Self-Correction for Eye-In-Hand Robotic Grasping using Action Learning

Muslikhin¹,², Jenq-Ruey Horng¹, Szu-Yueh Yang¹, and Ming-Shyan Wang¹
¹Department of Electrical Engineering, Southern Taiwan University of Education, Tainan 710, Taiwan
²Department of Electronics Engineering Education, Universitas Negeri Yogyakarta, Yogyakarta 55281, Indonesia

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

This work was funded by Higher Education Sprout project of Ministry of Education, Taiwan and MOST 110-2221-E-218-023- of Ministry of Science and Technology, Taiwan.

ABSTRACT Robotic grasping for cluttered tasks and heterogeneous targets is not satisfied by the deep learning that has been developed in the last decade. The main problem lies in intelligence, which is stagnant, even though it has a high accuracy rate in usual environment; however, the cluttered grasping environment is very irregular. In this paper, an action learning for robotic grasping using eye-in-hand coordination is developed to grasp the cluttered and wide range of various objects using 6 degree-of-freedom (DOF) robotic manipulator equipped with a three-finger gripper. To involve action learning in this system, k-Nearest Neighbor (kNN), Disparity Map (DM), and You Only Look Once (YOLO) were needed. After successfully formulating the learning cycle, an instrument assesses the robot's environment and performance with qualitative weightings. Some experiments of measuring the depth of the target, localization of target variations, target detection, and the gripping process itself were conducted. The entire process is spread out in plan, act, observe, and reflect for each action learning cycle. If the first cycle does not suffice the results according to the minimum pass standard, the cycle will renew until the robot succeeds in picking and placing. Furthermore, this study demonstrated that the action learning-based object manipulation system with stereo-like vision and eye-in-hand calibration can improve intelligence over previous errors with acceptable errors. Thus, action learning might be applicable to other object manipulation systems without having to define the environment first.

INDEX TERMS Action learning, deep learning, eye-in-hand manipulator, k-Nearest neighbor, robotic manipulator; robotic grasping; YOLOv3.

I. INTRODUCTION

Mimicking human behavior for object manipulation means to study the inherent interaction between fast feedback involving perception and action. It is like a complex manipulation task to extract a single object from messy objects. It can be ascertained that almost without prior planning, without tactile feedback, and no vision, the manipulations cannot be done very well [1]. In contrast, robotic manipulation tends to rely on initial analysis and planning, with the following trajectory feedback, to ensure adherence during execution. In other words, it usually uses multiple sensors, fusion sensors, or tactile sensors but this requires a certain approach before being used as continuous feedback. Continuous feedback is required in visual servo technique that requires feature identification [2]. Both open-loop perception and feedback features require calibration to determine the accurate geometric relationship between the end-effector of the robot and the camera [3], also involving some deep learning processes.

Latest decade in deep learning, AlexNet was added to the Convolutional Neural Network (CNN) by Krizhevsky et al. [5]. The Faster R-CNN (Region based CNN) has better precision speed than AlexNet, CNN, R-CNN, and Fast R-CNN. Redmon et al. [6] provided another highly capable method with the YOLO (You Only Look Once), the last one being YOLOv3 [7]. In YOLOv2, the speed of detection is even more significant than the Faster R-CNN. The prowess of deep learning needs to be supported by other techniques to be applicable in robotics.

Previous work by Levine et al. [8] from Google Inc. employed hand-eye coordination for the grasping robot that successfully grasped a new object through continuous servoing. The construction of hand-eye coordination has the
advantage of ease in estimating object localization, but the camera field of view (FoV) will be blocked by the robot arm itself [8],[9]. Besides, study [8] used huge data about 800,000 hand experiments involving at least 6 to 14 manipulator robots in parallel. This method is not practical in terms of time and needs many robot units. The interesting thing about this study is to grasp a new object that has not been recognized.

Although work [8] has involved many datasets in deep learning, the number of successful grasping experiments is unpredictable. Generally speaking, deep learning that has currently been developed still has weaknesses on applying to dynamic environments, such as in heterogeneous objects, wide range of targets, and cluttered objects. The nature of deep learning is very dependent on the learning rate at the training stage that has been given. However, the ability of deep learning that is specific and generalized turns out to have a weakness if the targets are overlapping and/or partially visible and is related to decision making. For that, deep learning generally needs to collaborate with other systems, as for supervised or unsupervised learning [10]–[12].

Some examples incorporating deep learning in several systems are becoming prevalent and have been widely applied, such as Shi et al. [13] and Tsai et al. [14] respectively implemented Deep Reinforcement Learning (DRL) and Deep CNN (DCNN) for mobile robots. Similar work was done by Chen et al. [15] by combining DRL, RNN (Recurrent Neural Network), and LSTM (Long Short-Term Memory), but its ability was less than 47%. Riviere et al. provided an outstanding achievement with end-to-end learning using the DCNN and Graph Neural Network (GNN) approaches that can be run on low-end microcontrollers but limited to the number of six obstacles and neighbors only [16]. In addition to the use of end-to-end techniques, incorporating deep learning is often found for Reinforcement Learning (RL).

Currently, [4], [14], and [17] worked with RL and followed up by deep network networks. Although RL is quite powerful after being combined with other techniques, its intelligence does not be improved because the environment determines the value at each stage and agents are trained with static data, which are not suitable for a changing environment. We try to solve RL's shortcomings by offering a novel action learning; it is an improved method without setting the value for each state. Action learning has been implemented in education for a long time ago [10], [11], [18], [19], but adapted to robotic or artificial intelligence (AI) has not been reported.

The action learning principle imitates the human learning method, where in addition to having past learning, the robot will also evaluate itself and the environment from several assessment indicators. In this way, the action learning will have a learning cycle repeated until it meets specific passing grade. Besides the robot's primary intelligence, it also learns to improve its capabilities by introducing the action learning. In practice, we will apply to the cluttered bin for the pick and place task. Therefore, we expect that our robot system, powered by an action learning in grasping, will be more effective.

Specifically, we propose to develop a vision-based object manipulation system using a standard robotic manipulator that is capable of picking and placing objects from cluttered positions and overlapping, which are frequently confronted while picking for the eye-in-hand manipulator. The following details are given in this paper:

- A stereo camera-like is employed to estimate the targeted depth, which is variable, heterogenous, and cluttered, also a localization method based on DM-kNN is proposed.
- We strive to be as accurate as possible recognition and detection objects with modified YOLOv3 as basic detection and self-correction validation.
- The learning independently from mistakes is considered by developing action learning for target-picking and placing tasks and depth collision problem for manipulators in layered environments. The environment as a reference value to make decision in a single cycle is assessed.
- The proposed action learning system in the task of picking targets applied to a six degree-of-freedom (DOF) robot manipulator with a three-finger gripper was performed and evaluated, it might provide alternative ways to similar robot cases.

In this paper, we discussed the system design overview in Section II. Section III introduces self-correction for robotic grasping and action learning on cluttered environment will be detailed in Section IV. The next Section V describes the experimental results. Finally, we conclude the work and offer ideas for possible future works in Section VI.

II. SYSTEM DESIGN OVERVIEW

A. PROPOSED SYSTEM DESIGN

The whole system is shown in Figure 1, where the dashed line box refers to the action learning process, while the green part provides preprocessing inputs of action learning and the red box is the robot goal. The goal to complete the moving and picking-placing task makes the gripper avoid confusing decision; hence, the procedures will cut off the time by assessing some indicators or inputs.

The inputs of DM (Disparity Map), YOLOv3, kNN, orientation/edge detection, and β are RGB images with a resolution of 640×480 pixels. The output of DM is far/near distance in the range of 270-300 mm. The YOLOv3 output is the result with a confidence level in percentage (%) and kNN output is in the form of coordinates (X, Y, Z). Output orientation is in the form of position degree (°) and environmental assessment value β and passing grade value β̂p are in the form of values 0–100. All these values will be fed into the plan in each cycle to proceed and then become a decision. Given the large number of inputs and a variety of targets, it is necessary to limit the specific scope of work from our proposed action learning.

VOLUME XX, 2020

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
B. THE SYSTEM LIMITATION

The developed action learning with eye-in-hand configuration is limited to being able to pick and place for the 10 target classes that have been trained, and the number of cycles in action learning cannot be predicted if we do not make a limitation. In this paper, we only limit twice. We did this to minimize the target dislocation due to the collision between the robot’s finger and the tray if there is no restriction on the retrieval experiment. Further explanations are discussed in Section IV.

III. SELF-CORRECTION FOR ROBOTIC GRASPING

A. STEREO CAMERA-LIKE WITH DEPTH ESTIMATION

The stereo camera was developed from a mono webcam Logitech C920 with the resolution 640×480 pixels as shown in Figure 2. The same camera is placed on the coordinates of the initial point \( O_1 \) (50, 450) and shifted along the x axis to become point \( O_2 \) (150, 450). The same camera is placed in an imaginary rigid surface aligned in the y axis and 100 mm apart in the x axis. The cameras must also be perfectly aligned to avoid the height offset generated on the resulting 3D image.

To measure the discrepancy of two cameras aligned, the blue dot positions of the object in 2D image plane are computed then the x values and y values between two images on left and right cameras are compared as illustrated in Figure 3. The difference value of \( y_1 \) and \( y_2 \) should be zero which indicates the two cameras aligned. Figure 3 shows the blue dot appeared on image plane of left camera with coordinate is \( p_1(x_1, y_1) \) and the dot point appeared on the right image plane with coordinate \( p_2(x_2, y_2) \). The distance between left camera center (optical center) and the right camera center (optical center) is called baseline \( (b) \). The distance between \( x_1 \) and \( x_2 \) is called disparity distance \( (d) \), as shown in Eqs. (1-3) where \( Z \) is the depth of point \( P \) and \( f \) is the camera focal length.

![Figure 1. Overall architecture diagram.](image1)

![Figure 2. Developed stereo camera-like using end-effector baseline.](image2)

![Figure 3. Stereo camera geometry.](image3)
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3129474, IEEE Access

where pixel locations on the 2D image are \( x_1 \) and \( x_2 \), and actual positions on the 3D image are \( X, Y, \) and \( Z \).

On the other hand, the camera's FoV can be used for depth verification by finding the \( dx \) value, see Eq. (7). If diagonal FoV (DFoV) is given in Figure 4, both vertical FoV (VFOV) and horizontal FoV (HFOV) for the C920 camera can be found. Because this camera employs a 16:9 CMOS sensor by default, we should convert it to the 4:3 aspect ratio using Eq. (8).

![Figure 4. The FOV of Logitech C920.](image)

The disparities between the left image and the right image are derived from Eq. (9) and Eq. (10), where \( \varepsilon_{\text{DFoV}}(x,y) \) is the normalized BM error with the horizontal disparity \( d \), \( W \) is window of the BM, \( D_{\text{max}} \) is the maximum value of disparity within the permissible limit, and \( u \) and \( v \) are the number of pixels in the \( xy \) camera image plane, respectively. To check the observed disparity, Eq. (11) expresses the disparity from the right image frame \( f_r \) to the left image frame \( f_l \), where \( \varepsilon_{\text{MME}}(x,y) \) determines how close the pair image values in the left \( (x,y) \) and right \( (x+d,y) \) images are to the same points. The MME is well-known from its effectiveness Eq. (12).

![Diagram](image)

The \( D(x,y) \) disparity map represents the displacement of the corresponding pixels between the left and right images. However, locating corresponding pixels is difficult. Some variables may cause problems in the non-occlusion pixels, such as non-textured, camera noise, homogeneity, and repeated texture. The disparity is calculated for all pixels using Block Matching (BM), and the validity of the disparity significance is defined as follows.

\[
D_{\text{L-R}}(x,y) = \arg \min_{d \in [0, D_{\text{max}}]} \varepsilon_{\text{DFoV}}^d(x,y)
\]

\[
\varepsilon_{\text{DFoV}}^d(x,y) = \sum_{(u,v) \in W} \left| f_l(x-u,y-v) - f_r(x-u+d,y-v) \right|
\]

\[
D_{\text{R-L}}(x,y) = \min_{d \in [-D_{\text{max}}, 0]} \varepsilon_{\text{DFoV}}^d(x,y)
\]

The Minimum Matching Error (MME) determines how close the pair image values in the left \( (x,y) \) and right \( (x+d,y) \) images are to the same points. The MME is well-known from its effectiveness Eq. (12).

\[
\text{MME}(x,y) = \varepsilon_{\text{DFoV}}^d(x,y)_{|d=D_{\text{R-L}}(x,y)}
\]

Apart from DM, other methods such as kNN are needed. The principle of kNN [21] is found in the application of robotic assistance, another study involved the Kinect sensor with the kNN algorithm [22]. kNN is applied for classification based on the closest distance to reference. At the same time, kNN is reported to have a weakness in distinguishing entities from each object, but the opportunity for classification with multiple references is open to this method.

**B. TARGETS DETECTION AND LOCALIZATION**

In this study, YOLOv3 was used as the basis for determining target localization. The results were in the form of confidence level and its bounding box. The square shape of the bounding box will be used as the basis for
Determining the target grip point. Therefore, detection using YOLOv3 is critical. Two crucial things related to target detection with YOLOv3 and the detected object’s orientation need to be explained further.

1) TARGET DETECTION

The essential part of a detection involves deep learning, in other words which one feature extractor will be chosen. So far, no one claims about a standard feature extractor for one algorithm like YOLO. This opens opportunities for developing customization of the algorithm into hardware [23], [24]. YOLOv3 can add more variations to the training data by utilizing data augmentation rather than increasing the number of labelled training samples, YOLOv3 with SqueezeNet shown in Figure 5. Data augmentation techniques include random horizontal flipping, random scaling by 10%, and color jitter augmentation in HSV space.

The cyan is a feature extraction network using SqueezeNet, the purple color indicates the first detection head, and the gray is the second detection head with their respective outputs. In this SqueezeNet, we use nine depth concatenation layers with an input size of 227×227×3 in the image form. There are 86 layers with a connection number of 75, and the output type is a classification of 10 classes. The basic idea of the YOLO architecture is to employ two networks simultaneously and the process to be quickly bypassed in certain parts. YOLOv3 uses logistic regression to estimate an objective score (confidence) for each bounding box.

The original SqueezeNet settings for the activation function are preserved by using the Rectified Linear Unit (ReLU) function in the fire modules [23]. The leaky ReLU function will be followed by the fully connected (FC) layers. Leaky ReLU is a modified version of ReLU with a slight slope in the function output for negative data [6]. So, the derivative is never zero; it can reduce the appearance of silent neurons, which solves the problem of ReLU failing to learn when negative intervals are encountered. The following is how the term leaky ReLU is defined as Eq. (13).

$$
\phi(x) = f(x) = \begin{cases} 
  x, & x > 0 \\
  0.1x, & x < 0 
\end{cases}
$$

During training, our model will be optimized using the categorical cross entropy loss function:

$$
\text{loss} = -\sum_{i=1}^{n} \hat{y}_{1i} \log y_{1i} + \hat{y}_{12} \log y_{12} + \cdots + \hat{y}_{im} \log y_{im}
$$

(14)

where n and m represent the number of samples and the number of categories, respectively. The $y$ represents the true value and $\hat{y}$ represents the prediction value.

In practice, it is necessary to make the loss function pay more attention to the categories with small samples, which will help solve the sample imbalance problem. To make the model training run smoothly and avoid overfitting, we add loss factors to the loss function as in Eq. (15):

$$
\text{loss} = -\sum_{i=1}^{n} \lambda_{1} \hat{y}_{1i} \log y_{1i} + \lambda_{2} \hat{y}_{12} \log y_{12} + \cdots + \lambda_{m} \hat{y}_{im} \log y_{im}
$$

(15)

The values of loss factor $\lambda$ have been listed for different target categories, calculated as Eq. (16):

$$
\lambda_{i} = \frac{C_{i}}{mN_{i}}
$$

(16)

where $C_{i}$ represents the total number of samples. The $N_{i}$ represents the sample amount of class $i$, while $n$ is the number of target categories.

2) TARGET ORIENTATION

After succeeding in identifying the grasping point, object orientation is necessary for robot grasping. If the target on tray position is cluttered, so orientation recognition is required. In a cluttered environment, orientation is necessary because overlapping or overlapped objects can form new orientations. On the other hand, picking up and placing objects such as circles and spheres or picking up using a vacuum gripper (non-finger) does not require orientation. Broadly, the traverses are grouped into five types based on the ratio of the longest axis to the horizontal $x$ axis.

The estimated region of the subject provided by the MATLAB function is used to calculate object orientation ranging from −90° to 90°. During the eye-in-hand adjustment process, these orientation data must be adjusted to the end-orientation effector’s so that the object can be grasped properly. The angle formed by the $x$ axis and the ellipse’s major axis, as shown in Figure 6, is known as object-orientation [25]. The relationship among the horizontal $x$, vertical line $y$, width $W$ and height $H$ of the object is given in Eq. (17),

$$
(a) \begin{cases} 
  0 \leq x \leq W' \\
  0.25 H' \leq y \leq 0.75 H'
\end{cases}
$$

(b) \begin{cases} 
  0.25 W' \leq x \leq 0.75 W' \\
  0 \leq y \leq W'
\end{cases}
$$

(17)
The ability to absorb and process information is crucial in various learning approaches. The learning process through previous experience and reflection ability forms the basis of Metacognition, where essentially knowing the meta-memory ensures the learner will most likely be repeated if a certain behavior is reinforced. It can also learn from other agents or resources. Essentially, the learner involves examples to be learned next from the syntax, so it can be reduced into a procedure or algorithm.

After the emergence of AI in the last decade, algorithms for robotic manipulators seem to increase again. It is commonly known that learning for robots is formerly imitated from human education learning. Several learning theories have been adopted, and each learning theory has its syntax, so it can be reduced into a procedure or algorithm. The emergence of action learning was introduced by Altrichter et al. and Dick et al. [28], [30], and [39], it had undergone several modifications by Bell, Aldridge, Whitehead, Mc Niff, Norton, Stringer et al., and some even called it classroom action research/action research. Details about action learning are discussed in the next subsection. So, the development of action learning in robotic manipulator has not been in scientific publications in engineering, and it is still limited to the field of education [29], [35], [40], and [41].

3) TARGET LOCALIZATION

The localization of the target is the combination of the X, Y coordinates, Z depth and orientation. Targets that YOLOv3 successfully recognizes will be used as an external reference in addition to the centroid of the target containing the XY coordinates. Meanwhile, FoV and disparity map order verify the results from camera-like stereo to Z depth. Both are combined, including orientation, so 3D points are formed with each orientation, as shown in Figure 7.

![Figure 7](image7.png)

FIGURE 7. DM kNN architecture with FoV in depth estimation.

C. ACTION LEARNING FOR SELF-CORRECTION

After the emergence of AI in the last decade, algorithms for robotic manipulators seem to increase again. It is commonly known that learning for robots is formerly imitated from human education learning. Several learning theories have been adopted, and each learning theory has its syntax, so it can be reduced into a procedure or algorithm.

1) APPROACH OF ACTION LEARNING IN ROBOTIC

Broadly, the emergence of action learning was introduced by Altrichter et al. and Dick et al. [28], [30], and [39], it had undergone several modifications by Bell, Aldridge, Whitehead, Mc Niff, Norton, Stringer et al., and some even called it classroom action research/action research. Details about action learning are discussed in the next subsection. So, the development of action learning in robotic manipulator has not been in scientific publications in engineering, and it is still limited to the field of education [29], [35], [40], and [41].

![Table 1](image1.png)

TABLE 1. Comparison among learning approaches in education practice.
To adopt action learning in the robotics field, it is necessary to understand the concepts of general learning approaches that have existed, including benchmarking them in Table 1. It should be emphasized that action learning is different from other learning approaches. Meanwhile it has some similarities in syntax, such as planning, acting, assessing, reflecting, evaluating, or reviewing in active learning, reinforcement learning, metacognitive learning, and experimental learning, but these are different as a whole process.

Action learning architecture contains cycles; there are four stages in one cycle. The number of cycles cannot be determined or limited. It is just the cycle will stop when it reaches a predetermined threshold. The threshold value is obtained from an evaluation instrument, and usually, in a single instrument, some items indicate performance indicators. The performance value of this instrument will continue to be evaluated in each cycle.

2) STEPPING IN EACH CYCLE
The four stages in one cycle are planning, acting, observing and reflecting. The first stage is a plan; some of the inputs are analyzed using a particular approach at this stage. The second stage, act, this part is a form of execution of an action that has been planned. Act in the robotic manipulator is described as a motion series starting from the initial position towards the target until the gripping process returns homing. The third stage observes the system's observations after the act is carried out through the assessment instrument. The last part is reflecting, which performs an evaluation for the robot, especially the success of the verification in this section. If the target grasping process is not successful, then the next cycle is recycled.

In order to verify an eye-in-hand configuration using action learning, the target pick and place task was performed as following details. Although we introduced action learning, reinforcement learning was an inspiration. The Bellman equation used by reinforcement learning gives a discounted value from the goal point; the possible paths are trained to get the maximum value. In this way, of course, the value of each state in reinforcement learning is defined previously. In contrast, on action learning, the value is removed and replaced with a real-time assessment based on the instrument, or we also call it a pass grade for learners. The pass grade value is denoted by $\beta_p$. The assessment value comes from the eight assessment instruments in Table 2 and the illustration in Figure 8.

It should be declared that the output of YOLOv3 detection is $\delta$ and $\varepsilon$ is the result of target localization, where $\delta, \varepsilon \in \mathbb{R}$. The value obtained from the eight assessment indicators in Table 2 could be written as Eq. (18),

$$
\beta = \sum_{i=1}^{n} n_i \omega_i + n_{i+1} \omega_{i+1} + \cdots + n_{i+7} \omega_{i+7}
$$

FIGURE 8. The flow of action learning with each cycle with input and output.
So, the plan in the first cycle can be written as Eq. (19), where the plan is symbolized by \( P_t \), action by \( a_t \), observe by \( o_t \), and reflect by \( r_t \).

\[
P_t = \beta \land \delta \land \varepsilon \Rightarrow a_t
\]  

If the \( P_t \) has fulfilled the conditions by \( \beta, \delta, \varepsilon \), it will continue to the \( a_t \) process, with conditions such as Eq. (20).

\[
a_t \neq 0 \Rightarrow o_t
\]  

From Eq. (20), we could write Eq. (21), and the value of the reflection result is dependent on \( o_t \) with binary properties,

\[
\begin{align*}
\{ & \alpha_t = \beta \land \delta \Rightarrow r_t; \beta, \delta, \varepsilon \neq 0 \\
& \{ \alpha_t = 0; \quad o_t \in \{0,1\} \}
\end{align*}
\]

If \( r_t = 1 \), then the cycle stopped; otherwise, if \( r_t = 0 \), it will scroll to the next cycle to evaluate \( \beta_{t+1} \) and return its value. In Eq. (21), when observation \( o_t \) collects inputs from the value \( \beta \land \delta \), it means that in this position, YOLOv3 works for the second time to make sure if the target has been grasped or not and turns into second cycle.

3) ASSESSMENT OF THE ENVIRONMENT

To develop action learning in manipulator robots, the robot’s perception of the target must be validated [34]. Assessment of robot perceptions is carried out using eight items of assessment instruments. Every single process of grasping attempt will obtain one assessment result with a range of 1-100. Recording and comparing the data \( \beta_n \) with the results obtained at that time are presented in a probability density curve. The normal distribution (also known as the Gaussian distribution) is a two-parameter curve family. The central limit theorem states (roughly) that as the sample size grows to infinity, the number of independent samples from any distribution with finite mean and variance converges to the normal distribution. The normal distribution curve has been widely used to generalize, predict, and analyze decision making [42]–[47].

Furthermore, the basics of normal distribution have been commonly used for the development of deep learning.

For action learning to work well, apart from the innate intelligence obtained by the manipulator robot through YOLOv3, other instruments are still needed. This assessment instrument is a function to assess environmental conditions in one cycle. This dissertation uses eight initial data to be generalized plus the latest data to update the latest environmental conditions based on eight indicators. Each datum has its scale and weighting. These scales and weights are not standardized but are arranged accordingly. Instrument details with all indicators are presented in Table 2, and the results of this instrument assessment on environment are denoted by \( \beta \) which is separated into two parts \( \beta_1 \) and \( \beta_2 \). The results of this assessment will be the decision-maker for action learning in stimulating the robot.

The results of the assessment in Table 2 are presented in the form of a bell curve. The normal distribution is popular for modeling unbiased uncertainties and additive random errors, as well as symmetrical distributions of many natural processes and phenomena [44]. A commonly cited rationale for assuming normal distributions is the central limit theorem, which states that the sum of independent observations asymptotically approaches a normal distribution regardless of the shape of the underlying distribution.

\[
PDF: f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2} \right); \quad -\infty \leq x \leq \infty
\]

where \( \mu \) is the mean and \( \sigma \) is the standard deviation.

Although a CDF is cumulative distribution function \( F(x) \) does not have a closed-form solution, it is frequently presented using the complementary error function solution. However, it can be expressed in terms of a standard normal CDF, \( G(\cdot) \),

\[
F(x) = G(x - \mu) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x-\mu} \exp \left( -\frac{1}{2} \frac{z^2}{\sigma^2} \right) dz
\]

The probability coverage corresponding to a given interval around the mean is often used to describe the symmetrical nature of the distribution. For example, the interval \([\mu \pm 1\sigma]\) corresponds to \( P(A) = 0.683 \), the interval \([\mu \pm 2\sigma]\) corresponds to \( P(A) = 0.954 \), and the interval \([\mu \pm 3\sigma]\) corresponds to \( P(A) = 0.997 \).

D. ROBOT COORDINATE TRANSFORMATION

Triangulation of methods for completing camera to robot manipulator coordinate transformation has been reported [7], [25], [48]. Triangulation does not require prior knowledge, calibration, training, and a wide variety of methods. The use of kNN, disparity map, HOG and FOV is a potential approach. The advantage of each method will complement the transformation of the camera coordinates to the coordinates of the manipulator robot, thereby overcoming the complexity of estimating the 3D position of the world target.

The camera is put on the end-effector with the eye-in-hand configuration and takes pictures in the cluttered 2D target coordinates of camera frame \( C \). It’s essential to transform frame \( C \) to end-effector frame \( E \) [49]. Figure 9 will make it simpler. The target frame \( P \) is the object to be grasped, in the image it is indicated by a blue ball that is on the chessboard frame \( B \). Suppose \( B \) is the location for

![Image](457x801 to 549x824)

**TABLE 2. Environmental assessment instrument on action learning.**

| Assessment Aspect (Indicators) | Scale/Probability |
|-------------------------------|------------------|
| Before Plan and Act (\( \beta_1 \)) | 0 1 2 3 oo |
| Measure the YOLOv3 result on a single process (\( \delta \)) | - | ≤80 81–95 ≥96 | 5 |
| Assess the current environment of \( \beta_2 \) value | - | ≤75 76–90 ≥91 | 3 |
| Estimate the XY-coordinate using kNN-FOV (\( \varepsilon \)) (mm) | - | ≥61 60–11 ≤10 | 3 |
| Estimate the Z coordinate using stereo camera-like and DM (mm) | - | ≥6 3–5 ≤2 | 4 |
| Before Observe and Reflect (\( \beta_2 \)) | No - - Yes | 15 |
| Ensure target is not overlap with each other | Yes - - No | 9 |
| Compare the current of \( \beta \) with the previous of \( \beta_2 \) (\%) | - | ≥21 6–20 ≤5 | 3 |
| The value CDF in the next cycle | - | ≤75 76–90 ≥91 | 5 |

VOLUME XX, 2020
cluttered target \( O \) in the frame. Let \( R^O \) represent the position of cluttered targets \( O \) in relation to the robotic base frame \( R \), and \( C^O \) represent the position of cluttered targets \( O \) in the \( C \) frame. Eq. (24) is used to express the transformation of the target coordinate from camera frame \( O \) to robotic base frame \( R \).

\[
\frac{\vec{O}}{R^S} = \frac{\vec{O} C}{E^S + E^R}
\]  

(24)

\( \frac{\vec{O}}{R^S} \) be obtained from the structure of the MELFA RV-3SD robot manipulator shown in Table 3, including the joint \( j \), angle between two connection rods \( \theta \), length of link \( l \), angle of torsion connected with rod \( \alpha \), and the distance between the two connection rods \( d \). The Denavit-Hartenberg (DH) parameters shown in Table 3 for manipulator control are the most common for inverse kinematics to control manipulators.

**IV. ACTION LEARNING ON CLUTTERED ENVIRONMENT**

**A. VARIOUS TARGETS ON CLUTTERED ENVIRONMENT**

The targets laid on the tray are within the gripper’s reach so that the centroid position of each detection result needs to be searched. The most uncomplicated technique is to calculate the centroids from the bounding box expressed in Eq. (25) below:

\[
B_{box} = \begin{bmatrix}
  a_{11} & a_{12} & a_{13} & a_{14} \\
  a_{21} & a_{22} & a_{23} & a_{24} \\
  a_{31} & a_{32} & a_{33} & a_{34} \\
  \vdots & \vdots & \vdots & \vdots \\
  a_{n1} & a_{n2} & a_{n3} & a_{n4}
\end{bmatrix}
\]  

(25)

The bounding box matrix \( B_{box} \) has four columns \( a_{i[1...4]} \) and the number of rows depends on the number of detected targets \( d_{[n]} \) on each coordinate. So, we could find the centroids \((X_{cen}, Y_{cen})\) from Eqs. (26) – (27) as follows.

\[
X_c = B_{box}(:, a_{11}); Y_c = B_{box}(:, a_{12})
\]

\[
a = B_{box}(:, a_{13}); b = B_{box}(:, a_{14})
\]

\[
X_{cen} = X_c + \frac{a}{2} \\
Y_{cen} = Y_c + \frac{b}{2}
\]  

(26)

(27)

From Eq. (27), the centroid can be calculated and becomes the reference point for a gripper to pick the target. The centroid point in this condition is still in 2D image, so it is necessary to add \( Z \) value obtained from stereo camera-like Eqs. (4)-(6) and verified by Eqs. (25)-(27).

**B. GRASPING THE LOCALIZED TARGET**

Before the target detection process and proceeding with localization, the parameter options for YOLOv3 need to be clarified. The difference in parameters certainly affects the results of deep learning itself. The value of the initial learning rate, the mini-batch size and maximum epoch applied will significantly affect detection accuracy and time consumption during training. For example, if the learning rate is too low, then training takes a long time. On the other hand, if the learning rate is too high, then training might reach a suboptimal result or diverge. The followings are the training option parameters applied in the paper; SGDM optimizer (Stochastic Gradient Descent with Momentum) as a solver for training network, initial learn rate 0.001, verbose set true, minibatch size of 16, max. epoch of 30, shuffle being never, and verbose frequency of 30.
A detector that has been formed from training can be seen in general performance based on the training loss for the required iteration numbers. Figure 10 shows the results of the YOLOv3 detector training with different optimizers (SGDM, ADAM (Adaptive Moment Estimation), and RMSProp (Root Mean Square Propagation)) in this paper. It was proved that, compared to the other two, SGDM was the best as displayed in Figure 10.a. It can be seen that in the 100th iteration, the value of training loss is almost close to zero and continues to tend to stagnate after more than 200 iterations in Figures 10.a,c,e, while the RMSE training can be seen in Figures 10.b,d,f. The precision of this detector is crucial for overall system testing verification. Ideally, the precision is one at all recall levels. Figure 10 is only a sample to see the detector’s performance from the attractor class and ring class. For that, we summarize it in the form of training RMSE for cautiously understanding. It can be seen that out of ten classes, the ring has better performance than other detectors, as shown in Figure 11.

We also compare detectors; in general, the process of making detectors is preceded by a labeling process. As we have done in our previous work [24], [50], modifications in labeling also have a significant effect on detection speed. In the paper, ten objects are used as targets in Figure 12. In a typical training detector, this can be done simultaneously for labeling ten targets to be only one detector for all ten targets. In contrast to the second method we used, where the ten targets have their own detectors, the number of detectors will increase. We call this method a parallel detector so that it can shorten the detection time by 1.34 times, and we applied in action learning.

In this paper, SGDM is the best as an optimizer and we use it. After that, after the detection and localization process went well, we determined the grasping point, as shown in Figure 13. Figure 13 shows the sequence of the recognition and detection of interfering target. Starting from Figure 13.a as the input image, Figure 13.b is the result of recognition by YOLOv3. The orientation of each object can be determined by using the traditional image processing technique, and now the target is marked with a (yellow ×) and a grip point (black +) as shown in Figures 13.d. The green circle shows the gripper finger range, including the five green circles in Figure 13.e that is the minimum number of obstacles assumed to be on four sides. It is clear the green circle on the ring and bottle overlaps the target (attractor). In the last process, Figure 13.f, the grasping point is given for each potential obstacle with a blue circle so that the bottle with the highest overlapping area will be shifted first. In Table 4, performance by three optimizers in YOLOv3, SGDM (S), ADAM (A), and RMSProp (R) are compared.

V. EXPERIMENTS

A. DETECTION METHOD EVALUATION

The following is a separate test of the YOLOv3 performance for ten targets. YOLOv3 testing includes the level of confidence, accuracy, precision, recall, performance, average precision, and computation time required [26]. A confidence value of 0.85 was set to
compute the detection result metrics. The results are shown in Table 4, in which can be seen that targets had a higher rate of detection precision.

Obviously, the annotation process simplifies the targets, whereas it is tougher to define the obstacles, such as piled up or sticky. This could not disturb the detection network but challenges to perform pick and place on that target. So, this experiment is necessary to measure the distance between the target and the obstacle using kNN. This measurement is done by assigning five closest points, assuming one closest point for the target to the centroid of the bounding box and the remaining four closest points for the right, left, front, and rear obstacles.

**B. EXPERIMENTS OF SELF-CORRECTION FOR GRASPING**

Regarding stereo camera-like coordinates, the coordinates of the 3D object's position are obtained using the verification method as discussed in Section IV. The green circle indicates the width of the gripper, and if there is an overlap with other circles, it means that other objects are disturbing. The overlap needs to be separated in order to grasp the target. In eye-in-hand coordination, we don't do any training. Because in this section, the system performs calculations based on the estimation results of the eye-in-hand camera. We have also included a video version at https://youtu.be/DJZ8oLop5E8 to provide a comprehensive understanding.

Figure 14 shows the sequence of targeting without using action learning. In Figure 14.a, the gripper approaches the object's position along with the estimation results of its orientation. The 6D (XYZABC) position object pose estimation has succeeded in estimating the position including the target orientation to the camera coordinates on the end effector, as follows: -10.1 mm (x axis), -474.3 mm (y axis), 268.0 mm (z axis), 178.0° (α axis), 1.0° (β axis), -51.6° (γ axis/orientation). The correction pen (square-shaped) is the target and β was set at 48. However, in Figure 14.b, it seems that the position of the y axis is a little less precise, and it is not good enough for the robot trying to grip the target. As a result (Figure 14.c), the robotic manipulator failed, and it does not try to repeat because it does not use action learning, in other hand β = 48 only.

The last two grasps attempt using action learning. The gripper is positioned parallel to the target orientation’s estimation results as shown in Figure 15.a. The 6D target pose has been successfully estimated, which includes the orientation of the target object to the camera coordinates on the end effector, as follows: 50 mm (x axis), 450 mm (y axis), 500 mm (z axis), -178.0° (α axis), 1.0° (β axis), -88° (γ axis/orientation) with value of β = 70 and attractor (cylindrical shape) as a target. The target is successfully grasped (Figure 15.b) and placed into black tray (Figure 15.c).

Figure 16 shows the sequence of targeting using action learning, but in Figure 16.a, the robot evaluates the attractor’s situation by recognizing the presence of overlap interference by the scissors. The system’s decision is made so an overlap scissors is shifted by gripper finger as shown in Figure 16.b and grasping try to re-identify, re-grasp (Figure 16.c), and place it into a black tray (Figure 16.d). The 6D object pose estimates are as follows: 101.8 mm (x axis), -453.5 mm (y axis), 380.0 mm (z axis), -178.0° (α axis), 1.0° (β axis), -88.6° (γ axis/orientation).

**TABLE 4. Performance of YOLOv3.**

| Parameters | Conf. | Acc. | Prec. | Rec. | FL | AP | T (s) |
|-----------|------|------|-------|------|----|----|------|
| Attractor | A    | 0.89 | 0.96  | 0.96 | 0.98| 0.94| 0.37 |
| Beam      | A    | 0.89 | 0.96  | 0.96 | 0.98| 0.94| 0.37 |
| Bottle    | S    | 0.91 | 0.89  | 0.92 | 0.95| 0.94| 0.31 |
| Cor. pen  | A    | 0.84 | 0.92  | 0.92 | 0.96| 0.93| 0.29 |
| Pliers    | A    | 0.84 | 0.94  | 0.96 | 0.98| 0.94| 0.24 |
| Ring      | S    | 0.94 | 0.88  | 0.88 | 0.96| 0.92| 0.24 |
| Scissors  | S    | 0.88 | 0.91  | 0.93 | 0.96| 0.86| 0.12 |
| Screwdriver| A   | 0.80 | 0.82  | 0.89 | 0.94| 0.76| 0.21 |
| Sponge    | A    | 0.72 | 0.85  | 0.92 | 0.92| 0.92| 0.24 |
| Wrench    | A    | 0.72 | 0.85  | 0.92 | 0.92| 0.92| 0.24 |
| Wrench    | R    | 0.81 | 0.85  | 0.88 | 0.91| 0.89| 0.68|
| μ         | A    | 0.84 | 0.92  | 0.94 | 0.97| 0.95| 0.28 |
| σ         | A    | 0.86 | 0.94  | 0.92 | 0.95| 0.94| 0.28 |

Note. S = SGDM, A = ADAM, and R = RMSProp.
C. EVALUATION ROBOTIC GRASPING USING ACTION LEARNING

Now we focused on Table 5 above that action learning which is discussed with 14 experiments. Evaluating action learning performance means recording four processes simultaneously. The success of action learning is judged by the robot's success in carrying out the gripping task. It is difficult to determine the value position $\beta$ in the first-time system running; for this reason, we create dummy data with $\sigma = 70$. Initially, action learning does not limit the number of cycles.

Nevertheless, this experiment has to limit to only two cycles. This consideration is based on the safety factor of the robot because the possibility of changing positions is very high. All details of the limited experimental results are totally of 21 grasping experiments performed in the paper. The 14 experiments employ action learning as mentioned above and the remaining seven without using action learning. The results are listed in Table 6. The success rate results without and with action learning (cycle limit = 2) are 0.142 and 0.642, respectively. After we increase the cycle limit to 3, the results are 0.285 and 0.857, respectively.

Every process of grasping by the robot in action learning, whether it fails or succeeds, the data are always stored by the robot. Re-reading the data becomes an essential part of observing and reflecting in a single cycle. The value of $\beta_p$ is set to 60 and 70, respectively, while the value of the instrument's assessment result is $\beta$. The system will continue to do repetition to reach $\beta_p = \beta$.

An experiment of 14 grip attempts with five failures required 19 cycles in total. The values of $\beta$, $\beta_p$, and $\beta_0$ if described in PDF will look like Figure 17. The value of $\beta_0$ when first set at 70, then after the robot works its value, is very dependent on the value of $\beta$. The last ten data are accumulated to calculate the PDF.

### Table 5. The results of object position and orientation test.

| No. | Measured | Actual | Error | Position (mm) |
|-----|----------|--------|-------|---------------|
|     | $X_m$    | $Y_m$  | $Z_m$ | $X_p$         | $Y_p$  | $Z_p$  | $C_p$ |
| 1   | 264.45   | 169.75 | 281.10| 270.32        | 173.14 | 282.72 | -77.67 | 2.26  | 3.63 |
| 2   | 296.50   | 138.89 | 280.90| 48.93         | 308.01 | 276.52 | 50.89  | 1.96  | 4.23 |
| 3   | 306.39   | 207.62 | 280.80| 67.95         | 311.52 | 222.15 | 283.22 | 71.34 | 3.40 |
| 4   | 295.40   | 468.34 | 281.80| -43.07        | 300.85 | 482.39 | 290.25 | -45.22| 2.15 |
| 5   | 319.97   | 266.39 | 276.70| 31.22         | 312.77 | 279.71 | 296.07 | 32.16 | 0.94 |
| 6   | 329.02   | 409.56 | 298.50| -47.16        | 330.09 | 408.23 | 301.47 | -48.58| 1.41 |
| 7   | 361.51   | 228.70 | 281.10| 7.86          | 377.66 | 234.71 | 282.34 | 7.94  | 0.08 |
| 8   | 364.42   | 373.75 | 279.30| 56.03         | 368.07 | 384.97 | 284.89 | 58.83 | 2.80 |
| 9   | 347.93   | 384.98 | 286.90| 43.57         | 477.17 | 392.68 | 298.38 | 44.88 | 1.31 |
| 10  | 394.72   | 374.21 | 280.90| 70.57         | 402.35 | 386.66 | 282.14 | 74.10 | 3.53 |
| 11  | 407.93   | 312.46 | 280.80| 52.01         | 406.48 | 321.83 | 275.46 | 54.61 | 2.60 |
| 12  | 455.66   | 248.18 | 278.50| 50.26         | 453.89 | 270.52 | 282.43 | 52.77 | 2.51 |
| 13  | 466.96   | 106.74 | 279.00| -3.55         | 463.63 | 114.21 | 298.53 | -3.73 | 0.18 |
| 14  | 475.89   | 288.59 | 281.80| 42.44         | 480.16 | 291.48 | 279.71 | 44.14 | 1.70 |

$$
\begin{align*}
\mu &= 1.92 \\
\Sigma &= 1.06
\end{align*}
$$
TABLE 6. The gripping test evaluation.

| Target       | $\beta_0$ | $\beta_p$ | $\beta$ | Obs. | $\sum$ cycle | Grasp Result | T (s) |
|--------------|-----------|-----------|---------|------|---------------|--------------|-------|
| Beam (25 g)  | 70        | 60        | 92      | 0    | 1             | 1            | 175   |
| Ring (141 g) | 92        | 60        | 99      | 0    | 1             | 1            | 166   |
| Scr. driver (68 g) | 99    | 60        | 73      | 0    | 1             | 1            | 182   |
| Attractor (45 g) | 73      | 60        | 88      | 0    | 1             | 1            | 264   |
| Wrench (237 g) | 88        | 60        | 39      | 75   | 2             | 0            | 253   |
| Bottle (184 g) | 75        | 60        | 46      | 83   | 2             | 0            | 173   |
| Beam (25 g)  | 83        | 60        | 73      | 1    | 1             | 1            | 177   |
| Corr. Pen (23 g) | 73      | 70        | 85      | 76   | 2             | 1            | 251   |
| Sponge (3 g) | 76        | 70        | 94      | 0    | 1             | 1            | 182   |
| Ring (141 g) | 94        | 70        | 97      | 0    | 1             | 1            | 173   |
| Scr. driver (68 g) | 97        | 70        | 69      | 0    | 1             | 1            | 170   |
| Scissors (84 g) | 69        | 70        | 59      | 74   | 2             | 0            | 234   |
| Pliers (79 g) | 74        | 70        | 51      | 78   | 2             | 0            | 246   |
| Attractor (45 g) | 78        | 70        | 77      | 1    | 1             | 1            | 179   |

TABLE 7. The action learning evaluation.

| Mode          | Cycle | Number of Successes | Number of Failures | Success Rate |
|---------------|-------|---------------------|--------------------|--------------|
| Non-Action Learning | n/a   | 1                   | 6                  | 0.142        |
| Action Learning           | 2     | 9                   | 5                  | 0.642        |
| Non-Action Learning       | n/a   | 2                   | 5                  | 0.285        |
| Action Learning           | 3     | 12                  | 2                  | 0.857        |
| Random Planner [51]       | -     | -                   | -                  | 0.480        |
| Dex-Net 4.0 [51]          | -     | -                   | -                  | 0.490        |
| Tactile-Visual [51]       | -     | -                   | -                  | 0.800        |

FIGURE 17. The PDF of 14 grip attempts with 19 cycles using action learning.

VI. CONCLUSION

This study has successfully developed action learning for grasping objects by deep learning and used a standard manipulator robot with a stereo-like camera in an eye-in-hand configuration. A robotic manipulator equipped with a gripper can pick up and place targeted objects at cluttered positions in the workspace. A camera stereo-like is created by shifting the initial position to the second position on the x axis by a baseline of 100 mm. The process of grasping targets in action learning consists of four steps; planning, acting, observing, and reflecting—several prerequisites; DM, kNN, YOLOv3, and orientation. However, the results show around 0.857 successful grasping task with self-correction using action learning, while separately tested; an accuracy for the YOLOv3 of 0.923, and depth estimation around 0.341 mm. This evaluation process calculates with limited cycle in action learning within three cycle and environmental pass grade of 60 and 70.

REFERENCES

[1] S. S. Srinivasa et al., “Herb 2.0: Lessons Learned From Developing a Mobile Manipulator for the Home,” Proc. IEEE, vol. 100, no. 8, pp. 2410–2428, Aug. 2012, doi: 10.1109/JPROC.2012.2200561.
[2] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, “Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection,” arXiv:1603.02199 [cs], Aug. 2016, Accessed: Nov. 10, 2020. [Online]. Available: http://arxiv.org/abs/1603.02199
[3] Y. Zou and R. Lan, “An End-to-End Calibration Method for Welding Robot Laser Vision Systems With Deep Reinforcement Learning,” IEEE Trans. Instrum. Meas., vol. 69, no. 7, pp. 4270–4280, Jul. 2020, doi: 10.1109/TIM.2019.2942533.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
[5] “R-CNN vs Fast R-CNN vs Faster R-CNN,” [Online]. Available: https://www.geeksforgeeks.org/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-ml/.
[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
[7] H. Yang et al., “Tender Tea Shoots Recognition and Positioning for Picking Robot Using Improved YOLO-V3 Model,” IEEE Access, vol. 7, pp. 180998–181011, 2019, doi: 10.1109/ACCESS.2019.2958614.
[8] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, “Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection,” arXiv:1603.02199 [cs], Aug. 2016, Accessed: Nov. 10, 2020. [Online]. Available: http://arxiv.org/abs/1603.02199
[9] J. Weisz and P. K. Allen, “Pose error robust grasping from contact wrench space metrics,” in 2012 IEEE International Conference on Robotics and Automation, May 2012, pp. 557–562. doi: 10.1109/ICRA.2012.6224697.
[10] C. Brook and M. Pedler, “Action learning in academic management education: A state of the field review,” The
International Journal of Management Education, vol. 18, no. 3, p. 100415, Nov. 2020, doi: 10.1016/j.ijme.2020.100415.

[11] R. Van Gasse, K. Vanlommel, J. Vanhoof, and P. Van Petegem, “Teacher interactions in taking action upon pupil learning outcome data: A matter of attitude and self-efficacy?,” Teaching and Teacher Education, vol. 89, p. 102989, Mar. 2020, doi: 10.1016/j.tate.2019.102989.

[12] G. Costello, “Simulation-Action Learning (SAL),” 2020, pp. 111–129, doi: 10.1007/978-3-030-41380-4_7.

[13] H. Shi, L. Shi, M. Xu, and K.-S. Hwang, “End-to-End Navigation Strategy With Deep Reinforcement Learning for Mobile Robots,” IEEE Trans. Ind. Inf., vol. 16, no. 4, pp. 2393–2402, Apr. 2020, doi: 10.1109/TII.2019.2936167.

[14] C.-Y. Tsai, Y.-S. Chou, C.-C. Wong, Y.-C. Lai, and C.-C. Huang, “Visually Guided Picking Control of an Omnidirectional Mobile Manipulator Based on End-to-End Multi-Task Imitation Learning,” IEEE Access, vol. 8, pp. 1882–1891, 2020, doi: 10.1109/ACCESS.2019.2962335.

[15] S. Chen, M. Wang, W. Song, Y. Yang, Y. Li, and M. Fu, “Stabilization Approaches for Reinforcement Learning-Based End-to-End Autonomous Driving,” IEEE Trans. Veh. Technol., vol. 69, no. 5, pp. 4740–4750, May 2020, doi: 10.1109/TVT.2020.2979493.

[16] B. Riviere, W. Honig, Y. Yue, and S.-J. Chung, “GLAS: Global-to-Local Safe Autonomy Synthesis for Multi-Robot Motion Planning With End-to-End Learning,” IEEE Robot. Autom. Lett., vol. 5, no. 3, pp. 4249–4256, Jul. 2020, doi: 10.1109/LRA.2019.2994035.

[17] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, “Data-driven grasping with partial sensor data,” in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 2009, pp. 1278–1283, doi: 10.1109/IROS.2009.5354078.

[18] H.-T. L. Chiang, A. Faust, M. Fiser, and A. Francis, “Learning Navigation Behaviors End-to-End With AutoRL,” IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 2007–2014, Apr. 2019, doi: 10.1109/LRA.2019.2989918.

[19] O. Serrat, “Action Learning,” 2017, pp. 589–594, doi: 10.1007/978-981-10-0983-9_62.

[20] 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. on Image Process., vol. 27, no. 7, pp. 3586–3598, Jul. 2018, doi: 10.1109/TIP.2018.2814217.

[21] M. Hashem, M. L. Mohammed, and A. E. Yousef, “Improving the Efficiency of Dental Implantation Process Using Guided Local Search Models and Continuous Time Neural Networks With Robotic Assistance,” IEEE Access, vol. 8, pp. 202755–202764, 2020, doi: 10.1109/ACCESS.2020.3034689.

[22] K.-T. Song and S.-C. Tsai, “Vision-based adaptive grasping of a humanoid robot arm,” in 2012 IEEE International Conference on Automation and Logistics, Zhengzhou, China, Aug. 2012, pp. 155–160, doi: 10.1109/ICAL.2012.6308189.

[23] W. Fang, L. Wang, and P. Ren, “Timer-YOLO: A Real-Time Object Detection Method for Constrained Environments,” IEEE Access, vol. 8, pp. 1935–1944, 2020, doi: 10.1109/ACCESS.2019.2961939.

[24] Muslihkin, J.-R. Horng, S.-Y. Yang, and M.-S. Wang, “Object Localisation and Depth Estimation for Eye-in-Hand Manipulator Using Mono Camera,” IEEE Access, vol. 8, pp. 121765–121779, 2020, doi: 10.1109/ACCESS.2020.3006843.

[25] Taryudi and M.-S. Wang, “Eye to hand calibration using ANFIS for stereo vision-based object manipulation system,” Microsystem Technologies, vol. 24, no. 1, pp. 305–317, Jan. 2018, doi: 10.1007/s00542-017-3115-y.

[26] K. Feng and F. Yuan, “Static hand gesture recognition based on HOG characters and support vector machines,” in 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), Dec. 2013, pp. 936–938, doi: 10.1109/IMSNA.2013.6743432.

[27] W. Ji, X. Meng, Z. Qian, B. Xu, and D. Zhao, “Branch localization method based on the skeleton feature extraction and stereo matching for apple harvesting robot,” International Journal of Advanced Robotic Systems, vol. 14, May 2017, doi: 10.1177/1729981417705276.

[28] B. Dick, E. Stringer, and C. Huxham, “Theory in action research,” Action Research, vol. 7, no. 1, pp. 5–12, Mar. 2009, doi: 10.1177/1476750308099594.

[29] J. Whitehead and J. McNiff, Action research living theory. London; Thousand Oaks: SAGE Publications, 2006.

[30] H. Altichter, S. Kemmis, R. McTaggart, and O. Zuber-Skerritt, “The concept of action research,” The Learning Organization, vol. 9, no. 3, pp. 125–131, Aug. 2002, doi: 10.1177/096978740210428840.

[31] H. Yu, X. Yang, S. Zheng, and C. Sun, “Active Learning From Imbalanced Data: A Solution of Online Weighted Extreme Learning Machine,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 4, pp. 1088–1103, Apr. 2019, doi: 10.1109/TNNLS.2018.2855446.

[32] N. Rubens, D. Kaplan, and M. Sugiyama, “Active Learning in Recommender Systems,” p. 31.

[33] A. J. McMurray, “Teaching action research: The role of demographics,” Active Learning in Higher Education, vol. 7, no. 1, pp. 37–50, Mar. 2006, doi: 10.1177/1469787406061146.

[34] M. Lambrou, “The pedagogy of stylistics: Enhancing practice by flipping the classroom, using whiteboards and action research,” Language and Literature, vol. 29, no. 4, pp. 404–423, Nov. 2020, doi: 10.1177/0963947020968665.

[35] E. T. Stringer, L. M. Christensen, and S. C. Baldwin, Integrating teaching, learning, and action research: enhancing instruction in the K-12 classroom. Thousand Oaks, Calif: Sage, 2010.

[36] Chuming Liu, Xin Xu, and Dewen Hu, “Multiobjective Reinforcement Learning: A Comprehensive Overview,” IEEE Trans. Syst. Man Cybern., Syst., vol. 45, no. 3, pp. 385–398, Mar. 2015, doi: 10.1109/TSMC.2014.2358639.

[37] F. Radmehr and M. Drake, “Exploring Students’ Metacognitive Knowledge: The Case of Integral Calculus,” Education Sciences, vol. 10, no. 3, 2020, doi: 10.3390/educsci10030055.

[38] B. Dick, “Action research literature 2006—2008: Themes and trends,” Action Research, vol. 7, no. 4, pp. 423–441, Dec. 2009, doi: 10.1177/1476750309350701.

[39] L. M. Bell and J. M. Aldridge, Student Voice, Teacher Action Research and Classroom Improvement. Rotterdam: SensePublishers, 2014. doi: 10.1007/978-94-6209-776-6.

[40] L. Norton, “Object Localization and Depth Estimation for Eye-in-Hand Manipulator Using Mono Camera,” IEEE Access, vol. 8, pp. 121765–121779, 2020, doi: 10.1109/ACCESS.2020.3006843.
[42] V. Digani, L. Sabattini, and C. Secchi, “A Probabilistic Eulerian Traffic Model for the Coordination of Multiple AGVs in Automatic Warehouses,” IEEE Robot. Autom. Lett., vol. 1, no. 1, pp. 26–32, Jan. 2016, doi: 10.1109/LRA.2015.2505646.

[43] “Fuzzy interval linguistic sets with applications in multi-attribute group decision making,” Journal of Systems Engineering and Electronics, vol. 29, no. 6, p. 1237, 2018, doi: 10.21629/JSEE.2018.06.11.

[44] S. Kay, “Model-based probability density function estimation,” IEEE Signal Process. Lett., vol. 5, no. 12, pp. 318–320, Dec. 1998, doi: 10.1109/97.735424.

[45] M. Srikanth, H. K. Kesavan, and P. H. Roe, “Probability density function estimation using the MinMax measure,” IEEE Trans. Syst., Man, Cybern. C, vol. 30, no. 1, pp. 77–83, Feb. 2000, doi: 10.1109/9326.827456.

[46] V. Kontorovich, V. Lyandres, and S. Primak, “The generation of diffusion Markovian processes with probability density function defined on part of the real axis,” IEEE Signal Process. Lett., vol. 3, no. 1, pp. 19–21, Jan. 1996, doi: 10.1109/97.475826.

[47] L. Wang, F. Qian, and J. Liu, “Shape control on probability density function in stochastic systems,” J. of Syst. Eng. Electron., vol. 25, no. 1, pp. 144–149, Feb. 2014, doi: 10.1109/JSEE.2014.00017.

[48] 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, doi: 10.1109/ACCESS.2019.2942144.

[49] X. Wang, X. Liu, L. Chen, and H. Hu, “Deep-learning damped least squares method for inverse kinematics of redundant robots,” Measurement, p. 108821, Dec. 2020, doi: 10.1016/j.measurement.2020.108821.

[50] M. Muslikhin, J.-R. Horng, S.-Y. Yang, M.-S. Wang, and B.-A. Awaluddin, “An Artificial Intelligence of Things-Based Picking Algorithm for Online Shop in the Society 5.0’s Context,” Sensors, vol. 21, no. 8, 2021, doi: 10.3390/s21082813.

[51] Q. Feng, Z. Chen, J. Deng, C. Gao, J. Zhang, and A. Knoll, “Center-of-Mass-based Robust Grasp Planning for Unknown Objects Using Tactile-Visual Sensors,” in 2020 IEEE International Conference on Robotics and Automation (ICRA), Aug. 2020, pp. 610–617. doi: 10.1109/ICRA40945.2020.9196815.

MUSLIKHIN received the B.Ed. and M.Ed. degrees in electronic engineering education, and technology and vocational education from Universitas Negeri Yogyakarta, Yogyakarta, Indonesia in 2011 and 2013, respectively. His Ph.D. in electrical engineering obtained from Southern Taiwan University of Sciences and Technology in Tainan City, Taiwan in 2021. His research interests include robotics, machine vision, artificial intelligence, and deep learning. He also won 1st place in the 2020 TIRT International Innovative Robotics Festival in Taoyuan City, Taiwan. In this competition, the team has implemented an AIoT robotic integration concept to help quarantine COVID-19 patients.

JENQ-RUEY HORNG received the B.S. and M.S. degrees in Electrical Engineering from the 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.S. and M.S. degree in electrical engineering from Southern Taiwan University of Science and Technology, Tainan City, Taiwan in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree at Southern Taiwan University of Science and Technology. His current research interests include positioning and navigation AGV and UAV and internet of things.

MING-SHYAN WANG received the B.S. degree in Electronic Engineering from the National Chiao Tung University, M.S. and Ph.D. degrees in electrical engineering from the National Cheng Kung University, in 1981, 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.