Proceedings of 2021 IEEE
International Conference on Mechatronics and Automation
August 8 - 11, Takamatsu, Japan

Visual Feedback Control and Transfer Learning-Based CNN for a Pick and Place Robot on a Sliding Rail

Fusaomi Nagata, Kohei Miki
Graduate School of Science & Engineering,
Sanyo-Onoda City University
Sanyo-Onoda, Japan
nagata@rs.socu.ac.jp

Keigo Watanabe
Graduate School of Natural Science and Technology
Okayama University
Okayama, Japan
watanabe@sys.okayama-u.ac.jp

Maki K. Habib
Mechanical Engineering Department
School of Sciences & Engineering
American University in Cairo, Cairo, Egypt
maki@aucegypt.edu

***Abstract*—Among the various types of deep neural networks (DNNs), convolutional neural networks (CNNs) have ingenious structures and are widely used for image recognition and/or defect inspection. The authors already developed a design, training and test tool for CNNs and support vector machines (SVMs) to support defect detection of various kinds of manufactured products, while showing the effectiveness and the userfriendliness through classification experiments using images of actual products. The tool further enables to view where the most activated area in each classified image is. Besides the tool, a desktop-sized pick and place (P&P) robot was also proposed while implementing a pixel-based visual feedback (VF) controller to autonomously reach target objects. In addition, a CNN designed based on transfer learning concept was developed to estimate objects' orientations. In this paper, a sliding rail is considered to allow the articulated robot to move around in a wider working range. The VF controller is extended to utilize the sliding rail. The usefulness and userfriendliness of the robot system using the sliding rail is confirmed through P&P experiments of randomly put objects on a table.**

***Index Terms*—convolutional neural network, transfer learning, pick and place, robot**

I. Introduction

Taryudi and Wang presented an application of a stereo vision system to sense the position and orientation of objects. By using the system, a robot arm equipped with an end effector could successfully pick up the objects and place them to desired positions in the workspace [1]. Han et al. analytically showed that conventional greedy approaches for P&P tasks did not ensure the time optimality. To address the shortcoming of the time optimality in applying classical solutions such as greedy approaches to practical P&P tasks, they developed algorithms that computed optimal object picking sequences for a predetermined finite horizon. The effectiveness was demonstrated by using Delta type and SCARA type robots [2].

For example, if an industrial robot is used to automate P&P tasks of small plastic articles, information of each object's position and orientation is essential for successful results. Simple image processing techniques such as binarization and segmentation may allow to recognize objects in an image and extract positions, however, that of orientation is not easy due to the variety in shape. Dolezel et al. reported the result concerning the fast 2D positioning of multiple complex objects for P&P applications using CNNs. The CNN generated schematic images representing the positions of objects with gradient circles of various colors [3]. Slavov introduced an object size estimation method by means of an electric gripper, in which neural network and machine learning approaches are applied for industrial robots' P&P sorting tasks [4].

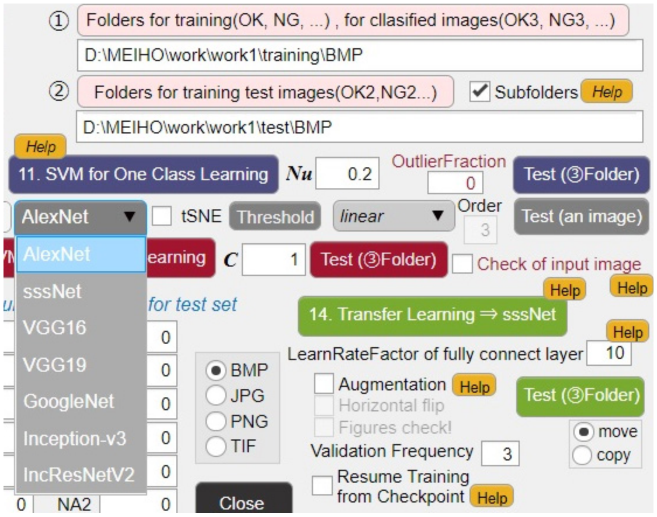
The authors presented a CNN&SVM design, training and test tool to detect defects included in images of small plastic articles. The usefulness and userfriendliness of the tool were confirmed through several design, training and evaluation experiments of CNNs and SVMs [5]. Moreover, the tool enabled to easily design powerful CNN models using transfer learning techniques. Then, a desktop-sized P&P robot with a pixel-based VF controller and a transfer learning-based CNN model was introduced [6]. The pixel-based VF control enabled to omit the complicated calibration between image and robot coordinate systems. Also the transfer learning-based CNN allowed the robot to estimate the orientation of target objects on a working table.

In this paper, a sliding rail is considered to allow the articulated robot to move around in wider working range. The VF controller is extended to utilize the sliding rail. The usefulness and userfriendliness of the robot system using the sliding rail are confirmed through P&P experiments of randomly put objects on a table.

II. Design, Training and Test Tool for Transfer-Learning Based CNNs

The authors have already proposed application to easily design and train CNNs and SVMs for visual inspection. In

978-1-6654-4098-1/21/$31.00 ©2021 IEEE



① Folders for training(OK, NG, ...) , for cllasified images(OK3, NG3, ...)

D:\MEIHO\work\work1\training\BMP

② Folders for training test images(OK2,NG2...) ☑ Subfolders Help

D:\MEIHO\work\work1\test\BMP

Help

11. SVM for One Class Learning *Nu* 0.2 OutlierFraction 0 Test (③Folder)

AlexNet ▼ □ tSNE Threshold linear ▼ Order 3 Test (an image)

AlexNet / sssNet / VGG16 / VGG19 / GoogleNet / Inception-v3 / IncResNetV2

earning *C* 1 Test (③Folder) □ Check of input image

for test set Help Help

14. Transfer Learning ⇒ sssNet Help

LearnRateFactor of fully connect layer 10

● BMP ○ JPG ○ PNG ○ TIF

□ Augmentation Help

□ Horizontal flip

□ Figures check!

Test (③Folder)

● move ○ copy

Validation Frequency 3

□ Resume Training from Checkpoint Help

0 0 0 0 0 0 0 NA2 0

Close

Fig. 1. A part of the main dailogue developed for efficiently designing and training transfer learning-based CNNs.

the training of CNN, pre-training using randomly initialize weights or additional (successive) training with already trained weights can be selected. As for SVM, one-class unsupervised learning and two class supervised learning can be selectively available. Besides, favorite CNNs such as AlexNet and GoogLeNet can be used for feature extractors, also kernel functions such as Gaussian and polynomial can be selected.

The tool provides another function to rapidly construct original CNNs using transfer learning concept. Figure 1 shows the part in the main dialogue developed for efficiently designing and training transfer learning-based CNNs. For example, the following main items can be set for the parameter conditions of transfer learning through the developed dialogue.

- Folders to put training images and test ones.
- CNNs used for the base of transfer learning such as AlexNet, VGG16, VGG19, GoogleNet, Inception-V3, IncResNetV2, NASNet-Mobile, NASNet-Large, EfficientNet-B0 and ResNet-50.
- Training parameters such as mini batch size, desired accuracy and loss, different learning rates for convolution layers and fully connected (FC) layers, max epochs, and so on.

Moreover, convolutional auto encoder (CAE) based on SegNet [7] can be designed to visualize defect parts which actually affect classification results. The user interface shown in Fig. 1 was developed on MATLAB system. Statistics and Machine Learning Toolbox, Deep Learning Toolbox, Parallel Computing Toolbox for GPU were optionally installed.

III. Design of Transfer Learning-Based CNN

A software was already developed to efficiently augment training images [5]. The number of images could be increased considering typical twelve orientations such as 0°, 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, and 165°. Examples of the category of 45° are shown in Figure 2. The resolution and channel are 200×200 and 1, respectively.

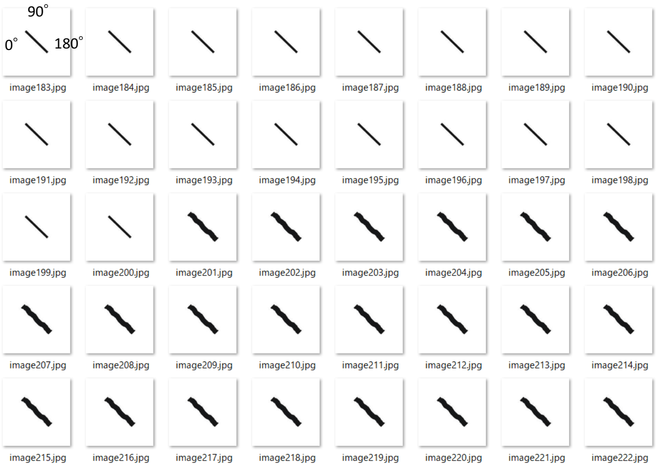


Fig. 2. Training image samples including the orientation of 45°.

A CNN transferred from AlexNet is designed to learn the orientation features included in images as shown in Figs. 2. The structure of the original AlexNet consisting of 25 layers is illustrated in Figure 3, which can classify input images into one of 1,000 categories. In order that the CNN could have an ability to classify input images into 12 categories of angles, the FC layers were replaced as shown in Fig. 4 before conducting transfer learning. 6889 images consisting of 12 categories were used for the transfer learning. As for training parameters, the mini batch size was set to 50. Iteration is the number of mini batches needed to complete one epoch, so that one epoch is composed of 6889/50≒137 iterations. Desired accuracy and loss were set to 1 and 0, respectively. Besides, different learning rates for convolutional layers and FC ones were set to 0.0001 and 0.001, respectively. It is effective for faster and more stable convergence to give the learning rate in convolutional layers smaller value than that of FC ones.

If nth training image is given to the CNN, then the softmax layer yields the probability $p_{ni}(i = 1, 2, \cdots, 12)$ as each score for twelve types of categories, written by

$$p_{ni} = \frac{e^{y_{ni}}}{\sum_{k=1}^{12} e^{y_{nk}}} \tag{1}$$

where $\boldsymbol{y}_n = [y_{n1}\ y_{n2} \cdots y_{n12}]^T$ is the output from the last FC layer corresponding to the nth image. The transferred CNN is trained using the back propagation algorithm using the loss function called cross entropy as

$$\bar{E} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{12} t_{nk}\log(y_{nk}) \tag{2}$$

where $\boldsymbol{t}_n = [t_{n1}\ \ t_{n2}\ \ \cdots\ \ t_{n12}]^T$ is the nth desired output vector for classification, i.e., only one element in $\boldsymbol{t}_n$ has 1, remained elements have 0. N is the number of images in the training data set.

The training to optimize the transfer learning-based CNN was processed using a single PC with a Core i7 CPU and a GPU (NVIDIA GeForce GTX 1060, 6GB) until both the

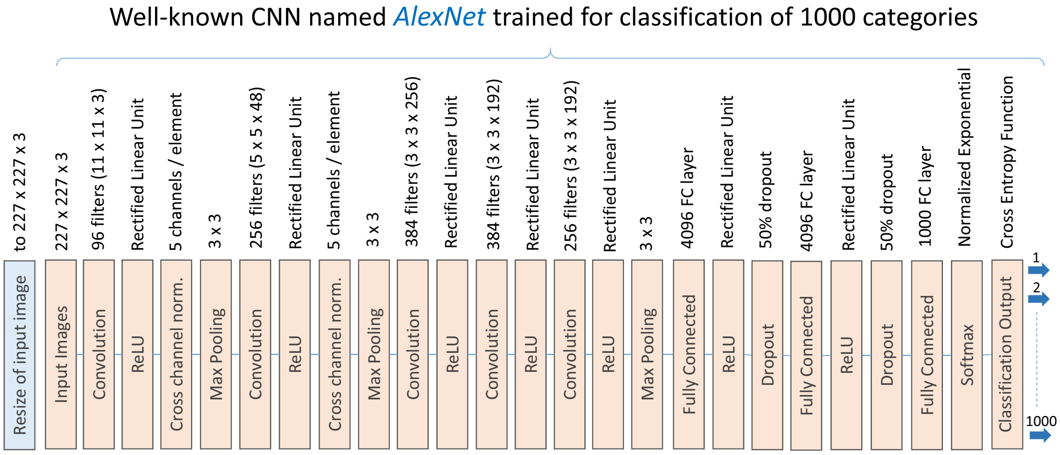


Fig. 3. Network structure of well-known CNN named AlexNet which can classify input images into one of 1000 kinds of categories.

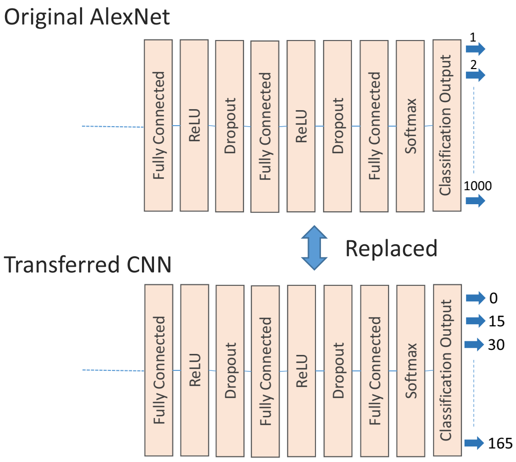


Fig. 4. Replacement of the last three layers for dealing with target classification task, i.e., 12 categories of angles.

training accuracy and loss converged to desired values. Note that the accuracy A_c at each iteration is obtained by

$$A_c = \frac{50 - N_m}{50} \times 100 \tag{3}$$

where 50 is the mini batch size, i.e., the number of sampled images processed during one iteration, N_m is the number of mis-classified images within the 50 sampled images. The training was stopped since both the accuracy and loss had not become better during 10 consecutive iterations or more. It actually took about 40 minutes until the training was stopped. Note that this training could be completed within one epoch by giving the different learning rates as 0.0001 and 0.001 in the convolutional layers and the FC one, respectively. A new CNN model constructed by transfer learning of AlexNet is presented to recognize the orientation of objects through the process explained above.

After the training, the generalization ability of the CNN was evaluated using more than 39 test images including small plastic articles and hand tools which had not been included in the training data set [6]. Figure 5 shows examples of classified images and those results, i.e., the angles written in the images are the outputs from the CNN. It took about a few ten milliseconds to classify one image. It is observed that the CNN has a promising generalization ability that can recognize the orientations of objects in the images. However, some visual inconsistencies, e.g., between "test7.jpg" (75°) and "test12.jpg" (60°); "test.jpg" (150°) and "test3.jpg" (150°) are observed. As can be clearly seen, some images are not complete square. Therefore, the main cause of these results seems to be the conversion of resolution before classification. The resolution of images given to the input layer was forced to be 227×227×3 specified by the input layer of the AlexNet, which brought out some undesirable deformation of images and the resultant ambiguities in classification.

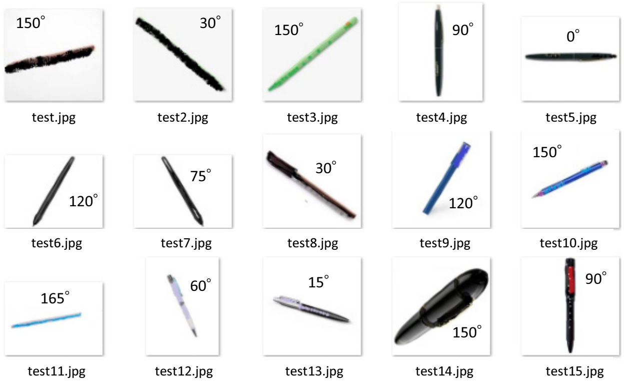


Fig. 5. Classification of test images using the transfer learning-based CNN.

IV. Pick and Place Experiment

A. Without VF Controller

An actual P&P experiment is conducted using a small articulated robot DOBOT Magician provided by Shenzhen Yuejiang Technology Co., Ltd. The experimental setup using a WEB camera is shown in Fig. 6. Position $[X\ Y\ Z]^T$ and yaw angle R of the gripper in the robot coordinate system can be controlled by an API function SetPTPCmd(X_d, Y_d, Z_d, R_d),

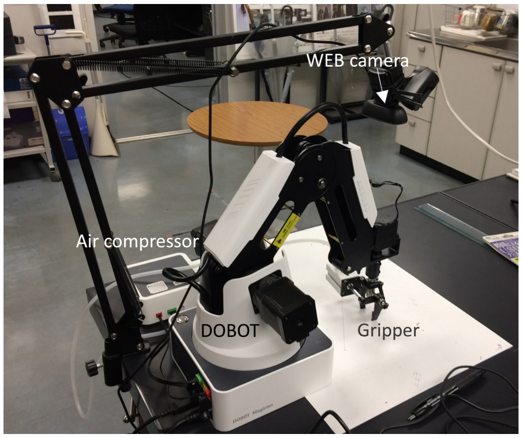
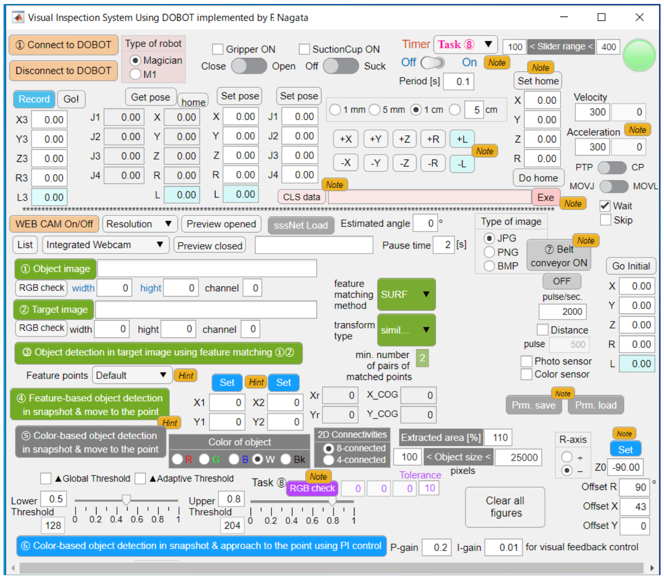


Fig. 6. Experimental setup based on an articulated robot using a WEB camera.


Visual Inspection System Using DOBOT implemented by F. Nagata

Type of robot: ● Magician ○ M1
□ Gripper ON Close / Open
□ SuctionCup ON Off / Suck
Timer Task ⑧ Off / On
Period [s] 0.1
100 < Slider range < 400
X3 0.00 Y3 0.00 Z3 0.00 R3 0.00 L3 0.00
J1 0.00 J2 0.00 J3 0.00 J4 0.00
X 0.00 Y 0.00 Z 0.00 R 0.00 L 0.00
X 0.00 Y 0.00 Z 0.00 R 0.00 L 0.00
J1 0.00 J2 0.00 J3 0.00 J4 0.00
○ 1 mm ○ 5 mm ● 1 cm ○ 5 cm
X 0.00 Y 0.00 Z 0.00 R 0.00
Velocity 300 0
Acceleration 300 0
PTP / CP
MOVJ / MOVL
☑ Wait
□ Skip
Resolution
Integrated Webcam
Estimated angle 0 °
Pause time 2 [s]
Type of image: ● JPG ○ PNG ○ BMP
RGB check width 0 hight 0 channel 0
RGB check width 0 hight 0 channel 0
feature matching method SURF
transform type simil...
min. number of pairs of matched points 2
Feature points Default
X1 0 X2 0
Y1 0 Y2 0
Xr 0 X_COG 0
Yr 0 Y_COG 0
OFF
pulse/sec. 2000
□ Distance
pulse 500
□ Photo sensor
□ Color sensor
X 0.00 Y 0.00 Z 0.00 R 0.00 L 0.00
Color of object: ○ R ○ G ○ B ● W ○ Bk
2D Connectivities: ● 8-connected ○ 4-connected
Extracted area [%] 110
100 < Object size < 25000 pixels
Tolerance 0 0 0 10
R-axis: ○ + ● –
Z0 -90.00
Offset R 90 °
Offset X 43
Offset Y 0
□ ▲Global Threshold □ ▲Adaptive Threshold
Lower Threshold 0.5 128
Upper Threshold 0.8 204
P-gain 0.2 I-gain 0.01 for visual feedback control

Fig. 7. Developed control dialogue for DOBOT and sliding rail.

where subscript d means desired values. Note that the yaw angle R is regulated to make it easy to pick a target object according to its orientation. Figure 7 shows the developed control dialogue for the robot. P&P task while recognizing the orientations of target objects can be executed through the dialogue. Figure 8 shows two flowcharts without and with the proposed VF controller, both of which are implemented in a timer interrupt routine.

In the timer interrupt, first of all, a snapshot of 1600×1200 resolution is captured. After binarized into black and white, a connected component with the largest area is found as a target object and then the COG position $[I_x \ I_y]^T$ $(1 \le I_x \le 1600, 1 \le I_y \le 1200)$ in image coordinate system is extracted by

$$I_x = \frac{\sum_{x=1}^{1600} \sum_{y=1}^{1200} x f(x,y)}{S} \tag{4}$$

$$I_y = \frac{\sum_{x=1}^{1600} \sum_{y=1}^{1200} y f(x,y)}{S} \tag{5}$$

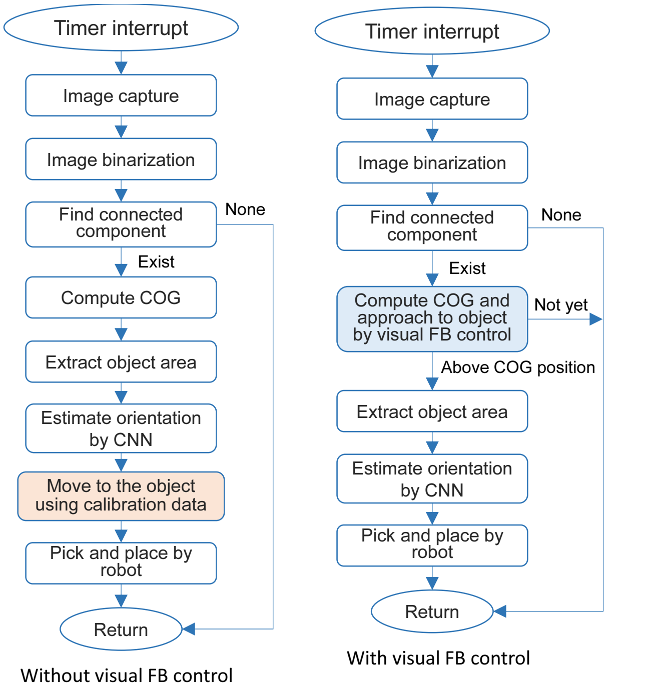


Fig. 8. Flowcharts to realize P&P task using transfer learning-based CNN, in which left and right figures show the process flow diagrams without and with a VF controller, respectively.

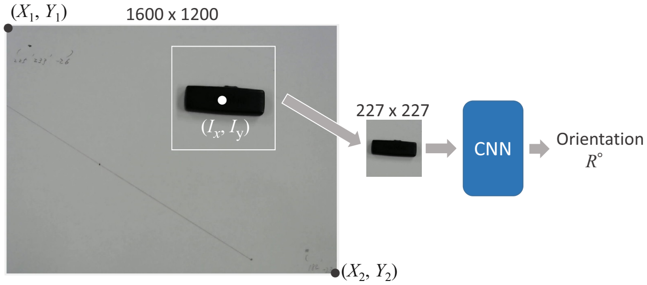


Fig. 9. Procedure to extract the orientation of a workpiece from a captured image.

where the position (x, y) are the variables of column and row in an image. $f(x, y)$ is the binary value, i.e., 1 or 0, at the coordinate (x, y). S is the area of an identified and extracted object, which is obtained by counting the number of pixels with a value of 1 forming the object. Consequently, desired position $[x_d \ y_d]^T$ in robot coordinate system to move the gripper to the COG position can be obtained by

$$x_d = X_1 + I_x \frac{X_2 - X_1}{1600} \tag{6}$$

$$y_d = Y_1 + I_y \frac{Y_2 - Y_1}{1200} \tag{7}$$

where $[X_1 \ Y_1]^T$ and $[X_2 \ Y_2]^T$ in robot coordinate system are the positions of left upper and right bottom of the snapshot as shown in Fig. 9, i.e., they are corresponding to pixels of $(0, 0)$ and $(1600, 1200)$, respectively. The part of the connected component is further cropped centering the COG from the original snapshot as shown in Fig. 9. The cropped image

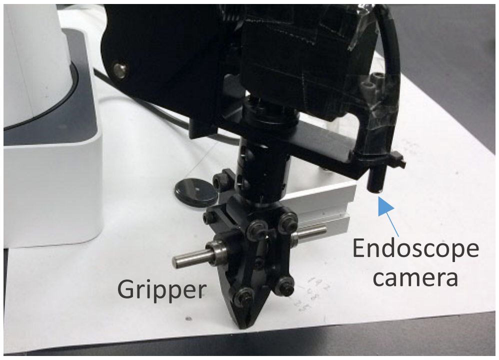


Fig. 10. Experimental setup based on an articulated robot DOBOT Magician with an endoscope camera.

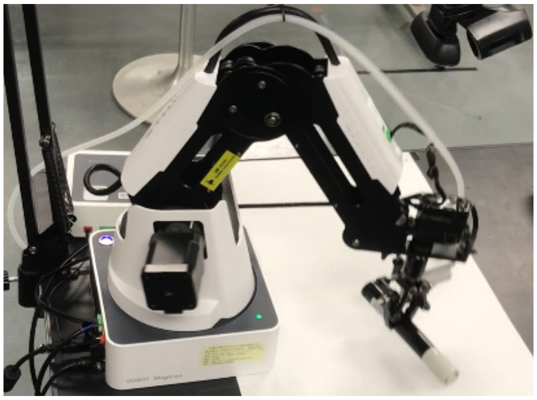


Fig. 11. Experimental scene just after picking.

is resized into 227×227 and given to the input layer of the transfer learning-based CNN designed in the previous section. Finally, the orientation of the object can be estimated by the CNN, which is used for the desired yaw angle R_d so that the robot can successfully grasp the object with a long-axis shape.

Figure 11 shows the scene just after picking, in which the gripper cloud well identify the COG position of the black part and the orientation, then pick the point up.

B. With VF Controller

The WEB camera and its initial configuration is not needed in VF control approach, which is different from that in the previous subsection. Instead of the WEB camera, a lightweight endoscope camera is attached close to the gripper as shown in Fig. 10. Manipulated variable $\boldsymbol{v}(k) = [v_x(k)\ v_y(k)]^T$ for VF is generated by proportional and integral actions given by

$$\boldsymbol{v}(k) = K_p \boldsymbol{e}(k) + K_i \sum_{n=1}^{k} \boldsymbol{e}(n) \tag{8}$$

where k is the discrete time. K_p and K_i are the gains for proportional and integral actions, respectively. $\boldsymbol{e}(k) = [e_x(k)\ e_y(k)]^T$ is the error vector in image coordinate system measured by

$$\boldsymbol{e}(k) = \boldsymbol{x}_d - \boldsymbol{I}(k) \tag{9}$$

where $\boldsymbol{x}_d = [\frac{x_r}{2}\ \frac{x_r}{2}]^T$ and $\boldsymbol{I}(k) = [I_x(k)\ I_y(k)]^T$ are the desired position and the measured object's COG position in image coordinate system, respectively. $[x_r\ y_r]$ is the resolution of captured image, so that the desired position $\boldsymbol{x}_d = [\frac{x_r}{2}\ \frac{y_r}{2}]^T$ is the center position of the image. The VF control functions to automatically move the gripper to the desired position just above the COG of the detected object.

It was confirmed from a similar experiment as shown in Fig. 11 that the VF controller enables the robot to execute the P&P task without setting $[X_1\ Y_1]^T$, $[X_2\ Y_2]^T$ and without using Eqs. (6) and (7).

V. Extended Pick and Place Robot

As introduced in the previous sections, the authors already proposed a P&P robot in which a VF controller and an orientation estimator based on AI were implemented, so that the robot could obtain a skillful picking ability. However, one of the negative weak points of such industrial robots is to be fixed on a hard floor or table, so that they cannot move even if needed. To cope with this need, a fusion with a sliding rail is considered to allow the robot to move in a wider space. Figure 12 shows the extended picking robot system with a sliding rail which can expand its functioning space. The effective motion range is 1000 mm in the Y-direction. Note that the WEB camera to overall monitor the working table is not needed due to the utilization of the sliding rail.

In operation of the presented P&P task, the robot basically moves around from side to side on the rail using the relative control input $[0, 0, 0, 0, L_d(k)]$ in order to find any objects using an endoscope. Note that L_d $(0 \leq L_d \leq 1000$ mm) is the desired position on the sliding rail. Figure 13 shows the flowchart of the robot system with the sliding rail. As can be seen, one block is added at the head of the flowchart shown in Fig. 8. This block allows the robot to repeatedly move from side to side until some objects are detected. If some objects with a designated ranged area, e.g., from 100 to 2000 pixels, are detected by using the image processing, then the robot executes pick motion by giving the control input $(X_d, 0, Z_d, R_d, 0)$ and place motion by giving $(X_f, Y_f, Z_f, R_f, L_f)$ one by one in descending order of the number of pixels, where $(X_d, 0, Z_d, R_d, 0)$ is the desired position for picking and $(X_f, Y_f, Z_f, R_f, L_f)$ is that for a final place set in advance.

P&P experiments as shown in Fig. 12 were conducted to check the ability of the robot on the sliding rail. If several stick-type objects were randomly put on the wide table, the robot could detect them, then pick them and place to a desired position one by one. It was confirmed that the flexibility and the ability of wide range could be enhanced for better P&P action.

VI. Conclusions

A CNN transferred from AlexNet has been introduced to detect the orientations of objects. Original AlexNet can classify input images into one of 1,000 kinds of objects, on the other hand, the transferred CNN can identify the orientation of an object in images with the resolution of 15 degrees. Also, a VF control technique has been implemented so that the gripper

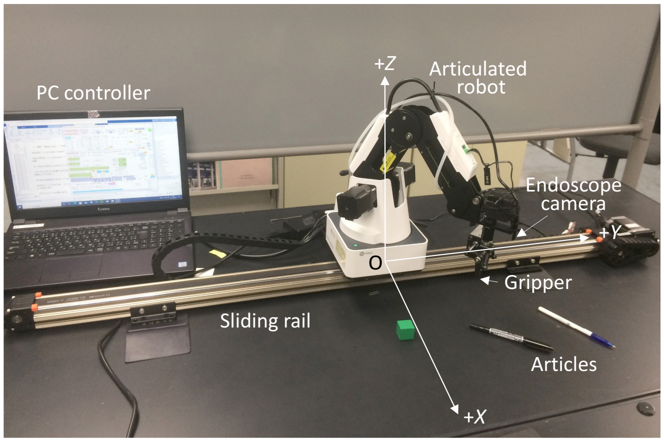


Fig. 12. Extended picking robot using a slider rail.

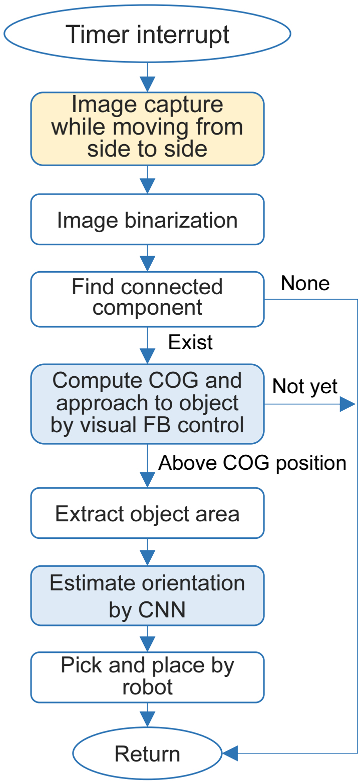


Fig. 13. Flowchart of the robot system with a sliding rail.

of the P&P robot can automatically move to the position above a target object. Due to the VF controller, the complicated calibration between image and robot coordinate systems could be omitted. Moreover, a sliding rail was considered to enable the desktop-sized robot to move around in wider working space. The VF controller was extended to utilize the sliding rail. The usefulness and userfriendliness of the robot system using the sliding rail were confirmed through P&P experiments of randomly put objects on a working table.

In future work, the authors will apply the proposed pixel-based VF controller and the CNN-based orientation detector into a SCARA type robot DOBOT M1 as shown in Fig. 14 to realize more fast and precise P&P motion. The SCARA type robot will be also operated through the controller shown in Fig. 7 by automatically switching dynamic link libraries DobotDll.dll for Magician or DobotDllM1.dll for M1.

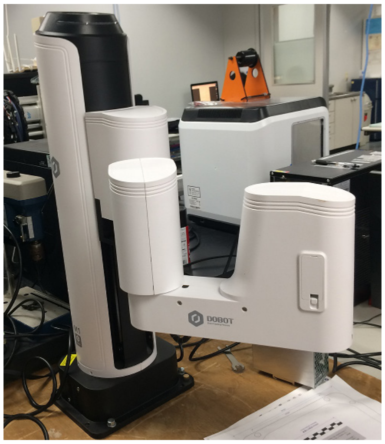


Fig. 14. More fast and precise SCARA type robot named DOBOT M1.

References

[1] Taryudi and M. S. Wang, “3D object pose estimation using stereo vision for object manipulation system,” *Procs. of 2017 International Conference on Applied System Innovation (ICASI)*, pp. 1532–1535, 2017.

[2] S. D. Han, S. W. Feng and J. Yu, “Toward fast and optimal robotic pick-and-place on a moving conveyor,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 446–453, April 2020.

[3] P. Dolezel, D. Stursa and D. Honc, “Rapid 2D positioning of multiple complex objects for pick and place application using convolutional neural network,” *procs. of 24th International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, pp. 213–217, 2020.

[4] D. Slavov, “Object size estimation with industrial robot gripper using neural network and machine learning,” *Procs. of 2020 International Conference Automatics and Informatics (ICAI)*, Varna, Bulgaria, pp. 1–4, 2020.

[5] F. Nagata, K. Tokuno, K. Mitarai, A. Otsuka, T. Ikeda, H. Ochi, K. Watanabe and M. K. Habib, “Defect detection method using deep convolutional neural network, support vector machine and template matching techniques,” *Artificial Life and Robotics*, vol. 24, no. 4, pp. 512–519, 2019.

[6] F. Nagata, K. Miki, A. Otsuka, K. Yoshida, K. Watanabe and M. K. Habib, Pick and place robot using visual feedback control and transfer learning-based CNN, *Procs. of the 2020 IEEE International Conference on Mechatronics and Automation (ICMA 2020)*, pp. 850–855, Beijing, China, October 13–16, 2020.

[7] V. Badrinarayanan, A. Kendall and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.