

Received November 16, 2019, accepted December 15, 2019, date of publication December 25, 2019, date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962335

Visually Guided Picking Control of an Omnidirectional Mobile Manipulator Based on End-to-End Multi-Task Imitation Learning

CHI-YI TSAI¹, (Member, IEEE), YUNG-SHAN CHOU¹, CHING-CHANG WONG¹,
YU-CHENG LAI¹, AND CHIEN-CHE HUANG¹

Department of Electrical and Computer Engineering, Tamkang University, New Taipei City 251, Taiwan

Corresponding author: Chi-Yi Tsai (chiyi_tsai@mail.tku.edu.tw)

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 108-2218-E-032-004, Grant MOST 108-2221-E-032-045, Grant MOST 108-2221-E-032-046, and Grant MOST 108-2622-8-032-001.

ABSTRACT In this paper, a novel deep convolutional neural network (CNN) based high-level multi-task control architecture is proposed to address the visual guide-and-pick control problem of an omnidirectional mobile manipulator platform based on deep learning technology. The proposed mobile manipulator control system only uses a stereo camera as a sensing device to accomplish the visual guide-and-pick control task. After the stereo camera captures the stereo image of the scene, the proposed CNN-based high-level multi-task controller can directly predict the best motion guidance and picking action of the omnidirectional mobile manipulator by using the captured stereo image. In order to collect the training dataset, we manually controlled the mobile manipulator to navigate in an indoor environment for approaching and picking up an object-of-interest (OOI). In the meantime, we recorded all of the captured stereo images and the corresponding control commands of the robot during the manual teaching stage. In the training stage, we employed the end-to-end multi-task imitation learning technique to train the proposed CNN model by learning the desired motion and picking control strategies from prior expert demonstrations for visually guiding the mobile platform and then visually picking up the OOI. Experimental results show that the proposed visually guided picking control system achieves a picking success rate of about 78.2% on average.

INDEX TERMS Omnidirectional mobile manipulator, visually guided picking control, deep learning, multi-task imitation learning, end-to-end control.

I. INTRODUCTION

In recent years, research on visual servoing of robot manipulators receives more and more attention because such a control method provides a robust solution for many robotic automation applications, e.g., agricultural harvesting [1], [2], bin picking [3], [4], and object grasping [5], [6]. Among these robotic control applications, the function of robot grasping and navigation control plays an important role in a robot manipulator system to achieve autonomous manipulation tasks [7], [8], which can be applied in several industrial and service scenarios. In order for the robot to have such an important capability, many studies on the visual servoing of robot

manipulators have been carried out, and we divide them into three categories: model-based, feature-based, and data-driven approaches. Figure 1 illustrates the relationship between the model-based, feature-based, and data-driven visual servoing systems. The model-based methods [5]–[8] require analyzing the three-dimensional (3D) pose information of the OOI in the environment and use the pose information to determine an optimal motion trajectory and grip position of the end-effector. However, the model-based methods often cost much time on scene interpretation, task-level reasoning, and object 3D pose estimation.

The feature-based visual servoing system can be regarded as a low-level shortcut through the model-based control architecture [9]. The feature-based methods provide an efficient and effective solution to deal with environment perception

The associate editor coordinating the review of this manuscript and approving it for publication was Xinyu Du¹.

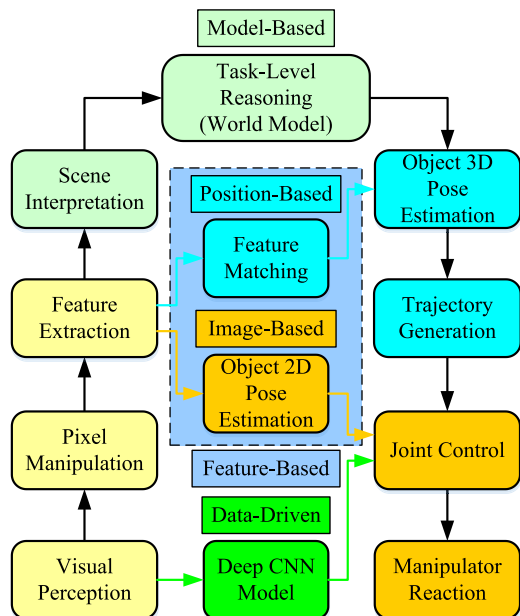


FIGURE 1. The relationship between model-based, feature-based and data-driven visual servoing approaches for robot manipulators. The feature-based approaches can further be divided into position-based and image-based approaches.

and can be divided into position-based and image-based methods. The position-based methods replace scene interpretation and task-level reasoning with feature matching [10] and object 3D pose estimation [11], [12]. On the other hand, the image-based methods skip scene interpretation, task level reasoning, 3D pose estimation, and trajectory generation tasks. However, they are still necessary to perform object 2D pose estimation to calculate the position and orientation of the OOI in the image plane.

Recently, data-driven control (DDC) and learning systems have become an emerging and rapidly growing topic [13]. Unlike the model-based and feature-based methods, the DDC method calculates the joint control commands from the visual perception data directly. This feature makes the robot can efficiently perform a visual servoing task more like human beings. Moreover, combining the DDC method with deep learning improves the visual servoing system to directly and accurately predict the grip action of the robot from the visual data through the deep CNN model. For instance, Lenz *et al.* proposed a deep learning approach to detect multiple robotic grasps for multiple objects contained in a single RGB-D view of a scene [14]. Watson *et al.* proposed a real-time robotic grasping method based on deep learning [15]. They trained a deep CNN model using supervised learning to fit the desired grasping positions from the captured RGB-D data. In [16], Levine *et al.* proposed a hand-eye coordination approach based on deep learning for the application of robotic grasping from monocular camera images. To learn the hand-eye coordination for grasping, they trained a CNN model to predict the probability of successful grasping under a given the gripper motion in the task-space using only monocular images.

Several deep learning-based end-to-end control methods also have been proposed to deal with the robotic grasp planning problem [17]–[19]. For example, Kumra *et al.* proposed a deep CNN-based robotic grasp detector, which predicts the best grasping pose of a parallel-plate robotic gripper using the RGB-D image of the scene [17]. Chu *et al.* [18] proposed a multi-grasp detector for multiple objects based on a deep learning architecture with RGB-D image input. Different from [17], the authors defined the learning problem to be classified with null hypothesis competition instead of regression. Recently, Zeng *et al.* presented a robotic pick-and-place system, which consists of a deep CNN-based multi-modal grasping framework and a CNN-based cross-domain image matching framework [19]. Both subsystems work hand-in-hand to handle a wide range of object categories without needing any task-specific training data for novel objects.

As to researches on the DDC-based robot navigation, Pfeiffer *et al.* proposed a DDC approach to make a robot to learn a navigation policy from the raw data of a 2D-laser scanner and the position of the desired target [20]. In this work, the authors proposed a CNN-based end-to-end architecture as a mapping function and trained it based on an expert demonstration to map the raw sensor data into the steering commands of the robot. Although this method can fit the expert demonstration efficiently, it still may overfit the desired actions during the end-to-end training process. To solve the overfitting problem encountered in end-to-end learning, Tai *et al.* used deep reinforcement learning (DRL) to train the robot by learning a navigation policy based on asynchronous deep deterministic policy gradient (ADDPG) algorithm [21], which extends the asynchronous off-policy Q-learning algorithm [22] using the deep deterministic policy gradient (DDPG) [23]. The DRL-based navigation method also takes the 2D-laser sensor information and the desired target position as the system inputs to predict the steering commands of the robot. The main advantage of the DRL-based navigation method is that it can provide a more robust navigation result when using lower-dimensional sensing information. However, there are two drawbacks to use the DRL algorithm. First, the model requires a fine-tuning process to update the network for each new desired target position. Second, the DRL needs a trial-and-error learning process, which is inefficient and not reliable during the model training process. To address these two issues, Zhu *et al.* proposed a DRL-based target-driven visual navigation method, which integrates an actor-critic model into an AI2-THOR framework to simulate a variety of agent actions and agent-object interactions [24]. This design allows users to collect a huge number of training samples efficiently. Recently, Pfeiffer *et al.* proposed a target-driven map-less navigation method based on reinforced imitation learning (RIL) [25], which leverages prior expert demonstrations to reduce sample complexity while avoiding distribution mismatching encountered in imitation learning and reinforcement learning. The RIL approach not only significantly improves the convergence rate of the DRL algorithm, but

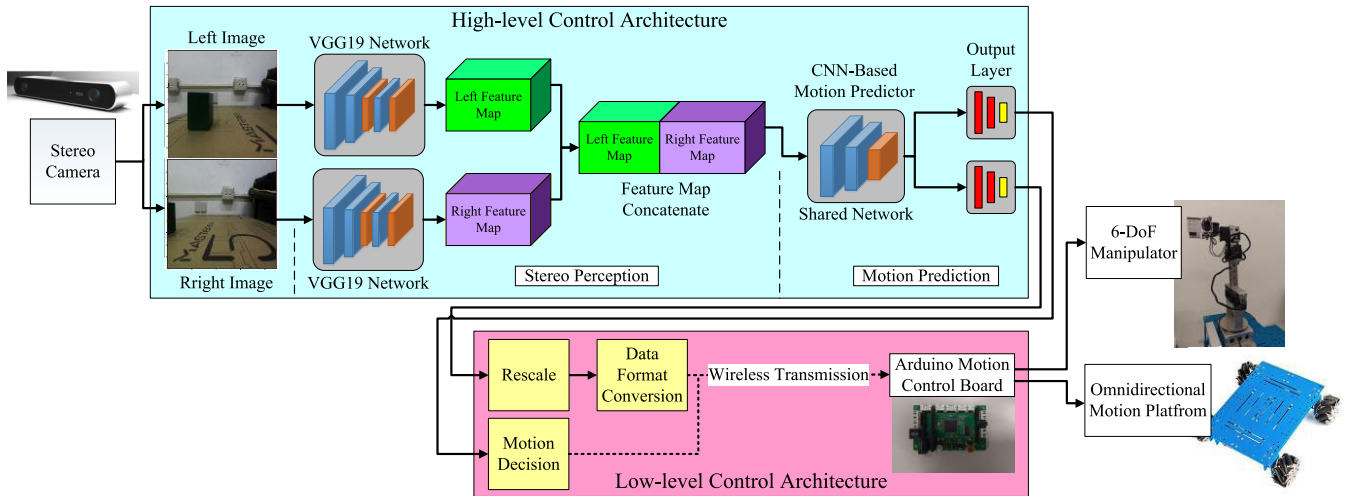


FIGURE 2. System architecture of the proposed data-driven visual guide-and-pick control system for a 6-DoF omnidirectional mobile manipulator based on the deep learning approach.

also generalizes the learned end-to-end navigation policy to unseen and real-world environments.

In this paper, we propose a novel data-driven visual guide-and-pick control method based on the end-to-end imitation learning technique [26]. The proposed DDC method is an extension of the authors' previous work [27], which presents a data-driven visual picking control design of a 6-DoF manipulator. This work extends the deep learning-based DDC method to control an omnidirectional mobile manipulator, which is a more challenging task than that of a stationary manipulator. The main contribution of this paper is twofold.

(1) We propose a new high-level multi-task control architecture based on CNN to learn the optimal guiding and picking actions of an omnidirectional mobile manipulator from stereo observations of the scene. By doing so, several computationally intensive processes can be omitted, such as depth map estimation, point cloud registration, object pose estimation, etc.

(2) We also propose a new CNN-based multi-task neural network model to learn a guide-and-pick control policy from prior expert demonstrations through end-to-end imitation learning. By combining the DDC method with a deep CNN multi-task model, the proposed control system makes the omnidirectional mobile manipulator able to predict the action for approaching the target and then picking up the object from the visual sensing data directly without the knowledge of the kinematic model of the robot. As far as we know, there is no existing paper that proposes such a modelless design.

From the experimental results, the average picking success rate of the proposed deep learning-based data-driven visual guide-and-pick control method reaches about 78.2% in the case of performing a random single-object picking task starting from at least 50 cm away from the object.

The remainder of this paper is organized as follows. Section II describes the system architecture of the proposed

data-driven visual guide-and-pick control system. Section III presents the design of the proposed CNN-based stereo perception network that provides the stereo feature map of the scene required in the following guide-and-pick control process. The proposed CNN-based guide-and-pick prediction network is introduced in Section IV. Section V reports several experimental results to validate the performance of the proposed data-driven visual guide-and-pick grasping control system. Finally, Section VI concludes the contributions of this work.

II. SYSTEM ARCHITECTURE

Figure 2 presents the system architecture of the proposed data-driven visual guide-and-pick control system, which is a CNN-based high-level multi-task controller consisting of stereo perception and motion prediction modules. The stereo perception module receives the captured stereo image to produce stereo feature maps. Most stereo vision systems include a depth estimation process to generate a depth map of the scene for related applications, such as 3D reconstruction, obstacle avoidance, or grasp planning, etc. In this work, we utilize the existing VGG19 network [28] as the backbone model to calculate the feature map of the captured stereo image. Moreover, we employ two VGG19 networks to extract the feature map of the left and right images, respectively. Next, a concatenating operation is applied on the left and right feature maps to generate a stereo feature map. The proposed CNN-based stereo perception module is described in detail in Section III.

On the other hand, the motion prediction module is a CNN-based multi-task network to predict the motion direction of the mobile platform and the joint angles of the 6-DoF manipulator for approaching and picking up the OOI placed in the working space based on the stereo feature map. In our implementation, an Arduino motion control card was used as a low-level controller to convert both platform motion and

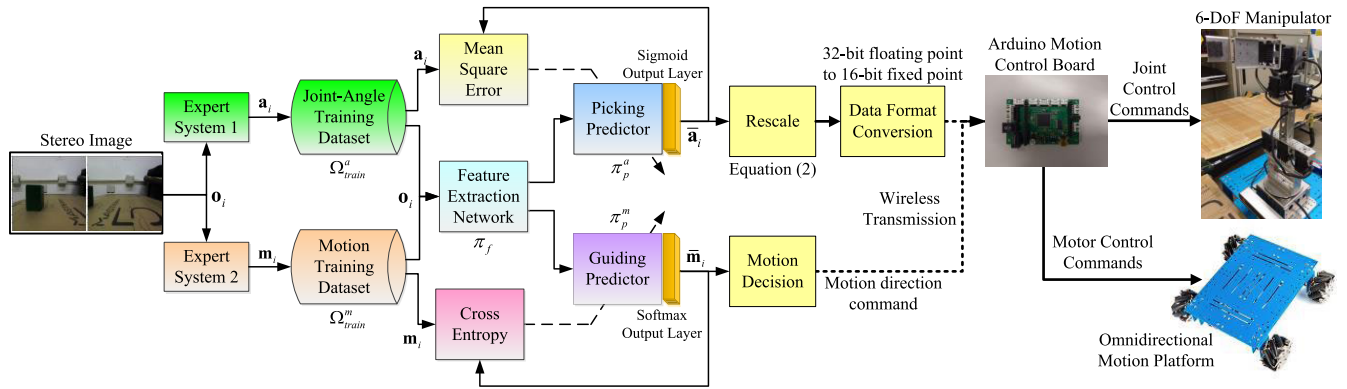


FIGURE 3. The multi-task imitation learning method used in this work to train the proposed CNN-based guide-and-pick predictor. An Arduino motion control board is used as a low-level controller to control the omnidirectional mobile manipulator.

joint angle commands into low-level control signals of the omnidirectional mobile manipulator.

Figure 3 shows the multi-task imitation learning method used for training the proposed CNN-based guide-and-pick predictor and the low-level control architecture of the proposed control system. There are two different training datasets collected from two different expert systems inputting the same observation. As shown in Figure 3, Expert System 1 and Expert System 2 provide the desired joint angles of the manipulator and the desired motion direction of the mobile platform, respectively. To train the proposed CNN-based motion predictor, we first have to select an appropriate feature extraction network as the stereo perception model. Next, we treat the training of the visual guiding and the visual picking model as a regression problem and a classification problem, respectively. We then apply the imitation learning technique to train both models. Section IV presents the details of the proposed CNN-based guide-and-pick predictor.

On the other hand, we use an Arduino motion control board to implement the low-level controller of the proposed control system for translating the high-level action predictions into the low-level control commands. Note that the data formats of the high-level action predictions and the low-level control commands are 32-bit floating-point and 16-bit fixed-point, respectively. Thus, we need a data format conversion to convert the data format of the rescaled joint angles obtained from the sigmoid output layer of the picking predictor from the 32-bit floating-point to the 16-bit fixed-point for wireless transmission over Bluetooth. Furthermore, the output from the softmax output layer of the guiding predictor is encoded into a motion direction command, which is a binary one-hot code to indicate the best action predicted by the guiding predictor. When the Arduino motion control board received the fixed-point command data, it sends the joint control commands and the motor control commands to the robot manipulator and the omnidirectional motion platform, respectively.

III. CNN-BASED STEREO PERCEPTION NETWORK

Extracting useful feature maps from the input stereo image is an essential task in the proposed CNN-based controller.

TABLE 1. Characteristics of the ResNet50, VGG16, and VGG19 models, and their MSE results recorded in the testing phase.

Model	Memory Size	Accuracy [30]		Training Time (s/epoch)	MSE Loss of Testing
		Top-1	Top-5		
ResNet50	99MB	0.715	0.901	80	20e-4
VGG16	528MB	0.727	0.910	32	13e-4
VGG19	549MB	0.759	0.929	32	4e-4

In order to find the best choice for the proposed CNN-based controller, we tested three commonly used CNN backbone models: ResNet50 [28], VGG16, and VGG19 [29]. Table 1 lists the characteristics of the three CNN backbone models validated on the ImageNet validation set [30].

To collect training and testing datasets, we manually controlled the mobile manipulator using a remote controller to approach and pick up an OOI placed in the workspace. At the same time, we also recorded all stereo observations and the corresponding control commands to form a training dataset and a testing dataset. Let \mathbf{o}_i denote the i -th observed stereo image and $\mathbf{a}_i = [a_0 \ a_1 \ \dots \ a_6]^T$ the i -th joint angle command vector, in which a_0 represents the normalized angle of the gripper and a_j for $j = 1 \sim 6$ indicate the six manipulator joints. Because the motor commands have a wide range of values, the value of each joint angle command is normalized to the range $[0, 1]$ as follows:

$$a_j = \frac{A_j - A_{\min}}{A_{\max} - A_{\min}} \quad \text{for } j = 0 \sim 6, \quad (1)$$

where A_{\min} and A_{\max} denote the minimum and maximum joint angle, respectively. These two parameters depend on the joint limits of the hardware system. A_j is the j -th actual joint angle command, which can be obtained from the following rescale operation

$$A_j = (A_{\max} - A_{\min})a_j + A_{\min}, \quad (2)$$

where a_j is the j -th joint angle output of the proposed CNN-picking predictor.

Let $\Omega_{train}^a = \{(\mathbf{o}_i, \mathbf{a}_i)\}_{i=1 \sim N_{train}^a}$ denote a joint angle training dataset and $\Omega_{test}^a = \{(\mathbf{o}_k, \mathbf{a}_k)\}_{k=1 \sim N_{test}^a}$ a joint angle testing dataset. In the data collection phase, we totally collected

304 training samples ($N_{train}^a = 304$) and 76 testing samples ($N_{test}^a = 76$). In the training phase, we randomly divided the training dataset Ω_{train}^a into several batches to train the proposed CNN-based picking predictor. Next, we used the mean squared error (MSE) as the loss function to train the CNN model of the picking predictor such that

$$L_{MSE}(\Omega_{batch}^a, \pi_p^a) \Big|_{\pi_f} = \frac{1}{N_{batch}^a} \sum_{i=1}^{N_{batch}^a} \left\| \mathbf{a}_i - \pi_p^a(\pi_f(\mathbf{o}_i)) \right\|^2, \quad (3)$$

where $\Omega_{batch}^a = \{(\mathbf{o}_i, \mathbf{a}_i)\}_{i=1 \sim N_{batch}^a} \subset \Omega_{train}^a$ is a batch of the joint angle training dataset. $N_{batch}^a \ll N_{train}^a$ denotes the batch size used in training. $\pi_f(\bullet)$ and $\pi_p^a(\bullet)$ represent the CNN model of the stereo perception module and the proposed picking predictor, respectively. For a given CNN model of the stereo perception module π_f , our goal is to optimize the picking predictor model π_p^a without updating the given model π_f for each batch such that

$$\hat{\pi}_p^a = \arg \min_{\pi_p^a} L_{MSE}(\Omega_{batch}^a, \pi_p^a) \Big|_{\pi_f}, \quad (4)$$

where $\pi_f = \{\pi_{VGG16}, \pi_{VGG19}, \pi_{ResNet50}\}$ is one of the tested CNN models.

In the testing phase, the testing dataset Ω_{test}^a was applied to the optimal picking predictor model $\hat{\pi}_p^a$ for computing the MSE loss of testing $L_{MSE}(\Omega_{test}^a, \hat{\pi}_p^a) \Big|_{\pi_f}$ with respect to each tested CNN model. Table 1 also records the MSE results of the testing, and we have some observations from the testing results.

(1) Although the ResNet50 model validated by the ImageNet dataset is more accurate than the other two models, it requires the most computational time during the training phase and the worst MSE result during the testing phase.

(2) The VGG16 and VGG19 models have the same computational time during the training phase, but the VGG19 model provides the best MSE result during the testing phase.

(3) Based on the testing results, we conclude that the VGG19 model is the best backbone model to be used as a feature map extractor for the proposed CNN-based high-level multi-task controller. Thus, we define $\pi_f \equiv \pi_{VGG19}$ as the feature extraction network in the proposed control system.

IV. CNN-BASED GUIDE-AND-PICK PREDICTION NETWORK

The proposed CNN-based high-level multi-task controller is a visual data-driven guide-and-pick predictor that uses the stereo feature map as the input to predict the best motion response to the current stereo observation. To train the CNN model, we adopted an end-to-end imitation learning technique [26], which aims to learn a new strategy model from a set of examples provided by demonstrators or human experts. Each example contains an observation and a corresponding demonstration action. By imitation learning, the machine can imitate the behavior of the expert strategy recorded in

the demonstration. In the training phase, the observations are used as input features, and the demonstration actions are used as desired targets to learn the best strategy model that is designed to match the observation-to-action policy generated by the model with the expert strategy.

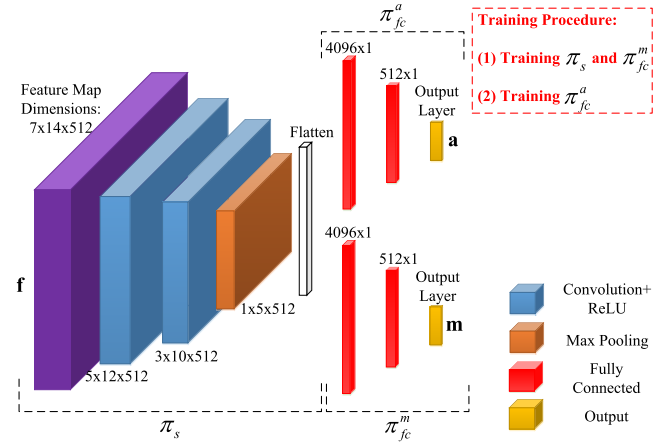


FIGURE 4. The network architecture of the proposed CNN-based guide-and-pick predictor.

Figure 4 shows the network architecture of the proposed CNN-based guide-and-pick predictor, which consists of a shared CNN model $\pi_s(\bullet)$ and two fully connected (FC) models $\pi_{fc}^a(\bullet)$ and $\pi_{fc}^m(\bullet)$. The shared CNN model consists of one max pooling layer and two CNN layers for adjusting the feature map obtained from the VGG19 model. Each of the following two FC models contains three FC layers to fit the modified feature map to the desired action. Let \mathbf{f} denote the stereo feature map obtained from the stereo perception module. In this study, the strategy model to be learned includes a guiding predictor model $\pi_p^m(\mathbf{f}) = \pi_{fc}^m(\pi_s(\mathbf{f}))$ and a picking predictor model $\pi_p^a(\mathbf{f}) = \pi_{fc}^a(\pi_s(\mathbf{f}))$. In the following, we present the training procedure of both predictor models based on the end-to-end imitation learning technique.

A. TRAINING OF THE GUIDING PREDICTOR MODEL

Figure 5 illustrates the mechanical structure of the four-Mecanum-wheeled omnidirectional motion platform used in this study. We apply a deep learning-based DDC method to control the motion of the platform in six motion directions, which are defined by four motor commands. As mentioned earlier, the proposed CNN-based guiding predictor model is constructed by the shared CNN model and three FC layers to fit the stereo feature map \mathbf{f} to the desired motion direction. Here, the final output layer is expected to be a softmax classifier [31] for the proposed guiding predictor model in order to predict the best motion direction corresponding to the current observation.

Let $\mathbf{m}_i = [b_{i1} \ b_{i2} \ \dots \ b_{iM}]^T$ denote the i -th motion direction command vector, which is a one-hot code to indicate one of the defined motion directions. In the data collection phase, we collected a platform motion training

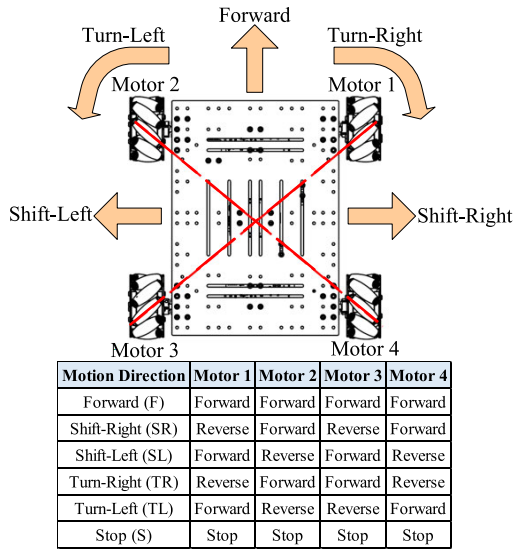


FIGURE 5. Mechanical structure and the corresponding motor actions of the four-Mecanum-wheeled omnidirectional motion platform [33].

dataset $\Omega_{train}^m = \{(\mathbf{o}_i, \mathbf{m}_i)\}_{i=1 \sim N_{train}^m}$ with a total sample number $N_{train}^m = 1264$ and a platform motion testing dataset $\Omega_{test}^m = \{(\mathbf{o}_k, \mathbf{m}_k)\}_{k=1 \sim N_{test}^m}$ with $N_{test}^m = 316$. In the training phase, we also randomly divided the training dataset Ω_{train}^m into several batches to train the proposed CNN-based guiding predictor. Next, we employed the cross-entropy (CE) as the loss function to train the CNN model of the guiding predictor such that

$$L_{CE}(\Omega_{batch}^m, \pi_{fc}^m(\pi_s)) \Big|_{\pi_{VGG19}} = - \sum_{i=1}^{N_{batch}^m} \sum_{j=1}^M b_{ij} \ln p(m_j | \mathbf{x}_i), \quad (5)$$

$$p(m_j | \mathbf{x}_i) = \frac{e^{\mathbf{x}_i^T \mathbf{w}_j}}{\sum_{k=1}^M e^{\mathbf{x}_i^T \mathbf{w}_k}}, \quad (6)$$

where b_{ij} denotes the j -th bit of the i -th motion direction command vector, and $\mathbf{x}_i = \pi_{fc}^m(\pi_s(\mathbf{f}_i)) \Big|_{\pi_{VGG19}}$ is the i -th feature vector associated with the i -th stereo feature map \mathbf{f}_i obtained from the i -th observation \mathbf{o}_i through the VGG19 network. $\pi_{fc}^m(\bullet)$ denotes the mapping function formed by the last FC layer of the proposed CNN-based guiding predictor model, and \mathbf{w}_j is the weight vector of the j -th output at the last FC layer. $\Omega_{batch}^m = \{(\mathbf{o}_i, \mathbf{m}_i)\}_{i=1 \sim N_{batch}^m} \subset \Omega_{train}^m$ is a batch of the platform motion training dataset. $N_{batch}^m \ll N_{train}^m$ denotes the batch size. At this stage, we optimized both the guiding FC model $\pi_{fc}^m(\bullet)$ and the shared CNN model such that

$$\hat{\pi}_{fc}^m, \hat{\pi}_s = \arg \min_{\pi_{fc}^m, \pi_s} L_{CE}(\Omega_{batch}^m, \pi_{fc}^m(\pi_s)) \Big|_{\pi_{VGG19}}. \quad (7)$$

Finally, the optimal guiding predictor model is given by $\hat{\pi}_p^m(\mathbf{f}) = \hat{\pi}_{fc}^m(\hat{\pi}_s(\mathbf{f}))$.

In the testing phase, we evaluated the performance of the optimized guiding predictor model based on the CE loss

function $L_{CE}(\Omega_{test}^m, \hat{\pi}_{fc}^m(\hat{\pi}_s)) \Big|_{\pi_{VGG19}}$ calculated from the testing dataset, and the testing accuracy was about 86.5%. Since the output value of the proposed CNN-based guiding predictor is also between 0 and 1, we need to perform a motion decision process to quantize the output value of the predictor into a binarized one-hot code, in which the single-bit “1” indicates the best motion direction corresponding to the current input observation.

Remark 1: As shown in Figure 5, the desired direction of motion includes a stop behavior for the robot to stop the guiding predictor when it is close enough to the target based on the current stereo perception. This behavior was also trained during the offline training stage. After the stop behavior occurs, the robot automatically switches to run the picking predictor model to perform the picking task.

B. TRAINING OF THE PICKING PREDICTOR MODEL

The proposed picking predictor model is also constructed by the same shared CNN model, as shown in Figure 4. Similar to the guiding predictor model, the picking predictor model also uses three FC layers to learn the mapping between the stereo feature map and the desired joint angles. However, we use the sigmoid classifier [32] as the final output layer of the proposed picking predictor model. Based on this design, the output range of the proposed predictive predictor is guaranteed to be 0 to 1, satisfying the condition of output normalization (1). Therefore, we use the sigmoid function $S(x) = (1 + e^{-x})^{-1}$ as the output layer activation function of the proposed picking predictor model. In the training phase, we employed the MSE loss function defined in (3) to train the CNN model of the picking predictor. However, at this stage, we only optimized the picking FC model $\pi_{fc}^a(\bullet)$ and kept the shared CNN model fixed during the training process. In other words, given the VGG19 feature extraction model, the MSE loss function used in the training phase of the picking predictor becomes $L_{MSE}(\Omega_{batch}^a, \pi_{fc}^a(\hat{\pi}_s)) \Big|_{\pi_{VGG19}}$.

In the testing phase, the performance of the optimized picking predictor model $\hat{\pi}_p^a(\mathbf{f}) = \hat{\pi}_{fc}^a(\hat{\pi}_s(\mathbf{f}))$ was evaluated on the testing dataset using the MSE loss function $L_{MSE}(\Omega_{test}^a, \hat{\pi}_{fc}^a(\hat{\pi}_s)) \Big|_{\pi_{VGG19}}$, and the testing error was about 2.875 degrees on average. Note that the proposed CNN-based picking predictor has an output range of 0 to 1.

Therefore, we must use equation (2) to rescale the output value of the predictor to the actual joint angle value.

V. EXPERIMENTAL RESULTS

We implemented the proposed CNN-based guide-and-pick control system using Tensorflow 1.5.0 running on a laptop equipped with 2.4GHz Intel Core i7-5500U, 8GB system memory, and Ubuntu 16.04 operating system. Figure 6 shows the stereo camera and the laboratory-made 6-DoF omnidirectional mobile manipulator used in the experiment. We placed the ZED stereo camera in front of the mobile manipulator for the robot to capture the scene stereo image in front of the platform easily. In the experiment, we selected the

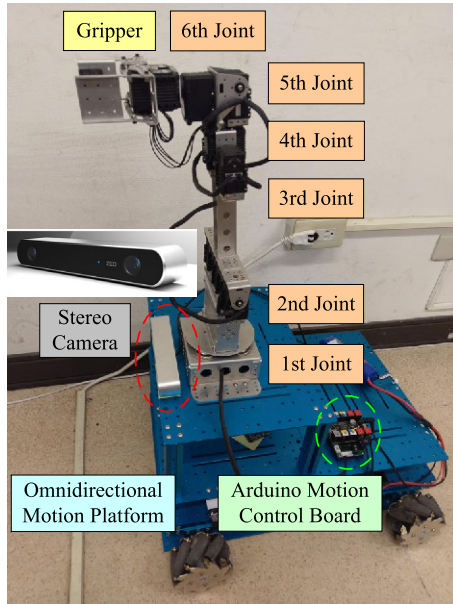


FIGURE 6. The stereo camera and the laboratory-made 6-DoF omnidirectional mobile manipulator used in the experiment.

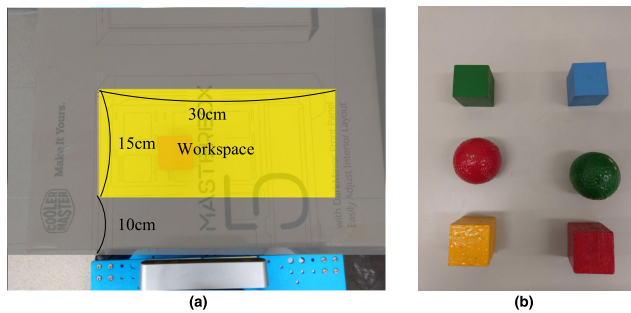


FIGURE 7. Experimental settings of the visual guide-and-pick control task: (a) the workspace of the mobile manipulator; (b) the objects used in the experiments.

video mode of the ZED camera as 720p, which provides enough image resolution for the proposed CNN-based guide-and-pick control system. Figure 7 presents the experimental settings of the visual guide-and-pick control task to validate the performance of the proposed control method. Due to the stereo camera and hardware limitations, the yellow region in Figure 7(a) illustrates the workspace of the manipulator. The six objects presented in Figure 7(b) were used in the visual guide-and-pick control experiment. We selected at least one object and randomly placed the object in the workspace of the manipulator to test the performance of the proposed visual guide-and-pick control system.

In the visual guide-and-pick control experiment, we totally performed 55 tests at three different initial distances, of which 15 times were initially at 50 cm away to the OOI, 15 times were at 100 cm, and 25 times were at 150 cm. Table 2 records the picking success rate of the proposed CNN-based data-driven visual servoing system tested in the guide-and-pick experiment. It is clear from Table 2 that the shorter the initial

TABLE 2. Picking success rate of the proposed control system tested in the visual guide-and-pick control task.

Initial Distance to the Object	Number of Successes	Number of Failures	Success Rate
50 cm	13	2	86.7%
100 cm	12	3	80.0%
150 cm	18	7	72.0%
Total Number	43	12	78.2%

distance to the object, the higher the picking success rate of the robot guide-and-pick control. The maximum picking success rate achieves 86.7% in the case of 50 cm initial distance to the object. As the initial distance increased from 50 cm to 150 cm, the picking success rate is decreased from 86.7% to 72.0%. Therefore, the average picking success rate of the proposed visual guide-and-pick controller is about 78.2% in 55 visual guide-and-pick control experiments. Note that most failure cases are that the target is not within the field of view (FoV) of the stereo camera due to a large motion variation caused by the wheel drift of the omnidirectional motion platform.

Figure 8 shows one experimental result of the proposed control system tested in the case of 50 cm initial distance. In the beginning, the OOI is placed at the right-hand side of the robot, as shown in the initial stereo observation. Thus, the first action of the mobile platform corresponding to the initial stereo observation is Turn-Right to make the robot facing to the OOI, as shown in the second stereo observation. Next, the robot attempts approaching the OOI using three Forward actions based on the 2nd, 3rd, and 4th stereo observations. When the robot is close enough to the OOI, the controller stops the action of the mobile platform and switches to control the 6-DoF manipulator to perform the visual picking task. Finally, the proposed CNN-based picking predictor calculates the required joint angles from the last stereo observation, and then the robot successfully picks up the OOI, as shown in the object picking result of Figure 8.

Figure 9 shows the experimental result of the proposed control system tested at 100 cm initial distance. In Figure 9, the OOI is initially placed at the left-hand side of the robot. Thus, in the beginning, the robot performs the Turn-Left action one time to correct the orientation angle between the robot and the OOI. When the orientation of the robot is corrected, the robot performs four Forward actions to approach the OOI based on the 2nd, 3rd, 4th, and 5th stereo observations. Finally, the proposed visual guide-and-pick controller stops the mobile platform and switches to control the 6-DoF manipulator to pick up the OOI based on the last stereo observation. Two video clips of the experimental result can refer to the online webpages of [34] and [35].

Remark 2: To drive the motion platform, we applied the motor commands for a fixed runtime of 3 seconds and a fixed speed of 12.5 cm/s. In this control mode, the Mecanum-wheeled motion platform may have a fairly strong-motion variation caused by wheel drift. However, the experimental




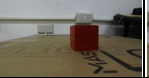








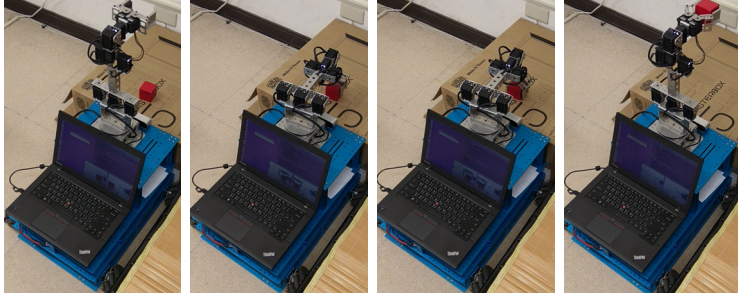
Visual Guiding Control of the Omnidirectional Motion Platform							
Stereo Observation	Left Image						
	Right Image						
Motion Direction		TR	F	F	F	F	S
Visual Picking Control of the 6-DoF Manipulator							
Predicted Joint Angles (in degrees)	a_0	a_1	a_2	a_3	a_4	a_5	a_6
	153.19	72.45	148.69	168.56	73.31	167.89	127.74
Object Picking Result							

FIGURE 8. Experimental result of the proposed visual guide-and-pick control system at 50 cm initial distance to the OOI.





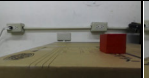







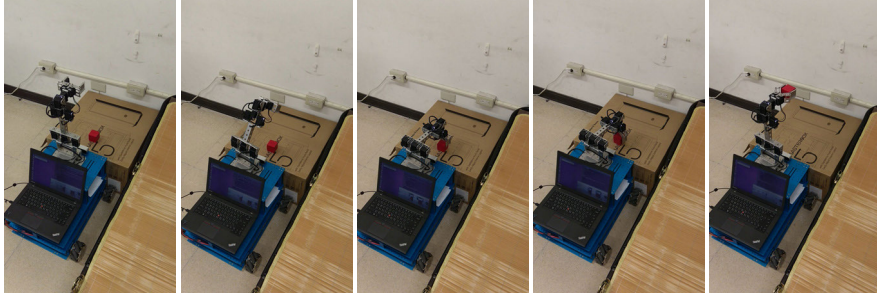
Visual Guiding Control of the Omnidirectional Motion Platform							
Stereo Observation	Left Image						
	Right Image						
Motion Direction		TL	F	F	F	F	S
Visual Picking Control of the 6-DoF Manipulator							
Predicted Joint Angles (in degrees)	a_0	a_1	a_2	a_3	a_4	a_5	a_6
	170.25	78.53	153.33	164.06	76.31	164.74	129.96
Object Picking Result							

FIGURE 9. Experimental result of the proposed visual guide-and-pick control system at 100 cm initial distance to the OOI.

results show that as long as the target remains within the FoV of the stereo camera, the proposed visual guide-and-pick controller can still overcome such a strong-motion variation to complete the guide-and-pick task.

Remark 3: As mentioned at the beginning of Section V, the proposed visual guide-and-pick control system was implemented on a laptop to test its performance. Thus, in the experiment, we did not have a GPU equipment to accelerate

the calculation of the deep CNN model. Since all high-level computations are performed only on the CPU, the processing speed of the proposed control system at this stage cannot achieve real-time performance. To constantly evaluate the stereo image and update the commands in real-time, the high-level control system requires a high-end laptop equipped with a high computing power GPU device to handle the massive calculation of the deep CNN model.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel CNN-based visual guide-and-pick control method, which helps to realize autonomous guide-and-pick control of an omnidirectional mobile manipulator using only the stereo visual sensing data. The proposed visual servoing system combines a CNN-based high-level multi-task control architecture with an Arduino-based low-level controller to implement a modelless visual data-driven guide-and-pick controller. The proposed high-level control architecture consists of two modules: one is a CNN-based stereo perception network, and the other one is a CNN-based guide-and-pick prediction network. The proposed stereo perception network employs the existing VGG19 model to extract the stereo feature map from the stereo visual sensing data directly without computationally stereo image processing, such as depth map estimation, 3D reconstruction, object pose estimation, etc. Moreover, we design a new CNN-based guide-and-pick prediction network to work with the stereo perception network and apply the end-to-end multi-task imitation learning method to train the proposed guide-and-pick prediction model. The experimental results show that the proposed CNN-based visual guide-and-pick control system can not only be used with the VGG19 model, but also can successfully control the omnidirectional mobile manipulator to approach and pick up the OOI placed in the robot workspace. Moreover, the picking success rate of the proposed vision-based DDC system is about 78.2% in 55 visual guide-and-pick control experiments.

In the future, we will extend the proposed CNN-based high-level multi-task controller to other mobile manipulator systems. In addition, the capabilities of the proposed control system applicable to other scenarios and other types of objects will also be studied to assess its general value in other situations.

REFERENCES

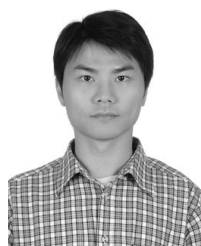
- [1] B. Joffe, K. Ahlin, A.-P. Hu, and G. McMurray, "Vision-guided robotic leaf picking," *EasyChair Preprints*, no. 250, pp. 1–6, Jun. 2018.
- [2] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "A review of key techniques of vision-based control for harvesting robot," *Comput. Electron. Agricult.*, vol. 127, pp. 311–323, Sep. 2016.
- [3] K. Kim, J. Kim, S. Kang, J. Kim, and J. Lee, "Vision-based bin picking system for industrial robotics applications," in *Proc. 9th Int. Conf. Ubiquitous Robots Ambient Intell.*, Daejeon, South Korea, Nov. 2013, pp. 515–516.
- [4] J.-K. Oh, S. Lee, and C.-H. Lee, "Stereo vision based automation for a bin-picking solution," *Int. J. Control Automat. Syst.*, vol. 10, no. 2, pp. 362–373, Apr. 2012.
- [5] V. Andaluz, R. Carelli, L. Salinas, J. M. Toibero, and F. Roberti, "Passivity-based visual feedback control with dynamic compensation of mobile manipulators: Stability and L2-gain performance analysis," *Robot. Auton. Syst.*, vol. 66, pp. 64–74, 2015.
- [6] Y. Wang, G.-L. Zhang, H. Lang, B. Zuo, and C. W. De Silva, "A modified image-based visual servo controller with hybrid camera configuration for robust robotic grasping," *Robot. Auton. Syst.*, vol. 62, no. 10, pp. 1398–1407, Oct. 2014.
- [7] R. Pieters, Z. Ye, P. Jonker, and H. Nijmeijer, "Direct motion planning for vision-based control," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1282–1288, Oct. 2014.
- [8] V. Andaluz, R. Carelli, L. Salinas, J. M. Toibero, and F. Roberti, "Visual control with adaptive dynamical compensation for 3D target tracking by mobile manipulators," *Mechatronics*, vol. 22, no. 4, pp. 491–502, Jun. 2012.
- [9] P. I. Corke, *Visual Control of Robots: High-Performance Visual Servoing*. New York, NY, USA: Wiley, 1997.
- [10] C.-Y. Tsai, A.-H. Tsao, and C.-H. Huang, "Graphics processing unit-accelerated multi-resolution exhaustive search algorithm for real-time keypoint descriptor matching in high-dimensional spaces," *IET Comput. Vis.*, vol. 10, no. 3, pp. 212–219, Apr. 2016.
- [11] C.-Y. Tsai and S.-H. Tsai, "Simultaneous 3D object recognition and pose estimation based on RGB-D images," *IEEE Access*, vol. 6, pp. 28859–28869, 2018.
- [12] 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.
- [13] Z. Hou, H. Gao, and F. Lewis, "Data-driven control and learning systems," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4070–4075, May 2017.
- [14] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 705–724, Apr. 2015.
- [15] J. Watson, J. Hughes, and F. Iida, "Real-world, real-time robotic grasping with convolutional neural networks," in *Towards Autonomous Robotic Systems*. Cham, Switzerland: Springer, 2017, pp. 617–626.
- [16] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 421–436, Apr. 2018.
- [17] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, BC, Canada, Sep. 2017, pp. 769–776.
- [18] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3355–3362, Oct. 2018.
- [19] A. Zeng, S. Song, and K.-T. Yu, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *Proc. IEEE Int. Conf. Robot. Automat.*, Brisbane, QLD, Australia, Aug. 2018, pp. 3750–3757.
- [20] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, May 2017, pp. 1527–1533.
- [21] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, BC, Canada, Sep. 2017, pp. 31–36.
- [22] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, May 2017, pp. 3389–3396.
- [23] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*. [Online]. Available: <https://arxiv.org/abs/1509.02971>
- [24] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 3357–3364.
- [25] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4423–4430, Oct. 2018.
- [26] A. Attia and S. Dayan, "Global overview of imitation learning," 2018, *arXiv:1801.06503*. [Online]. Available: <https://arxiv.org/abs/1801.06503>
- [27] C.-Y. Tsai, C.-C. Huang, and Y.-S. Chou, "Data-driven visual picking control of a 6-DoF manipulator using end-to-end imitation learning," in *Proc. CACS Int. Autom. Control Conf.*, Taoyuan, Taiwan, Nov. 2018.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
- [29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [31] X. Qi, T. Wang, and J. Liu, "Comparison of support vector machine and softmax classifiers in computer vision," in *Proc. 2nd Int. Conf. Mech., Control Comput. Eng.*, Harbin, China, 2017, pp. 151–155.

- [32] S. Banerjee, S. S. Chaudhuri, and S. Roy, "Fuzzy logic based vision enhancement using sigmoid function," in *Proc. IEEE Calcutta Conf.*, Kolkata, India, Dec. 2017, pp. 41–45.
- [33] Makeblock. *Mecanum-Wheel-Robot-Kit*. Accessed: Sep. 2, 2019. [Online]. Available: <https://www.makeblock.com/project/mecanum-wheel-robot-kit>
- [34] *Experimental Result of the Deep Learning-Based Visually Guided Picking Control of an Omnidirectional 6-DoF Mobile Manipulator*. Accessed: May 3, 2019. [Online]. Available: <https://www.youtube.com/watch?v=3595fO300ZQ>
- [35] *Experimental Result of the Deep Learning-Based Visually Guided Picking Control Method Without Being Aligned With the Box*. Accessed: Sep. 15, 2019. [Online]. Available: <https://www.youtube.com/watch?v=uT8MG14yHR0>



CHING-CHANG WONG received the B.S. degree in electronic engineering from Tamkang University, New Taipei City, Taiwan, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan, in 1986 and 1989, respectively.

In 1989, he joined the Department of Electrical Engineering, Tamkang University, where he is currently a Professor. His research interests include fuzzy systems, intelligent control, SOPC design, and robot design.



CHI-YI TSAI received the B.S. and M.S. degrees in electrical engineering from the National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in electrical and control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2008.

In 2010, he joined the Department of Electrical Engineering, Tamkang University, New Taipei City, Taiwan, where he is currently a Professor. His research interests include image processing, color image enhancement processing, visual tracking, visual servoing, computer vision, and deep learning.



YU-CHENG LAI received the B.S. and M.S. degrees in electrical engineering from Tamkang University, New Taipei City, Taiwan, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include robotics, computer vision, and deep learning.



YUNG-SHAN CHOU received the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1996. He joined the Electrical Engineering Department, Tamkang University, in 1999, and where he is currently an Associate Professor. His research interests include robust control and machine learning control theory.



CHIEN-CHE HUANG received the M.S. degree in electrical engineering from Tamkang University, New Taipei City, Taiwan, in 2018. He is currently a Product Engineer with the King Yuan Electronics Company, Ltd., Taiwan. His research interests include robotics, computer vision, and deep learning.

...