IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 7, 2021, accepted June 19, 2021, date of publication June 25, 2021, date of current version July 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3091737

Convolutional Neural Network-Based Visual Servoing for Eye-to-Hand Manipulator

FUYUKI TOKUDA[1], (Student Member, IEEE), SHOGO ARAI[1], (Member, IEEE), AND KAZUHIRO KOSUGE[2,3], (Fellow, IEEE)

[1]Department of Robotics, Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan
[2]Center for Transformative AI & Robotics, Tohoku University, Sendai 980-8579, Japan
[3]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong

Corresponding author: Fuyuki Tokuda (tokuda@irs.mech.tohoku.ac.jp)

ABSTRACT We propose a CNN based visual servoing scheme for precise positioning of an eye-to-hand manipulator in which the control input of a robot is calculated directly from images by a neural network. In this paper, we propose Difference of Encoded Features driven Interaction matrix Network (DEFINet), a new convolutional neural network (CNN), for eye-to-hand visual servoing. DEFINet estimates a relative pose between desired and current end-effector from desired and current images captured by an eye-to-hand camera. DEFINet includes two branches of the same CNN that share weights and encode target and current images, which is inspired by the architecture of Siamese network. Regression of the relative pose from the difference of the encoded target and current image features leads to a high positioning accuracy of visual servoing using DEFINet. The training dataset is generated from sample data collected by operating a manipulator randomly in task space. The performance of the proposed visual servoing is evaluated through numerical simulation and experiments using a six-DOF industrial manipulator in a real environment. Both simulation and experimental results show the effectiveness of the proposed method.

INDEX TERMS Visual servoing, neural network, manipulator.

I. INTRODUCTION

Visual servoing [1], [2] is a method of controlling a robot by the feedback of features extracted from images in real-time. Various studies of visual servoing have been conducted so far such as end-effector pose control of a manipulator [3]–[6], formation control of multiple mobile robots [7]–[9], position and attitude control of unmanned aerial vehicle (UAV) [10]–[12], control of surgical manipulator [13], [14], and so on.

Visual servoing is generally classified into two types: position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [1], [2]. PBVS is a method of positioning a robot through the minimization of the difference between target and current poses of the robot which is estimated from captured images. PBVS has been attracting attention due to the recent price reduction and spread of 3D sensors, and the progress of 3D measurement [15], [16] and pose estimation [17]–[19] technology. However, PBVS requires an intrinsic parameter, which results in the vulnerability to the errors of the camera parameters. Moreover, the robot and the camera coordinate systems must be calibrated beforehand, which limits the practical usage of PBVS.

IBVS is a technique to position a robot by minimizing the difference between the features extracted from the current image and target image. The identification of camera parameters and the calibration between the robot and the camera coordinate systems are unnecessary by computing the matrix called interaction matrix that projects the image feature vector in image space to a robot motion in Euclidean space. In traditional IBVS, the positions/poses of geometric features, such as points and straight lines [1], in the image plane are commonly chosen as the image features and utilized to analytically compute the interaction matrix.

Hand-crafted image features such as points and straight lines can only be applied in some limited objects and scenes. Instead of extracting geometric features from images, a method of using the luminance values of images as features has been proposed [20], [21]. These methods no longer require image features matching. However, its convergence domain is small due to the high nonlinearity between the

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

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

image feature space and the workspace of the robot and the positioning accuracy is sensitive to lighting conditions and occlusions. To overcome such limitation, many researches have been conducted including methods based on Monte Carlo sequential importance sampling [22], Kalman filter [23], and Q-learning [24].

Convolutional neural networks (CNN) have shown superior performance with state-of-the-art method in some areas, such as object identification [25], [26], camera relocalization [27], [28], and pose estimation of objects [29], [30], etc.. Recently, CNN has been applied to visual servoing scheme [31]–[33] in order to overcome the limitation of visual servoing, such as the requirement of hand-crafted image features, and the sensitivity to lighting conditions and occlusions.

Saxena *et al.* [31] trained FlowNetSimple [34] by synthetic image data to position a camera mounted on UAV and accomplished visual servoing through various scenes and target poses. The networks used in [31] extracts image features from concatenated two images. To the best of our knowledge, their research is the first to servo a robot toward a target by minimizing the estimated relative camera pose between the target and current.

Bateux *et al.* [32] trained AlexNet [25] and VGG16 [35] by image data generated from a single image to position an eye-in-hand manipulator through perturbation of lighting conditions and occlusions. The main contribution of the paper is to propose a method to generate a training dataset for an eye-in-hand manipulator from a single image.

Yu *et al.* [33] proposed a new network based on Siamese architecture [36] for camera pose estimation to position an eye-in-hand manipulator. The network proposed by Yu *et al.* [33] processes images through two branches of convolutional layers which have the same structure and weights. The network regresses the camera pose from concatenated two flattened image features that are extracted from two backbones.

In this paper, we propose Difference of Encoded Features driven Interaction matrix Network (DEFINet) (Fig.3) for CNN based eye-to-hand visual servoing that utilizes subtracted image features extracted from Siamese architecture for a regression, which results in efficient performance. DEFINet consists of two parts, the feature extraction part and the regression part. Inspired by the architecture proposed in [33], the feature extraction part consists of two networks with the same structure that share weights to process two images in parallel. The biggest difference from the network in [33] is that the difference between the two encoded features is fed into the regression part to regress the relative pose, which results in high positioning accuracy. The architecture of the network is further discussed in Section III. The network is trained by a dataset for eye-to-hand configuration (Fig.1) generated from a sample dataset of images collected by operating a manipulator automatically for a given task space. The positioning evaluation is conducted through various types of architecture to reveal the effectiveness of DEFINet.

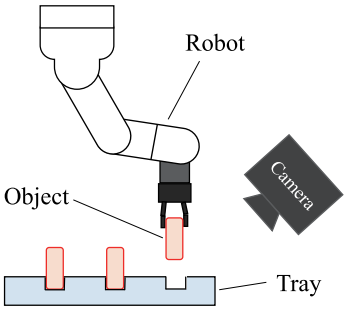

FIGURE 1. A system to position a grasped object using a camera. The camera is fixed to the ground.

More precisely, the contributions of this paper are as follows:

1) We propose DEFINet, a new CNN for visual servoing, that utilizes subtracted image features extracted from Siamese architecture based network, which results in high positioning accuracy.
2) We evaluate the performance of various networks through numerical simulation and show that the positioning accuracy and convergence domain of DEFINet is superior compared to the other networks and direct visual servoing.
3) We demonstrate that the end-effector of a 6-DoF manipulator can be positioned in high accuracy in a real environment. We show that DEFINet can generalize to unseen lighting conditions, unseen objects, and unseen occlusions in a real environment.

In the following, Section II overviews existing research regarding CNN based visual servoing. Section III proposes DEFINet for a high precision visual servoing. The experimental results of the proposed visual servoing technique are presented in Section IV. Section V concludes this paper.

II. RELATED WORKS

Deep learning, especially CNN has been applied to visual feedback control for the position control of robots. Levine *et al.* [37] achieved a complex task by a dual manipulator from captured images by using reinforcement learning and CNN. The reinforcement learning and the CNN are used to estimate the motor torques to complete tasks such as opening a lid of a bottle and hanging a hanger to a pole. CNN is trained by reinforcement learning to learn the policy of the robot motion for each task. Rahmatizadeh *et al.* [38] proposed a method for multi-task picking and placing tasks by applying CNN and long short-term memory (LSTM). Images and a task selection vector are input to CNN and LSTM to generate robot arm trajectories for the selected task. Levine *et al.* [39] applied CNN to estimate the quality of the various candidate of the command velocity to servo a manipulator towards the object to grasp it. The manipulator in eye-to-hand configuration moves toward an object by applying the command velocity with the highest quality.

Some researches [31]–[33] use CNN to estimate the relative pose between target and current images to servo a robot

toward a given target pose. Note that these methods differ from [37]–[39] in the sense that the positioning task is accomplished by servoing the robot so as to reduce the difference between the pre-captured target image and the current image. As mention in Section I, A. Saxena *et al.* [31] achieved camera mounted quadrotor (eye-in- hand) positioning into different target poses in various scenes. CNN takes a concatenated matrix of a current image and a target image and outputs the relative pose between the current and target camera. The positioning error evaluated in the synthetic environment is $(x, y, z) = (5.1$ mm, 2.8 mm, 0.5 mm), while the posture error is $(x, y, z) = (0.28$ deg., 0.42 deg., 0.42 deg.). An experiment in a real environment is accomplished using the neural network trained by a publicly available dataset.

As well, Bateux *et al.* [32] proposed a new method to generate a dataset from a single image to position an eye-in-hand manipulator. The micro-meter-order positioning accuracy is achieved using AlexNet [25] without disturbance and about 10 cm accuracy is achieved for the worst case disturbance considered in the paper. VGG16 [35] is trained by 100k images to extend their method toward the scene-agnostic scheme. The concatenated matrix of the current image and target image is fed into the neural network to estimate the relative pose between the current and target camera pose. The authors succeeded in positioning the manipulator in various scenes with the same neural network.

Yu *et al.* [33] proposed a new network for CNN based visual servoing to position eye-in-hand manipulator for a VGA-connector insertion task. The desired image and the current image are fed into Siamese architecture. The two image features of desired and current images extracted from the backbones are concatenated to regress the relative pose between the current and camera pose. The network can reduce positioning error to 0.6 mm in translation and 0.4 deg. in rotation, from initial errors of 10 mm in translation and 5 deg. in rotation.

Saxena *et al.* [31] and Bateux *et al.* [32] showed that the convergence domain of the CNN based visual servoing is larger than that of the conventional visual servoing methods. However, the major limitation was the positioning accuracy in the vicinity of the desired pose. To overcome the limitation, Yu *et al.* [33] focused on the positioning accuracy near the desired pose and proposed new architecture. In this paper, we also focus on the positioning accuracy in the vicinity of the desired pose. We propose a new network based on Siamese architecture by considering the basics of visual servoing. The performance of our method is evaluated in both simulation and real environments.

III. PROPOSED VISUAL SERVOING

In this section, a new CNN based visual servoing scheme is proposed for a precise positioning task of an industrial object grasped by a manipulator for a kitting task using an eye-to-hand system as shown in Fig.1.

The kiting task considered in this paper is an assembly task including the peg-in-hole problem. For the assembly task, the motion of the manipulator can be divided into two types of motions: gross-motion and fine-motion [40]. First, the object is picked up and transferred to the vicinity of the assembly pose by the gross-motion of the manipulator. The visual servoing proposed in this paper is for the fine motion, which is used for precise positioning of the object for the assembly of the object. The task space is pre-defined considering the position of the assembly pose, the location of the camera fixed to the task space, singular points of the manipulator, and joint constraints.

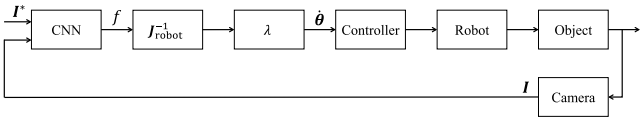

FIGURE 2. Block diagram of the proposed visual servoing. The relative pose between the desired and the current end-effector is estimated by a neural network.

This section proposes a new visual servoing scheme for the eye-to-hand system which corresponds to the fine-motion of the manipulator.

A. CONTROL LAW

Fig.2 shows a block diagram of the proposed visual servoing system. In contrast to the networks proposed in [31], [32], and [33], whose outputs are the relative pose of the camera, our network is trained to estimate the difference between the current end-effector pose and the desired end-effector pose $\boldsymbol{r}-\boldsymbol{r}^*$ from the current image $\boldsymbol{I}$ and target image $\boldsymbol{I}^*$, where $\boldsymbol{r} = (\boldsymbol{t}_f, \boldsymbol{\eta}_f)^T$ and $\boldsymbol{r}^* = (\boldsymbol{t}_f^*, \boldsymbol{\eta}_f^*)^T$ are the current and desired poses of the end-effector with respect to the base coordinate of the robot, respectively. $\boldsymbol{t}_f$ and $\boldsymbol{t}_f^*$ is defined as the current and desired translational vector, respectively. $\boldsymbol{\eta}_f$ and $\boldsymbol{\eta}_f^*$ is defined as the current and desired rotation vector that is represented in the XYZ Euler angle, respectively. The output of the network is expressed as

$$\boldsymbol{r} - \boldsymbol{r}^* = f(\boldsymbol{I}^*, \boldsymbol{I}). \tag{1}$$

The visual servoing can be realized by the following control law

$$\dot{\boldsymbol{\theta}} = -\lambda \boldsymbol{J}_{robot}^{-1}(\boldsymbol{r} - \boldsymbol{r}^*), \tag{2}$$

where $\dot{\boldsymbol{\theta}} \in \mathbb{R}^6$ is the joint velocity commanded to the velocity servo controller of the manipulator, $\lambda \in \mathbb{R}$ is the gain, and $\boldsymbol{J}_{robot}^{-1}$ is the inverse of the manipulator Jacobian. The manipulator Jacobian, $\boldsymbol{J}_{robot}$, is defined as a differentiation of the six-dimensional pose vector with respect to the joint angle vector. The inverse of the manipulator Jacobian, $\boldsymbol{J}_{robot}^{-1}$, relates the endpoint velocity vector to the manipulator joint velocity vector. The end-effector positioning control is achieved by servoing the robot toward the target pose by controlling the joint velocity by (2) in real-time.

B. NETWORK ARCHITECTURE

To estimate the relative pose of the end-effector between the target and current images, we apply a Siamese network

architecture [36]. As shown in Fig.3, DEFINet consists of two parts, the feature extraction part and the regression part.

The feature extraction part contains two parallel CNN architectures, which share the weights and parameters. Each branch of the feature extraction part is designed based on VGG16. The network accepts an input image of size 512 × 512 × 3 pixels to extract 16 × 16 × 512 feature tensor. Note that we choose VGG16 as a feature extractor from the perspective of the estimation accuracy and the prediction time.

The regression part is composed of a subtraction process, a global average pooling layer [41], and a fully connected (FC) layer. The two feature maps (16 × 16 × 1664) extracted from each branch are fed into the subtraction process to obtain a feature map that represents the difference of the extracted features.

Merging the two feature maps by a concatenating process could be another solution to process two extracted features. Although considering the conventional networks proposed for CNN based visual servoing, we think that the high nonlinearity between the image feature space of the concatenated tensor and the end-effector space causes the decrease of the estimation accuracy. The subtraction process constrains the extracted features to be zero when two input images are the same. Such constrain mitigates the high nonlinearity between the image feature space and the end-effector space, which makes the network easy to learn the feature embedding.

The global average pooling is applied to the subtracted feature map and produces a 512-dimensional vector. The common way of connecting the convolutional layers to the FC layer is the flattening of the feature maps extracted from convolutional layers, although the FC layers are prone to overfitting. In [41], global average pooling is proposed to solve the problem of over-fitting of FC layers. We experimentally confirmed that the global average pooling is more effective for over-fitting than the flatten process when trained with our dataset.

The last layer of the network is the FC layer with 6 units regressing the XYZ translation vector and XYZ Euler angle vector. For the FC layer, the Linear activation function is applied. The output of the network is directly calculated after the global average pooling. Experimentally, we found out that it is not necessary to use the fully connected layers between the layers after the backbone and the output layer. The high non-linearity between the image feature space and the output of the network is mitigated by the subtraction process, therefore, any additional fully connected layers are unnecessary to learn the feature embedding, which reduces the parameters of the network and the training time.

C. DATASET

It is essential to have a large number of training data for precise regression, though gathering such data is time-consuming. Therefore, we generate a dataset from a few sampled image data collected using a robot. The proposed approach is divided into two steps. The first step is to sample a relatively small amount of image data by operating a robot randomly in the given task space. The second step is to generate a dataset based on the combinatorial theory using the data sampled in the first step. By the generated dataset, the robot can be positioned in the given task space.

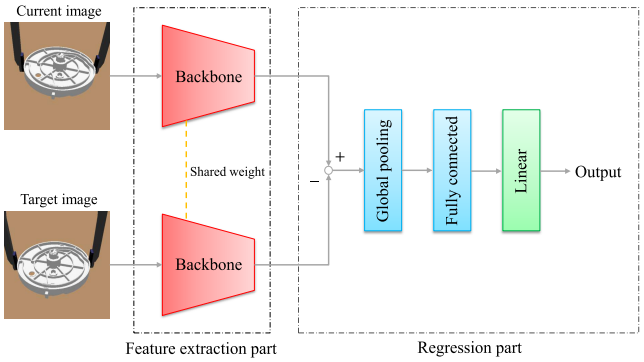

FIGURE 3. The architecture of DEFINet. The two image features extracted from the feature extraction part are fed into the regression part to estimate the relative pose between the current and desired end-effector.

In the first step, the hand of the manipulator which grasps an object is randomly moved i times in a task space around a reference hand pose. The image $\boldsymbol{I}_i$ captured by a camera and the hand pose $\boldsymbol{r}_i$ of the manipulator are stored in every iteration to form a sample data $(\boldsymbol{I}_i, \boldsymbol{r}_i)$. In the second step, two data are chosen randomly from the sample data and a training data $([\boldsymbol{I}^*, \boldsymbol{I}], \Delta\boldsymbol{r})$ (set of two images and a difference of the hand pose) is generated using two selected sample data. For example, if $(\boldsymbol{I}_1, \boldsymbol{r}_1)$ and $(\boldsymbol{I}_2, \boldsymbol{r}_2)$ are chosen from the sample data, training data is $([\boldsymbol{I}_1, \boldsymbol{I}_2], \boldsymbol{r}_1 - \boldsymbol{r}_2)$. At this point, $(\boldsymbol{I}_1, \boldsymbol{r}_1)$ corresponds to the target state and $(\boldsymbol{I}_2, \boldsymbol{r}_2)$ corresponds to the current state. Using the i-sample data, i^2-training data are collected by brute force. Finally, n data are chosen without duplication from the i^2 training data to reduce the number of training data. Both input images and outputs are normalized by their maximum and minimum values, respectively.

D. LEARNING

The Euclidean loss between the estimated vector and the ground truth vector is computed to regress the relative pose of the end-effector between the target and current images. The loss function is defined as

$$E = \alpha \|\Delta\tilde{\boldsymbol{t}}_f - \Delta\boldsymbol{t}_f\|_2 + \beta \|\Delta\tilde{\boldsymbol{\eta}}_f - \Delta\boldsymbol{\eta}_f\|_2 \tag{3}$$

where $\Delta\tilde{\boldsymbol{t}}_f$, $\Delta\tilde{\boldsymbol{\eta}}_f$, $\Delta\boldsymbol{t}_f$, and $\Delta\boldsymbol{\eta}_f$ are the ground-truth of the relative translation, the ground-truth of the relative orientation, the predicted relative translation, and the predicted relative orientation. α and β are parameters to adjust the training speed of translation and rotation vector. In this paper, $\alpha = 1.0$ and $\beta = 1.0$ are used. The network is trained for 100 epochs by Adadelta [42] using Keras library [43].

IV. EXPERIMENTS AND RESULTS

We first measure the positioning accuracy of the proposed method and the conventional method through numerical

TABLE 1. Specification of the PC.

OS	Windows 10 Home 64 bit
CPU	Intel Core i7-6950X
Memory	128 GB (DDR4-2400)
GPU	NVIDIA GeForce GTX1080

simulation to evaluate the performance of each visual servoing scheme without any disturbances such as flickering of lights and sunlight from windows. We then implement the proposed method in a real system and demonstrate the proposed visual servoing in a real environment. All the experiments are performed using PC shown in Table 1.

A. EXPERIMENT THROUGH NUMERICAL SIMULATIONS

In this experiment, we evaluate the positioning accuracy of the proposed visual servoing using DEFINet through numerical simulation. The experiments are divided into three parts. The first part evaluates the positioning accuracy of the object at the reference pose using DEFINet. The second part evaluates the validation loss of DEFINet and the other networks. The third part evaluates the average positioning accuracy to position the object into various target poses using the proposed DEFINet, the other networks, and direct IBVS.

OpenGL is used to render a six-DOF manipulator (DENSO VS-068), a parallel gripper, target objects, and the working space using CAD models. Three objects, "Object A", "Object B", and "Object C", illustrated in Fig.5, are chosen for the simulation experiments. The rendered environment and the target objects are shown in Fig.4 and Fig.5, respectively. The base coordinate system is attached at the base of the manipulator as shown in Fig.4 (b). 1,000 sample data are collected by moving the manipulator randomly in the task space to generate 3,000 training data for each object as described in Section III-C. 3,000 data of each object are stacked to generate 9,000 training data. The reference pose is $r = (-440$ mm, 75 mm, -1024 mm, 180 deg., 0 deg., -180 deg.). The task space is defined as the reference pose ± 5 mm in translations along X, Y, Z axes and ± 5 deg. around X, Y, Z axes. Fig.6 shows the example images of the training image data.

In the experiments, we consider a high precision positioning task and the small task space is defined. Note that any dimension of task space can be defined as long as the robot singular points and physical joint angle limits are not included in the task space.

1) POSITIONING AT THE REFERENCE POSE

In this section, the positioning accuracy of the object at the reference pose is evaluated. The initial pose of the visual servoing is $r = (-460$ mm, 55 mm, -1044 mm, 160 deg., -20 deg., -200 deg.) (Fig.7 (a), Fig.8 (a), Fig.9 (a)) and the desired pose is $r = (-440$ mm, 75 mm, -1024 mm, 180 deg., 0 deg., -180 deg.) (Fig.7 (b), Fig.8 (b), and Fig.9 (b)). It is worth noting that the initial displacement $\Delta r = (-20$ mm, -20 mm, -20 mm, -20 deg., -20 deg., -20 deg.) is much larger than the displacement given by the training dataset (Fig.6). $\lambda = 1.5$ is used as a visual servoing gain during the simulation experiments, which is determined by trial and error.

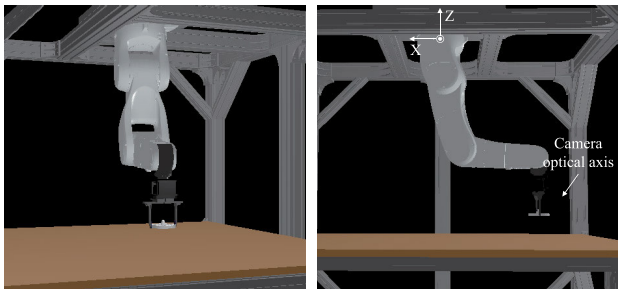

FIGURE 4. A manipulator and positioning object rendered by OpenGL. (a) The overview of the system. A positioning object is grasped by a gripper. (b) The base coordinate system is attached to the base of the manipulator. The camera captures images of the object grasped by a gripper.

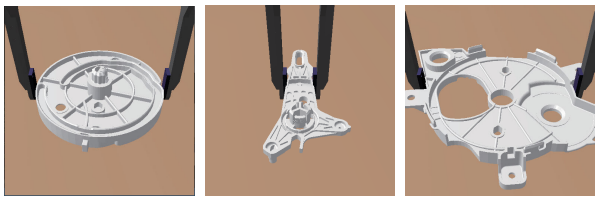

FIGURE 5. Three rendered object using CAD models. (a) Object A. (b) Object B. (c) Object C.

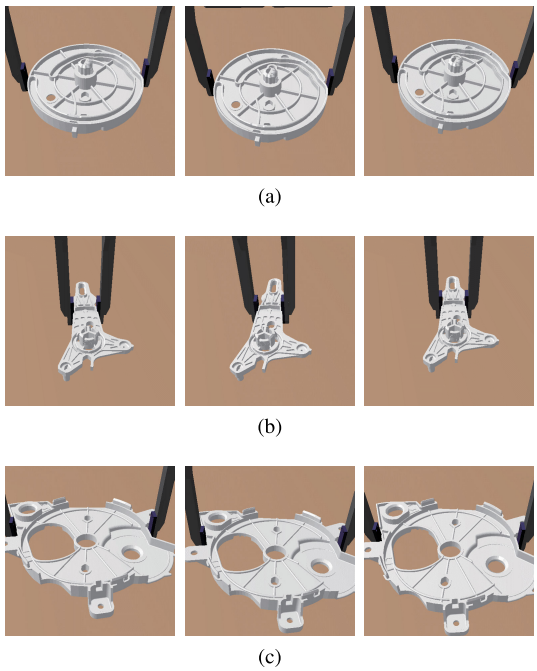

FIGURE 6. The example images of the sampled data. The sample data are collected by moving the end-effector of the manipulator randomly from a reference pose. (a) Example sample data of object A. (b) Example sample data of object B. (c) Example sample data of object C.

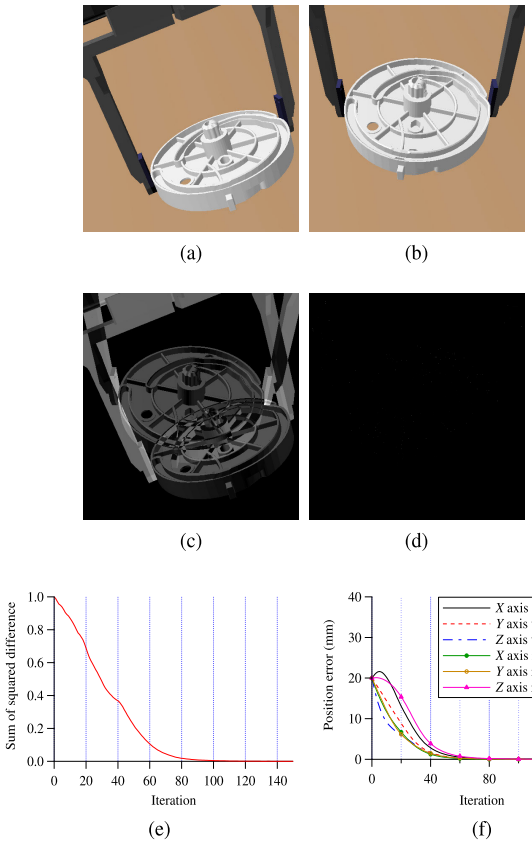

FIGURE 7. The result of positioning object A. (a) Initial pose. The initial displacement is much larger than the displacement given by the training dataset. (b) Desired pose. (c) Initial error image. (d) Error image after visual servoing. (e) The sum of the squared difference between the desired image and the current image. (f) The absolute difference between the desired and current pose.

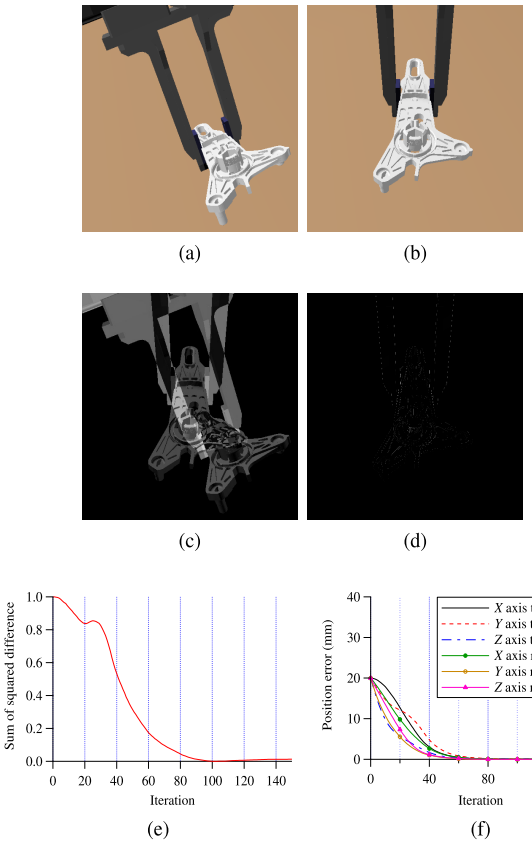

FIGURE 8. The result of positioning object B. (a) Initial pose. The initial displacement is much larger than the displacement given by the training dataset. (b) Desired pose. (c) Initial error image. (d) Error image after visual servoing. (e) The sum of the squared difference between the desired image and the current image. (f) The absolute difference between the desired and current pose.

Fig.7 (c), Fig.8 (c), and Fig.9 (c) show the image error of the initial state of the visual servoing. Fig.7 (d), Fig.8 (d), and Fig.9 (d) show the image error of the final state of the visual servoing. Note that the image error is an absolute value of the difference between desired and current images. Using the proposed method, the pose of all objects converged to the desired pose as shown in Fig.7 (e), (f), Fig.8 (e), (f), and Fig.9 (e), (f). Note that the sum of squared differences (SSD) are normalized by the value of the initial SSD. The positioning error of Object A, Object B, and Object C is $|\Delta \boldsymbol{r}| =$ (0.003 mm, 0.001 mm, 0.004 mm, 0.008 deg., 0.001 deg., 0.007 deg.), $|\Delta \boldsymbol{r}| =$ (0.051 mm, 0.084 mm, 0.039 mm, 0.022 deg., 0.013 deg., 0.023 deg.), and $|\Delta \boldsymbol{r}| =$ (0.001 mm, 0.006 mm, 0.002 mm, 0.031 deg., 0.006 deg., 0.027 deg.), respectively. In spite that the initial pose is located outside of the task space, the positioning is accomplished by the generalization of the network.

2) COMPARISON OF VALIDATION LOSS

In this section, the validation loss of DEFINet is compared with the other networks: Network α and Network β, using 1,000 data prepared for validation. Network α is designed based on Siamese architecture as same as the networks proposed in [33]. Each branch of the backbone takes the desired and current images, respectively. Two extracted image features are flattened and concatenated with each other and fed into fully connected layers to output the pose difference of the end-effector. Network β is designed based on networks proposed in [31] and [32], which is composed of backbone and fully connected layers. The backbone takes a concatenated image of the desired and current images to output the extracted image feature. The image feature is flattened and fed into two fully connected layers that consist of 1024 units to output the pose difference of the end-effector. Both of the backbones of Network α and Network β is VGG16 as same as DEFINet. Network α and Network β are trained using Adagrad [44] due to the trap to a local minimum when using Adadelta [42].

Fig.10 shows the validation loss of each network. The validation loss of DEFINet is especially low compared to Network α and Network β, which indicates that the architecture of DEFINet is effective for visual servoing usage. Furthermore, the comparison between Network α and Network β reveals that Siamese architecture improves the accuracy of the estimation.

The training time of the proposed DEFINet is 39.2 hours using the PC specified in Table 1. The training time of the

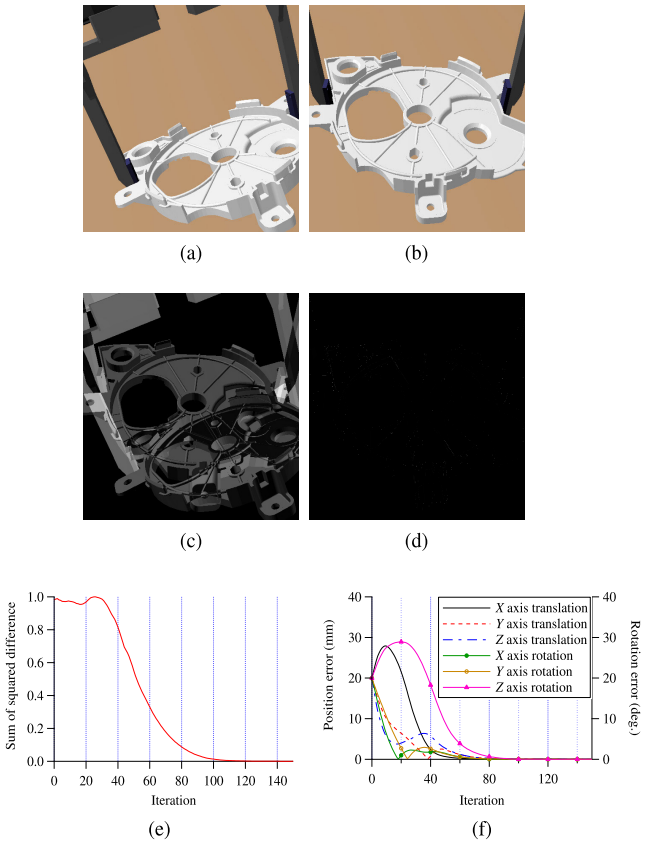

FIGURE 9. The result of positioning object C. (a) Initial pose. The initial displacement is much larger than the displacement given by the training dataset. (b) Desired pose. (c) Initial error image. (d) Error image after visual servoing. (e) The sum of the squared difference between the desired image and the current image. (f) The absolute difference between the desired and current pose.

other two networks, Network α and Network β are 55.0 hours and 26.4 hours, respectively. The prediction time of the proposed network, Network α, and Nework β are 0.20 s, 0.26 s, and 0.15 s, respectively. The computation time of the proposed DEFINet is slightly larger than Network β, but the validation loss of the proposed network is better than Network β as shown in Fig.10.

Through the validation, we found that the fully connected layers in the regression part do not affect the prediction accuracy. DEFINet does not include the fully connected layers except the output layer. The number of parameters of DEFINet, Network α, and Network β are 14,717,766, 284,206,918, and 149,990,918, respectively. The number of the proposed network is less than 10% of the other networks. Therefore, the proposed network with two backbone networks can be trained even using a single GPU such as GeForce GTX1080.

3) COMPARISON OF POSITIONING ACCURACY

Direct IBVS based on estimated Jacobian [45] is implemented for comparison experiments. The image Jacobian for each object is estimated from 3,000 data of relative pose between current and desired end-effector and the

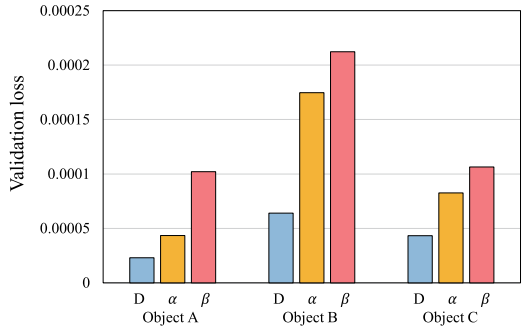

FIGURE 10. The validation loss of DEFINet, Network α, and Network β. The performance of DEFINet is superior in all of the case. Validation loss of network that is trained by training data generated from 1,000 sample data. D, α and β stands for DEFINet, Network α and Network β, respectively.

difference of the image intensity. Note that Photometric visual servoing [20] is also implemented using ViSP library [46], although the pose of the end-effector often converged to local minima.

In this section, the average positioning accuracy to position the object into various target poses using DEFINet, Network α, Network β, and direct IBVS is evaluated. The initial pose and the desired pose are randomly chosen within the range of ± 5 mm in translation and ± 5 deg. in rotation from the reference pose.

The average positioning accuracy of 50 times of trial using DEFINet, Network α, Network β, and direct IBVS are shown in Fig.11. Fig.11 (a), (b), and (c) show the experimental results of positioning Object A, Object B, and Object C, respectively. Three networks and direct visual servoing succeeded in positioning all three objects. The positioning accuracy of DEFINet is especially high for all the three objects compared to Network α and Network β. Direct IBVS also succeeded in positioning Object A and Object C in high accuracy, however, the positioning accuracy of Object B is low compared to the other objects. DEFINet succeeded in positioning all the objects under 0.073 mm translation error and 0.042 deg. rotation error, while the positioning error of the other networks and direct IBVS depends on the objects. The positioning accuracy of each method is shown in Table 2.

Next, the initial pose and the desired pose are randomly chosen from the range of $[-10, -5]$ mm and [5, 10] in translation and $[-10, -5]$ deg. and [5, 10] deg. in rotation from the reference pose, which is outside of the task space. The objective of this experiment is to evaluate the performance of positioning outside of the task space.

The average positioning accuracy and the success ratio through 50 times of trial using DEFINet, Network α, Network β, and direct IBVS are shown in Fig.12. Fig.12 (a), (b), and (c) shows the experimental results of positioning Object A, Object B, and Object C, respectively. The average positioning accuracy is calculated only using the "success positioning" attempt in which the positioning error is smaller than the initial error since the pose of the end-effector did not always converge. The success ratio of positioning Object A

TABLE 2. Positioning accuracy inside of the task space.

Object	Network	Trans. X (mm)	Trans. Y (mm)	Trans.X (mm)	Rot. X (deg.)	Rot. Y (deg.)	Rot. Z (deg.)
A	DEFINet	0.016	0.010	0.018	0.012	0.010	0.021
	Network α	0.075	0.048	0.073	0.062	0.053	0.096
	Network β	0.095	0.038	0.102	0.052	0.089	0.081
	Direct IBVS	0.037	0.011	0.030	0.032	0.038	0.052
B	DEFINet	0.061	0.048	0.073	0.033	0.033	0.031
	Network α	0.117	0.071	0.129	0.092	0.064	0.071
	Network β	0.152	0.028	0.167	0.044	0.074	0.057
	Direct IBVS	0.254	0.017	0.178	0.033	0.122	0.026
C	DEFINet	0.031	0.016	0.030	0.021	0.020	0.042
	Network α	0.110	0.061	0.090	0.094	0.080	0.089
	Network β	0.139	0.056	0.209	0.065	0.066	0.088
	Direct IBVS	0.050	0.012	0.037	0.017	0.025	0.035

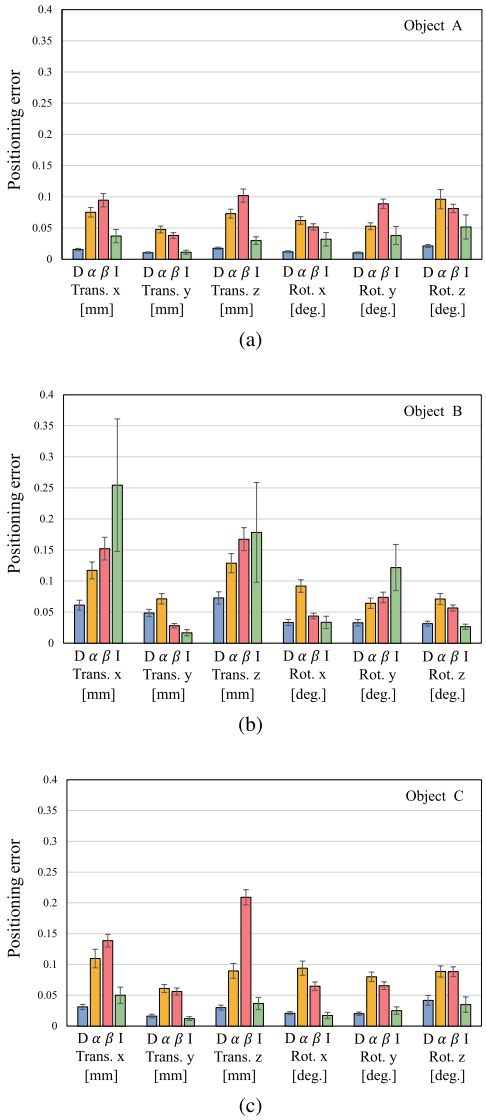

FIGURE 11. The average positioning error of each network and direct IBVS. The positioning is conducted inside of the task space. D, α, β, I stand for DEFINet, Network α, Network β, and direct IBVS, respectively. (a) Positioning error of object A. (b) Positioning error of object B. (c) Positioning error of object C.

using DEFINet, Network α, Network β, and direct IBVS are 100 %, 72 %, 82 %, and 66 %. The success ratio of positioning Object B using DEFINet, Network α, Network β, and direct IBVS are 100 %, 62 %, 64 %, and 48 %. The success ratio of positioning Object C using DEFINet, Network α, Network β, and direct IBVS are 98 %, 74 %, 78 %, and 72 %. The success ratio of positioning reveals that the convergence domain of DEFINet is larger than that of Network α, Network β, and direct IBVS.

As for the positioning accuracy, DEFINet succeeded in positioning all the objects under 1.259 mm translation error and 0.485 deg. rotation error, while the positioning error of the other networks vary depending on the object. The positioning accuracy of Network α is low compared to Network β, which indicates that Siamese architecture is fragile against unseen initial pose and desired pose. However, the architecture of the regression part of DEFINet improves the robustness against the unseen initial pose and desired pose, which leads to high positioning accuracy outside of the task space. The positioning accuracy of each method is shown in Table 3.

B. EXPERIMENT THROUGH REAL ENVIRONMENT

In this section, we measure the performance of positioning into a reference pose in a real environment to evaluate the positioning accuracy of DEFINet. Fig.13 shows the experimental system used for the evaluation. The experimental system consists of a 6-DOF manipulator (DENSO VS-068), a parallel gripper (TAIYO ESG2) attached to the manipulator, and an industrial camera (The Imaging Source DMK33UX265) which captures an image of 512 pixels $\times$ 512 pixels with 60 fps. The experimental system is equipped with a LED lighting device attached to the frame as shown in Fig.13 and covered by a black curtain to remove the effect of the external lighting source. The brightness of the LED lighting device is kept constant for each lighting condition.

The robot end-effector grasping the object is first positioned at the target pose to capture the target image. Then the end-effector of the robot is moved randomly in a predefined area for visual servoing in the task space. After the visual servoing is completed, the pose of the manipulator is acquired by solving the forward kinematics from the joint angles. The positioning error is measured from the difference between the desired and final end-effector pose.

TABLE 3. Positioning accuracy outside of the task space.

Object	Network	Trans. X (mm)	Trans. Y (mm)	Trans.X (mm)	Rot. X (deg.)	Rot. Y (deg.)	Rot. Z (deg.)
A	DEFINet	0.266	0.199	0.468	0.118	0.145	0.223
	Network α	3.853	1.382	3.246	1.803	2.190	5.436
	Network β	1.811	0.443	2.098	1.856	0.537	2.263
	Direct IBVS	5.898	3.002	5.698	5.012	3.261	5.989
B	DEFINet	0.295	1.259	0.586	0.485	0.248	0.451
	Network α	2.128	1.934	3.104	2.149	3.059	3.528
	Network β	2.001	1.053	2.863	2.477	1.148	1.612
	Direct IBVS	6.409	2.510	5.465	3.924	4.035	4.983
C	DEFINet	0.370	0.190	0.654	0.189	0.195	0.426
	Network α	6.121	3.353	5.684	2.933	3.613	4.937
	Network β	2.012	0.500	2.311	1.221	1.287	2.431
	Direct IBVS	7.814	4.455	6.085	4.475	4.032	5.923

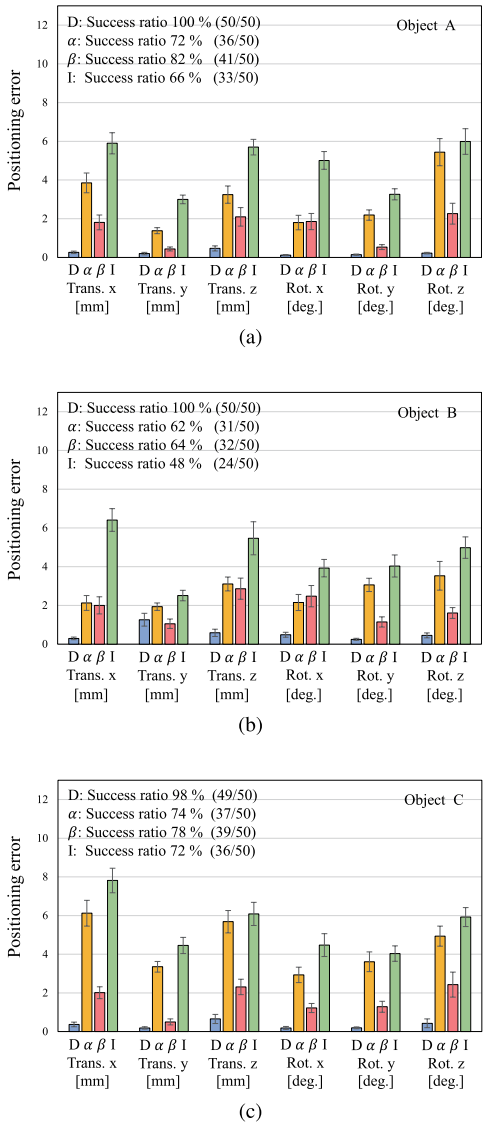

FIGURE 12. The average result of the positioning error of each network and direct IBVS. The positioning is conducted outside of the task space. D, α, β, I stand for DEFINet, network α, network β, and direct IBVS, respectively. (a) Positioning error of object A. (b) Positioning error of object B. (c) Positioning error of object C.

The dataset is generated in the same manner described in Section III-C. 1,000 sample data are collected to generate

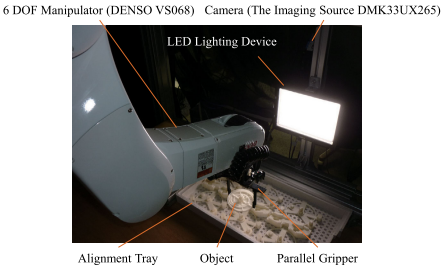

FIGURE 13. Experimental setup in a real environment.

3,000 training data. The network is trained for a single object, which we name "Real Object A" (Fig.14 (a)). The task space is defined in the range of $[-5, 5]$ mm in translations along X, Y, Z axes and [-5, 5] deg. around X, Y, Z axes. The reference pose is $r = (-420.93$ mm, 69.17 mm, -1056.53 mm, 180 deg., 0 deg., -180 deg.), where the base coordinate system is set at the base of the manipulator.

The initial and desired pose of the visual servoing are $r = (-440.93$ mm, 49.17 mm, -1076.53 mm, 160 deg., -20 deg., -200 deg.) and $r = (-420.93$ mm, 69.17 mm, -1056.53 mm, 180 deg., 0 deg., -180 deg.), respectively, where the difference between initial pose and the desired pose is given by $\Delta r = (-20$ mm, -20 mm, -20 mm, -20 deg., -20 deg., -20 deg.). The visual servoing gain $\lambda = 1.0$ is adjusted by trial and error so that the calculated joint velocity does not exceed the joint angular velocity constraint. The velocity command is updated at about every 65 ms.

Fig.15 shows the time-series images of visual servoing. The pose of the manipulator changes dramatically through the visual servoing to position the object. The object converged into the desired pose, in spite that the difference between initial pose Fig.16 (a) and the desired pose Fig.16 (b) is much larger than the difference given for the training dataset.

Fig.16 (c) and (d) show the error image between the desired and current images of the initial and final state of the visual servoing, respectively. Despite the large displacement in the initial state, the desired and current images match exactly in the final state. The corresponding behavior can be also observed from Fig.16 (e), where the error of the sum of the squared difference between the desired and current image

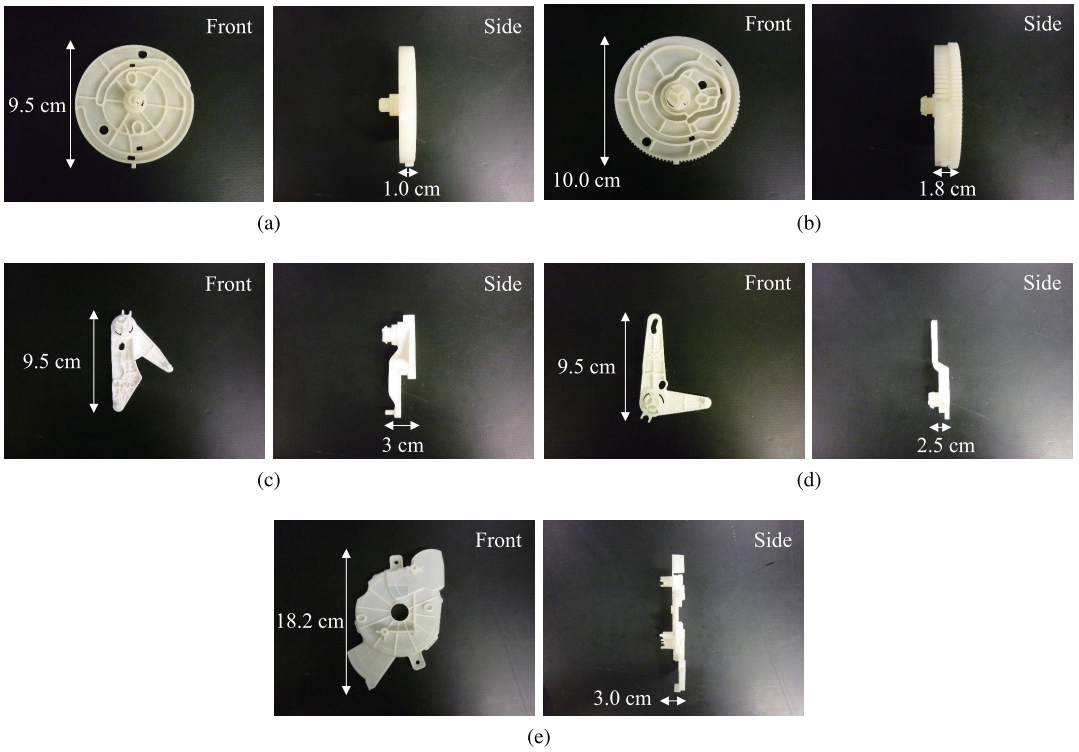

FIGURE 14. The dimension of each object used in the experiment shown in section IV-B and C. (a) Real Object A. (b) Real Object B. (c) Real Object C. (d) Real Object D. (e) Real Object E.

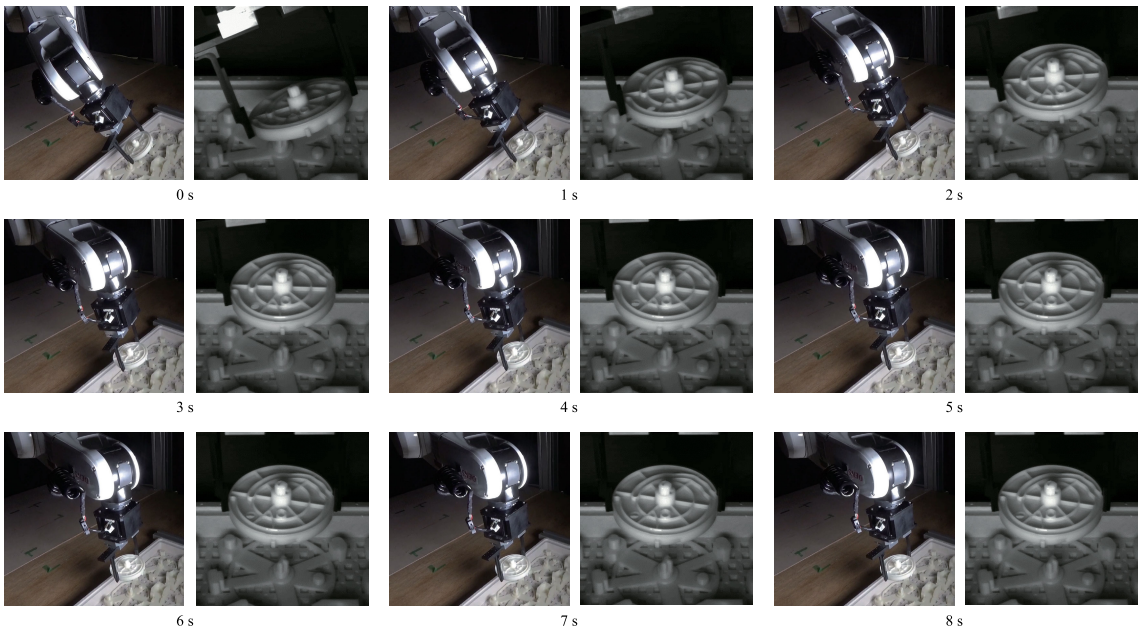

FIGURE 15. Time-series images of the visual servoing from 0 s to 8 s. The pose of the manipulator changes dramatically through visual servoing.

converges near zero. Fig.16 (f) presents the result of the error of the hand pose. The position error is $(x, y, z) = (0.324$ mm, 0.110 mm, 0.046 mm) and the posture error is $(x, y, z) =$ (0.005 deg., 0.075 deg., 0.038 deg.), which can be said that the proposed method can be used in a practical scene.

C. TOWARDS UNSEEN ENVIRONMENTS

1) POSITIONING UNDER UNSEEN LIGHTING CONDITIONS

In this section, we evaluate the generalization ability of DEFINet to unseen lighting conditions. The network is trained by the same training dataset as described in previous

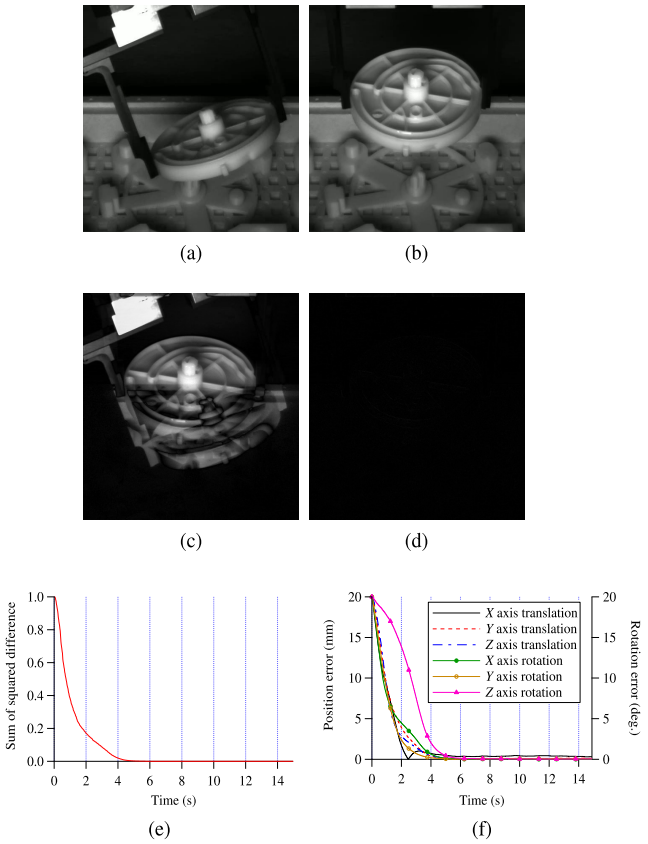

FIGURE 16. An example of the experimental results of visual servoing in a real environment using DEFINet. (a) Initial pose. The initial displacement is much larger than the displacement given by the training dataset. (b) Desired pose. (c) Initial error image. (d) Error image after visual servoing. (e) The sum of the squared difference between the desired image and the current image. (f) The absolute difference between the desired and current pose.

Section IV-B. Note that the network is trained by dataset only for Real Object A under the constant light condition.

The positioning experiment is conducted in darker and brighter lighting conditions than that of the training dataset using Real Object A to evaluate the generalization to the unseen lighting condition. Bright and dark lighting conditions are created by changing the brightness of the LED lighting device. The initial pose and the desired pose is $r = (-430.93$ mm, 59.17 mm, -1066.53 mm, 170 deg., -10 deg., -190 deg.) and $r = (-420.93$ mm, 69.17 mm, -1056.53 mm, 180 deg., 0 deg., -180 deg.), respectively, where the difference between initial pose and the desired pose is given by $\Delta r = (-10$ mm, -10 mm, -10 mm, -10 deg., -10 deg., -10 deg.).

Fig.17 shows the initial image, desired image, final image, initial error image, and final error image. The first and second row shows the result of the positioning experiment using Real Object A under dark environment and bright environment, respectively. The positioning accuracy for each lighting condition is shown in Table 4. The network succeeded in positioning Real Object A under unseen lighting conditions under 1.068 mm error in translation and 0.224 deg. error in rotation.

The positioning accuracy under the dark lighting condition is higher than that of the bright lighting condition as shown in Table 4. The color of the objects used for experiments is almost white and brighter than that of the background. Under the dark lighting condition, the image of the object was not affected so much and was captured well. Under the bright lighting condition, the background image was not affected so much but the object image was affected to a certain extent. Actually, 23.9% of the image pixels related to the object were saturated under the bright lighting condition. This caused the positioning accuracy under the bright condition lower than that of the bright condition. However, the positioning accuracy of objects with unseen lighting condition can be improved by including different lighting conditions in the training dataset.

2) POSITIONING OF UNSEEN OBJECTS

Visual servoing is conducted using four unseen objects: Real Object B, Real Object C, Real Object D, and Real Object E, under the same light condition as the training dataset to evaluate the generalization ability of NEFINet to unseen objects. The shape of Real Object B is similar to Real Object A, and the shapes of Real Object C, Real Object D, and Real Object E are completely different from Real Object A. The dimension of each object is shown in Fig.14. The initial pose and the desired pose are $r = (-425.93$ mm, 64.17 mm, -1061.53 mm, 175 deg., -5 deg., -185 deg.) and $r = (-420.93$ mm, 69.17 mm, -1056.53 mm, 180 deg., 0 deg., -180 deg.), respectively, where the difference between the initial pose and the desired pose is given by $\Delta r = (-5$ mm, -5 mm, -5 mm, -5 deg., -5 deg., -5 deg.).

Fig.18 shows the result of the positioning experiment using Real Object B, C, D, and E under the same lighting condition as the training dataset. The positioning accuracy for each object is shown in Table 5. The positioning accuracy of Real Object B, Real Object C, and Real Object D are high even though the shapes are completely different from Real Object A. The positioning error of Real Object E is larger than the other objects. The resin material of the Real Object A (Seen object), B, C, and D is the same and only the Real Object E is made of different resin material and has different reflectance and transparency. There is a certain limitation of generalization to unseen objects with unseen reflectance and transparency. We expect that the positioning accuracy of objects with unseen reflectance and transparency can be improved by including objects with different resin materials in the training dataset.

3) POSITIONING OF UNSEEN OBJECTS UNDER UNSEEN LIGHTING CONDITIONS

To evaluate the generalization ability of DEFINet to both unseen objects and lighting conditions, we conducted visual servoing using four different objects: Real Object B, Real Object C, Real Object D, and Real Object E, under darker and brighter lighting conditions than that of the training dataset. The positioning of unseen objects under unseen lighting

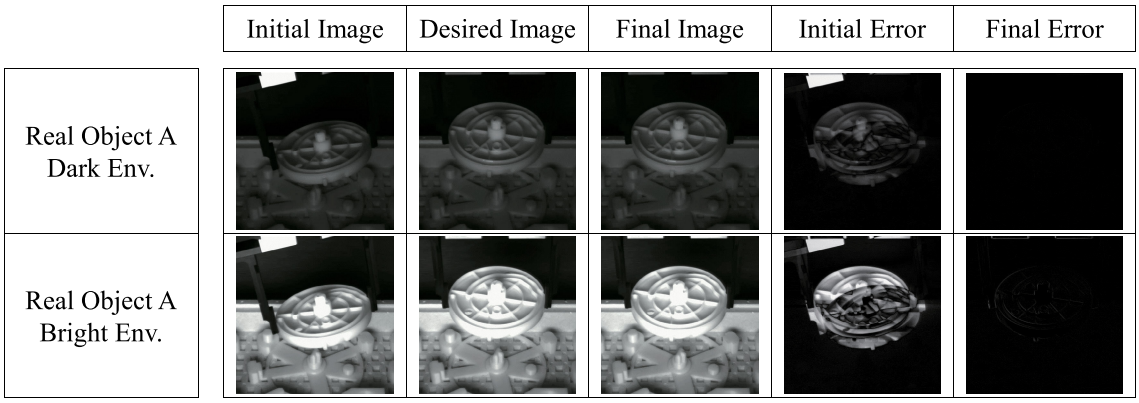

FIGURE 17. Experimental results of positioning experiments under unseen lighting conditions using a seen object (real object A). The network is trained by a dataset of real object A. The first row corresponds to the result shown in section IV-B. The second and third row corresponds to the result of the positioning experiments under unseen lighting conditions. The network managed to position the object under the unseen dark environment and bright environment.

TABLE 4. Positioning accuracy of the seen object under unseen lighting conditions.

Object	Trans. X (mm)	Trans. Y (mm)	Trans.X (mm)	Rot. X (deg.)	Rot. Y (deg.)	Rot. Z (deg.)
Real Object A in Dark Env.	0.250	0.088	0.043	0.035	0.001	0.049
Real Object A in Bright Env.	1.068	0.188	1.037	0.149	0.018	0.224

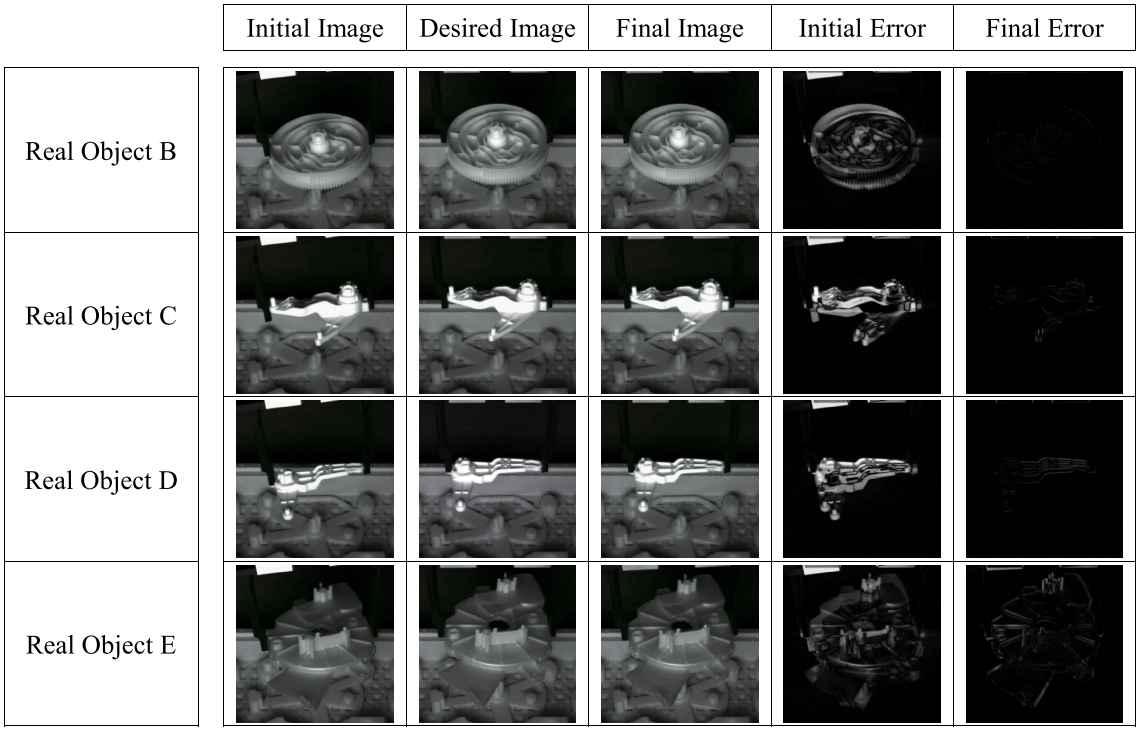

FIGURE 18. Experimental result of positioning experiment using unseen objects (real object B, real object C, real object D, and real object E) under the same lighting condition as the training dataset. Note that the network is trained by a dataset of real object A. The network succeeded in positioning real object B, real object C, and real object D. However, the positioning error of real object E is larger compared to the other objects, which can be mitigated by including various objects to the training dataset.

conditions is a challenging task and the network could not position the objects from initial error of ±5mm in translation and ±5 deg. in rotation, therefore we chose a slightly smaller initial error than former experiments. The initial pose and the desired pose for the unseen object is $r = (-423.93$ mm, 66.17 mm, -1059.53 mm, 177 deg., -3 deg., -183 deg.) and $r = (-420.93$ mm, 69.17 mm, -1056.53 mm, 180 deg., 0 deg., -180 deg.), respectively, where the difference between initial

TABLE 5. Positioning accuracy of the unseen objects under seen lighting condition.

Object	Trans. X (mm)	Trans. Y (mm)	Trans.X (mm)	Rot. X (deg.)	Rot. Y (deg.)	Rot. Z (deg.)
Real Object B	0.208	0.059	0.010	0.014	0.076	0.173
Real Object C	0.473	0.013	0.200	0.184	0.100	0.733
Real Object D	0.567	0.108	0.317	0.296	0.094	0.100
Real Object E	3.526	0.185	3.989	1.295	0.566	2.073

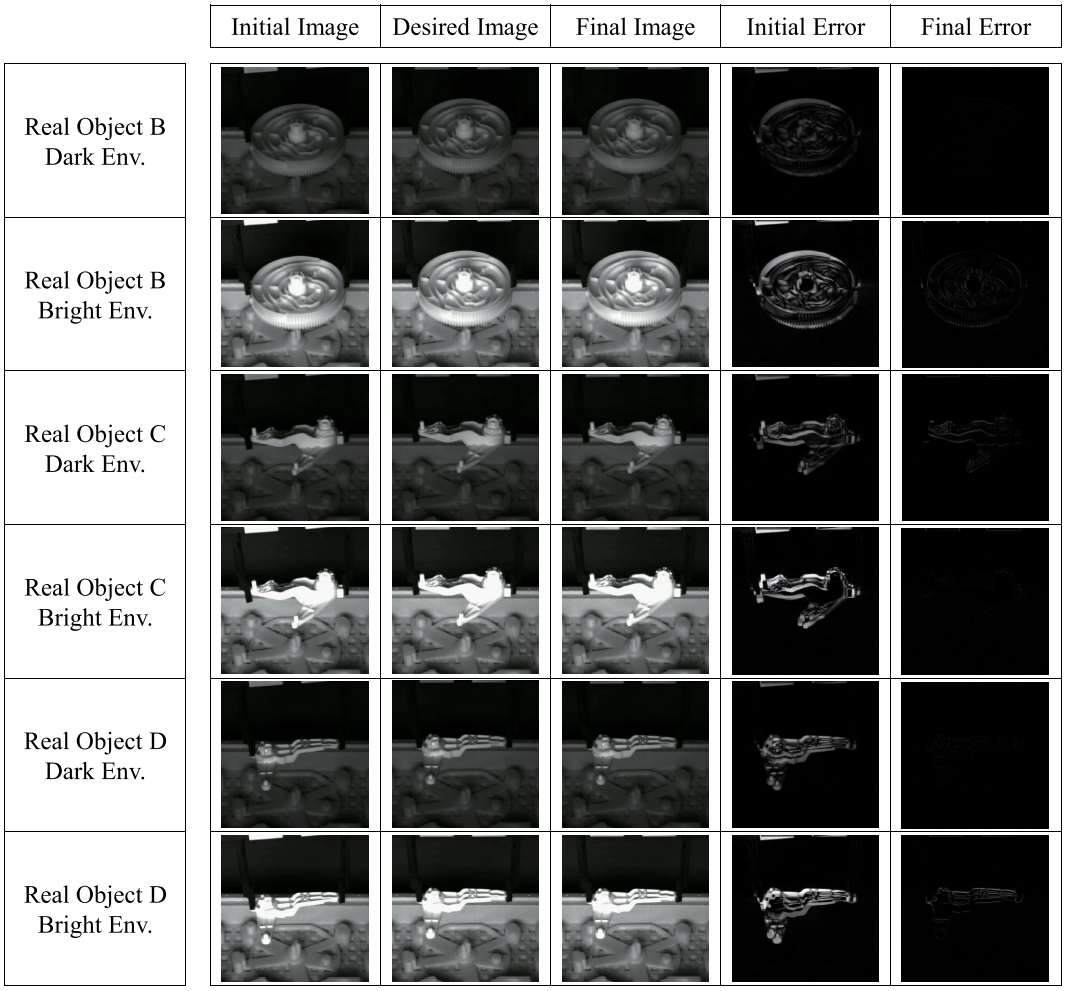

FIGURE 19. Experimental result of positioning using unseen objects (real object B, real object C, real object D) under unseen lighting condition (dark env. and bright env.). The network positioned unseen objects under unseen lighting conditions.

TABLE 6. Positioning accuracy of the unseen objects under unseen lighting conditions.

Object	Trans. X (mm)	Trans. Y (mm)	Trans.X (mm)	Rot. X (deg.)	Rot. Y (deg.)	Rot. Z (deg.)
Real Object B in Dark Env.	0.063	0.047	0.044	0.142	0.003	0.241
Real Object B in Bright Env.	0.423	0.270	0.516	0.242	0.023	0.232
Real Object C in Dark Env.	0.175	0.140	0.505	0.080	0.180	0.630
Real Object C in Bright Env.	0.337	0.043	0.138	0.195	0.148	0.107
Real Object D in Dark Env	0.128	0.238	0.081	0.004	0.128	0.120
Real Object D in Bright Env	0.155	0.264	0.571	0.256	0.452	0.220

pose and the desired pose is given by $\Delta r = (-3\text{ mm}, -3\text{ mm}, -3\text{ mm}, -3\text{ deg.}, -3\text{ deg.}, -3\text{ deg.})$.

Fig.19 shows the result of the positioning experiments using Real Object B, C, and D under two types of unseen lighting conditions: dark environment and bright environment. The positioning accuracy of each object is shown in Table 6. The network positioned Real Object B, C, and D under both dark and bright environments under 0.571 mm

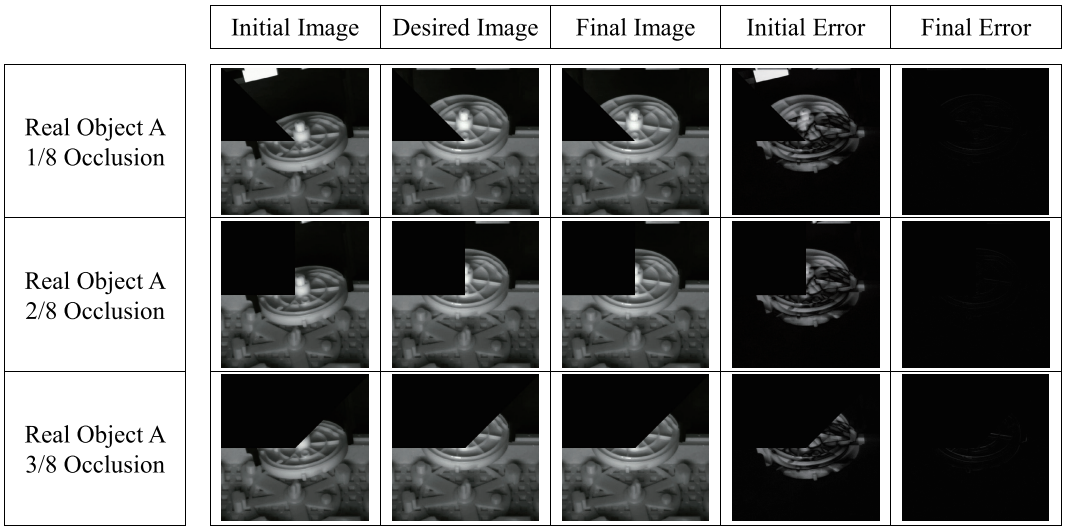

FIGURE 20. Experimental result of positioning using real object A under seen lighting condition. The network positioned the object with occlusion.

TABLE 7. Positioning accuracy under occlusion.

Object	Trans. X (mm)	Trans. Y (mm)	Trans.X (mm)	Rot. X (deg.)	Rot. Y (deg.)	Rot. Z (deg.)
Real Object A with 1/8 occlusion	0.027	0.145	0.059	0.110	0.062	0.049
Real Object A with 2/8 occlusion	0.120	0.073	0.210	0.067	0.003	0.102
Real Object A with 3/8 occlusion	0.500	0.528	1.178	0.823	0.439	0.203

error in translation and 0.630 deg. in rotation. As for Real Object E, the network could not position the object under unseen lighting conditions. As discussed in Section IV-C-(2), the effect of the differences in reflectance and transparency of Real Object E could not be completely removed by the generalization of the proposed network. The positioning accuracy can be improved by increasing the objects with different resin materials included in the training dataset.

4) POSITIONING WITH OCCLUSIONS

To evaluate the generalization to occlusion, we conducted visual servoing using Real Object A with occlusion, under the same lighting condition as the training dataset. The captured images are occluded by image processing. The initial pose and the desired pose is $r = (-430.93$ mm, 59.17 mm, -1066.53 mm, 170 deg., -10 deg., -190 deg.) and $r = (-420.93$ mm, 69.17 mm, -1056.53 mm, 180 deg., 0 deg., -180 deg.), respectively, where the difference between initial pose and the desired pose is given by $\Delta r = (-10$ mm, -10 mm, -10 mm, -10 deg., -10 deg., -10 deg.).

Fig.20 shows the result of the positioning experiments using Real Object A with occlusion. The first, second, and third row show the experimental result when 1/8, 2/8, and 3/8 of the captured images are occluded, respectively. The positioning accuracy of the object with each occlusion is shown in Table 7. The network positioned Real Object A with 1/8 occlusion and 2/8 occlusion with the error of less than 0.210 mm in translation and 0.110 deg. in rotation. The positioning accuracy of Real Object A with 3/8 occlusion is under 1.178 mm error in translation and 0.823 deg. error in rotation. Note that the network could not position Real Object A with 4/8 occlusion where half of the image is occluded. We can conclude that the restoration process of the occluded image pixel is one of a solution to overcome the limitation, which is left for future work.

V. CONCLUSION

We presented a CNN based visual servoing scheme for an eye-to-hand manipulator. We proposed DEFINet that estimates a relative pose between the desired and current end-effector from the desired and current images captured by a camera. DEFINet regresses a relative pose from a difference of target image feature and current image feature, which results in efficient and high accuracy positioning. The dataset is generated from a small amount of sample data collected by operating a manipulator.

Numerical simulation shows that DEFINet is able to position an object from a large displacement between the initial and desired pose that the network had never learned. Furthermore, we compared the positioning accuracy of DEFINet inside and outside of the task space with other networks and direct IBVS. We confirmed that the positioning accuracy and the convergence domain is larger than that of the other networks and direct IBVS

We demonstrated the positioning of an object from a large displacement of the initial pose and the desired pose in a real environment. We confirmed that the proposed method is also effective in a real environment. The proposed method achieved micro order accuracy using seen lighting conditions and objects. We further confirmed that DEFINet is robust against unseen lighting conditions, unseen objects, and occlusions.

The future works include increasing the positioning accuracy of the unseen objects under unseen lighting conditions in a larger task space with large occlusion by using a larger dataset and considering Sim2Real methods.

REFERENCES

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
[2] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
[3] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative end-effector control using Cartesian position based visual servoing," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 684–696, Oct. 1996.
[4] R. Kelly, "Robust asymptotically stable visual servoing of planar robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 759–766, Oct. 1996.
[5] K. Hashimoto, T. Ebine, and H. Kimura, "Visual servoing with hand-eye manipulator-optimal control approach," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 766–774, Oct. 1996.
[6] C. Kingkan, S. Ito, S. Arai, T. Nammoto, and K. Hashimoto, "Model-based virtual visual servoing with point cloud data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 5549–5555.
[7] R. Vidal, O. Shakernia, and S. Sastry, "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2003, pp. 584–589.
[8] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 87–100, Feb. 2007.
[9] X. Zhang, Y. Fang, and X. Liu, "Motion-estimation-based visual servoing of nonholonomic mobile robots," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1167–1175, Dec. 2011.
[10] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 331–340, Apr. 2008.
[11] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 971–976.
[12] H. Zhang and J. P. Ostrowski, "Visual servoing with dynamics: Control of an unmanned blimp," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1999, pp. 618–623.
[13] R. Ginhoux, J. A. Gangloff, M. F. de Mathelin, L. Soler, M. M. A. Sanchez, and J. Marescaux, "Beating heart tracking in robotic surgery using 500 hz visual servoing, model predictive control and an adaptive observer," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Apr. 2004, pp. 274–279.
[14] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation," *IEEE Eng. Med. Biol. Mag.*, vol. 16, no. 1, pp. 40–45, Jan. 1997.
[15] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, "A state of the art in structured light patterns for surface profilometry," *Pattern Recognit.*, vol. 43, no. 8, pp. 2666–2680, Aug. 2010.
[16] N. Chiba, S. Arai, and K. Hashimoto, "Feedback projection for 3D measurements under complex lighting conditions," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 4649–4656.
[17] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.
[18] D. Liu, S. Arai, Z. Feng, J. Miao, Y. Xu, J. Kinugawa, and K. Kosuge, "2D object localization based point pair feature for pose estimation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 1119–1124.
[19] D. Liu, S. Arai, J. Miao, J. Kinugawa, Z. Wang, and K. Kosuge, "Point pair feature-based pose estimation with multiple edge appearance models (PPF-MEAM) for robotic bin picking," *Sensors*, vol. 18, no. 8, p. 2719, Aug. 2018.
[20] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 828–834, Aug. 2011.
[21] K. Deguchi, "A direct interpretation of dynamic images with camera and object motions for vision guided robot control," *Int. J. Comput. Vis.*, vol. 37, no. 1, pp. 7–20, 2000.
[22] X. Zhong, X. Zhong, H. Hu, and X. Peng, "A nonparametric-learning visual servoing framework for robot manipulator in unstructured environments," *Neurocomputing*, vol. 437, pp. 206–217, May 2021.
[23] J. Qu, F. Zhang, Y. Tang, and Y. Fu, "Dynamic visual tracking for robot manipulator using adaptive fading Kalman filter," *IEEE Access*, vol. 8, pp. 35113–35126, 2020.
[24] M. Kang, H. Chen, and J. Dong, "Adaptive visual servoing with an uncalibrated camera using extreme learning machine and Q-leaning," *Neurocomputing*, vol. 402, pp. 384–394, Aug. 2020.
[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[27] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 512–519.
[28] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.
[29] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3364–3372.
[30] S. Mahendran, H. Ali, and R. Vidal, "3D pose regression using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 494–495.
[31] A. Saxena, H. Pandya, G. Kumar, A. Gaud, and K. M. Krishna, "Exploring convolutional networks for end-to-end visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3817–3823.
[32] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3307–3314.
[33] C. Yu, Z. Cai, H. Pham, and Q.-C. Pham, "Siamese convolutional neural network for sub-millimeter-accurate camera pose estimation and visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 935–941.
[34] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.
[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[36] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," in *Proc. Adv. Neural. Inf. Process. Syst.*, 1994, pp. 737–744.
[37] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, 2015.
[38] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3758–3765.
[39] 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.
[40] T. Lozano-Perez, J. L. Jones, E. Mazer, and P. A. O'Donnell, "Task-level planning of pick-and-place robot motions," *Computer*, vol. 22, no. 3, pp. 21–29, Mar. 1989.
[41] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400
[42] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: http://arxiv.org/abs/1212.5701
[43] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io
[44] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
[45] H. Sutanto, R. Sharma, and V. Varma, "Image based autodocking without calibration," in *Proc. Int. Conf. Robot. Autom.*, vol. 2, 1997, pp. 974–979.
[46] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: A generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.

FUYUKI TOKUDA (Student Member, IEEE) received the B.S. degree in engineering from the Nagoya Institute of Technology, Nagoya, Japan, in 2017, and the M.S. degree in engineering from Tohoku University, Sendai, Japan, in 2019, where he is currently pursuing the Ph.D. degree in engineering with the Department of Robotics.

His research interests include the development of visual feedback control using visual servoing techniques and deep learning for bin-picking, assembling, and aligning tasks of industrial objects.

Mr. Tokuda received the Outstanding Presentation Award from the SICE Tohoku 55th Anniversary Conference, in 2020; the Research Fellowship from the Tohoku University Graduate Program for Integration of Mechanical Systems, in 2018; and the Research Fellowship from the Japan Society for the Promotion of Science (JSPS), in 2021.

SHOGO ARAI (Member, IEEE) received the B.S. degree in aerospace engineering and the M.S. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2005, 2007, and 2010, respectively.

From 2010 to 2016, he was an Assistant Professor with the Intelligent Control Systems Laboratory, Tohoku University. He joined the System Robotics Laboratory, Department of Robotics, Tohoku University, as an Associate Professor, in 2016, where he is currently an Associate Professor. His research interests include robot vision, machine vision, 3D measurement, production robotics, networked control systems, and multi-agent systems.

Dr. Arai received the Best Paper Award from FA Foundation, in 2019; the 32th Best Paper Award from the Robotics Society of Japan, in 2019; the Certificate of Merit for Best Presentation from the Japan Society of Mechanical Engineers, in 2019; the Excellent Paper Award from the Institute of Systems from Control and Information Engineers, in 2010; the Best Paper Award Finalist at IEEE International Conference on Mechatronics and Automation, in 2012; the SI2019 Excellent Presentation Award from the Society of Instrument and Control Engineers, in 2019; the SI2018 Excellent Presentation Award from the Society of Instrument and Control Engineers, in 2018; the SI2017 Excellent Presentation Award from the Society of Instrument and Control Engineers, in 2017; and the Graduate School Research Award from the Society of Automotive Engineers of Japan, Inc., in 2007.

KAZUHIRO KOSUGE (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in control engineering from the Tokyo Institute of Technology, in 1978, 1980, and 1988, respectively.

After having served as a Research and Development Staff with the Production Engineering Department, Nippon Denso Company, Ltd.; a Research Associate at the Tokyo Institute of Technology; and an Associate Professor at Nagoya University. He joined Tohoku University as a Professor, in 1995, and served as a Distinguished Professor, from 2018 to 2021. He is currently the Director of the Center for Transformative AI and Robotics and a Specially Appointed Professor with the Graduate School of Engineering, Tohoku University, Japan. He recently joined The University of Hong Kong as the Chair Professor of the Department of Electrical and Electronic Engineering.

Dr. Kosuge received the Medal of Honor and the Medal with Purple Ribbon from the Government of Japan, in 2018—a national honor in recognition of his prominent contributions to academic and industrial advancements. He also received the IEEE RAS George Saridis Leadership Award in Robotics and Automation, in 2021, for his exceptional vision of innovative research and outstanding leadership in the robotics and automation community through technical activity management. He is a JSME Fellow, a SICE Fellow, a RSJ Fellow, a JSAE Fellow, and a member of the Engineering Academy of Japan. He was the President of the IEEE Robotics and Automation Society, from 2010 to 2011; the IEEE Division X Director, from 2015 to 2016; and the IEEE Vice President of Technical Activities, in 2020.