# Pick and Place Robot Using Visual Feedback Control and Transfer Learning-Based CNN


Fusaomi Nagata, Kohei Miki, Kento Nakashima
Akimasa Otsuka, Takeshi Ikeda, Hiroyuki Ochi
*Graduate School of Science & Engineering,*
*Sanyo-Onoda City University*
Sanyo-Onoda, Japan
{nagata,otsuka_a,t-ikeda,ochi}@rs.socu.ac.jp

Keigo Watanabe
*Department of Intelligent Mechanical Systems*
*Division of Industrial Innovation Sciences*
*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*—Artificial neural network (ANN) which has four or more layers structure is called deep NN (DNN) and it is recognized as a promising machine learning technique. Convolutional neural network (CNN) is widely used and powerful structure for image recognition. It is also known that support vector machine (SVM) has a superior ability for binary classification in spite of only having two layers. The authors already have developed a CNN&SVM design and training tool for defect detection of resin molded articles, while the effectiveness and the validity have been proved through several CNNs design, training and evaluation. The tool further enables to facilitate the design of a CNN model based on transfer learning concept. In this paper, a pick and place robot is introduced while implementing a visual feedback control and a transfer learning-based CNN. The visual feedback control enables to omit the complicated calibration between image and robot coordinate systems, also the transfer learning-based CNN allows the robot to estimate the orientation of target objects. The effectiveness of the system is evaluated through experimental pick and place tests using an articulated robot named DOBOT.**



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


## I. Introduction

Artificial neural network (ANN) which has four or more layers structure is called deep NN (DNN) and is recognized as a promising machine learning technique. Convolutional neural network (CNN) has the most used and powerful structure for image recognition. It is also known that support vector machine (SVM) has a superior ability for binary classification in spite of only two layers. Nagi et al. designed max-pooling convolutional neural networks (MPCNN) for vision-based hand gesture recognition [1]. The MPCNN could classify six kinds of gestures with 96% accuracy and allow mobile robots to perform real-time gesture recognition. Weimer et al. also designed a deep CNN architectures for automated feature extraction in industrial inspection process [2]. The CNN automatically generates features from massive amount of training image data and demonstrates excellent defect detection results with low false alarm rates. Faghih-Roohi et al. presented a different type of deep CNN for automatic detection of rail surface defects [3]. It was concluded that the large CNN model performed a better classification result than the small and medium CNN, although the training required a longer time. Zhou et al. used a CNN to classify the surface defects of steel sheets [4]. The CNN could directly learn better representative features from labeled images of surface defects. Further, Ferguson et al. presented a system to identify casting defects in X-ray images based on the Mask Region-based CNN architecture [5], [6]. It is reported that the proposed system simultaneously performed defect detection and segmentation on input images making it suitable for a range of defect detection tasks. We have developed a CNN&SVM design and training tool for defect detection of resin molded articles and the effectiveness and validity have been proved through several CNNs design, training and evaluation [7–9]. The tool further enables to easily design a CNN model based on transfer learning concept. When industrial robots are applied to pick and place tasks of resin molded articles, information of each object ' s position and orientation is essential. Recognition and extraction of object position in an image are not so difficult if using image processing technique, however, that of orientation is not easy due to the variety in shape.In this paper, a pick and place robot is introduced while implementing a visual feedback control and a transfer learning-based CNN. The visual feedback control enables to omit the complicated calibration between image and robot coordinate systems, also the transfer learning-based CNN allows the robot to estimate the orientation of target objects. The effectiveness of the system is evaluated through experimental pick and place tests

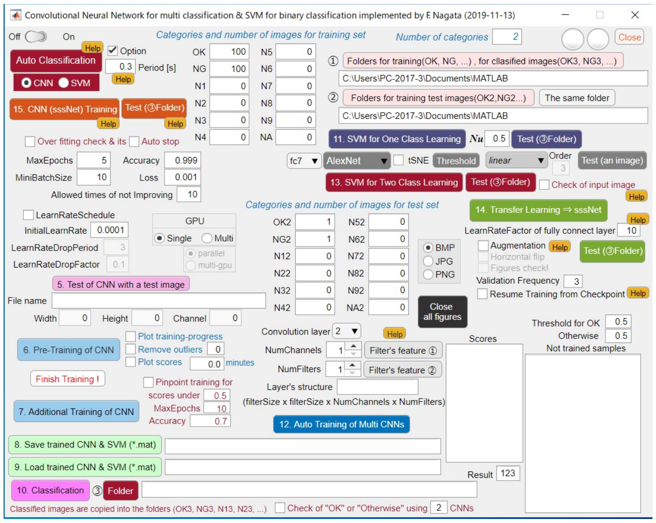


Fig. 1. Main dailogue developed for efficiently designing and training of CNN and SVM.

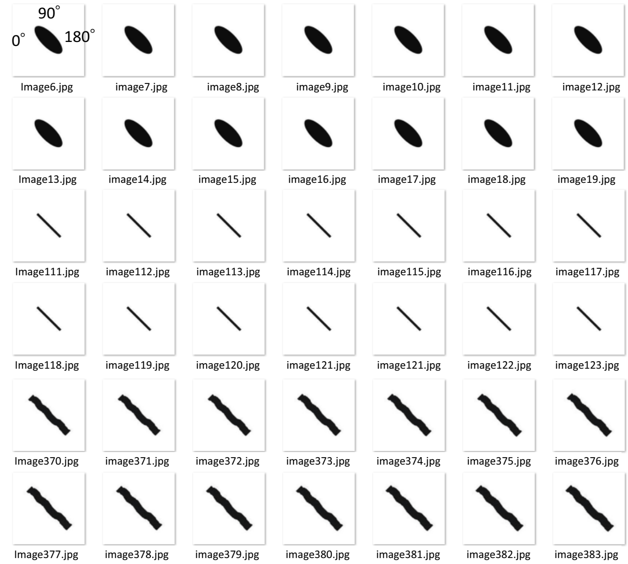


Fig. 2. Examples of training images for the orientation of 45°.

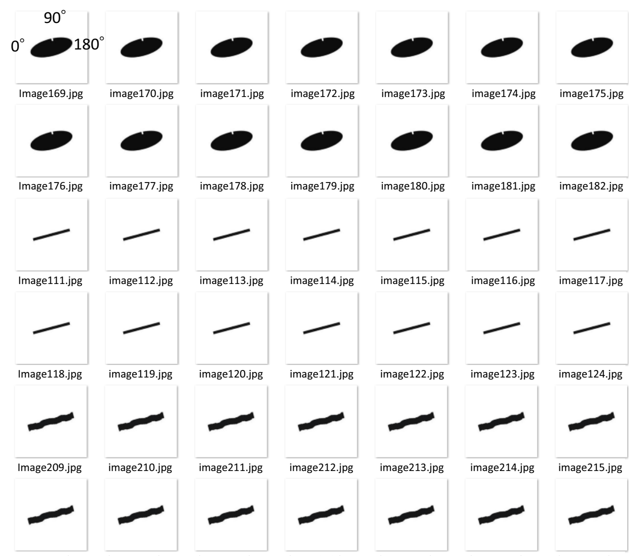


Fig. 3. Examples of training images for the orientation of 165°.

using an articulated robot named DOBOT.

## II. Design & Training Tool for CNN and SVM

Figure 1 shows the main dialogue of the developed CNN&SVM design tool. In training of CNN, pre-training using randomly initialize weights and additional (successive) training with once trained weights can be selected. As for SVM, one-class unsupervised learning and two class supervised learning can be selectively executed. Also, favorite CNN, which is used for a feature extractor, and Kernel function are selected. The tool has another promising function to design original CNNs based on transfer learning. For example, the following main items can be set for the operation of transfer learning through the dialogue.

- Folders for training and test images.
- Base CNNs used for transfer learning such as AlexNet, VGG16, VGG19, GoogleNet and Inception-V3.
- Learning parameters such as max epochs, mini batch size, desired accuracy and loss, learning rates for convolution layers and fully connected layers.

The software shown in Fig. 1 is developed on MATLAB system optionally installed with Neural Network Toolbox, Parallel Computing Toolbox for GPU, Deep Learning Toolbox, Statistics and Machine Learning Toolbox.

## III. Images for Training and Test

Training image generator was already proposed to efficiently augment limited number of training images [7]. By using the generator, images for training are prepared considering typical twelve orientations, i.e., 0°, 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150° and 165°. Figures 2 and 3 show examples of the training images for the categories of 45° and 165°, respectively. The resolution and channel are $200\times200$ and 1, respectively.

## IV. Transfer Learning Based CNN

### A. Design and Training

In this section, a transfer learning based CNN is designed to learn the feature of orientation included in images as shown in Figs. 2 and 3. Figure 4 illustrates the structure of the original AleXnet consisting of 25 layers, which can classify input images into one of 1,000 categories. In order to make the CNN have an ability to classify input images into 12 categories as 0°, 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150° and 165°, the fully connected layers are replaced as shown in Fig. 5 before executing transfer learning. 6,889 images consisting of 12 categories are used for the transfer learning. As for training parameters, mini batch size is given 50. Iteration is the number of mini batches needed to complete one epoch, so that one epoch in this transfer learning is composed of 6,889/50 ≒ 137 iterations. Desired accuracy and loss are set to 1 and 0, respectively. Besides, learning rates of convolutional layers and fully connected ones are set to 0.0001 and 0.001, respectively. It is important for fast and stable convergence in transfer learning to set the learning rate in convolutional layers smaller than that of fully connected layers.

If the $n$th image for training is given to the input layer of the transferred CNN, then the softmax layer produces the probability $p_{ni}(i = 1, 2, \cdots, 12)$ called the score for twelve categories, which is given 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 fully connected layer corresponding to the $n$th image. The transferred CNN is trained based on the back propagation algorithm using the loss function called cross entropy. The cross entropy is calculated by

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

where $\boldsymbol{t}_n = [t_{n1}\ t_{n2}\ \cdots\ t_{n12}]^T$ means the $n$th desired output for twelve categories, i.e., only one element in $\boldsymbol{t}_n$ has 1, remained elements have 0. $N$ is the total number of samples in the training set.

The training was conducted using a single PC with a Core i7 CPU and a GPU (NVIDIA GeForce GTX 1060, 6GB). The training progress is shown in Fig. 6, in which both the training accuracy and loss seem to well converge to desired values. It actually took about 40 minutes until the learning was stopped since both the accuracy and loss had not been improved during 10 consecutive iterations or more. Note that this training could be completed within one epoch by severally giving different learning rates in convolutional layers and fully connected layers. Through the process explained above, an original CNN model acquired by transfer learning of AlexNet, which is the winner of ImageNet LSVRC2012, is presented to recognize the orientation of objects.

### *B. Generalization Ability*

After the training, the generalization ability of the transfer learning based CNN is checked using 15 test images imitating resin molded articles which have not been included in the training data set. Figure 7 shows the photos and their classification results, i.e., the angles shown in the JPEG images are the outputs from the CNN. It is observed from the results that the obtained 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 in Fig. 7 are not complete square. That is the reason why the main

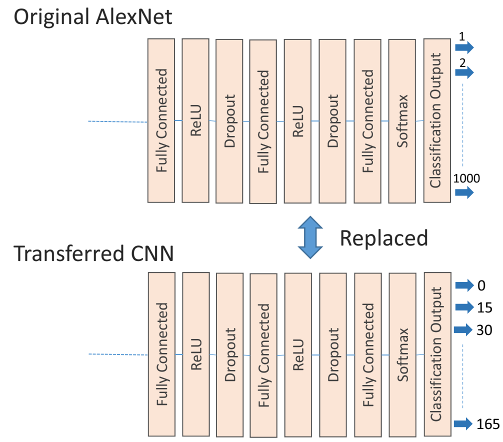


Fig. 5. Replacement of fully connected layers for dealing with target classification task, i.e., 12 categories.

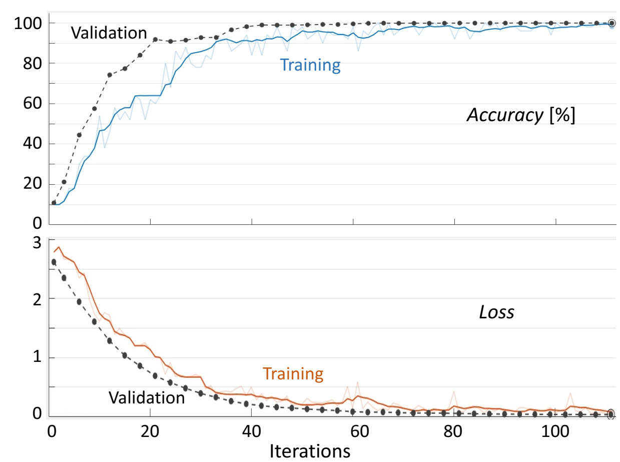


Fig. 6. Training progress of transfer learning shown in Fig. 5.

cause of these results seems to be the conversion of resolution before classification. The resolution of images given to the input layer is forced to be converted to 227×227×3 fixed according to the input layer of the AlexNet, which brings out some undesirable deformation of images and the resultant ambiguities in classification.

## V. Experiment of Pick and Place

### *A. Without Visual Feedback Control*

An actual pick and place experiment is conducted using a small articulated robot named DOBOT. The experimental setup is shown in Fig. 8. Position $[x\ y\ z]^T$ and yaw angle $R$ of the gripper in robot coordinate system can be controlled by an API function SetPTPCmd($x_d, y_d, z_d, R_d$). Note that the yaw angle $R$ is dealt with the orientation of a target object in this experiment. Figure 9 shows the developed control dialogue for the robot. Pick and place task while recognizing the orientations of target objects can be executed through the dialogue. Figure 10 illustrates the flowchart of the pick and place task, which is implemented in a timer interrupt routine. In the timer

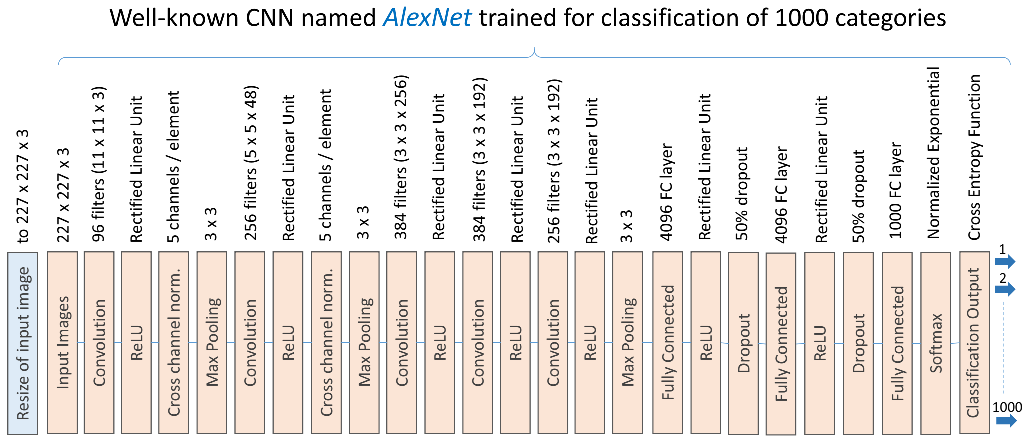


Fig. 4. Network structure of well-known CNN named AlexNet which can classify input images into one of 1000 kinds of categories.

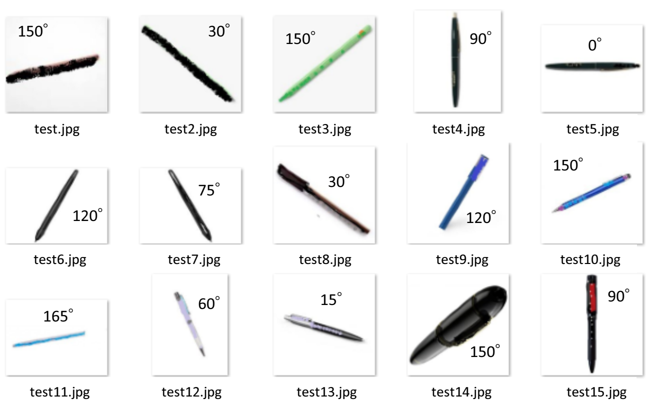


Fig. 7. Classification results of test images using transfer learning based CNN.

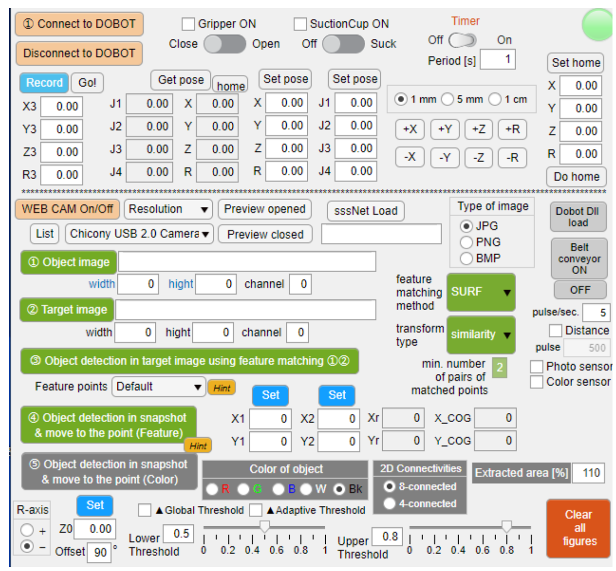


Fig. 9. Developed control dialogue for DOBOT.

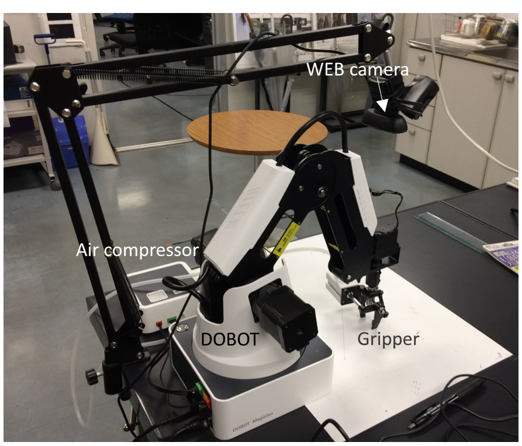


Fig. 8. Experimental setup based on an articulated robot.

interrupt, first of all, a snapshot of $1600 \times 1200$ resolution is captured. After binarized into black and white, a connected component with the largest area is found as a target object and the COG position $[I_x \; I_y]^T$ $(1 \leq I_x \leq 1600, 1 \leq I_y \leq 1200)$ in image coordinate system is extracted. Consequently, desired position $[x_d \; y_d]^T$ in robot coordinate system to move the gripper to the COG position can be obtained by

$$x_d = X_1 + I_x \frac{X_2 - X_1}{1600} \tag{3}$$

$$y_d = Y_1 + I_y \frac{Y_2 - Y_1}{1200} \tag{4}$$

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. 11, 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

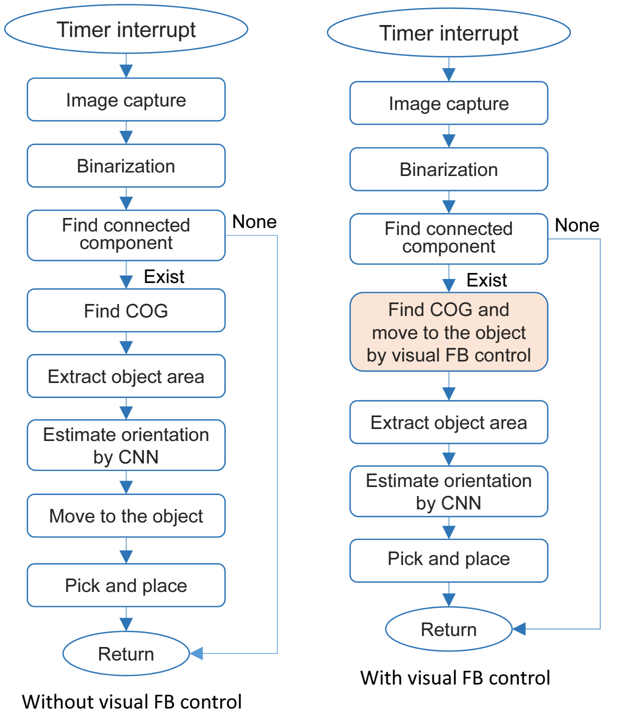


Fig. 10. Flowcharts to realize pick and place task using transfer learning-based CNN, in which left and right figures show the process flow diagrams without and with a visual feedback control, respectively.

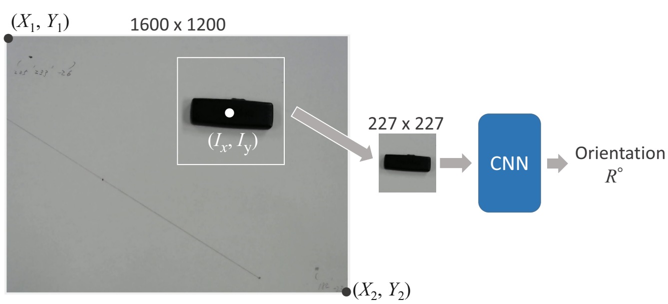


Fig. 11. Procedure to extract the orientation of a workpiece from a captured image.

original snapshot as shown in Fig. 11. The cropped image 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.

### *B. With Visual Feedback Control*

Camera configuration in visual feedback control is different from that in the previous subsection. A lightweight endoscope camera is attached close by the gripper as shown in Fig. 12. Manipulated variable$\boldsymbol{v}(k) = [v_x(k) \ v_y(k)]^T$ for visual feed-

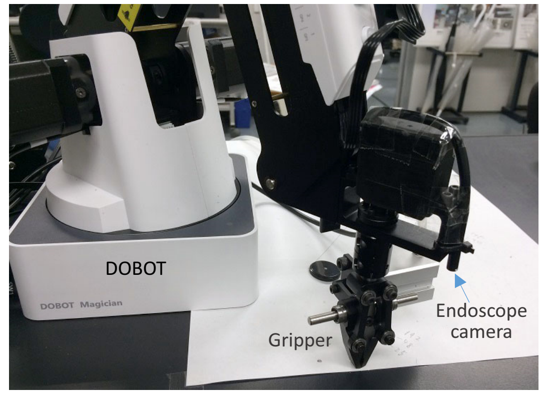


Fig. 12. Experimental setup based on an articulated robot with a endoscope camera.

back is generated by a PI-action given by

$$\boldsymbol{v}(k) = K_p\boldsymbol{e}(k) + K_i \sum_{n=1}^{k} \boldsymbol{e}(n) \tag{5}$$

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

$$\boldsymbol{e}(k) = \boldsymbol{X}_d - \boldsymbol{I}(k) \tag{6}$$

where $\boldsymbol{X}_d = [\frac{X_r}{2} \ \frac{Y_r}{2}]^T$ 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. $[X_r \ Y_y]$ 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.

## VI. Conclusions

In this paper, a CNN acquired by transfer learning of AlexNet, which had been trained on approximately 1.2 million images of ImageNet database, was introduced to recognize the orientations of objects. Originally, the AlexNet had been able to classify input images into one of 1,000 kinds of objects, however the transferred CNN has been able to recognize the orientation of an object in images with 12 kinds of degrees. Then, a visual feedback control has been implemented so that the gripper of the pick and place robot can move to the position nearly just above an target object. Due to the visual feedback control, the complicated calibration between image and robot coordinate systems can be omitted. The effectiveness of the system is evaluated through experimental pick and place tests using an articulated robot named DOBOT.

## Acknowledgment

This work was partially supported by Mitsubishi Pencil Co., Ltd. and Meiho Co., Ltd.

## REFERENCES


[1] J. Nagi, F. Ducatelle, G.A.D. Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L.M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA2011)*, pp. 342–347, 2011.

[2] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals – Manufacturing Technology*, Vol. 65, No. 1, pp. 417–420, 2016.

[3] S. Faghih-Roohi, S. Hajizadeh, A. Nunez, R. Babuska, and B.D. Schutter, "Deep convolutional neural networks for detection of rail surface defects," *Procs. of the 2016 International Joint Conference on Neural Networks (IJCNN2016)*, Vancouver, Canada, pp. 2584–2589, 2016.

[4] S. Zhou, Y. Chen, D. Zhang, J. Xie, and Y. Zhou, "Classification of surface defects on steel sheet using convolutional neural networks," *Materials and Technology*, Vol. 51, No. 1, pp. 123–131, 2017.

[5] M. Ferguson, R. Ak, Y. Lee, and K. Law, "Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning," *Smart and Sustainable Manufacturing Systems*, Vol. 2, No. 1, pp. 137–164, 2018.

[6] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask R-CNN," *Procs. of 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[7] F. Nagata, K. Tokuno, H. Tamano, H. Nakamura, M. Tamura, K. Kato, A. Otsuka, T. Ikeda, K. Watanabe, M.K. Habib, "Basic application of deep convolutional neural network to visual inspection," *Procs. of International Conference on Industrial Application Engineering (ICIAE2018)*, pp. 4–8, Okinawa, 2018.

[8] F. Nagata, K. Tokuno, K. Nakashima, A. Otsuka, T. Ikeda, H. Ochi, K. Watanabe, M. K. Habib, "Fusion method of convolutional neural network and support vector machine for high accuracy anomaly detection," *Procs. of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA 2019)*, pp. 970–975, Tianjin, China, 2019.

[9] F. Nagata, K. Tokuno, K. Mitarai, A. Otsuka, T. Ikeda, H. Ochi, K. Watanabe, 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.