Convolutional Neural Network-Based Robot Control for an Eye-in-Hand Camera

Jia Guo, Huu-Thiet Nguyen, Chao Liu, Chien Chern Cheah

Convolutional Neural Network-Based Robot Control for an Eye-in-hand Camera

Jia Guo, Huu-Thiet Nguyen, Chao Liu, Chien Chern Cheah

***Abstract*—In past decades, much progress has been obtained in vision-based robot control theories with traditional image processing methods. With the advances in deep learning based methods, Convolutional Neural Network (CNN) has now replaced the traditional image processing methods for object detection and recognition. However, it is not clear how the CNN-based methods can be integrated into robot control theories in a stable and predictable manner for object detection and tracking, especially when the aspect ratio of the object is unknown and also varies during manipulation. In this paper, we develop a vision-based control method for robots with an eye-in-hand configuration, which can be directly integrated with existing CNN-based object detectors. The task variables are generated based on parameters of the bounding box from the output of any real-time CNN object detector such as You Only Look Once (Yolo). To address the chattering problem of bounding box, Long Short-Term Memory (LSTM) is used to provide smoothed bounding box information. A vision-based controller is then proposed following task-space motion control design formulation in order to keep the object of unknown aspect ratio in the center of field of view of the camera. The stability of the overall closed-loop control system is analyzed rigorously using Lyapunov-like approach. Experimental results are presented to illustrate the performance of the proposed CNN-based robot controller.**

***Index Terms*—CNN, Vision-based control, Robot control.**

I. Introduction

VISION system constitutes an important part of robotic systems as it provides useful information for decision and control. With the help of visual information, the objects to be tracked or manipulated can be identified, localized and their geometric relationships with respect to the robotic systems can also be obtained. Based on that, visual servoing and vision-based robot control methods have been developed for various applications. Robot control is known to be a challenging control problem because of the non-linearity and uncertainty of the kinematics and dynamics. Most of the robot control theories with dynamic uncertainty (see [1]–[4] and the references therein) were inspired by the pioneer work in [5] where Lyapunov method was first introduced in robot control.

In general, robot motion control [6] can be mainly classified into joint space control [2] and task space [5] or operational space [7] control. Traditionally, robots have been mostly used in factory automation where the environment is structured and fixed. In such scenarios, the control tasks are less challenging in the sense that the target objects can usually be detected with sufficient accuracy as uncertain factors can be greatly suppressed and therefore joint space control methods can be directly applied. With the recent advances in robotic and sensing technologies, robots have found their way to many new applications in many emerging industries/areas such as constructions, logistics, transportation and healthcare. In these applications, the control tasks face new challenges including complex working environments, noisy and inaccurate sensor measurements, calibration or kinematic errors etc.

To address the kinematic uncertainty issues in motion control problems caused by these challenges, some earlier works have been proposed [8]–[10]. Approximate Jacobian controllers [8], [9] were first developed for setpoint control of robot with uncertain kinematics and dynamics. The first adaptive Jacobian controller for tracking control of robots with uncertain kinematics and dynamics was developed in [10], by using the concept of modular adaptive law to update the kinematic and dynamic parameters separately. Motivated by these works [8]–[10], considerable achievements have been obtained in understanding the setpoint and trajectory tracking control problems with kinematic uncertainty later on [11]–[16]. A setpoint control problem with amplitude limited control inputs was considered in [11]. A prediction error based adaptive Jacobian controller was developed in [12] for tracking control tasks. In vision-based control tasks, the parameters of depth information between the features and the camera could not be adapted together with other kinematic parameters due to the non-linearity property. To overcome this problem, a depth-independent control method was developed in [13]. Later on, it was found in [14] that the parameters of the depth information could be updated separately based on the concept of modular adaptive law. To isolate the design and analysis of the kinematic control system in task-space control, a separation approach was developed in [15], [16]. Besides the traditional setpoint control and trajectory tracking control, the concept of region reaching control was proposed in [17] where the desired control objective was specified as a region instead of a desired point or trajectory. In these works [8]–[17], the structure of the Jacobian is assumed to be known. Recently, a deep neural-network based robot controller [18] was developed for robot control with unknown Jacobian based on fully connected neural networks. These methods [8]–[18] were developed based on the assumption that external sensor measurements, such as vision systems using traditional image processing, are available with fidelity.

Traditional image processing techniques have enabled the development of kinematics based visual servoing [19], [20]

J. Guo, Huu-Thiet Nguyen and C. C. Cheah are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798 (email: ECCCheah@ntu.edu.sg).

Chao Liu is with Department of Robotics, LIRMM, CNRS - University of Montpellier, France.

This work was supported by the Ministry of Education (MOE) Singapore, Academic Research Fund (AcRF) Tier 1, under Grant RG65/22.

which does not consider the effects of robot dynamics in the control process. This field has mainly been considered as a standalone research field in robotics but some recent efforts have been made to integrate the kinematic servoing laws with dynamic control theories based on the separation approach [15], [16]. Some research efforts have also been devoted to solving visual servoing problems with visibility constraints, which inherently arise owing to the limited field of view of the camera. The authors in [21] presented a prescribed performance visual servoing scheme with a predefined visibility constraint. Similar to other pixel-based methods based on image processing, this method would also fail if the object features cannot be distinguished easily from the background. Moreover, Miao et al. [22] considered a vision-based formation control problem with field of view constraints. But again, pixel-based method with a fixed constraint was used. The vision-based control problems with limited field of view [23] could also be solved by using the concept of region-based control [17], [24], [25]. Although using a desired region is more robust, it is also hard to pre-set the desired region for unknown objects, as it shares a similar problem as using a fixed constraint.

All aforementioned robot control methods [8]–[23] were developed based on traditional image processing. However, it is noted that current state-of-the-art techniques for object detection are mainly based on deep neural networks [26]–[28]. As one of the most effective networks in the deep learning field for image classifications, Convolutional Neural Network (CNN) [29]–[32] have made impressive achievements in many areas, thanks to their advantages of fast training, sharing weights, and downsampling dimensionality reduction. Inspired by CNN, Region Convolutional Neural Network (R-CNN) was proposed in 2014, which means Regions with CNN features [33]. Compared with traditional CNN which is mainly used for object classifications, R-CNN can also achieve object detection and tracking. Other types of object detection algorithms based on CNN include Single-shot detector (SSD) [34] and You Only Look Once (Yolo) [35], [36]. With these methods, bounding boxes of the target objects and their class probabilities of prediction can be generated from camera image synchronously in real time. But these works have been carried out purely for vision-based object detection purpose and no link to robot control theory has been considered. Although CNN has made significant impacts especially in signal processing domains like image processing and voice recognition, little literature has been reported about CNN based visual servoing for robots [37]–[39]. Most of the reported works focus on obtaining the relationship between the image and the expected output by training a CNN-based model. The authors in [37] found that CNN based visual servo commands could be generated for unmanned autonomous vehicle (UAV), by minimizing the estimated relative camera pose based on the target and current images. However, it was assumed that a desired image for the target camera pose could be obtained in advance. In [38], a deep neural network-based method which combines AlexNet [40] and VGG16 [31] was proposed. The authors focused on how to create a dataset automatically and efficiently for the network training rather than robot control. A network [39] called difference of encoded features driven interaction matrix network (DEFINet) was proposed to estimate the relative pose for an eye-to-hand camera. This method cannot be easily expanded to the eye-in-hand system as it is difficult to obtain the model due to changes in the field of view. In these methods [37]–[39], a desired position/pose or target image must be specified for the robot in order to define the control task. However, in actual implementations, little information can be obtained in advance for the target, such as desired or target image, size, aspect ratio of target objects, etc. If the prior information of the target object cannot be obtained, then the control tasks cannot be accomplished by these methods.

Although existing CNNs play an important role in object detection in the domain of computer vision, to the best of our knowledge, there is no robot controller which incorporates existing CNN-based object detector for object detection and tracking purpose and meanwhile guarantees the control performance under the Lyapunov analysis framework. Therefore, the stable vision-based robot control method integrating CNN-based object detector with rigorous theoretic support remains an open problem. The main difficulty lies in that the bounding box of an object generated by CNN-based object detector has different aspect ratios in the field of view of camera due to different viewing distance and orientation angle. Therefore, for an uncertain object, it is impossible to pre-define the exact bounding box aspect ratio before the control task.

The contributions and novelties of this paper are therefore listed as follows:

(1) A novel CNN-based robot control method is proposed to detect and track objects with unknown aspect ratio by positioning the object within a desired region in the field of view. To the best of our knowledge, it is the first study to integrate CNN-based object detectors into the synthesis of a stable vision-based robot controller with rigorous theoretic support under the Lyapunov analysis framework.

(2) By setting a desired range instead of exact values for the object bounding box, the proposed method can tolerate uncertainty and changes in the object geometric shape within the camera image and thus provides a flexible and controllable strategy for the object detection and tracking.

As CNN-based detector output may contain random noises, chattering in the detector output is inevitable. Long Short-Term Memory (LSTM) is then used to provide a smoothed output for the task variables and it is easy to implement together with the CNN object detector. A series of experiments with different objects have been conducted to verify the effectiveness of the proposed CNN-based robot control method for object detection with eye-in-hand configuration.

II. Problem Formulation

In this paper, we consider a vision-based robot control problem by using a CNN object detector. A robot manipulator mounted with a camera is used to detect a target object as illustrated in Fig.1 and a CNN based robot controller is developed to move the camera. This configuration is known as the eye-in-hand configuration [19], [20], [41], [42] in the literature. The main objective in this paper is to control the

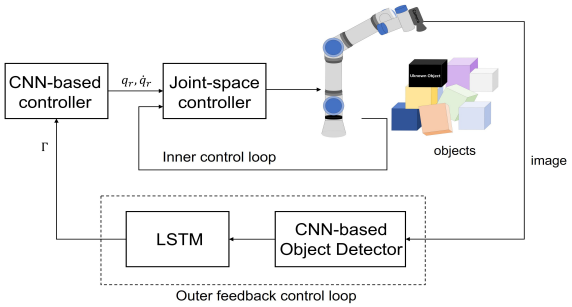


Fig. 1. An overall block diagram of the robot control system with an eye-in-hand configuration.

robot based on the CNN object detector so that the target object is positioned within a desired area of the image field of view without knowing the position and aspect ratio of the object.

A. CNN based Objector Detector

As a classic representative deep neural network, CNN has shown great success in image classification tasks. A typical structure of CNN which consists of several convolutional layers and fully connected layers is shown in Fig. 2(a). Given an input image, the network generates an output value which represents the probability that the image belongs to a certain class. A higher probability is more meaningful for correct detection. Besides classification problems, CNN can also be extended to object detection tasks. In image classification, a class label of each image is assigned based on the output value; whereas in object detection, the location of the object of interest in the image is detected by generating a bounding box around the object, in addition to assigning a class label. Inspired by the development of R-CNN [33], several CNN-based methods such as faster-RCNN [43], SSD [34], Yolo [35], [36] were developed to achieve object detection in real time. These CNN-based object detectors possess different structures and Fig.2(b) shows an illustration based on Yolov3 [36]. It is also possible to convert any CNN into an object detector by constructing an image pyramid [44].

B. Kinematic mapping and Jacobian Matrix between Image Space and Robot Joint Space

To achieve vision-based robot control using CNN-based objector detector, the relationship between camera image space and robot joint space should first be introduced. In this section, the kinematic mapping between image space and robot joint space and the associated Jacobian matrix are described.

Let $x_i \in R^2$ represents the ith feature point's position of the object in the camera image space. $\dot{x}_i$ represent its velocities in image space, while $\dot{r}$ denote velocity in robot base frame. The relationship between velocities in image space and robot base frame is given as [14], [16], [19], [45]:

$$\dot{x}_i = \frac{1}{z_i(q)} \mathcal{J}_i(r_i)\dot{r} \tag{1}$$

where matrix $\frac{1}{z_i(q)}\mathcal{J}_i(x_i)$ represents the Jacobian matrix of mapping from end-effector base to image space, $z_i(q) \in R$ denotes as the depth of the feature point with respect to the camera frame [45].

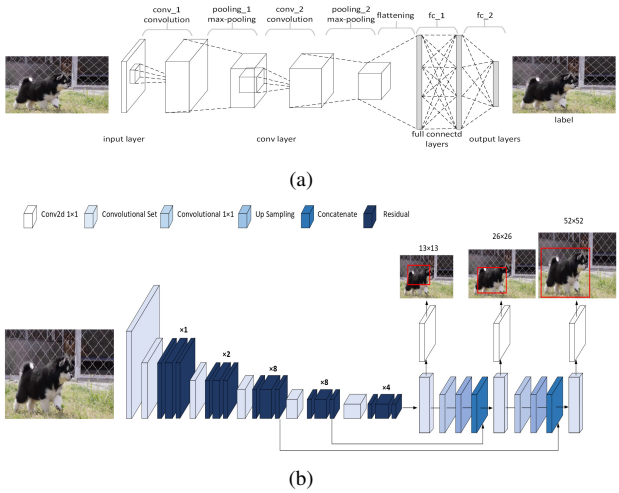


Fig. 2. Object recognition based on CNN. (a) Object classification; (b) Object detection.

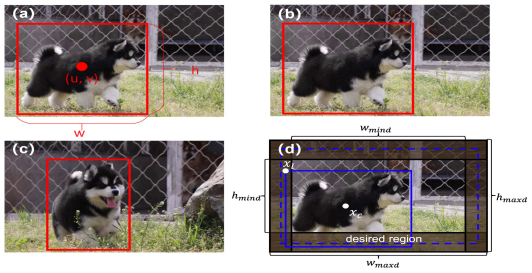


Fig. 3. (a) An illustration of bounding box and its parameters. w and h are the corresponding width and height of detected object. (u, v) denotes the pixel coordinate of center point of the bounding box. (b)∼(c) Illustration on change of aspect ratio of bounding box. (d) Scenario of FOV fitting. w and h are the width and height of current target bounding box. w_{mind} and h_{mind} are the minimum desired width and height of target bounding box, while w_{maxd} and h_{maxd} are the maximum desired width and height of target bounding box. x_c is the pixel coordinates of the center of the target bounding box, while x_l is the top left pixel coordinates of the target bounding box. The dotted line indicates a desired bounding box.

The relationship between velocity vector $\dot{r}$ and joint velocity $\dot{q}$ can be expressd as:

$$\dot{r} = \mathcal{J}_r(q)\dot{q} \tag{2}$$

where $\mathcal{J}_r(q)$ is the Jacobian matrix of manipulator from the joint space to Cartesian space.

C. Control objective

We consider an eye-in-hand configuration where the target object is within the focal length and field of view (FOV) of the camera as illustrated in Fig. 1. The control objective is to move the camera so as to keep the object of unknown aspect ratio in the center of FOV and simultaneously satisfy a constraint of width or height to maximise the view (see Fig.3(d)), which therefore achieves a better display of the object under monitoring. Generally, better view of the object leads to better detection result by the detector which comes with higher output probability value of confidence level.

The main difficulty in achieving the control objective is that the geometric shape and position of the object in image is usually unknown in advance and the aspect ratio of the bounding box may vary depending on the position and pose of the camera with respect to the object of interest. Fig.3(b)∼(c) show another example where the aspect ratio of the bounding box changes because of the movement of the target.

III. CNN-based Robot Control Method

To address the problem as described in previous section, a CNN-based control scheme is proposed in this work resorting to a novel reference joint velocity design which aims to minimize a potential energy function and hence leads to convergence of location and size of the target bounding box to the desired region. The overall block diagram of the control system is shown in Fig. 1.

A. Reference Joint Velocity Generation based on CNN-based Task Space Feedback

The target object is detected by using any CNN-based object detector which provides information of the bounding box and class label of the object of interest. To achieve smooth measurement of the object states, a LSTM network is trained based on the ground truth and actual information of the bounding box from the CNN-based object detector. The task vector is defined as the output of LSTM (see Fig. 1) as follows:

$$\Gamma = [u, v, w, h]^T \tag{3}$$

where u and v are the pixel coordinates of the center of the target bounding box and w and h are the width and height of the target bounding box.

Define the coordinates of the center point of the target bounding box as $x_c = [u_c\ v_c]^T$, while the top-left pixel vertex of the target bounding box is denoted as $x_l = [u_l\ v_l]^T$. The image variable x is defined as the two feature points in this work $x_i\ (i = 1, 2)$, where $x_1 = x_c$ and $x_2 = x_l$. Then, from eqn. (1), we obtain

$$\dot{x} = Z^{-1}(q)\mathcal{J}(r)\dot{r} \tag{4}$$

where

$$\dot{x} = [\dot{x}_1^T, \dot{x}_2^T]^T$$
$$\mathcal{J}(r) = [\mathcal{J}_1^T(r_1), \mathcal{J}_2^T(r_2)]^T$$
$$Z^{-1}(q) = \begin{bmatrix} \frac{1}{z_1(q)}I & 0 \\ 0 & \frac{1}{z_2(q)}I \end{bmatrix}$$

Substituting eqn. (2) into eqn. (4), the relationship between image feature point and joint can be expressed as:

$$\dot{x} = Z^{-1}(q)\mathcal{J}(r)\mathcal{J}_r(q)\dot{q} = Z^{-1}(q)\mathcal{A}(q)\dot{q} \tag{5}$$

where $\mathcal{A}(q) = \mathcal{J}(r)\mathcal{J}_r(q)$.

According to the definition of task variable Γ, we obtain

$$\Gamma = \mathcal{P}x, \text{where } \mathcal{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \end{bmatrix} \tag{6}$$

where $x = [x_1^T, x_2^T]^T$. The task variable is therefore expressed by the variable x and parameter matrix $\mathcal{P}$.

Differentiating eqn. (6) with respect to time and using eqn. (5) yields

$$\dot{\Gamma} = \mathcal{P}\dot{x} = \mathcal{P}Z^{-1}(q)\mathcal{A}(q)\dot{q} = J^*\dot{q}, \tag{7}$$

where $J^* = \mathcal{P}Z^{-1}(q)\mathcal{A}(q)$. Next, let us define the objective functions for the task as follows:

$$\begin{aligned} f_1(\Delta\Gamma_1) &= (u - u_d)^2 - e_u^2 \leq 0 \\ f_2(\Delta\Gamma_2) &= (v - v_d)^2 - e_v^2 \leq 0 \\ f_3(\Delta\Gamma_3) &= w - w_{max} \leq 0 \\ f_4(\Delta\Gamma_3) &= w_{min} - w \leq 0 \\ f_5(\Delta\Gamma_4) &= h - h_{max} \leq 0 \\ f_6(\Delta\Gamma_4) &= h_{min} - h \leq 0 \end{aligned} \tag{8}$$

where w_{max} and h_{max} are the maximum desired values of width and height of the target bounding box, respectively, w_{min} is the minimum desired value for width and h_{min} is the minimum desired value for height of the target bounding box, e_u and e_v are the thresholds. u_d and v_d are defined as the desired center value of the horizontal and vertical coordinates for image. The functions f_1, f_2 are used to define the desired region of center, which is a rectangular area with (u_d, v_d) as the center and $2e_u$ and $2e_v$ as the width and height, respectively. When $e_u = e_v = 0$, the desired region reduces to a desired point (u_d, v_d). Similarly, the desired range of w is defined by f_3, f_4, while the desired range of h is defined by f_5, f_6. According to the above definitions of f_3, f_4, when the variable w is in the range $[w_{min}, w_{max}]$, the values of the functions are non-positive. Similarly, for f_5, f_6, the values of the functions are non-positive when h is in the range $[h_{min}, h_{max}]$,. The parameters $u_d, v_d, w_{max}, w_{min}, h_{max}, h_{min}$ are pre-defined thus known and can be adjusted by users.

Next, a potential energy function $P_l(\Gamma)$ associated with each objective function f_l is introduced as

$$P_l(\Delta\Gamma) = \frac{1}{N}k_l[\max(0, f_l(\Delta\Gamma))]^N, N > 2 \tag{9}$$

where $k_l > 0$ and $l = 1, 2, 3, 4, 5, 6$. Partial differentiation of eqn. (9) with respect to $\Delta\Gamma$, we obtain

$$\frac{\partial P_l(\Delta\Gamma)}{\partial\Delta\Gamma} = \begin{cases} 0, & f_l(\Delta\Gamma) \leq 0 \\ k_l f_l^{N-1}(\Delta\Gamma)\left(\dfrac{\partial f_l(\Delta\Gamma)}{\partial\Delta\Gamma}\right)^T, & f_l(\Delta\Gamma) > 0, \end{cases} \tag{10}$$

Hence, the $(\partial P_l(\Delta\Gamma)/\partial\Delta\Gamma)$ is continuous and eqn. (10) can be expressed as

$$\frac{\partial P_l(\Delta\Gamma)}{\partial\Delta\Gamma} = k_l[\max(0, f_l(\Delta\Gamma))]^{N-1}\left(\frac{\partial f_l(\Delta\Gamma)}{\partial\Delta\Gamma}\right)^T \tag{11}$$

According to the definition of the potential energy function in eqn. (9), when Γ is within the desired region, $\frac{\partial P_l(\Delta\Gamma)}{\partial\Delta\Gamma}$ is zero.

Ideally, to achieve a better display, the target object should be positioned within a desired area of the image field of view so that both the width and height of the bounding box fill up the entire image as much as possible. However, since the shapes of the unknown objects may vary significantly

according to different tasks and even for a fixed object, the aspect ratio of the bounding box may also vary when the robot moves, it is therefore difficult to define a desired area in advance by using existing methods in the literature. For example, while tracking a dog as illustrated in Fig. 3, the desired width or height of the bounding box cannot be defined in advance and the robot may need to adjust the camera's position to fill up either the width or height within the image depending on the movements of the dog. In this paper, we introduce a potential energy P_T to achieve a better display in a flexible and feasible way. The potential energy P_T is defined as

$$P_T = P_1 + P_2 + (P_3 + P_4) * (P_5 + P_6) \tag{12}$$

It can be inferred that the value of total potential energy function is 0, when the task variables u, v, w, h meet the following conditions:

$$\begin{cases} -e_u + u_d \leq u \leq e_u + u_d \\ -e_v + v_d \leq v \leq e_v + v_d \\ w_{min} \leq w \leq w_{max} \end{cases} \text{ or } \begin{cases} -e_u + u_d \leq u \leq e_u + u_d \\ -e_v + v_d \leq v \leq e_v + v_d \\ h_{min} \leq h \leq h_{max} \end{cases} \tag{13}$$

From eqn. (12), it can be seen that the total potential energy is consist of three part, in which $(P_3 + P_4) * (P_5 + P_6)$ is treated as one item. When the task variables u, v and w meet the condition on the left side of eqn. (13), $(P_3 + P_4)$ is equal to 0. On the contrary, if the task variables u, v and h meet the condition on the right side of eqn. (13), $(P_5 + P_6)$ is equal to 0. In this way, the problem of obtaining a desired best fit view for any aspect ratio can be resolved.

Substituting eqn. (11) into eqn. (12) and replacing P_1, P_2 with $u - u_d$, $v - v_d$, respectively, the total potential energy P_T is:

$$\begin{aligned} P_T =& \frac{1}{2}k_1(u-u_d)^2 + \frac{1}{2}k_2(v-v_d)^2 + \frac{1}{N}\left(k_3[\max(0,\right. \\ & f_3(\Delta\Gamma_3))]^N + k_4[\max(0, f_4(\Delta\Gamma_3))]^N\big) * \frac{1}{N}(k_5 \\ & \left.[\max(0, f_5(\Delta\Gamma_4))]^N + k_6[\max(0, f_6(\Delta\Gamma_4))]^N\right) \end{aligned} \tag{14}$$

The partial differentiation of total potential energy with respect to $\Delta\Gamma_i$ can be calculated as follow:

$$\begin{aligned} &\frac{\partial P_T(\Delta\Gamma_1)}{\partial\Delta\Gamma_1} = \frac{\partial P_T(\Delta u)}{\partial\Delta u} = k_1 * (u - u_d) \\ &\frac{\partial P_T(\Delta\Gamma_2)}{\partial\Delta\Gamma_2} = \frac{\partial P_T(\Delta v)}{\partial\Delta v} = k_2 * (v - v_d) \\ &\frac{\partial P_T(\Delta\Gamma_3)}{\partial\Delta\Gamma_3} = \frac{\partial P_T(\Delta w)}{\partial\Delta w} \\ &= \left(k_3[\max(0, f_3(\Delta\Gamma_3))]^{N-1} * \left(\frac{\partial f_3(\Delta\Gamma_3)}{\partial\Delta\Gamma_3}\right)^T\right. \\ &\left. + k_4[\max(0, f_4(\Delta\Gamma_3))]^{N-1} * \left(\frac{\partial f_4(\Delta\Gamma_3)}{\partial\Delta\Gamma_3}\right)^T\right) \\ &* \frac{1}{N}\left(k_5[\max(0, f_5(\Delta\Gamma_4))]^N + k_6[\max(0, f_6(\Delta\Gamma_4))]^N\right) \\ &\frac{\partial P_T(\Delta\Gamma_4)}{\partial\Delta\Gamma_4} = \frac{\partial P_T(\Delta h)}{\partial\Delta h} \\ &= \frac{1}{N}\left(k_3[\max(0, f_3(\Delta\Gamma_3))]^N + k_4[\max(0, f_4(\Delta\Gamma_3))]^N\right) \\ &* \left(k_5[\max(0, f_5(\Delta\Gamma_4))]^{N-1} * \left(\frac{\partial f_5(\Delta\Gamma_4)}{\partial\Delta\Gamma_4}\right)^T\right. \\ &\left. + k_6[\max(0, f_6(\Delta\Gamma_4))]^{N-1} * \left(\frac{\partial f_6(\Delta\Gamma_4)}{\partial\Delta\Gamma_4}\right)^T\right) \end{aligned} \tag{15}$$

where $\Gamma = [u, v, w, h]^T$. We define the gradient of potential function $P_T(\Delta\Gamma)$ as variable $\Delta\varepsilon$, the expression is defined as follows, which can be considered as region error.

$$\Delta\varepsilon = \left[\frac{\partial P_T(\Delta\Gamma_1)}{\partial\Delta\Gamma_1}\ \frac{\partial P_T(\Delta\Gamma_2)}{\partial\Delta\Gamma_2}\ \frac{\partial P_T(\Delta\Gamma_3)}{\partial\Delta\Gamma_3}\ \frac{\partial P_T(\Delta\Gamma_4)}{\partial\Delta\Gamma_4}\right]^T. \tag{16}$$

A reference joint velocity $\dot{q}_r$ is proposed as follows.

$$\dot{q}_r = -\alpha J^{*T}\Delta\varepsilon \tag{17}$$

where α is a positive constant.

For the inner feedback control loop, the velocity tracking error can be denoted as: $\Delta\dot{q}_{in} = \dot{q} - \dot{q}_r$. As the boundedness of the velocity tracking error $\Delta\dot{q}_{in}$ is ensured by the inner control loop, we can define a positive constant β so that it satisfies the following condition [15], [16]:

$$\int_0^t \Delta\dot{q}_{in}^T(\tau)\Delta\dot{q}_{in}(\tau)d\tau \leq \beta, \quad \forall t \geq 0 \tag{18}$$

By multiplying $\Delta\dot{q}_{in}$ with J^*, we can obtain

$$J^*\Delta\dot{q}_{in} = J^*\dot{q} - J^*\dot{q}_r \tag{19}$$

Substituting eqn. (17) into eqn. (19), we have

$$J^*\Delta\dot{q}_{in} = J^*\dot{q} - J^*\dot{q}_r = J^*\dot{q} + \alpha J^*J^{*T}\Delta\varepsilon \tag{20}$$

Therefore, from eqn. (7), $\dot{\Gamma}$ can be derived as

$$\dot{\Gamma} = J^*\Delta\dot{q}_{in} - \alpha J^*J^{*T}\Delta\varepsilon \tag{21}$$

Theorem 1: Let the reference joint velocity be chosen as in eqn. (15), eqn. (16) and eqn. (17) with the total potential function P_T and the objective functions defined by eqn. (12), the system described in eqn. (21) guarantees the convergence of region error $\Delta\varepsilon \to 0$ as $t \to \infty$.

Proof: To prove the stability of the controller, a Lyapunov-like function candidate V_1 is proposed as follows.

$$V_1 = P_T(\Delta\Gamma) + \frac{1}{\alpha}\left[\beta - \int_0^t \Delta\dot{q}_{in}^T(\tau)\Delta\dot{q}_{in}(\tau)d\tau\right] \tag{22}$$

Differentiating eqn. (22) and substituting eqn. (16) and eqn. (21) into it

$$\begin{aligned} \dot{V}_1 &= \frac{\vartheta P_T(\Delta\Gamma)}{\vartheta\Delta\Gamma}\dot{\Gamma} - \frac{1}{\alpha}\Delta\dot{q}_{in}^T\Delta\dot{q}_{in} \\ &= -\alpha\Delta\varepsilon^T J^*J^{*T}\Delta\varepsilon + \Delta\varepsilon^T J^*\Delta\dot{q}_{in} - \frac{1}{\alpha}\Delta\dot{q}_{in}^T\Delta\dot{q}_{in} \end{aligned} \tag{23}$$

Since $\Delta\varepsilon^T J^* \Delta\dot{q}_{in} \leq \frac{\alpha}{2}\Delta\varepsilon^T J^* J^{*T}\Delta\varepsilon + \frac{1}{2\alpha}\Delta\dot{q}_{in}^T\Delta\dot{q}_{in}$, substituting this inequality into eqn. (23) yields

$$\dot{V}_1 \leq -\frac{1}{2\alpha}\Delta\dot{q}_{in}^T\Delta\dot{q}_{in} - \frac{\alpha}{2}\Delta\varepsilon^T J^* J^{*T}\Delta\varepsilon \leq 0 \tag{24}$$

Since $V_1 \geq 0$ and $\dot{V}_1 \leq 0$, V_1 is bounded and hence $P_T(\Delta\Gamma)$ is bounded. The boundedness of $P_T(\Delta\Gamma)$ ensures the boundedness of the functions $f_i(\Delta\Gamma_i)$. Therefore, Γ is also bounded. In addition, it can be concluded from eqn. (24) that $\Delta\varepsilon \in L_2(0, +\infty)$. Since J^* consists of image Jacobian and manipulator Jacobian, while J_r is trigonometric functions of q and image Jacobian $Z^{-1}(q)J(r)$ is bounded based on finite camera parameters, according to eqn. (17), $\dot{q}_r$ is therefore bounded. Since the boundedness of $\Delta\dot{q}_{in} = \dot{q} - \dot{q}_r$ is ensured by the inner controller, the boundedness of $\dot{q}_r$ also ensures the boundedness of $\dot{q}$. Thus, $\dot{x}$ is bounded, which ensures the boundedness of the time derivative of the region error $\Delta\dot{\varepsilon}$. Therefore, it can be concluded that $\Delta\varepsilon$ is uniformly continuous. Then it follows from [2] (Lemma C1 in its Appendix C) that $\Delta\varepsilon \to 0$ as $t \to \infty$.

Remark 1: The joint reference input described by eqn. (17) can be applied to a robot with an inner control loop which guarantees condition (18). In the above analysis, the effects of tracking error in the inner loop is taken into consideration in the analysis. In the literature of kinematic visual servoing, it is commonly assumed that $\Delta\dot{q}_{in} = 0$ for all t and the joint velocity vector is treated as the control input. In this case, β in eqn. (19) equals to 0, and the Lyapunov-like function candidate is simplified to

$$V_1 = P_T(\Delta\Gamma), \tag{25}$$

The derivative of V_1 is therefore

$$\dot{V}_1 = \frac{\vartheta P_T(\Delta\Gamma)}{\vartheta\Delta\Gamma}\dot{\Gamma} = -\alpha\Delta\varepsilon^T J^* J^{*T}\Delta\varepsilon \leq 0 \tag{26}$$

Remark 2: In this paper, we focus on the case where the Jacobian matrix is known. However, it is important to note that the problem of kinematic and Jacobian uncertainty has been extensively studied in the literature of robot control [10], [11], [14]–[16] and this result can also be similarly extended to deal with kinematic uncertainty.

B. Joint Velocity Control based on Dynamic Robot Control Method

The results in section III.A can be employed for robots with closed control architecture such as industrial robots where only joint reference commands are accessible by users. That is, the convergence of the joint velocity is ensured by the inner control loop.

In the case of robots with open control architecture, an inner control loop can be designed to force the robot joint velocity $\dot{q}$ to track the reference one $\dot{q}_r$. In this work, an adaptive control method taking into account dynamics uncertainty of the robot arm is employed.

The dynamics of a manipulator with n degrees of freedom can be expressed in joint space as [1], [2]:

$$M(q)\ddot{q} + \left[\frac{1}{2}\dot{M}(q) + S(q,\dot{q})\right]\dot{q} + g(q) = \tau \tag{27}$$

where $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix of manipulator, $\frac{1}{2}\dot{M}(q) + S(q,\dot{q}) \in \mathbb{R}^n$ denotes the centripetal and Coriolis matrix where the vector of gravitational force and moments denotes as $g(q) \in \mathbb{R}^{n\times n}$. $M(q)$ is symmetric and positive definite and $S(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is skew-symmetric. $\tau \in \mathbb{R}^n$ stands for the control input. The manipulator dynamic parameters can be expressed as [3] :

$$M(q)\ddot{q} + \left[\frac{1}{2}\dot{M}(q) + S(q,\dot{q})\right]\dot{q} + g(q) = W_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\theta_d, \tag{28}$$

where $W_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)$ is the dynamic regressor matrix and θ_d is the vector of dynamic parameters. With the presence of dynamics uncertainty, only an estimation of the dynamic parameters is available and denoted as $\hat{\theta}_d$ such that

$$\hat{M}(q)\ddot{q} + \left[\frac{1}{2}\dot{\hat{M}}(q) + \hat{S}(q,\dot{q})\right]\dot{q} + \hat{g}(q) = W_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\hat{\theta}_d. \tag{29}$$

The adaptive joint velocity controller with the proposed reference velocity $\dot{q}_r$ in eqn. (17) is proposed as:

$$\tau = -K_s\Delta\dot{q}_{in} + W_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\hat{\theta}_d. \tag{30}$$

The overall controller is different from a standard trajectory tracking controller as the reference velocity is defined by eqn. (17), (16) and (15). Note that here the reference motion signals $\dot{q}_r$ and $\ddot{q}_r$ are used in the dynamic regressor matrix $W_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)$. The adaptive vector of dynamic parameters is updated by the follow adaptation law:

$$\dot{\hat{\theta}}_d = -L_d W_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Delta\dot{q}_{in} \tag{31}$$

where L_d is a diagonal positive definite matrix.

Substituting equations (28), (29) and (30) into eqn. (27), the closed loop robot dynamics can be obtained as:

$$\begin{aligned} M(q)\Delta\ddot{q}_{in} + \left[\tfrac{1}{2}\dot{M}(q) + S(q,\dot{q}) + K_s\right]\Delta\dot{q}_{in} \\ + W_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Delta\theta_d = 0, \end{aligned} \tag{32}$$

where $\Delta\theta_d = \theta_d - \hat{\theta}_d$.

Theorem 2: With the proposed reference joint velocity $\dot{q}_r$ as defined in eqn. (17), the designed joint motion controller τ as in eqn. (30) and the dynamic parameter adaptation law in eqn. (31) guarantee that the robot velocity $\dot{q}$ converges asymptotically $\dot{q} \to \dot{q}_r$ and also the region error $\Delta\varepsilon \to 0$ as $t \to \infty$.

Proof: To analyse the convergence of the robot joint velocity $\dot{q}$ to its reference signal $\dot{q}_r$, a Lyapunov-like function candidate V_2 can be chosen as:

$$V_2 = \frac{1}{2}\Delta\dot{q}_{in}^T M(q)\Delta\dot{q}_{in} + \frac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d \tag{33}$$

Differentiating V_2 with respect to time, it has

$$\dot{V}_2 = \Delta\dot{q}_{in}^T M(q)\Delta\ddot{q}_{in} + \frac{1}{2}\Delta\dot{q}_{in}^T\dot{M}(q)\Delta\dot{q}_{in} + \Delta\hat{\dot{W}}_d^T L_d^{-1}\Delta\hat{\dot{W}}_d \tag{34}$$

Using the closed loop system equation (27) and the adaptation law of dynamic parameter vector, the above equation (34) can be simplified to:

$$\dot{V}_2 = -\Delta\dot{q}_{in}^T K_s\Delta\dot{q}_{in} \leq 0 \tag{35}$$

From equations (33) and (35), it can be seen that V_2 is positive definite in $\Delta\dot{q}_{in}$ and $\Delta\theta_d$ and $\dot{V}_2$ is negative definite in $\Delta\dot{q}_{in}$, therefore it is easy to conclude that $\Delta\dot{q}_{in} \to 0$ asymptotically, i.e. $\dot{q} \to \dot{q}_r$ asymptotically.

An overall Lyapunov function candidate V can be proposed based on V_1 and V_2 as in following to analyse the overall system performance:

$$V = V_1 + V_2 = P_T(\Delta\Gamma) + \frac{1}{\alpha}\left[\beta - \int_0^t \Delta\dot{q}_{in}^T(\tau)\Delta\dot{q}_{in}(\tau)d\tau\right] + \frac{1}{2}\Delta\dot{q}_{in}^T M(q)\Delta\dot{q}_{in} + \frac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d$$

From eqn. (24) and (35), it has

$$\begin{aligned}\dot{V} &= -\alpha\Delta\varepsilon^T J^* J^{*T}\Delta\varepsilon + \Delta\varepsilon^T J^*\Delta\dot{q}_{in} - \frac{1}{\alpha}\Delta\dot{q}_{in}^T\Delta\dot{q}_{in} \\ &\quad - \Delta\dot{q}_{in}^T K_s\Delta\dot{q}_{in} \\ &\le -\frac{1}{2\sigma}\Delta\dot{q}_{in}^T\Delta\dot{q}_{in} - \frac{\sigma}{2}\Delta\varepsilon^T J^* J^{*T}\Delta\varepsilon - \Delta\dot{q}_{in}^T K_s\Delta\dot{q}_{in} \\ &= -\Delta\dot{q}_{in}^T(\frac{1}{2\alpha}I + K_s)\Delta\dot{q}_{in} - \frac{\alpha}{2}\Delta\varepsilon^T J^* J^{*T}\Delta\varepsilon \le 0\end{aligned} \tag{36}$$

From eqn. (33), it is seen that V_2 is lower bounded. From the proof of Theorem 1, V_1 is also lower bounded. Therefore, the overall Lypunov function V is lower bounded. From the boundedness of V_2, $\Delta\dot{q}_{in}$ and $\Delta\theta_d$ must be bounded which leads to the boundedness of $\Delta\ddot{q}_{in}$ according to eqn. (32) so that $\Delta\dot{q}_{in}$ is uniformly continuous. From eqn. (36), it can be seen that both $\Delta\dot{q}_{in}^T \in L_2(0,+\infty)$ and $\Delta\varepsilon \in L_2(0,+\infty)$. Noting that from the proof of Theorem 1 $\Delta\varepsilon$ is shown to be uniformly continuous. Then similar as in proof of Theorem 1, the asymptotic convergence of $\Delta\dot{q}_{in}$ and $\Delta\varepsilon$ as $t \to \infty$ can be concluded which completes the proof.

Remark 3: Various motion control methods exist in literature to achieve desired joint position or velocity with or without consideration of kinematic uncertainties [8]–[17]. Recent research has also shown that external motion controllers can be designed for commercial robots with closed built-in motion controllers to accomplish joint space or task space control tasks [46]. In this paper, a two-step design approach like [15], [46] is used but since the proposed methodology in this paper is general, other existing works can also be integrated and developed according to specific application requirements. However, the formulations in these works [8]–[17], [46] are based on the traditional trajectory tracking control problem where a desired trajectory is first defined, and a controller is designed to track the trajectory. Comparatively, the proposed method in this work focuses on developing a controller which can integrate any existing real-time CNNs object detector to achieve better detection and tracking of object with unknown aspect ratio. The main contribution of this paper is the development of the reference joint command described by eqn. (15) $\sim$ (16) and the construction of the potential function described by eqn. (8) $\sim$ (14) so that stability of the CNN based robot control systems can be ensured while analysing the closed-loop systems using Lyapunov-like methods.

Remark 4: This paper considers a manipulator mounted with one camera but it is important to note that CNN based object detectors are capable of detecting multiple targeted objects simultaneously. As only one eye-in-hand camera is used, the proposed method cannot be used to track multiple objects independently. Nevertheless, since the aspect ratio of the bounding box is not required to be known, the CNN-based robot control can be extended to track all the objects together by generating a super bounding box which encloses all the objects. This can be achieved by taking the extreme ends or corners of all the bounding boxes of the objects to form an overall bounding box as illustrated in Fig. 4.

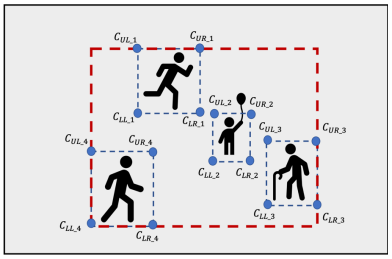


Fig. 4. Tracking of multiple objects by using an overall bounding box.

Remark 5: Recently, several works [18], [47]–[49] have been devoted to the development of stable deep learning techniques. The result in [47] was focusing on training the convolutional networks for image classification tasks rather than robot control. For control tasks, deep learning methods were developed for control systems with unknown kinematics [18] and unknown dynamics [48], [49], with or without the use of visual feedback. However, these results were developed for conventional tracking control rather than CNN-based control with unknown aspect ratio of target object.

C. Effects of Disturbances

In the presence of a disturbance $\tilde{d}_1$ in the dynamic system, the closed-loop dynamics is described according to eqn. (32) as:

$$M(q)\Delta\ddot{q}_{in} + \left[\frac{1}{2}\dot{M}(q) + S(q,\dot{q}) + K_s\right]\Delta\dot{q}_{in} + W_d(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Delta\theta_d = \tilde{d}_1, \tag{37}$$

If an disturbance or fluctuation $\tilde{d}_2$ also exists in the kinematic system, then the derivative of V in eqn. (36) becomes

$$\dot{V} = -\Delta\dot{q}_{in}^T\left(\frac{1}{2\alpha}I + K_s\right)\Delta\dot{q}_{in} - \frac{\alpha}{2}\Delta\varepsilon^T J^* J^{*T}\Delta\varepsilon + \Delta\dot{q}_{in}^T\tilde{d}_1 + \Delta\varepsilon^T\tilde{d}_2.$$

Integrating the equation yields

$$\begin{aligned}V - V(0) &= -\int_0^t \Delta\dot{q}_{in}^T(\tau)\left(\frac{1}{2\alpha}I + K_s\right)\Delta\dot{q}_{in}(\tau)d\tau \\ &\quad - \frac{\alpha}{2}\int_0^t \Delta\varepsilon^T(\tau)J^* J^{*T}\Delta\varepsilon(\tau)d\tau + \int_0^t \Delta\dot{q}_{in}^T(\tau)\tilde{d}_1(\tau)d\tau \\ &\quad + \int_0^t \Delta\varepsilon^T(\tau)\tilde{d}_2(\tau)d\tau.\end{aligned} \tag{38}$$

Since $\int_0^t \Delta\dot{q}_{in}^T(\tau)\tilde{d}_1(\tau)d\tau \le \frac{1}{2}\int_0^t \Delta\dot{q}_{in}^T(\tau)\Delta\dot{q}_{in}(\tau)d\tau + \frac{1}{2}\int_0^t \tilde{d}_1^T(\tau)\tilde{d}_1(\tau)d\tau$ and $\int_0^t \Delta\varepsilon^T(\tau)\tilde{d}_2(\tau)d\tau \le \frac{1}{2}\int_0^t \Delta\varepsilon^T(\tau)$

$\Delta\varepsilon(\tau)d\tau + \frac{1}{2}\int_0^t \tilde{d}_2^T(\tau)\tilde{d}_2^T(\tau)d\tau$, then we can obtain

$$V - V(0) \leq -\int_0^t \Delta\dot{q}_{in}^T(\tau)\left(K_s + \frac{1}{2\alpha}I - \frac{1}{2}I\right)\Delta\dot{q}_{in}(\tau)d\tau - \frac{\alpha-1}{2}\int_0^t \Delta\varepsilon^T(\tau)J^*J^{*T}\Delta\varepsilon(\tau)d\tau + \frac{1}{2}\int_0^t \tilde{d}_1^T(\tau)\tilde{d}_1(\tau)d\tau + \frac{1}{2}\int_0^t \tilde{d}_2^T(\tau)\tilde{d}_2(\tau)d\tau. \quad (39)$$

Since V is non-negative, the above inequality can be rewritten as:

$$\int_0^t \Delta\dot{q}_{in}^T(\tau)\left(K_s + \frac{1}{2\alpha}I - \frac{1}{2}I\right)\Delta\dot{q}_{in}(\tau)d\tau + \frac{\alpha-1}{2}\int_0^t \Delta\varepsilon^T(\tau)J^*J^{*T}\Delta\varepsilon(\tau)d\tau \leq \frac{1}{2}\int_0^t \tilde{d}_1^T(\tau)\tilde{d}_1(\tau)d\tau + \frac{1}{2}\int_0^t \tilde{d}_2^T(\tau)\tilde{d}_2(\tau)d\tau + V(0). \quad (40)$$

Let

$$\frac{1}{\gamma^2} \triangleq \min\left\{\lambda_{max}[K_s] + \frac{1}{2\alpha} - \frac{1}{2}, \frac{\alpha-1}{2}\right\} > 0, \quad (41)$$

where $\alpha > 1, \lambda_{max}[K_s] > \frac{1}{2}$ and $\lambda_{max}[K_s]$ denotes the maximum eigenvalue of K_s, then eqn. (40) can be rewritten as:

$$\int_0^t \Delta\dot{q}_{in}^T(\tau)\Delta\dot{q}_{in}(\tau)d\tau + \int_0^t \Delta\varepsilon^T(\tau)J^*J^{*T}\Delta\varepsilon(\tau)d\tau \leq \frac{\gamma^2}{2}\int_0^t \tilde{d}_1^T(\tau)\tilde{d}_1(\tau)d\tau + \frac{\gamma^2}{2}\int_0^t \tilde{d}_2^T(\tau)\tilde{d}_2(\tau)d\tau + \gamma^2 V(0)$$

Therefore, we can conclude that H_∞ tuning [2] (see chapter 7) with the errors $\Delta\dot{q}_{in}$, $\Delta\varepsilon$ is established for the disturbances if K_s and α are chosen as in condition (41). To eliminate the errors, a switching control terms [50] can be added but it may result in chattering of the control inputs.

IV. Experiment

To verify the performance of proposed method, several experiments were performed by implementing the controller on a 6-degree-of-freedom (DoF) robotic manipulator-UR5e [51]. The CNN based object detector used in the experiments was Yolov3 with a AP_{50} of 57.9 [36].YOLOv3 is used as it is a representative CNN detector which is commonly used in many real-time applications. The bounding box information for generating task variables can be automatically obtained online. This section is organised into three parts: first, the training of the LSTM network is presented; second, the implementation of proposed controller based on the LSTM output is presented; third, two applications employing the proposed control method for human tracking and crack detection are provided.

A. LSTM Output of bounding box

In order to achieve high-performance object detection, we use a high frame-rate camera - Intel Realsense D435i, which can capture images with more than 30 FPS. The image resolution is 640*480. However, the high frame-rate and together with missed detection in some situations may result in chattering of the bounding box. Therefore, LSTM is first used to obtain the state information of the bounding box.

As the current task variables are mainly related to its neighboring past variables, the network input number of LSTM is therefore set as 10. The network structure of LSTM used in this work is a classical three-layer network: input layer, hidden layer and output layer. Among them, there are 4 LSTM neurons in hidden layer. The activation function is sigmoid function. The model is trained for 10 epochs with 1 batch size. The input data is collected by Realsense camera. RGB image and depth information from Realsense are used in this work. In order to better model the phenomenon of chattering bounding box, training data based on both static and moving target objects was collected. Note that the use of LSTM is independent of the specific CNN detector used and hence can also be used with any real-time CNN based object detector to smoothen the output chattering.

The dataset consists of 500 frames in total, with the training set and test set being divided into the approximate proportion of 2:1. The input variables are the detection results $[u_{yolo}, v_{yolo}, w_{yolo}, h_{yolo}]^T$ obtained from Yolov3. The ground truths are obtained by manual labeling.

To show that the proposed method can be easily integrated with existing CNN-based object detector, a pretrained Yolov3 model downloaded from https://github.com/pjreddie/darknet is used. The target object in this experiment is a human. To better illustrate the chattering problem of bounding box for Yolov3, we chose 8 consecutive frames from the test set (See Fig. 5).

A comparison between actual bounding box variables and output variables of LSTM is shown in Fig. 6. The bounding box information obtained from Yolov3 were rather noisy, and it was noted that the chattering mainly appeared when the object moved. It can be seen from Fig. 6 that the LSTM output is more stable than the result provided by Yolov3 and it is more consistent with the ground truth.

B. CNN based Robot Control

To illustrate the performance of the proposed CNN-based task-space control method, we performed a series of experiments. The maximum desired region was specified as the entire image field of view (640*480) and the minimum desired width and height were specified as 384 and 288 respectively. The gains k_1, k_2, k_3, k_4, k_5 and k_6 are set as $1e^{-4}$, $1.5e^{-4}$, $1e^{-6}, 1e^{-6}, 1e^{-6}, 1e^{-6}$, and the value of α is set as 0.03. The desired value for u and v is 320 and 240, which corresponds to the horizontal and vertical coordinates of the center of an image. In the experiment, a human is chosen as the target where the height of the bounding box is much larger than the width in general.

The plots of task variables versus time are shown in Fig. 7. It can be seen from Fig. 7(a) and Fig. 7(b) that the task variables u and v gradually converge to pixel 320 and 240 respectively, which means that the target is positioned in the center of the field of view. The task variable w is not in the range [384, 640] since the shape of bounding box for a human who is standing is a vertical rectangle. However, it can be seen

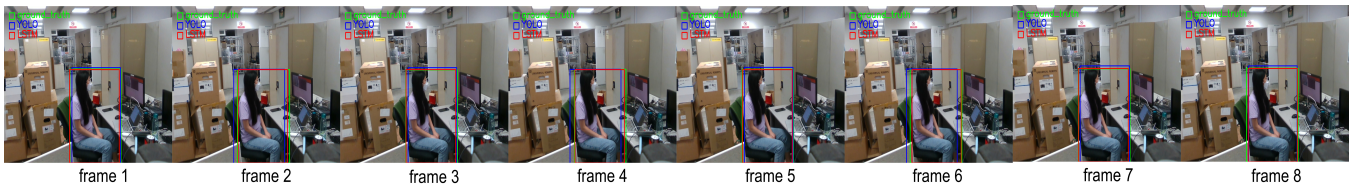
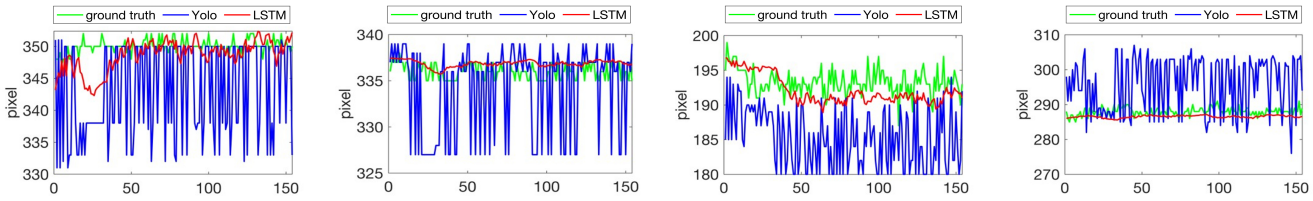
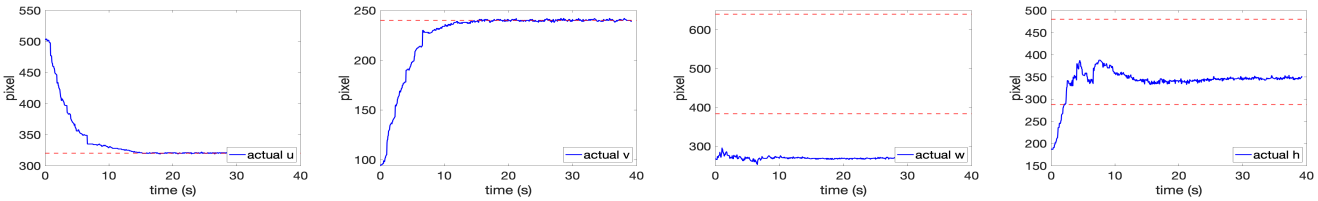


Fig. 5. 8 consecutive frames from test set.



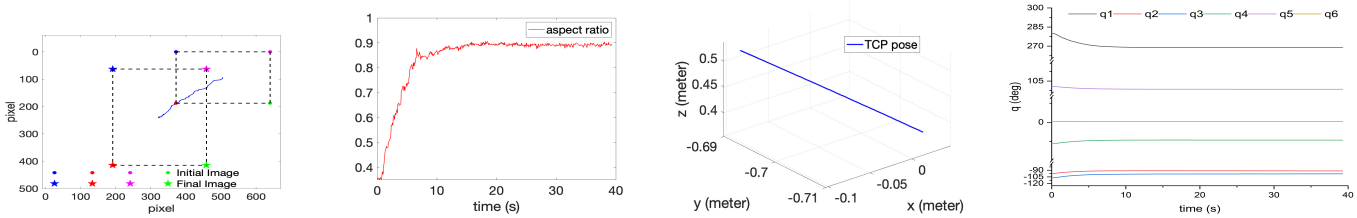
(a) task variable u (b) task variable v (c) task variable w (d) task variable h

Fig. 6. Comparisons between ground truth of task variables, task variables from Yolov3 and task variables from LSTM.



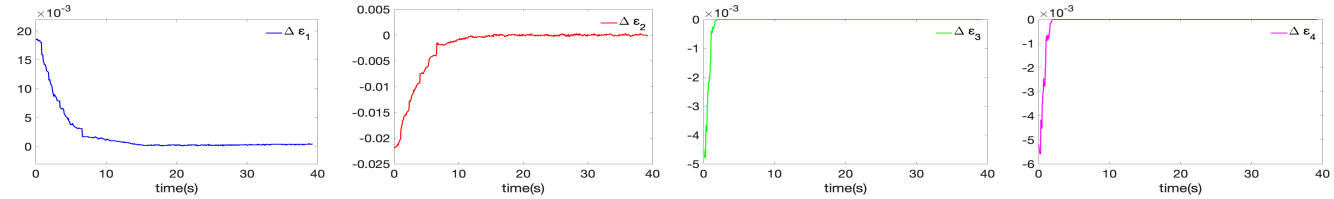
(a) task variable u (b) task variable v (c) task variable w (d) task variable h

Fig. 7. Plots of task variables over time. (a) task variable u; (b) task variable v; (c) task variable w; (d) task variable h.



(a) (b) (c) (d)

Fig. 8. Changes in trajectory and aspect ratio of the detected object; (a) Path of bounding box; (b) Plot of aspect ratio versus time; (c) A 3D path of robot end-effector in Cartesian space during control; (d) Joint angles related to robot control (q_i denotes the ith joint).



(a) (b) (c) (d)

Fig. 9. Region errors.

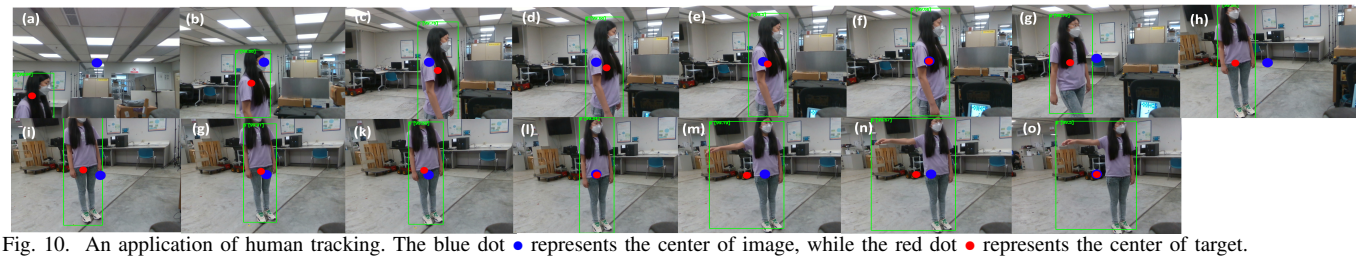
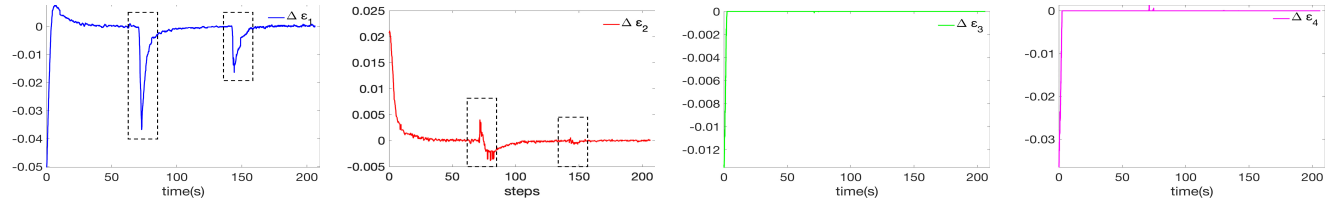


Fig. 10. An application of human tracking. The blue dot ● represents the center of image, while the red dot ● represents the center of target.



(a) (b) (c) (d)

Fig. 11. Region errors of the human tracking application.

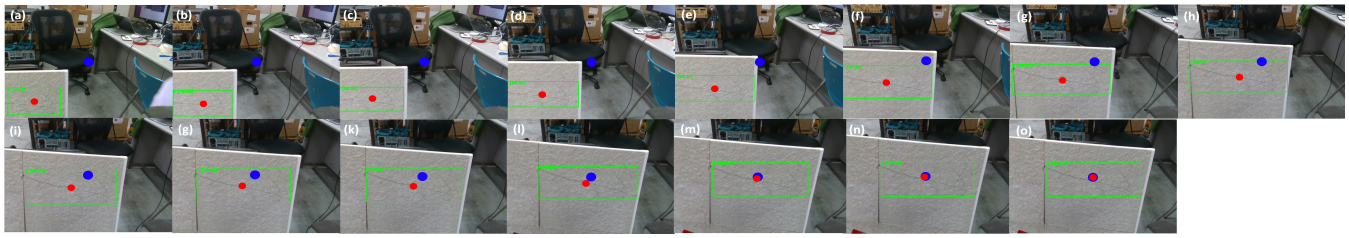
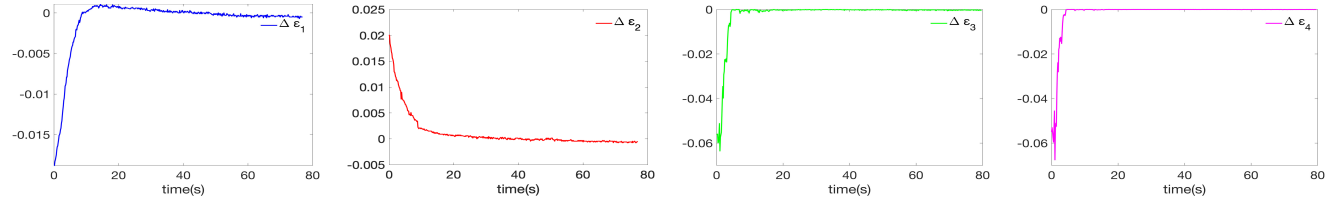


Fig. 12. An application to improve crack detection. The blue dot ● represents the center of image, while the red dot ● represents the center of target.



(a) (b) (c) (d)

Fig. 13. Region errors of the crack detection application.

that the task variable h finally reaches the desired height range [288, 480]. This illustrates the case that when the aspect ratio of the object is relatively large, the larger one is controlled to reach the desired range. Therefore, in this experiment, the task variable u, v, h reach the desired range which proves the effectiveness of the proposed method.

The trajectory of bounding box in pixels is shown in Fig. 8. According to the aspect ratio (See Fig 8.(b)), the initial shape of bounding box is a horizontal rectangle due to the initial field of view but with the movement of the manipulator, the final shape of bounding box becomes a vertical rectangle. In spite of changes in aspect ratio the bounding box over time, the target object can still be positioned at the center of the image with the desired coverage of the the field of view. A 3D path of robot end-effector in Cartesian space and the joint angles are shown in Fig 8.(c) and Fig 8.(d) respectively. The region errors $\Delta\varepsilon$ are shown in Fig.9.

C. Applications

1) Human Detection and tracking: The model used in detection and tracking of human was also the pre-trained Yolov3. The results of the detection and tracking are shown in Fig. 10. The blue dot represents the center of the image (640*480), which is a fixed point with a pixel coordinate [320, 240], and the red dot denotes the center of the bounding box for the target object, which is generated by LSTM and YOLOv3. At the beginning, the controller automatically tracked the human as seen from Fig. 10(a) to Fig. 10(f). The robot moved so that the target was positioned at the middle of the field of view as seen in Fig. 10(f). Next, the human took a step back to introduce a sudden change in position as seen in Fig. 10(g). Nevertheless, the target could still be tracked by the controller as seen from Fig. 10(g) to Fig. 10(l). Lastly, a sudden change in aspect ratio of the human was introduced by raising the hand and the robot adjusted its position to ensure that the hand remained visible within the field of view (Fig. 10(o)). The target center represented by red dot in Fig. 10(o) coincides with the image center represented by blue dot. The region errors of detection and person tracking are shown in Fig. 11. The increases in position errors indicated by the dashed boxes in Fig. 11(a) and Fig. 11(b) were caused by the changes in position and posture of the human but these errors reduced to zero eventually. In addition, the camera could simultaneously track the target automatically and stopped when the height of the target had reached the desired region (Fig. 11(d)).

2) Crack Detection: Next, instead of using pre-trained weights, we trained a Yolov3 model for detection of cracks. The total number of training images was 905, which was collected by using the Realsense camera directly.

The result of crack detection by using the proposed method is shown in Fig. 12. As the cracks are tiny and hence not easy to be detected, a better view or display of the cracks can usually lead to a better detection result. Fig. 12 shows that a deletion result with a very low confidence level of 34.5 percent at the beginning stage due to a poor view of the cracks (Fig. 12(a)). With the use of the proposed controller, the robot eventually moved to a better position so that the crack was detected with a higher confidence level of 99.97 percent (Fig. 12(o)). From Fig. 12, the center of target represented by red dot gradually tends towards and eventually coincides with the

center of image, which is represented by blue dot. In addition, the camera also moved from the initial position with partial view of the cracks (see Fig. 12(a)) indicated with a small bounding box in the lower left corner of the field of view, to a final position where the width of the bounding had reached the desired region (see Fig. 12(o)) so that the crack was more visible within the field of view. Fig. 13 shows the convergence of the region errors.

V. Conclusion

In this paper, we have proposed a CNN-based robot control framework for eye-in-hand configuration. The proposed methodology is general and can be integrated with existing CNN-based object detector. The CNN-based robot controller can be used to track objects with unknown aspect ratio by positioning the object within a desired region in the FOV. Experimental results have been presented to demonstrate the feasibility and applications of the proposed method. In this paper, the orientation of the object is not considered and therefore the bounding box may not enclose the object closely if it is rotated. Future work would include extending the method to the case of oriented bounding box so as to enclose the object more closely and thus render a more accurate object detection. This paper focuses on single object tracking using an eye-in-hand configuration and multiple objects are treated as a group (see remark 4). Future work would also be carried out to develop multi-robot coordination technique for tracking of multiple objects independently.

References

[1] F. Lewis, C. Abdallah, and D. Dawson, "Control of robot manipulators," *Editorial Maxwell McMillan, Canada*, 1993.

[2] S. Arimoto, "Control theory of nonlinear mechanical systems:a passivity-based and circuit-theoretic approach," *Claremdon Press*, 1996.

[3] J. J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.

[4] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: A tutorial," *Automatica*, vol. 25, no. 6, pp. 877–888, 1989.

[5] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *J. Dyn. Sys., Meas., Control*, vol. 103, no. 3, pp. 119–125, 1981.

[6] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Motion control," *Robotics: Modelling, Planning and Control*, 2009.

[7] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[8] C. C. Cheah, S. Kawamura, and S. Arimoto, "Feedback control for robotic manipulator with uncertain kinematics and dynamics," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 1998, pp. 3607–3612.

[9] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, "Approximate jacobian control for robots with uncertain kinematics and dynamics," *IEEE transactions on robotics and automation*, vol. 19, no. 4, pp. 692–702, 2003.

[10] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive tracking control for robots with unknown kinematic and dynamic properties," *The International Journal of Robotics Research*, vol. 25, no. 3, pp. 283–296, 2006.

[11] W. E. Dixon, "Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 52, no. 3, pp. 488–493, 2007.

[12] H. Wang and Y. Xie, "Prediction error based adaptive jacobian tracking of robots with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2889–2894, 2009.

[13] Y. H. Liu, H. Wang, C. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 804–817, 2006.

[14] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive jacobian vision based control for robots with uncertain depth information," *Automatica*, vol. 46, no. 7, pp. 1228–1233, 2010.

[15] H. Wang, "Adaptive control of robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 948–954, 2016.

[16] Y. Li, H. Wang, Y. Xie, C. C. Cheah, and W. Ren, "Adaptive image-space regulation for robotic systems," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 850–857, 2019.

[17] C. C. Cheah, D. Q. Wang, and Y. C. Sun, "Region-reaching control of robots," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1260–1264, 2007.

[18] H.-T. Nguyen and C. C. Cheah, "Analytic deep neural network-based robot control," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 2176–2184, 2022.

[19] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[20] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[21] C. P. Bechlioulis, S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, "Robust image-based visual servoing with prescribed performance under field of view constraints," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 1063–1070, 2019.

[22] Z. Miao, H. Zhong, J. Lin, Y. Wang, Y. Chen, and R. Fierro, "Vision-based formation control of mobile robots with fov constraints and unknown feature depth," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 2231–2238, 2020.

[23] X. Li, C. C. Cheah, X. Yan, and D. Sun, "Robotic cell manipulation using optical tweezers with limited fov," in *2014 IEEE International Conference on Robotics and Automation*, 2014, pp. 4588–4593.

[24] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009.

[25] S. Hou and C. Cheah, "Dynamic compound shape control of robot swarm," *IET control theory & applications*, vol. 6, no. 3, pp. 454–460, 2012.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] Y. LeCun *et al.*, "Lenet-5, convolutional neural networks," *URL: http://yann. lecun. com/exdb/lenet*, vol. 20, no. 5, p. 14, 2015.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[34] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016, pp. 21–37.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

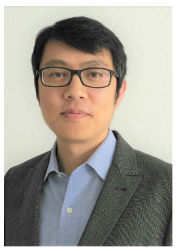
[36] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[37] A. Saxena, H. Pandya, G. Kumar, A. Gaud, and K. M. Krishna, "Exploring convolutional networks for end-to-end visual servoing," in *2017 IEEE International Conference on Robotics and Automation*, 2017, pp. 3817–3823.

[38] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *2018 IEEE international conference on robotics and automation*, 2018, pp. 3307–3314.
[39] F. Tokuda, S. Arai, and K. Kosuge, "Convolutional neural network-based visual servoing for eye-to-hand manipulator," *IEEE Access*, vol. 9, pp. 91 820–91 835, 2021.
[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
[41] S. He, Y. Xu, D. Li, and Y. Xi, "Eye-in-hand visual servoing control of robot manipulators based on an input mapping method," *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2022.
[42] L. Cui, H. Wang, X. Liang, J. Wang, and W. Chen, "Visual servoing of a flexible aerial refueling boom with an eye-in-hand camera," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 10, pp. 6282–6292, 2020.
[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.
[44] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
[45] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME transactions on mechatronics*, vol. 5, no. 1, pp. 39–48, 2000.
[46] H. Wang, W. Ren, C. C. Cheah, Y. Xie, and S. Lyu, "Dynamic modularity approach to adaptive control of robotic systems with closed architecture," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2760–2767, 2019.
[47] H.-T. Nguyen, S. Li, and C. C. Cheah, "A layer-wise theoretical framework for deep learning of convolutional neural networks," *IEEE Access*, vol. 10, pp. 14 270–14 287, 2022.
[48] D. M. Le, M. L. Greene, W. A. Makumi, and W. E. Dixon, "Real-time modular deep neural network-based adaptive control of nonlinear systems," *IEEE Control Systems Letters*, vol. 6, pp. 476–481, 2021.
[49] O. S. Patil, D. M. Le, M. L. Greene, and W. E. Dixon, "Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network," *IEEE Control Systems Letters*, vol. 6, pp. 1855–1860, 2021.
[50] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991.
[51] P. M. Kebria, S. Al Wais, H. Abdi, and S. Nahavandi, "Kinematic and dynamic modelling of ur5 manipulator," in *2016 IEEE international conference on systems, man, and cybernetics*, 2016, pp. 229–234.

Jia Guo received the B.S. degree in electronic and information engineering from Qilu University of Technology in 2013. In 2020, she received her Ph.D. degree of Intelligent Communication and Information System in Ocean University of China. From 2018 to 2019, she has been a visiting student in National University of Singapore. Now she is a research fellow in Nanyang Technological University. Her research interests include intelligent navigation system, control, machine learning and robotics.

Huu-Thiet Nguyen received the degree of engineer in control and automation engineering from Hanoi University of Science and Technology, Hanoi, Vietnam in 2015, and the PhD degree in electrical and electronic engineering from Nanyang Technological University, Singapore in 2022. His research interests include robot control, robot learning, and machine learning in robotics and physical systems.



Chao Liu received his Ph.D. degree in Electrical & Electronic Engineering from Nanyang Technological University, Singapore in 2006. He joined French National Center for Scientific Research (CNRS) as CR2 Research Scientist in 2008 and was promoted to CR1 Research Scientist in 2012. Dr. Liu's research interests include surgical robotics, teleoperation, haptics, nonlinear control theory and computer vision.

Chien Chern Cheah was born in Singapore. He received Ph.D. degrees in electrical engineering from Nanyang Technological University, Singapore, in 1996. He was a Research Fellow with the Department of Robotics, Ritsumeikan University, Japan, from 1996 to 1998. Professor Cheah is currently with Nanyang Technological University. He serves as an Associate Editor for Automatica. He has served as an Associate Editor for IEEE TRANSACTIONS ON ROBOTICS, from 2010 to 2013 and the Lead Guest Editor for IEEE TRANSACTIONS ON MECHATRONICS, in 2021.