneumatically Actuated, Soft Robotic Gripper with a Telescopic Palm”, 2020, Corpus ID: 215541431
[9] SetupRobotics Quickchange Dual Toolbase, Retrieved from:https://www.universal-robots.com/plus/urplus-components/accessories/setuprobotics-quickchange-dual-toolbase-for-all-jobs/
[10] Hyeong Ryeol Kamm, Sung-Ho Lee, Taejung Park, Chang-Hun Kim, “RViz: a toolkit for real domain data visualization”, *Telecommunication Systems*, 60, pp. 337-345, 2015, DOI:10.1007/s11235-015-0034-5
[11] Zhaowei Cai, Nuno Vasconcelos, “Cascade R-CNN: High-Quality Object Detection and Instance Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-14, 2019, DOI:10.1109/tpami.2019.2956516
[12] Instance segmentation on COCO test-dev, https://github.com/open-mmlab/mmdetection/tree/master/configs/cascade_rcnn
[13] Sertac, K., & Emilio, F., “Sampling-based Algorithms for Optimal Motion Planning”, *IEEE International Journal of Robotics Research*, Vol. 30, No. 7, pp. 846-894, 2011, DOI:10.1177/0278364911406761