IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 24, 2020, accepted June 29, 2020, date of publication July 8, 2020, date of current version July 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006843

Object Localization and Depth Estimation for Eye-in-Hand Manipulator Using Mono Camera

MUSLIKHIN [1,2], JENQ-RUEY HORNG[1], SZU-YUEH YANG[1], AND MING-SHYAN WANG[1]

[1]Department of Electrical Engineering, Southern Taiwan University of Science and Technology, Tainan 710, Taiwan
[2]Department of Electronics Education Engineering, Universitas Negeri Yogyakarta, Yogyakarta 55281, Indonesia

Corresponding author: Ming-Shyan Wang (mswang@stust.edu.tw)

This work was supported in part by the Higher Education Sprout Project of Ministry of Education, Taiwan, and in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-218-029.

ABSTRACT This paper proposes the object localization and depth estimation to select and set goals for robots via machine vision. An algorithm based on a deep region-based convolution neural network (R-CNN) will recognize targets and non-targets. After the targets are recognized, we employed both the k-nearest neighbors (kNN) and the fuzzy inference system (FIS) to localize two-dimension (2D) positions. Moreover, based on the field of view (FoV) and a disparity map, the depth is estimated by a mono camera mounted on the end-effector with an eye-in-hand manipulator structure. Although using a single mono camera, the system can easily find the camera baseline by only shifting the end-effector a few millimeters towards the x-axis. Thus, we can obtain and identify the depth of the layered environment in 3D points, which form a dataset to recognize the junction box covers on the table. Experimental tests confirmed that the algorithm could accurately distinguish junction box covers or non-targets and could estimate whether the targets are within the depth for grasping by three-finger grippers. Furthermore, the proposed optimized depth error of -0.0005%, and localization method could precisely position the junction box cover with recognizing and picking error rates 0.993 and 98.529% respectively.

INDEX TERMS Region-based convolution neural network, eye-in-hand manipulator, machine vision, robotics, automation.

I. INTRODUCTION

Nowadays, no doubt that Artificial Intelligence (AI) has been a fundamental portion of industrial robots. After the expansion of deep learning in the last decade, traditional image processing methods are no longer used alone, for instance, the techniques based on color thresholding, color transformation, morphology, and images segmentation, these are not adequate to overcome the growing industrial environments in term on object recognition [1]–[3].

Since Krizhevsky *et al.* [4] introduced AlexNet using convolutional neural network (CNN) techniques, the image processing performance has been increasing rapidly and numerous CNNs have subsequently developed. The outstanding model proposed by Redmon *et al.* [5] in their experiment attempted to find other methods; YOLO (You Only Look Once) was presented, then Girshick [6] introduced Fast Region with CNN (Fast R-CNN). Similarly, Lin *et al.* [7] applied CNN to detect various objects to be grasped by the robot manipulator. Choi *et al.* [8] used CNN for the detection of various objects to be grabbed by the soft gripper. Ge *et al.* [9] applied a faster detection capability using R-CNN for strawberry picking robots by localizing three environments first. Another applied Faster R-CNN has been reported to detect and localize objects (e.g., pallets) based on 2D rangefinder data [10]. It has rapid detection and localizes objects accurately, but this applied in an automated guided vehicles (AGV). VoxelNet's approach [11] is entirely accurate by giving XYZ space to the target; unfortunately, the Z value is not calculated quantitatively and it employs the LiDAR as a sensor.

CNN has been refined for other applications in the robotics field. Mao *et al.* [12] used the YOLO-v3 technique with modifications to reduce Floating-Point Operations (FLOPs) that can improve performance 2.5 times faster. YOLO-v3 also includes K-means to separate clusters in each target, and its accuracy is more than 90% [13]. While CNN has been massively equipped for the recognition and detection of objects,

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

its applications in pick-and-place robots have been limited published. Many of these approaches centered on image processing and were thus not appropriate for a particular industrial robot system. Even today, grasping approaches still use traditional image processing techniques because of their flexibility to combine deep learning [14], [15].

To achieve reliable and valid picking of the targets, people need to localize them after recognition and detection. Some techniques to achieve these goals use a mono camera, stereo cameras, or a depth camera. However, most depth cameras are chosen because they have features that considerably facilitate image processing [16], [18].

Zhao *et al.* [19] estimated depth with the MSCN_{NS} (Multi-Scale Sub-Pixel Convolutions and a Smoothness Constraint) using a mono camera. A related method, Multi-Scale Dilated Convolution Network (MSDC-Net), was carried out by Tian *et al.* [20], while Xiong *et al.* [21] adopted RGB camera (Red, Green, and Blue). Using a mono camera is challenging the depth determination. Conversely, if we practice the eye-in-hand manipulator mode, it will benefit from the weight, dimension, and budget.

Many industrial robots use a stereo camera for localization and recognition because of its clarity. Taryudi and Wang [22] used a stereo camera for the detection bottle cap by eye-to-hand manipulator configuration. Cai *et al.* [23] used the same configuration and proposed a detection method for obstacle avoidance in six-dimension (6-D) poses using stereo cameras. Chen *et al.* [24] combined the geometry constraint with the epipolar constraint to achieve 3-D recovery of the fiber optic in compact eye-to-hand manipulator environment.

Even though stereo cameras have many conveniences, their construction tends to be non-concise as the RGB-D (RGB and Depth) cameras, which are now popularly used along with abundant image processing features. The Light Detection and Ranging (LiDAR) and environmental imaging of outdoor vehicles are incorporated by Reina *et al.* [25]. Ge *et al.* [9] retained the traditional Hough Transform image processing technique to recognize the classifications of strawberry environments. References [7], [9], [26]–[28] also used RGB-D images to estimate the targets and sense obstacles in each environment. All of those works used Kinect, Xtion, RealSense, ZED, MultiSense, or FRAMOS as their RGB-D cameras.

Depth estimation is crucial for industrial robots to ensure safe picking objects in a layered environment. Although many have used stereo cameras and RGB-D cameras, they are suitable for eye-to-hand configuration. This configuration is not affected by the weight and size of the camera. While eye-in-hand configuration may regard weight and size as obstacles, in some cases, robots can move to take things in a gap or a narrow space. How to reduce weight and size is not only the overriding solution, but depth estimation accuracy also remains the important goal.

References [29], [30], [32], [33] also used CNN to sense depths in both outdoor and indoor environments. Indeed, most of the prevailing environmental perception systems are used for vehicle navigation, whose conditions are very different from robots grasping objects in a layered position. Moreover, the depth data displayed in [29], [32], [33] are still in quantitative perception and are not possible to apply to manipulators. Caglayan and Can [31] used CNN with depth estimation through image disparity technique and results depicted in quantitative perception. Unfortunately, that experiment was conducted in outdoor environments and used an RGB-D camera. To ensure accurate performance in an indoor environment with multi-layered object positions, the robot requires the right depth estimation.

In our previous work [34], object localization and depth estimation have adopted machine learning systems based on eye-to-hand using a stereo camera with a color thresholding method. This method calculates the object centroid and the depth (Z) while the other two coordinates (XY) are gotten from the disparity between the left and right cameras. Differently, Lin *et al.* [7] mentioned that color-based image processing is not capable of changing environments, but we could utilize the crucial functions, such as the Adaptive Neuro-Fuzzy Inference System (ANFIS), in the system to improve the performance. Our challenge lies in using a mono camera with basic FoV and disparity point clustering, so we need to blend several approaches such as kNN, FIS, and the disparity map. These works have focused on deep learning, which is applied to industrial robot picking, as previously mentioned. In this paper, we are ensuring secure picking in a layered environment, which is the focal point.

Specifically, we propose to resolve localization and depth estimation on layered-environment problems, which are frequently confronted while picking for the eye-in-hand manipulator. The following details are given in this paper:

- We employ deep learning to recognize the targeted junction box covers, which depend on the layered environment, and we propose a localization method based on R-CNN.
- We consider the depth collision problem for manipulators in layered environments. We solve this problem by proposing the FIS and kNN algorithms. We classify layers as a cluster points cloud to detect the depth as the pickable junction box covers are on the layer and verified by disparity map.
- The localization and depth estimation methods are performed and evaluated on our certain robot manipulator conditions thus, it provides a reference for eye-in-hand manipulator systems concerning localization and depth estimation for similar industrial robots.

In this paper, we discussed the overall system design in Section II. Section III introduces localization and depth estimation and grasping a target on layered environment until it is placed in a box in Section IV. The next Section V describes the experimental results. Finally, we conclude the work and offer ideas for possible future work in Section VI.

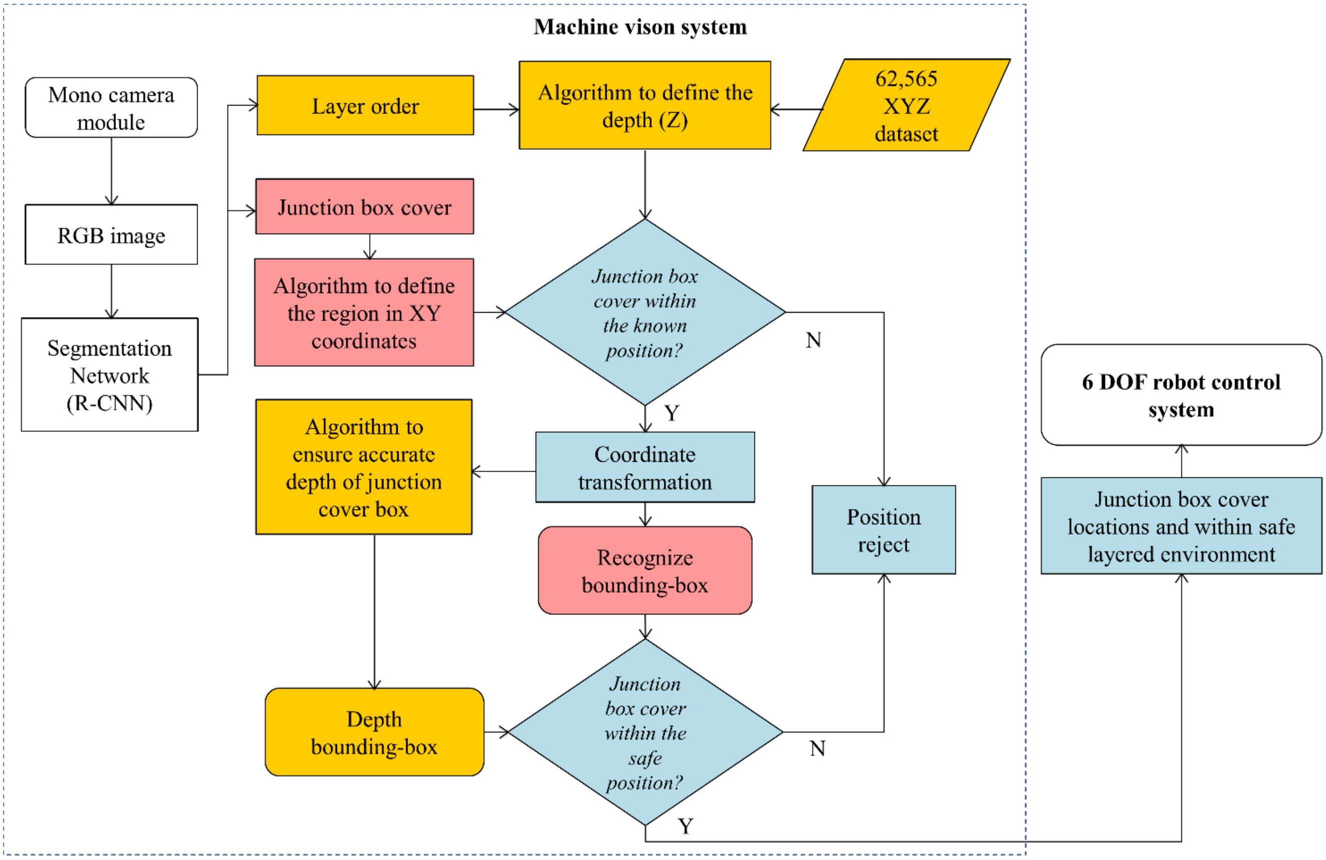
Machine vison system
Mono camera module
RGB image
Segmentation Network (R-CNN)
Layer order
Algorithm to define the depth (Z)
62,565 XYZ dataset
Junction box cover
Algorithm to define the region in XY coordinates
Junction box cover within the known position?
N
Y
6 DOF robot control system
Algorithm to ensure accurate depth of junction cover box
Coordinate transformation
Position reject
Junction box cover locations and within safe layered environment
Recognize bounding-box
Depth bounding-box
Junction box cover within the safe position?
N
Y

FIGURE 1. Overall architecture diagram.

II. OVERALL SYSTEM DESIGN

A. PROPOSED SYSTEM DESIGN

Our eye-in-hand robot manipulator takes a position from its initial position, where it takes a picture, processes the input image, and then sends commands to the robot controller. Hence, the robot in the initial position is static. The obtained RGB image is used to determine the depth and recognized target acquired from a mono camera mounted on the end-effector in a machine vision.

The architecture of the proposed machine learning is shown in Fig. 1. The instance segmentation in the R-CNN network is used to identify targets; junction box covers and non-target. Thereafter, the recognized junction box cover undergoes safe operation checking in XY coordinate algorithm, the layered environment algorithm, the depth algorithm with a dataset, coordinate transformation, and location verification algorithm to obtain an accurate final position. The recognized-bounding box and depth-bounding box are labeled into locations within the safe layered environment, thus realizing actual picking.

The proposed depth estimation algorithm involves identifying the area by target location in specific layered environments on 2D images. In Fig.1, the shaded pink are ideas refer to junction box cover localization, while those shaded in yellow are correlated to the layered environment. These two goals coordinate with each other to finalize the positions of junction box covers within the accurate layer; hence, the procedures associating with both goals are shaded in cyan. These explained that both localization and depth algorithms would be specified in Section III.

B. LIMITATION OF THE SYSTEM

Before the localization and depth algorithms reported in the next section, we need to reveal some limitations of the system. This will promote the process of facing the minimum requirements in system evaluation. The limitation includes the minimum number of actual objects (target and non-target) of four. The target is limited to the junction box cover and non-target in the form of wooden blocks that have almost the same area. Limit input number to a minimum of four objects to guarantee the variation of the arrangement of objects, far apart (full and semi) and concise (single and double), the system will fulfill all four scenarios. Meanwhile, the maximum number that can be executed by the proposed system is 18, and there are12 real objects for the maximum and minimum in the widest and narrowest area, respectively. The work area in our system refers to the FoV of the camera (Fig. 3) or the top of the truncated pyramid. The widest area is 500 mm × 375 mm at 495 mm, and the narrowest area is 417 mm × 313 mm at 413 mm camera depth. Further explanations are discussed in Section IV.

III. LOCALIZATION AND DEPTH ESTIMATION

A. CAMERA FoV

We have decided to use a mono as well as optimal camera, Logitech C920, in an eye-in-hand manipulator with the considerations mentioned above. Based on the initial position and FoV, the capture area can be calculated. The relationship between depth and capture area forms a linear relationship. So, the same object will be known in each dimension. Table 1 is a comparison between the C920's camera with several RGB-D cameras.

TABLE 1. Comparison between RBG-D cameras and C920's camera.

Parameters	RGB-D cams.			Mono cam.
	Kinect v2	Xtion Pro	SR305	C920
Res. color (px)	*1920×1080*	1280×1024	1920 × 1080	1080×720
Res. depth (px)	512×424	*640×480*	*640×480*	n/a
Frame rate (fps)	30	30	*60*	30
Max. dist. (mm)	4500	3500	1500	n/a
Size (mm)	66×249×67	48×178×38	*26×139×12*	29×94×24
HFoV/VFoV (°)	*70/60*	58/45	69/54	70/43
Weight (g)	1400	540	255	*217*

The 78° value for FoV is valid at 16:9 aspect ratio; the FoV value itself in the product is measured from a diagonal position or only written as FoV. On the C920 camera datasheets, it should be written DFoV (Diagonal Field of View). While for the 4:3 aspect ratio, DFoV needs to be elaborated with (1) to find HFoV and VFoV, where H means horizontal and V represents vertical, see Fig. 2.

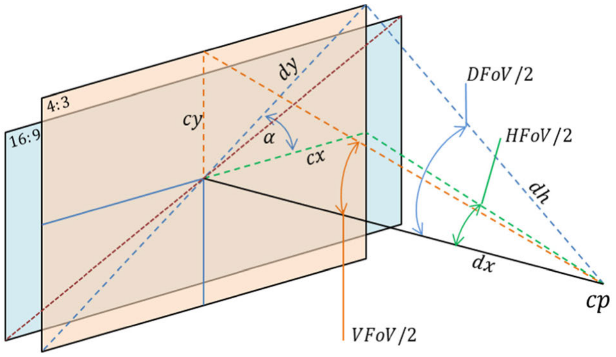

FIGURE 2. The FoV of C920's camera.

In Fig. 2, if DFoV is given, both VFoV and HFoV can be found for the C920 camera. By default, this camera works with a 16:9 CMOS sensor, so we need to convert it to the 4:3 aspect ratio as the following (1), where dx denotes the length between the camera pinhole cp and the center of the frame and dh is the length between the camera pinhole and the vertex of the frame. While cx is half length of the horizontal line, cy is half length of the vertical line, and dy is half length of the diagonal line. Thus, the difference in the 4:3 aspect ratio and 16:9 concerns the length of dy.

$$\begin{cases} DFoV = cos^{-1}(\dfrac{dx}{dh}) \times 2 \\ dx = dh \times \cos(\dfrac{DFoV}{2}) \\ dy = dh \times \sin(\dfrac{DFoV}{2}) \\ cx = dy \times \cos(\alpha) \\ cy = dy \times \sin(\alpha) \end{cases} \tag{1}$$

Referring to (1), we could find the values of HFoV and VFoV by the angle $\alpha = atan\,(3/4)$, so obtained in (2).

$$\begin{cases} HFoV = 2 \times \operatorname{atan}\left(tan\left(\dfrac{DFoV}{2}\right)\right) \times \cos\left(\operatorname{atan}\left(\dfrac{3}{4}\right)\right) \\ VFoV = 2 \times \operatorname{atan}\left(tan\left(\dfrac{DFoV}{2}\right)\right) \times \sin\left(\operatorname{atan}\left(\dfrac{3}{4}\right)\right) \end{cases} \tag{2}$$

With HFoV and VFoV, then to recognize the depth of a position can be done through a comparison of the perimeter or volume of an object. Illustration of distance, object, and camera has a linear relationship in the FoV. In Fig. 2, we could see the C920 camera projection with HFoV and VFoV of 70° and 43° respectively obtained from (2).

B. TARGETS RECOGNITION

R-CNN [9] was often applied for the recognition and segmentation of fruits. As explained above, various networks possible for the detection are valid, reliable, and fast for grasping with depth estimation [5]–[7], [35]. Furthermore, we intend to estimate the junction box cover location in xyz-space as validly as reasonable.

The most popular one is to adopt the Mask R-CNN [9], [36], [37], but this method has heavy computation as well as has to find the target centroid. Therefore, a lighter burden way can use the traditional image processing technique by centroids function, which is provided in MATLAB, as we apply to the system. In this case, although bounding boxes also include other object pixels, the addition of bounding boxes in an object can provide detailed information. In short, the required depth value on the second bounding box is needed by the robot as an input.

To achieve the goal of recognizing targets needs training and testing processes as we apply the R-CNN. In the training section, datasets (images) are labeled using the Image Labeler App. Each has two classes of RoIs (Regions of Interests) namely "JunctionBox" and "non_target" including a dataset of 62,525-point mappings of FoV. Whereas the testing or object detection section that has features and detectors from the training results is used to test the target capture results from the mono camera, see Fig. 3. At the object detection stage, we have at least four steps. The three times of convolution and pooling could be seen; 5 × 5 kernel and 3 × 3 kernel were applied respectively on the system.

The junction box cover is the object's target, while the non-targets present potential gadflies with machine vision to make

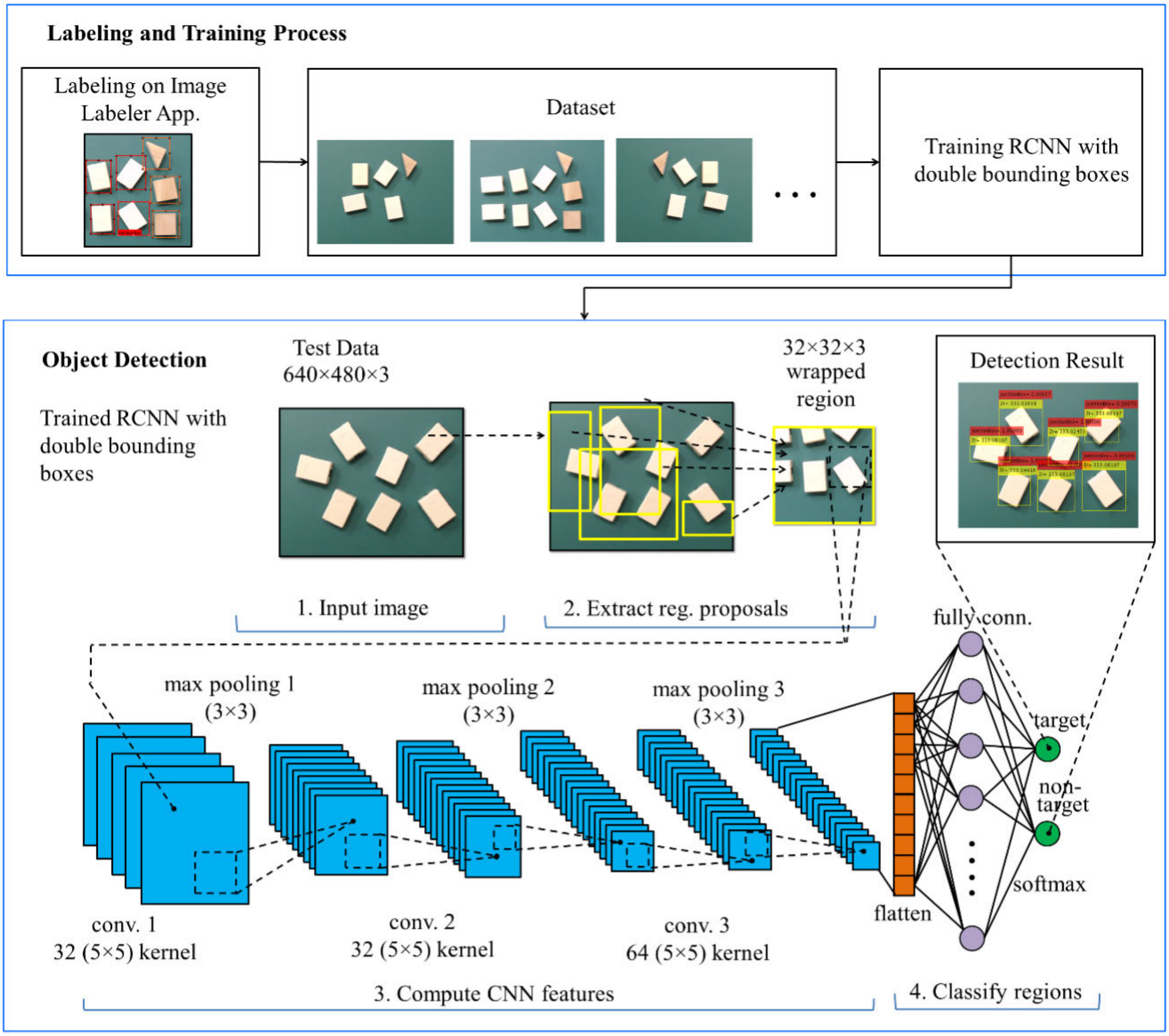

FIGURE 3. Structure of R-CNN with double bounding boxes (red: confidence level and yellow: depth value). Upper box is the training process including labeling and training itself. Lower box is detection process with fourth steps of R-CNN process.

decision while in manipulation and are, therefore, also non-targets that should be detected. For instance, the recognition and detection results are provided in Fig. 5 to show the couple bounding boxes, and each color represents a confidence level and depth on a layered environment. A detailed discussion about the layered environment will be presented in the next section.

C. DEPTH ESTIMATION FOR RECOGNIZED TARGETS

For the junction box covers, after several segmented targets were created, in which one segment represented a recognized target via image processing. The segments were calculated to get centroid points, areas, and depths of the recognized targets in the frame of camera C. Our workflow in the depth estimation is illustrated in Fig. 4. The depths were extracted from the centroids, areas, layered environments, the FoV, the closest point of XYZ-dataset, and image disparities. In addition, the depths have been transformed from the target frame O to the C frame using an intrinsic camera parameter.

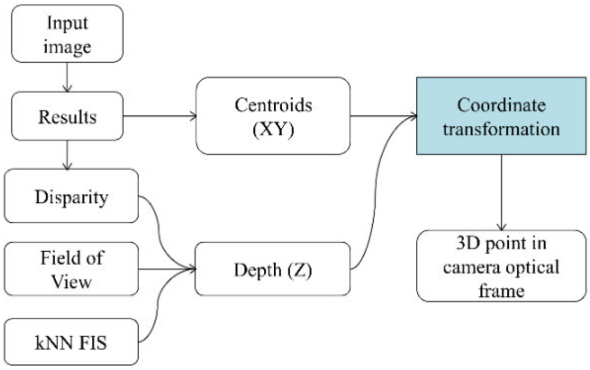

FIGURE 4. Flow diagram of the coordinate transformation.

Samples of the depth estimation process and its effects can be found in Fig. 5. Figs. 5(a) and (d) are the original

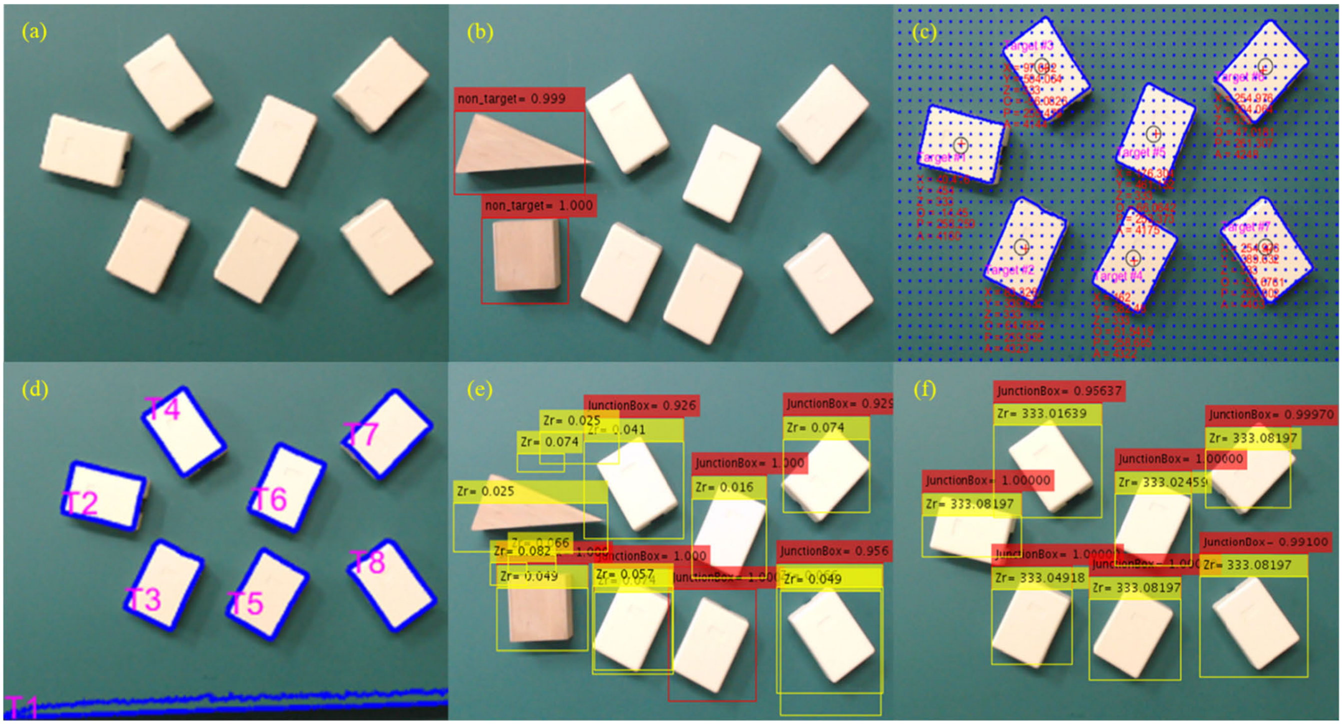

FIGURE 5. Recognition and detection results. (a) and (d) show the input images and all targeted objects, while T1 is a noise or not a real target; (b) shows the recognition of the non-target; (c) displays the edge detection-n results on end-effector without vibration besides the segmented closest class point with detail of XYZ coordinates, perimeters, and areas; (e) displays the location (xy), perimeters, areas, and orientations in normal condition with non-targets; (f) displays recognized objects with double bounding boxes without noised vibration from (c).

images and the raw corresponding targets effected by lighting. Figs. 5(b) and (e) are the images with noise caused by non-targeting. Figs. 5(c) and (f) are sequence of the final localization then signed by double bounding boxes. Each bounding box contains the level of confidence ranging from 0 to 1 and depth mark in millimeters, but the positions of the junction box covers were sorted and sent to the manipulator. Therefore, the shortest coordinate of the junction box covers will be executed first.

D. TARGET LOCATION ESTIMATION METHODS

1) LAYERS CLUSTERING

In this picking and placing robotic manipulator system, once the target 3D is obtained, the machine vision system requires to send the positions of all junction box covers to the robot controller system. Nevertheless, it was decided that the raw points transformed from the segmented images were not sufficiently accurate, particularly for the z-axis such as depth camera frame Z_c and depth target frame Z_t. Consequently, post-processing procedures were carried out on the raw points of the target centroids in order to achieve a layered environment that could accurately represent the actual target location.

The inaccuracy of the points transformed was induced by a variety of factors. Such as, the target points perhaps projected into the background scene due to the incorrect light-dependent sensing of the camera shown in Figs. 5(b) and (d) respectively. Another factor was noise from the camera vibration due to end-effector moving to initial position, and, also, there may have been inaccurate from edge detection.

In Figs. 5(c) and (e), disparity clustering is needed to verify the layer's position. We adopted the concept of the disparity map, which is commonly applied in stereo cameras, even though in this paper, using a mono camera. The first capture has taken before the end-effector reaches the initial position, and the second capture has taken shortly after the initial position, thus as there are left and right cameras as camera baseline.

The disparity map $D(x, y)$, represents the displacement among the left and right images in terms of corresponding pixels. However, in a real application, it is challenging to find corresponding pixels. In the nonocclusion pixels, some factors such as nontextured-homogeneous, repeated-texture, and camera noise may cause trouble. The disparity estimation is accomplished by block matching for all pixels, and the disparity validity value shall be determined as follows,

$$D_{L\rightarrow R}(x,y) = \underset{d\epsilon[0,D_{max}]}{\arg\min}\ \varepsilon^{d}_{L\rightarrow R}(x,y) \qquad (3)$$

$$\varepsilon^{d}_{R\rightarrow L}(x,y) = \frac{\sum_{(u,v)}\sum_{\in W}|f_r(x-u,y-v)-f_l(x-u+d,y-v)|}{\sum_{(u,v)}\sum_{\in W}|f_r(x-u,y-v)+f_l(x-u+d,y-v)|} \qquad (4)$$

The left image and right image disparities are obtained from (3) and (4), where $\varepsilon^{d}_{R\rightarrow L}(x,y)$ is the normalized block

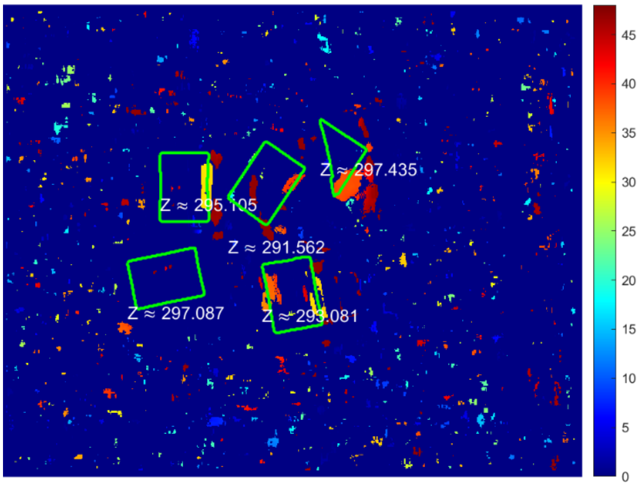

FIGURE 6. Disparity map using block matching with disparity range 0-48. The raw depth estimation varies from 291.562 mm to 297.435 mm, while the actual $Z = 295$ mm.

matching error with a horizontal disparity d, W is the block matching window, and D_{max} is the disparity of maximum value in an allowable limit. In order to verify the observed disparity, the following will obtain the disparity from the right image frame f_r to the left image frame f_l, while u and v are the number of pixels in xy-camera image plane, respectively:

$$D_{R\rightarrow L}(x, y) = \underset{d\epsilon[-D_{max},0]}{\arg\min} \; \varepsilon^{d}_{R\rightarrow L}(x, y) \tag{5}$$

The minimum matching error (MME) then estimates how similar the pair image values at (x, y) in the left image and the corresponding point $(x + d, y)$ in the right image are. The MME is construed as (6).

$$MME(x, y) = \varepsilon^{d}_{L\rightarrow R}(x, y)|_{d=D_{R\rightarrow L}(x,y)} \tag{6}$$

Depth calculation results are still not wholly accurate using the disparity map obtained from equations (3)-(6), and the result can be seen in Fig. 6. In industrial robot settings, the safe limit between the gripper fingertip and the table is a maximum of 3 mm, see Figure 11(e). For this reason, verification needs to be done with a dataset. Meanwhile, the pixel density on the FoV is used to generate these datasets, and then it is applied in Algorithm 1, so the depths are accurate.

Therefore, the disparity of layers clustering algorithm was utilized to filter out ambiguous layers. Verifying by the disparity map-based clustering, Algorithm 1 is a method in which disparity in each layer variation has different densities to fix Z_c to be Z_r.

2) TARGET POSITION OPTIMIZATION

Double bounding box targets from RGB camera frame contain real robot coordinates $(X_r Y_r Z_r)$ sent to the manipulator. In Figs. 5(e) and (f), the raw points obtained after layer clustering and matching of xy-coordinate are regional points. It is proven that bounding boxes can only show the target position roughly, but centroids can clarify the actual position of the junction box cover. The target surface facing the camera perpendicularly feels better than other surfaces away from the midpoint of DFoV. In this scenario, the camera angle is on the top view. Fig. 7 shows the junction box cover in the FoV. We have generated a dataset manually to localize targets more accurately and three methods will explain it in the next subsections.

Algorithm 1 Ensure Layered Environment Through Disparity Map

```
o1:    the disparity map of the objects in certain
       layered manipulator $D(x, y)_{O_i}$;
for every detected object $O_{i=1:n}$ do
    verifying the Zc data to the disparity clustering
    $\forall D(x, y), Z_{c[i]} D(x, y)_{[i]}$;
    if the verifications are closed $D(x, y)_{[i]} \approx$ dataset
    then
        saving the world camera's depth as world robot's
        depth $Z_{r[i]} \leftarrow Z_{c[i]}$;
        $i++$;
    else
        return o1;
    end
end
```

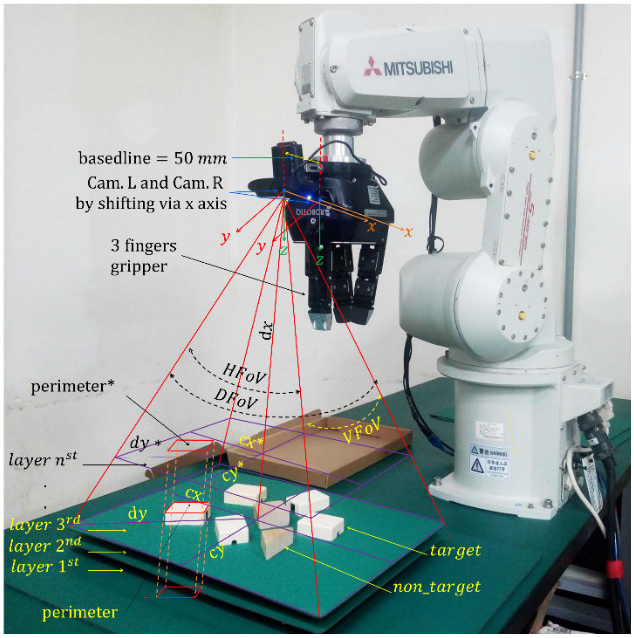

FIGURE 7. The FoV utilized in eye-in-hand manipulator with layered environment.

Based on Fig. 7, we develop FIS-Sugeno rules which consist of four-perimeter inputs and the output Z, while five Gaussian membership functions (MFs) are given for each variable. The number of MF depends on the data characteristics as polynomials and object patterns of interplaying. The Gaussian membership function is given as follows:

$$\mu(x) = e^{\frac{-(1-c)^2}{2s^2}}, \tag{7}$$

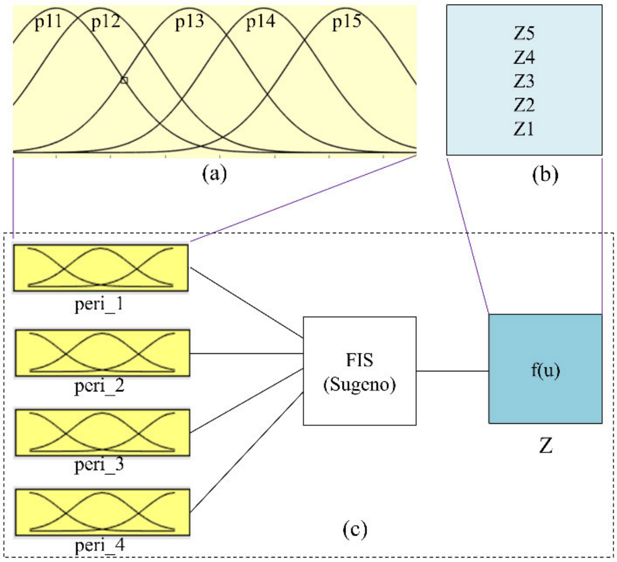

FIGURE 8. Structure of FIS; (a) the five MFs of perimeter; (b) the output of Z; (c) the four inputs of perimeters to FIS Sugeno with output of Z.

where c and s are the set of parameters that modifies the MF form and x is the input. According to (7), we can illustrate the composition of MF as shown in Fig. 8 and generate rules (8).

$$
\begin{aligned}
&\{1\}\ \mathit{If}\ (peri1\ is\ p11)\ \&\ (peri2\ is\ p21)\ \&\ (peri3\ is\ p31)\\
&\qquad \&\ (peri4\ is\ p41) = Z1;\\
&\{2\}\ \mathit{If}\ (peri1\ is\ p12)\ \&\ (peri2\ is\ p21)\ \&\ (peri3\ is\ p31)\\
&\qquad \&\ (peri4\ is\ p41) = Z1;\\
&\{3\}\ \mathit{If}\ (peri1\ is\ p13)\ \&\ (peri2\ is\ p21)\ \&\ (peri3\ is\ p31)\\
&\qquad \&\ (peri4\ is\ p41) = Z1;\\
&\{4\}\ \mathit{If}\ (peri1\ is\ p14)\ \&\ (peri2\ is\ p21)\ \&\ (peri3\ is\ p31)\\
&\qquad \&\ (peri4\ is\ p41) = Z1;\\
&\{5\}\ \mathit{If}\ (peri1\ is\ p15)\ \&\ (peri2\ is\ p21)\ \&\ (peri3\ is\ p31)\\
&\qquad \&\ (peri4\ is\ p41) = Z1;\\
&\qquad \vdots\\
&\{84\}\ \mathit{If}\ (peri1\ is\ p15)\ \&\ (peri2\ is\ p25)\ \&\ (peri3\ is\ p35)\\
&\qquad \&\ (peri4\ is\ p44) = Z5;\\
&\{85\}\ \mathit{If}\ (peri1\ is\ p15)\ \&\ (peri2\ is\ p25)\ \&\ (peri3\ is\ p35)\\
&\qquad \&\ (peri4\ is\ p45) = Z5;
\end{aligned}
\tag{8}
$$

Eq. (8) is used to determine the form of membership of FIS inputs. In total, four inputs that represent a minimum group in each layer with a random orientation. However, without the orientation, the grasping process is challenging to achieve; in this approach, the orientations are processed separately by comparing the longest edge pixels of the object with horizontal pixels x. However, the purpose of FIS is not accurate enough to know the exact layered-environment order, so classification is still needed.

One of the powerful classification methods is kNN, which strengthens our adoption. The case of determining the position of an object with the nearest neighbor can be resolved by centroid using Euclidean E_i. The E_i is chosen in this paper and obtained from (9), the length of the x-axis is from the disparity of x_1 and the end of x_2-axis. The width of the y-axis is reached from the initial y_1-axis and y_2-axis, while the height of the z-axis is found from z_1 and z_2.

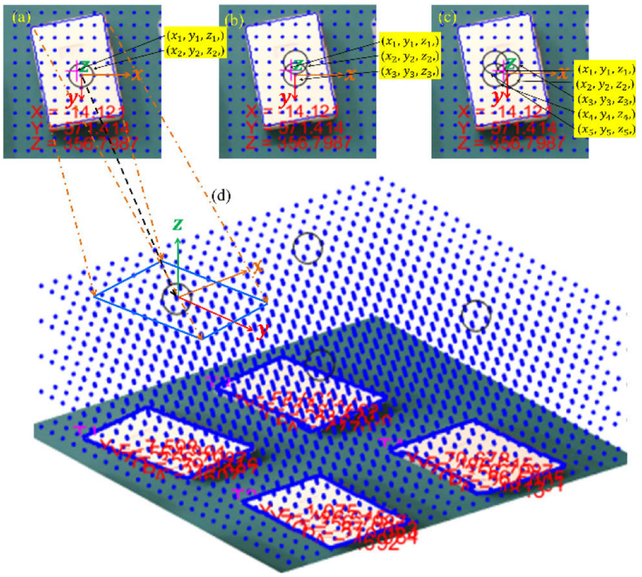

FIGURE 9. kNN classification with a closest point; (a) kNN with k = 1; (b) kNN with k = 2; (c) kNN with =4; and (d) 3D plotting with layer cloud points.

$$
\mathrm{E_i} = \sqrt{(\mathrm{x_2} - \mathrm{x_1})^2 + (\mathrm{y_2} - \mathrm{y_1})^2 + (\mathrm{z_2} - \mathrm{z_1})^2} \tag{9}
$$

Fig. 9 (a) shows each point representing a class with a resolution of 5 pixels × 5 pixels. The magenta (+) is the centroid of the junction box covers, while the black circle denotes the class area where the centroid position is. In Figs. 9(a)-(c), the numbers of k are respectively given by 1, 2, and 4; however, in kNN, the closest distance is at $k = 1$. In contrast, $k = 2$ means it has two closest distances, or in other words, the second closest distance as a runner up the distance. It means if the variables just only include x_1, y_1, z_1 and x_2, y_2, z_2 for $k = 1$ but at $k = 2$, another x_3, y_3, z_3 are needed. Therefore, we use $k = 1$ to reduce computing in the estimation of the xy-position. Grasping decision for objects is sorted by target number, which is pixel order in the xy-image plane.

3) TARGET ORIENTATION

Most CNN application in fruit harvesters does not pay attention to orientation of fruits such as oranges, strawberries, or eggplants because they are hanging [9], [23], [33]. Nevertheless, object orientation needs to be addressed for applications in industry. For this reason, in addition to localization, orientation becomes an inherent part that the gripper could hold accurately.

Object orientations ranging from −90° to 90° are determined from the region of the subject given by the MATLAB function. Throughout the eye-in-hand adjustment process, these orientation data must be modified to approximate the

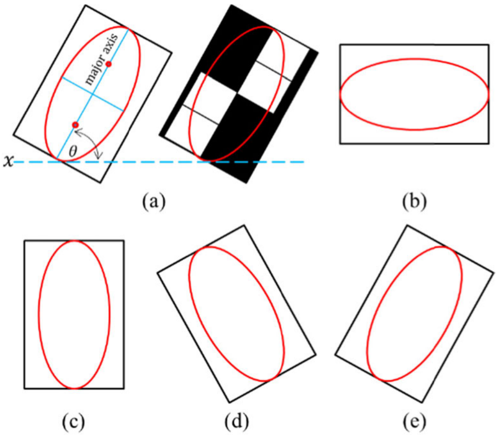

FIGURE 10. **Detecting object orientation; (a) comparison between major axis to x-axis; (b) horizontal; (c) vertical; (d) left diagonal; (e) right diagonal.**

orientation of the end-effector so that the object can be properly grasped. The angle between the x-axis and the major axis of the ellipse as illustrated in Fig. 10 (a) is as an object orientation. The left side of the figure displays the same ellipse as the blue lines showing the axes, the red dots are the center of the blue line, and the orientation is the angle between the horizontal dashed line and the major axis. The image region and its corresponding ellipse are shown on the right side of the figure. Classifying each map feature is based on four categories; (a) horizontal, (b) vertical, (c) left diagonal, and (d) right diagonal. The relationship between the values of horizontal line x and vertical line y and the width W$'$ and the height H$'$ of the object is given in (10) for each classification.

$$
\begin{aligned}
&(a)\begin{cases} 0 \le x \le W' \\ 0.25H' \le y \le 0.75H' \end{cases}\\
&(b)\begin{cases} 0.25W' \le x \le 0.75H' \\ 0 \le y \le W' \end{cases}\\
&(c)\begin{cases} y \ge \dfrac{H'}{W'}x - \dfrac{1}{2}H' \\ y \le \dfrac{H'}{W'}x + \dfrac{1}{2}H' \\ 0 \le y \le H' \\ 0 \le x \le W' \end{cases}\\
&(d)\begin{cases} y \ge -\dfrac{H'}{W'}x + \dfrac{1}{2}H' \\ y \le -\dfrac{H'}{W'}x + \dfrac{3}{2}H' \\ 0 \le y \le H' \\ 0 \le x \le W' \end{cases}
\end{aligned} \tag{10}
$$

E. TARGET TO WORLD COORDINATE TRANSFORMATION

The mono camera is designed to capture the 2D coordinates of the junction box cover in camera frame C, and converting the points from C into the end-effector frame E was necessary. The relationship among the frames is shown in Fig. 11, in which O is the cover for the junction box cover, C for the camera frame, R for the robotic base frame and B for the chessboard.

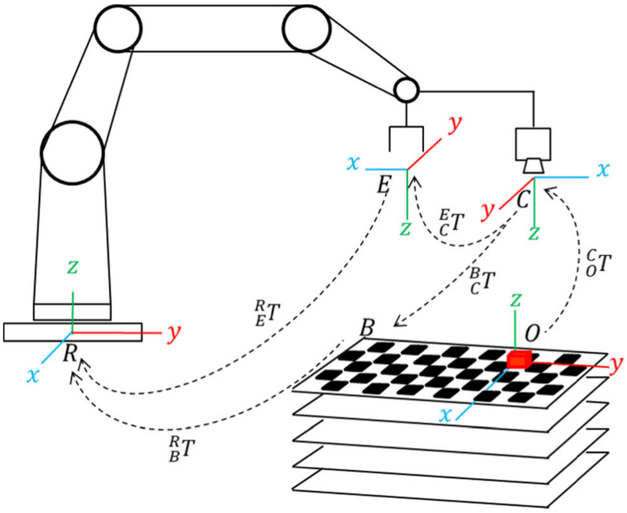

FIGURE 11. **Coordinate frames for eye-in-hand robot system with mono camera. The transformation matrix from the target coordinate frame *O* to the end-effector coordinate frame *E*, the transformation matrix from the camera coordinate frame *C* to *E* frame, and the transformation *E* to the robot base coordinate frame *R*.**

Let ${}^{R}O$ be the location of the junction box cover O with respect to the robotic base frame R and ${}^{C}O$ be the location of junction box cover O in the C frame. The transformation of the target coordinate from camera frame C to robotic base frame R could be expressed in (11) according to:

$$
{}^{O}_{R}T = {}^{O}_{C}T^{C}_{E}T^{E}_{R}T \tag{11}
$$

where ${}^{O}_{R}T$ is transformation from O frame to R frame, the target frame and the robotic base frame, respectively. The ${}^{B}_{C}T$ depict in Fig. 11 can be obtained from camera calibration while ${}^{R}_{B}T$ is known. Based on parameters of ${}^{B}_{C}T$ and ${}^{R}_{B}T$, ${}^{R}_{C}T$ could be obtained.

To obtain ${}^{E}_{R}T$ basically, the structure of MELFA RV-3SD robot manipulator is shown in Table 2. The number of joint j, angle between two connection rods θ, length of link l, angle of torsion connected with rod α, and the distance between the two connection rods d. Although there is no standard for manipulator control, the most common way of adopting the Denavit-Haternberg (DH) parameters is that for Table 2, inverse kinematics for control manipulators are used.

TABLE 2. **The DH parameter for MELFA RV-3SD manipulator.**

j_i	θ$_i$	l_i (mm)	d_i (mm)	a_i (°)	Joint
1	θ$_1$	350	95	-90°	waist
2	θ$_2$	0	245	0°	shoulder
3	θ$_3$	0	135	90°	elbow
4	θ$_4$	270	0	-90°	forearm
5	θ$_5$	0	0	90°	wrist
6	θ$_6$	85	0	0°	tool

Based on Table 2, we can obtain (12) for the homogeneous transformation of the link ${}^{i-1}_{i}T$.

$$
{}^{i-1}_{i}T = \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & \alpha_{i-1} \\ sin\theta_i cos\theta_{i-1} & cos\theta_i cos\alpha_i & -sin\alpha_{i-1} & -sin\alpha_{i-1} d_i \\ sin\theta_i cos\theta_{i-1} & cos\theta_i cos\alpha_i & cos\theta_{i-1} & cos\theta_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}
$$

The transformation of a robotic manipulator matrix is based on the method of DH parameters. We refer to (12) for each transformation at each joint as follows:

$$
\begin{aligned}
{}^{0}_{1}T &= \begin{bmatrix} cos\theta_1 & -sin\theta_1 & 0 & 0 \\ sin\theta_i cos\theta_{i-1} & cos\theta_i cos\alpha_i & 0 & 0 \\ sin\theta_i cos\theta_{i-1} & cos\theta_i cos\alpha_i & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{1}_{2}T &= \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ -sin\theta_2 & -cos\theta_2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{2}_{3}T &= \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & l_2 \\ sin\theta_3 & cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{3}_{4}T &= \begin{bmatrix} cos\theta_4 & -sin\theta_4 & 0 & l_3 \\ 0 & 0 & 1 & d_4 \\ -sin\theta_4 & -cos\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{4}_{5}T &= \begin{bmatrix} cos\theta_5 & -sin\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ sin\theta_5 & cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{5}_{6}T &= \begin{bmatrix} cos\theta_6 & -sin\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -sin\theta_6 & -cos\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{13}
$$

From (13), a forward kinematics equation can be obtained as follows:

$$
{}^{0}_{6}T = {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T \cdot {}^{4}_{5}T \cdot {}^{5}_{6}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{14}
$$

where (n_x, n_y, n_z), (o_x, o_y, o_z), (a_x, a_y, a_z), and (p_x, p_y, p_z), stand for the normal vector, orientation vector, approach vector, and the position of the end-effector, respectively. For inverse kinematics (15) can be derived from (14).

$$
\begin{aligned}
{}^{0}_{1}T^{-1} \cdot {}^{0}_{6}T &= {}^{1}_{6}T \\
{}^{1}_{2}T^{-1} \cdot {}^{0}_{1}T^{-1} \cdot {}^{0}_{6}T &= {}^{2}_{6}T \\
{}^{2}_{3}T^{-1} \cdot {}^{1}_{2}T^{-1} \cdot {}^{0}_{1}T^{-1} \cdot {}^{0}_{6}T &= {}^{3}_{6}T \\
{}^{3}_{4}T^{-1} \cdot {}^{2}_{3}T^{-1} \cdot {}^{1}_{2}T^{-1} \cdot {}^{0}_{1}T^{-1} \cdot {}^{0}_{6}T &= {}^{4}_{6}T \\
{}^{4}_{5}T^{-1} \cdot {}^{3}_{4}T^{-1} \cdot {}^{2}_{3}T^{-1} \cdot {}^{1}_{2}T^{-1} \cdot {}^{0}_{1}T^{-1} \cdot {}^{0}_{6}T &= {}^{5}_{6}T
\end{aligned} \tag{15}
$$

IV. GRASPING IN LAYERED ENVIRONMENT

The industrial robot needs to be aware of its layered environment to prevent manipulation and gripper from hitting the table. Hence, the robotic system has to triangulate the manipulation plan and precisely grab the targeted object so that it could not cause possible damage [38].

During the experiment, junction box covers are scattered on the tray and sometimes sprawled and stuck together. At the same time, our gripper had a three-finger model, one middle finger, and the other two fingers. Therefore, the (10) is used to identify the orientation and the gripper pinches *gpp* the target in a transverse position. The area marked by the dashed line in Fig. 12 represents a gripper works in three-finger pinching. Fig. 12(a) depicts a target in a concise position, while Fig. 12(b) displays far apart position, and Fig. 12 (c) shows a far apart with noise caused by end-effector vibration during movement. Therefore, the junction box covers should be pinched in a safe area.

A. TARGETS ON THE LAYERED ENVIRONMENT

A critical output obtained by the FoV model is the sensing layer. There are five layers on the table containing the junction box covers with random numbers and scattered arrangements. Most junction box covers are arranged in irregularly, far apart, or even coincided with one another. During pinching, it can be dangerous for the three fingers of the gripper. In this section, we offer three methods to identify the position of the targeted junction box cover among the whole.

1) METHOD 1: FAR APART

To classify junction box covers in the same layer, it is necessary to calculate the perimeter and area, as shown in Fig. 12(a). Thereafter, the first four targets in the image plane are analyzed to get the decision that the junction box cover is on a specific layer that as the same. Perimeter and area of the target tend to be high, compared to targets that are perpendicular to the camera or the *dx*-axis (2). Because the scenery targets were arranged randomly, the targets can be far apart. For illustration, if there are four targets, three targets close collectively, and one separate target has a higher perimeter and area. Furthermore, the perimeter and the area are compared to the pixel density according to the capture area obtained from the FoV (3) and FIS is applied to overcome this case.

We perceive, however, that our method was not always quite appropriate, as there were some conditions in scattered targets causing the FIS output to exceed the safe gripper limit, as shown in Fig. 12 (b). In this fact, the FIS method does not apply to junction box covers that are >185 mm apart from three target groups that are perpendicular to the dx-axis, therefore, the case in Figs. 12 (b) and (c) may be rated a nonoptimal condition by applying this method.

2) METHOD 2: CONCISE

Contrary to method 1, the targets that are close together or concise tend to be homogeneous for the perimeter and area.

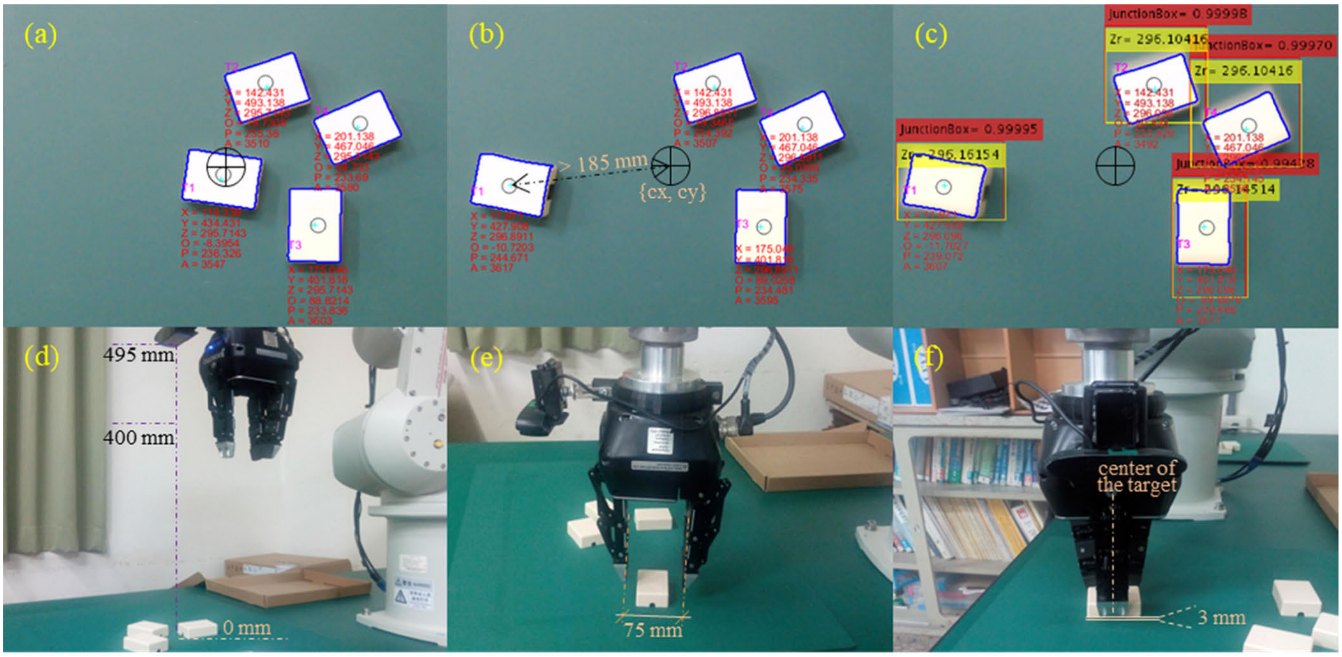

FIGURE 12. Three-finger pinching; (a) the results of localization when the targets were concise by method 2; (b) the results of localized targets far apart by method 1; (c) the results of localized far apart and blurred targets using method 3; (d) the initial position of end-effector; (e) straight gripper opening gpo (75 mm) before pinching; (f) pinching the target transverse target in the center of the object and kept fingertips 3 mm above the table.

For example, if there are four targets on a layer, the perimeter value and the target area are in the average range even though their positions are not always perpendicular to the *dx*-axis. Method 2 is accurate as handled by FIS but still risks to camera vibrations at the end-effector, as in Fig. 12 (c).

To succeed in the above problems that appear in method 1 and method 2, first, the kNN algorithm is applied to ensure all the *x*, *y*, and *z* points of each target of the five layers. The target that has been detected by the edge and calculated by the area we give a red sign (+) is as the closest point *Ei*. To verify the junction box cover is in one layer as shown in Fig. 11. While the vibration effect of the end-effector causes the image to become blurry *ib* and the perimeter and area become larger. This vibrational effect is rare; however, we say that this is also part of the weakness of method 2.

3) METHOD 3: CONFIDENCE

Although the weakness of method 1 is resolved by method 2, method 2 also has disadvantages because of the blurred image due to the vibration of the end-effector. Blurred images could be still recognized by R-CNN; in this case, we use the 0.86 confidence level threshold, α. Method 3 can eliminate the possibility of reading errors for a blurred image, *ib* such as Fig. 12 (c) in a layered environment.

The performance of three stratified methods is a kind of triangulation which conducts rules. However, if the layer is not interpreted correctly, the pinching the junction box cover by the gripper will fail even though the exact *x* and *y* positions are known. Furthermore, accuracy with the utilization of triangulation, the error rate in the interpretation of layered environments can be suppressed well.

B. GRIPPER PINCHING THE TARGET

Pinching a target with a three-finger gripper means that not only centroid must be known but also orientation. We use (10) for a system capable of calculating orientation effectively, even in this case, using conventional image processing methods. Before the gripper pinches the target, the orientation is first recognized in the initial position while this is part of the anticipated system failure, see Algorithm 2. The finger gripper condition is also in a straight open state to avoid the failure of the gripper controller system, which is a system separated from the robot manipulator controller.

The gripper movement can be divided into three parts, such as Algorithm 3. First, when the gripper is in the initial position of the finger pinch condition that obstacles to the FoV camera can be avoided. Second, the gripper moves from the initial position to the target. Gripper movement depends entirely on the 3D point resulting from localization; in such conditions, the gripper finger opens straight. When approaching the target, the gripper will stop in the safe pinch position, 3 mm on the table. Failure to interpret Z certainly has a fatal impact on the safe pinch position. Finally, the junction cover box is pinched by the gripper in a transverse position towards the target. After the pinch position, the gripper does not move directly to the placing position, but the target is raised about 400 mm measured from the lowest layer. This increase of specific pinch is to avoid the impact of the gripper tip against surrounding targets, and this is important for layered environment because the picking point is higher than the placing point.

Algorithm 3 in pinching the target that we saw is safe enough to avoid two collisions. There are at least two

Algorithm 2 Ascertain Whether Junction Box Covers Are Within the Layered Environment

```
o2:    detect regions $\alpha_{Z_i}$ of the target junction box
       cover $T_i$ in certain layered environment $Z_i$;
for every identified confidence level of junction box
cover $\alpha_{T_{i=1}}$ to $\alpha_{T_{i=4}}$ do
    thresholding the confidence level of the junction box
    $O_i \leftarrow \alpha_{T_{i=n}}$;
    if the confidence level $\alpha_{T_{i=n}} > 0.85$ then
        taking first four sample from whole targets
        $O_{i=1:4}$;
        calling the Algorithm 1;
    else if the first of four targets are far apart
    $20 \geq (O_{i=1} \vee O_{i=2} \vee O_{i=3} \vee O_{i=4}) \leq 180\ mm$ then
        applying the FIS function;
        if all perimeters are known $\forall peri_n = 1$ then
            applying 85 rules FIS $\leftarrow rules_{1:85}$;
            outputting of FIS $\rightarrow Z_{1:5}$;
        else
            increment to read the next target $\alpha_{T_{i=1}} ++$;
        end
    else if the first of four targets are concise
    $(O_{i=1} \vee O_{i=2} \vee O_{i=3} \vee O_{i=4}) \leq 19\ mm$ then
        applying the FIS function FIS $\leftarrow rules_{1:85}$;
    else if the FIS outputs are known
    $(O_{i=1} \vee O_{i=2} \vee O_{i=3} \vee O_{i=4}) \geq 20$ :
    $180\ mm \&\& O_{1:4} = ib$ then
        applying the FIS function FIS $\leftarrow rules_{1:85}$;
        outputting the depth FIS $\rightarrow Z_{1:5}$;
        if the FIS outputs are known FIS $\rightarrow Z_{1:5}$ then
            corresponds the depth with each detection
            result $\forall Z_n \leftrightarrow RCNN_n$;
        else if the confidence level > 0.85 and
        perimeters[i] in the range $\alpha_{T_i} > 0.85$ &&
        $peri_n = 230 : 5000$ then
            outputting the target layered env.; $T_{i[Z_n]}$;
            calling the Algorithm 3;
            increment to complete the target; computing
            in a layered env.;
            $i++$;
        else
            return o2.
    end
end
```

Algorithm 3 Ascertain Whether Junction Box Covers Are within the Gripper Pinching

```
o3:    coordinates of junction box cover $T_i$
       in certain layered environment of $Z_i$;
for every detected junction box $T_{i=1}$ to $T_{i=n}$ do
    $\perp \leftarrow \exists p_{[x,y,z]}$; adjusting the gripper transverse to the
    junction box cover's orientation;
    opening the gripper straight $gpo \leftarrow p_{[x,y,z]}$;
    if the gripper is moving down to the target
    $gpo\perp \rightarrow T_{i=1:n} = 75mm$ then
        stopping the gripper −3 mm before reaching the
        target $gpo\perp_{T_i} \downarrow\rightarrow (Z_{r[i=1]} - 3mm)$;
        adjusting the orientation based on targets
        $O_r \leftarrow o_{[x,y,z]}$
    else if $Z_{r[i]} > (Z_{r[i]} - 3)\ \&\&\ Z_{r[i]} \leq (Z_{r[i]} - 3)$ then
        heading the gripper to the target's order
        $gpo\perp_{T_i} \downarrow\rightarrow Z_{r[i=1:n]}$;
        lifting the target
        $gpp\perp_{T_i} \uparrow\rightarrow Z_{r[i=1:n]} = 400mm$;
        moving the target to home position $h_{pos}$;
        placing the target to the box $gpp\perp_{T_i}$;
        $i++$;
    else
        return o3.
    end
end
```

potential collisions during pinching work. The first collision occurs as the gripper finger moves to hit the layer, and the second collision between the gripper fingertip and another cover near the target. As a result, it is necessary to lift the gripper first to a certain point and heading to place a little bit longer to avoid a collision. The use of Algorithms 1 to 3 is an alternative to the Iterative Closest Point algorithm (ICP) that is commonly used [39]. Although ICP has weaknesses, it needs an initial guess, preprocessing steps, and not so fast because of finding the closest point in pairs. In addition to the specific performance of our system, please watch the following video link https://youtu.be/z3pdEWU-u80.

V. EXPERIMENTS

A. DETECTION METHOD EVALUATION

The detection result metrics [40] of precision, recall, F1 score, and Average Precision (AP) are evaluated as described in (16). To test the method of detection, the numbers of True Positive (TP) and False Positive (FP) were involved in 500 recorded images in total. A confidence value of 0.85 was set to compute the precision, recall, F1 score, and AP.

$$\begin{cases} precision = \dfrac{TP}{TP+FP} \\ recall = \dfrac{TP}{TP+FN} \\ F_1 = \dfrac{2 \times precision \times recall}{precission + recall} \\ AP = \displaystyle\int_0^1 p\,(r)\,dr \end{cases} \tag{16}$$

The results are shown in Table 3. It can be seen that the closest junction box covers had a higher rate of detection precision. It was obvious that the annotation process makes it simple to identify the junction box covers whereas it is more difficult to define the non-target (square blocks) ones because they have similarities in shape and color. This could be disturbing to the detection network.

B. EXPERIMENTS OF ESTIMATION FOR THE DEPTH

Technically, the solution to the depth evaluation employs the disparity map, the FoV alignment, and dataset matched by

TABLE 3. Evaluation of tests from process of detection.

Parameters	Classes	
	Target	Non-target
Confidence level	0.850	0.850
Precision	0.992	0.997
Recall	0.993	0.997
Performance (F1)	0.993	0.997
AP	**0.977**	**0.988**
Time detection (s)	0.573	0.716

kNN and FIS. The dataset consist of number of class, camera coordinate frame $X_cY_cZ_c$, robotic coordinate frame $X_rY_rZ_r$, and target coordinate frame $X_tY_tZ_t$. The secure distance for gripper finger was set to +3 mm above the target position stand on empirical experience. Around 62,525 datasets were collected with 1,684 junction box covers and 500 non-targets were tested. The results of the identification are shown in each layer order as shown in Table 4. Different to the detection results of the junction box covers, depths were found significant in a small difference, only in <0.9 mm and the overall percent error was −0.0005% with the average time detection 0.7056 s for each target.

TABLE 4. Depth metrics for each layered environment.

Layer no.	Actual depth	Depth estimation			Times (s)	
		μ	σ	% err.	μ	σ
1	291	291.044	0.023	0.0002	0.814	0.024
2	323	312.045	0.023	-0.0031	0.813	0.023
3	333	333.044	0.023	0.0001	0.759	0.023
4	353	353.047	0.025	0.0001	0.638	0.024
5	373	373.042	0.025	0.0001	0.504	0.025
Ave.				**-0.0005**	**0.7056**	

The accuracy of the depth was based on method 1, method 2, and method 3 and localization of the junction box covers was tested in each layer. Hence, the tests focused primarily on machine learning that the layered system had been identified precisely. The evaluation of the depth may not adequately accurate when testing for higher than the fifth layer or lower than the first layer.

C. GRASPING IN THE INDUSTRIAL ROBOT EVALUATION

We have tested the junction box cover, depth estimation, and localization method on actual MELFA RV-3SD robotic manipulator in industrial-like settings. This industrial robot comprises a robotic arm, a camera, and a three-finger gripper for picking junction box covers as shown in Fig. 7. An onboard GPU (Intel R UHD Graphics 630) is equipped for running the whole system and a PC computes and sends commands to the manipulation controller. The average processing time for one image frame, including running the detection network and coordinate transformation, was 0.7056 s, as shown in Table 4. The picking and placing 89 actual objects (68 of the junction box covers and 21 of non- targets) are evaluated in 640 × 480 pixels resolution. Table 5 listed the number of object testing as actual objects; from this testing, we can know raw detection and finally get output from method 3. The output of method 3 still consists of two kinds of object detection, targets and non-targets, and accuracy is the success rate of pick-and-place targets.

TABLE 5. Picking-placing success rate with localization and depth estimation method.

Test no.	Num. of objects			Output our method	
	Actual objects	Raw detect.	Output of Method 3	T/NT*	No. Picked
1	10	12	10	8/2	8/0
2	10	12	10	7/3	7/0
3	8	10	8	6/2	6/0
4	14	15	14	11/3	11/0
5	12	13	12	10/2	10/0
6	11	11	11	8/3	7/0
7	10	11	10	7/3	7/0
8	14	15	14	11/3	11/0
Sum.	89	99	89	68/21	67/0
Acc.	100%			**98.529%**	

T/NT = target / non-target.

Table 5 consists of an object number and output of our project. The eight tests were involving 89 actual objects (target and non-target). Totally 99 of raw detection results were produced from 89 actual objects. This means at least 10 noises from eight attempts, as in Fig. 5 (d) where there are seven actual objects T1 to T8, but T1 is a noise. Therefore, the outputs of method 3 not only are layered security of method 1 and method 2, but also ensure the amount of output as the input (actual object). Column T/NT describes as numbers of the target and non-target objects; for instance, 8/2 means there are eight targets and two non-targets. In our system, only targets are taken while non-targets are not picked up, as shown in the last column. In the sixth test, there was one target grasped but detached. Allegedly, that junction box cover began to crack, and it became rickety and detached when moving to place point.

In this evaluation, we use these three methods simultaneously. The picking rates of success in the localization method depend on method 1 and method 2, while if localization by method 2 fails, it will be treated by method 3. In which the scenario of moving down and going to homing or targeting was based on Algorithm 1. Each successful pinching to the home position *hpos* was considered as a successful picking-and-placing task.

The tests were conducted in varied situations, including those where the junction box covers were limited to 12 pcs and those where layers were replaced with a flat sheet, A3 sized. In this test, the dimension of junction box cover is 53 mm × 38 mm × 18 mm, and the number of successfully detected and pinched junction box covers of 85 trials are recorded in Table 4. It can be shown that in the changed setting, the optimized method of localization and an error rate of depth estimation achieved of 0.977 and −0.0005% respectively.

VI. CONCLUSION

This study intended a localization method and depth estimation algorithms for junction box cover grasping robots. The localization approach was based on CNN's segmented and mono camera depth images. To enhance localization accuracy, we applied the FoV, while disparity map-based layering was used to verify the depth points. The junction box covers were detected using R-CNN and determined using the kNN and FIS, and their locations were compared with the dataset of junction box covers to double-checking. The test results indicated that the optimized localization method could precisely locate the junction box cover, with a picking rate of 98.529% in industrial-like settings. The overall of the error rates for the junction box cover and an error rate of depth estimation were 0.993 and -0.0005%, respectively.

More sophisticated challenges, such as piled up and mounting positions like objects poured out from the container, will be considered in mini PC or AI embedded systems as our future work.

REFERENCES

[1] Y. Xiong, Y. Ge, Y. Liang, and S. Blackmore, "Development of a prototype robot and fast path-planning algorithm for static laser weeding," *Comput. Electron. Agricult.*, vol. 142, pp. 494–503, Nov. 2017.

[2] S. Hayashi, S. Yamamoto, S. Saito, Y. Ochiai, J. Kamata, M. Kurita, and K. Yamamoto, "Field operation of a movable strawberry-harvesting robot using a travel platform," *Jpn. Agricult. Res. Quart.*, vol. 48, no. 3, pp. 307–316, 2014.

[3] Y. Xiong, C. Peng, L. Grimstad, P. J. From, and V. Isler, "Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper," *Comput. Electron. Agricult.*, vol. 157, pp. 392–402, Feb. 2019.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Red Hook, NY, USA, Dec. 2012, pp. 1097–1105.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

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

[7] C.-M. Lin, C.-Y. Tsai, Y.-C. Lai, S.-A. Li, and C.-C. Wong, "Visual object recognition and pose estimation based on a deep semantic segmentation network," *IEEE Sensors J.*, vol. 18, no. 22, pp. 9370–9381, Nov. 2018.

[8] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, "Learning object grasping for soft robot hands," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2370–2377, Jul. 2018.

[9] Y. Ge, Y. Xiong, G. L. Tenorio, and P. J. From, "Fruit localization and environment perception for strawberry harvesting robots," *IEEE Access*, vol. 7, pp. 147642–147652, 2019.

[10] I. S. Mohamed, A. Capitanelli, F. Mastrogiovanni, S. Rovetta, and R. Zaccaria, "Detection, localisation and tracking of pallets using machine learning techniques and 2D range data," *Neural Comput. Appl.*, vol. 32, pp. 8811–8828, Aug. 2019.

[11] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4490–4499.

[12] Q.-C. Mao, H.-M. Sun, Y.-B. Liu, and R.-S. Jia, "Fast and efficient non-contact ball detector for picking robots," *IEEE Access*, vol. 7, pp. 175487–175498, 2019.

[13] H. Yang, L. Chen, M. Chen, Z. Ma, F. Deng, M. Li, and X. Li, "Tender tea shoots recognition and positioning for picking robot using improved YOLO-V3 model," *IEEE Access*, vol. 7, pp. 180998–181011, 2019.

[14] Q. Zhang and G. Gao, "Grasping point detection of randomly placed fruit cluster using adaptive morphology segmentation and principal component classification of multiple features," *IEEE Access*, vol. 7, pp. 158035–158050, 2019.

[15] S. S. Mehta and T. F. Burks, "Vision-based control of robotic manipulator for citrus harvesting," *Comput. Electron. Agricult.*, vol. 102, pp. 146–158, Mar. 2014.

[16] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on mask-RCNN," *Comput. Electron. Agricult.*, vol. 163, Aug. 2019, Art. no. 104846.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988.

[18] S. Bargoti and J. P. Underwood, "Image segmentation for fruit detection and yield estimation in apple orchards," *J. Field Robot.*, vol. 34, no. 6, pp. 1039–1060, Sep. 2017.

[19] S. Zhao, L. Zhang, Y. Shen, S. Zhao, and H. Zhang, "Super-resolution for monocular depth estimation with multi-scale sub-pixel convolutions and a smoothness constraint," *IEEE Access*, vol. 7, pp. 16323–16335, 2019.

[20] Y. Tian, Q. Zhang, Z. Ren, F. Wu, P. Hao, and J. Hu, "Multi-scale dilated convolution network based depth estimation in intelligent transportation systems," *IEEE Access*, vol. 7, pp. 185179–185188, 2019.

[21] Y. Xiong, Y. Ge, and P. J. From, "An obstacle separation method for robotic picking of fruits in clusters," *Comput. Electron. Agricult.*, pp. 1–9, 2020, doi: 10.1016/j.compag.2020.10539.

[22] M. S. Wang, "Eye to hand calibration using ANFIS for stereo vision-based object manipulation system," *Microsyst. Technol.*, vol. 24, no. 1, pp. 305–317, Jan. 2018.

[23] C. Cai, N. Somani, and A. Knoll, "Orthogonal image features for visual servoing of a 6-DOF manipulator with uncalibrated stereo cameras," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 452–461, Apr. 2016.

[24] Z. Chen, D. Zhou, H. Liao, and X. Zhang, "Precision alignment of optical fibers based on telecentric stereo microvision," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 1924–1934, Aug. 2016.

[25] G. Reina, A. Milella, W. Halft, and R. Worst, "LIDAR and stereo imagery integration for safe navigation in outdoor settings," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Linkoping, Sweden, Oct. 2013, pp. 1–6.

[26] P. Nguyen and V. A. Ho, "Grasping interface with wet adhesion and patterned morphology: Case of thin shell," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 792–799, Apr. 2019.

[27] C. Song and A. Boularias, "Inferring 3D shapes of unknown rigid objects in clutter through inverse physics reasoning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 201–208, Apr. 2019.

[28] M. Gupta, J. Muller, and G. S. Sukhatme, "Using manipulation primitives for object sorting in cluttered environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 608–614, Apr. 2015.

[29] M. Feng, Y. Wang, J. Liu, L. Zhang, H. F. M. Zaki, and A. Mian, "Benchmark data set and method for depth estimation from light field images," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3586–3598, Jul. 2018.

[30] C. Zhou, Y. Liu, Q. Sun, and P. Lasang, "Joint object detection and depth estimation in multiplexed image," *IEEE Access*, vol. 7, pp. 123107–123115, 2019.

[31] A. Caglayan and A. B. Can, "Volumetric object recognition using 3-D CNNs on depth data," *IEEE Access*, vol. 6, pp. 20058–20066, 2018.

[32] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[33] H. Yan, X. Yu, Y. Zhang, S. Zhang, X. Zhao, and L. Zhang, "Single image depth estimation with normal guided scale invariant deep convolutional fields," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 80–92, Jan. 2019.

[34] Y.-C. Du, M. Muslikhin, T.-H. Hsieh, and M.-S. Wang, "Stereo vision-based object recognition and manipulation by regions with convolutional neural network," *Electronics*, vol. 9, no. 2, p. 210, Jan. 2020.

[35] J. Fu, J. Liang, and Z. Wang, "Monocular depth estimation based on multi-scale graph convolution networks," *IEEE Access*, vol. 8, pp. 997–1009, 2020.

[36] X. Liu, D. Zhao, W. Jia, W. Ji, C. Ruan, and Y. Sun, "Cucumber fruits detection in greenhouses based on instance segmentation," *IEEE Access*, vol. 7, pp. 139635–139642, 2019.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[38] J. H. Park and H. W. Park, "Fast view interpolation of stereo images using image gradient and disparity triangulation," *Signal Process., Image Commun.*, vol. 18, no. 5, pp. 401–416, May 2003.

[39] X. Zhang, C. Glennie, and A. Kusari, "Change detection from differential airborne LiDAR using a weighted anisotropic iterative closest point algorithm," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 7, pp. 3338–3346, Jul. 2015.

[40] W. Ji, X. L. Meng, Z. Qian, B. Xu, and D. A. Zhao, "Branch localization method based on the skeleton feature extraction and stereo matching for apple harvesting robot," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 3, pp. 1–9, May 2017.

MUSLIKHIN received the B.Ed. degree in electronic educational engineering and the M.S. degree in technology and vocational education from Universitas Negeri Yogyakarta, Yogyakarta, Indonesia, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the Southern Taiwan University of Science and Technology, Tainan, Taiwan. His research interests include robotics, machine vision, and deep learning.

JENQ-RUEY HORNG received the B.S. and M.S. degrees in electrical engineering from National Cheng Kung University, in 1981 and 1983, respectively. He is currently an Associate Professor with the Department of Electrical Engineering, Southern Taiwan University of Science and Technology. His research interests include microcontroller-based systems design and DSP-based applications.

SZU-YUEH YANG received the B.Sc. and M.Sc. degrees in electrical engineering from the Southern Taiwan University of Science and Technology, Tainan, Taiwan, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His current research interests include positioning and navigation AGV and UAV, and the Internet of Things.

MING-SHYAN WANG received the B.S. degree in electronic engineering from National Chiao Tung University, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University, in 1985 and 1993, respectively. He is currently a Distinguished Professor with the Department of Electrical Engineering, Southern Taiwan University of Science and Technology. His research interests include servomotor drive design, robotics, and neural network control.