Pick and Place Robot Using a Transfer Learning-Based CNN*

Fusaomi Nagata
Gradate School of Sci. and Eng.
Sanyo-Onoda City University
Sanyo-Onoda, Japan
nagata@rs.socu.ac.jp

Kohei Miki
Gradate School of Sci. and Eng.
Sanyo-Onoda City University
Sanyo-Onoda, Japan
f116056@ed.socu.ac.jp

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

Maki K. Habib
Mech. Eng. Dep., Sch. of Sci. and Eng.
The American University in Cairo
New Cairo, Egypt
maki@aucegypt.edu

***Abstract*—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. 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. The tool further enables to easily design a CNN model based on transfer learning concept. In this paper, a CNN acquired by transfer learning of AlexNet, which is the winner of ImageNet LSVRC2012, is designed to recognize the orientation of objects. The effectiveness of the transfer learning based CNN is evaluated through an actual pick and place test using an articulated robot named DOBOT.**

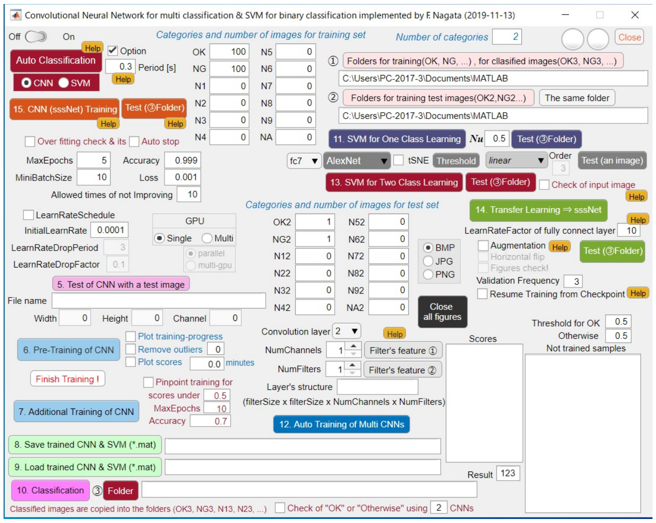
convolutional neural network, transfer learning, pick and place, robot ***Index Terms*—**

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 CNN acquired by transfer learning of AlexNet, which is the winner of ImageNet LSVRC2012, is introduced to recognize the orientation of objects in images. The effectiveness of the CNN is evaluated using test image data set of thin resin mold articles.

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


Convolutional Neural Network for multi classification & SVM for binary classification implemented by F. Nagata (2019-11-13)

Off ○ On

Categories and number of images for training set — Number of categories: 2

☑ Option — Period [s]: 0.3

● CNN ○ SVM

OK: 100, NG: 100, N1: 0, N2: 0, N3: 0, N4: 0, N5: 0, N6: 0, N7: 0, N8: 0, N9: 0, NA: 0

① Folders for training(OK, NG, ...) , for cllasified images(OK3, NG3, ...): C:\Users\PC-2017-3\Documents\MATLAB

② Folders for training test images(OK2,NG2...) — The same folder: C:\Users\PC-2017-3\Documents\MATLAB

□ Over fitting check & its □ Auto stop

MaxEpochs: 5, Accuracy: 0.999

MiniBatchSize: 10, Loss: 0.001

Allowed times of not Improving: 10

Nu: 0.5

fc7 ▼ — AlexNet ▼ — □ tSNE — linear ▼ — Order: 3

□ Check of input image

□ LearnRateSchedule

InitialLearnRate: 0.0001

LearnRateDropPeriod: 3

LearnRateDropFactor: 0.1

GPU: ● Single ○ Multi (● parallel ○ multi-gpu)

Categories and number of images for test set

OK2: 1, NG2: 1, N12: 0, N22: 0, N32: 0, N42: 0, N52: 0, N62: 0, N72: 0, N82: 0, N92: 0, NA2: 0

● BMP ○ JPG ○ PNG

LearnRateFactor of fully connect layer: 10

□ Augmentation □ Horizontal flip □ Figures check!

Validation Frequency: 3

□ Resume Training from Checkpoint

File name:

Width: 0, Height: 0, Channel: 0

□ Plot training-progress

□ Remove outliers: 0

□ Plot scores: 0.0 minutes

□ Pinpoint training for scores under: 0.5, MaxEpochs: 10, Accuracy: 0.7

Convolution layer: 2 ▼

NumChannels: 1

NumFilters: 1

Layer's structure: (filterSize x filterSize x NumChannels x NumFilters)

Threshold for OK: 0.5, Otherwise: 0.5

Scores

Not trained samples

Result: 123

Classified images are copied into the folders (OK3, NG3, N13, N23, ...) □ Check of "OK" or "Otherwise" using 2 CNNs

Help
Auto Classification
Help
15. CNN (sssNet) Training
Test (③Folder)
Help
Help
Close
11. SVM for One Class Learning
Test (③Folder)
Threshold
Test (an image)
13. SVM for Two Class Learning
Test (③Folder)
Help
14. Transfer Learning ⇒ sssNet
Help
Help
Test (③Folder)
Help
5. Test of CNN with a test image
Close all figures
Help
6. Pre-Training of CNN
Filter's feature ①
Filter's feature ②
Finish Training !
7. Additional Training of CNN
12. Auto Training of Multi CNNs
8. Save trained CNN & SVM (*.mat)
9. Load trained CNN & SVM (*.mat)
10. Classification
③ Folder

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

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 200×200 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

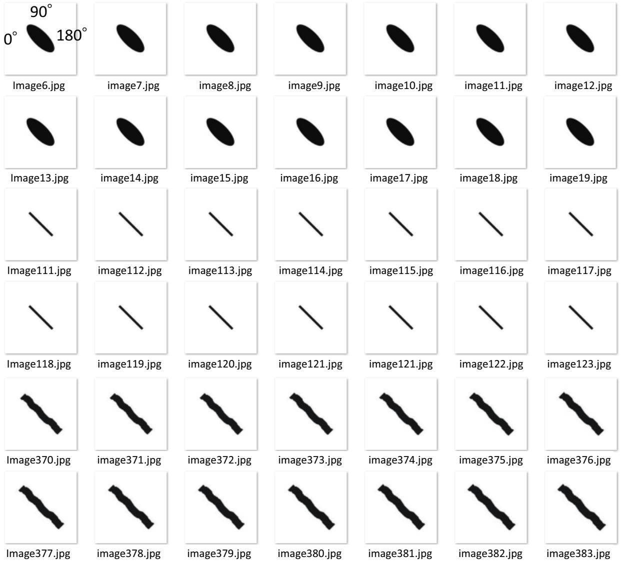


Fig. 2. Examples of training images for the orientation of 45°.

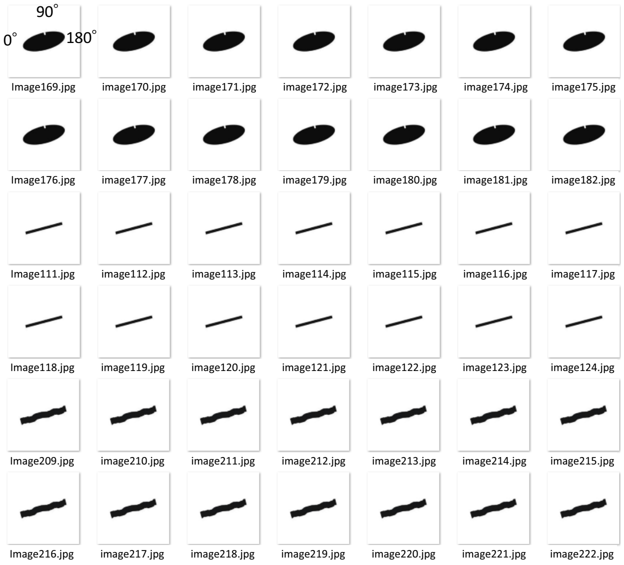


Fig. 3. Examples of training images for the orientation of 165°.

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 nth image for training is given to the input layer of the transfered CNN, then the softmax layer produces the probability $p_{ni}(i = 1, 2, \cdots, 12)$ called the score for twelve

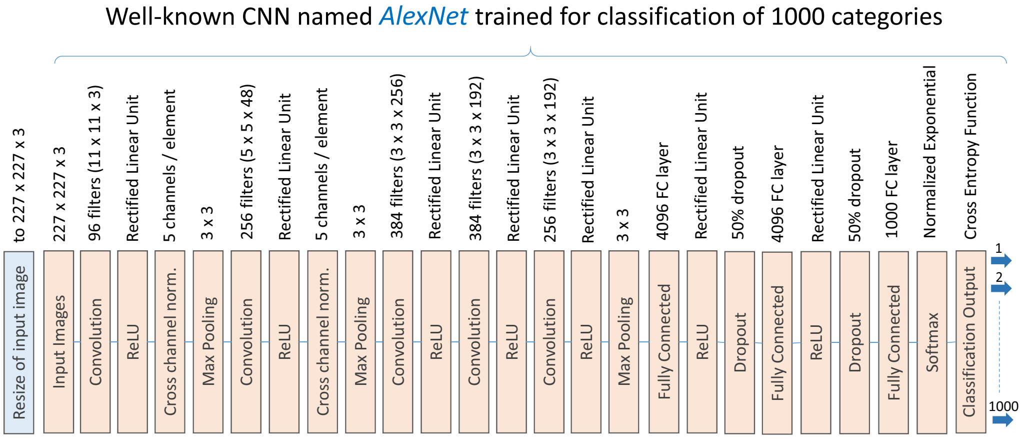


Fig. 4. Network structure of well-known CNN named AlexNet which can categorize input images into one of 1000 kinds of objects.

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 nth image. The transfered 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 nth 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.

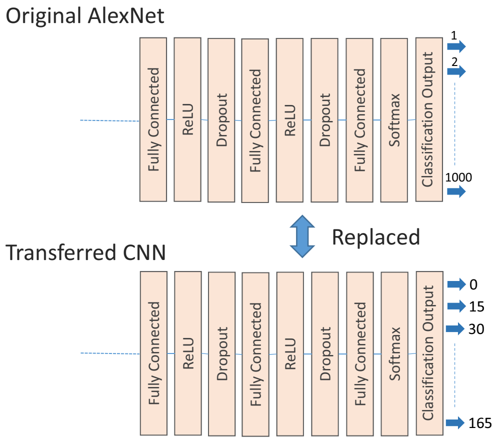


Fig. 5. Replacement of fully connected layers for dealing with target classification task, i.e., 12 categories.

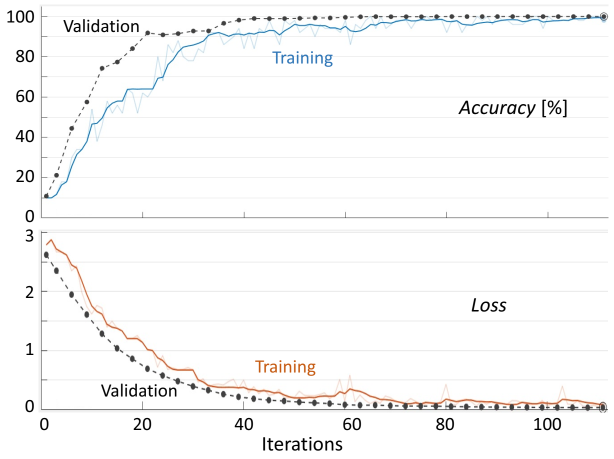


Fig. 6. Training progress of transfer learning shown in Fig. 5.

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°)

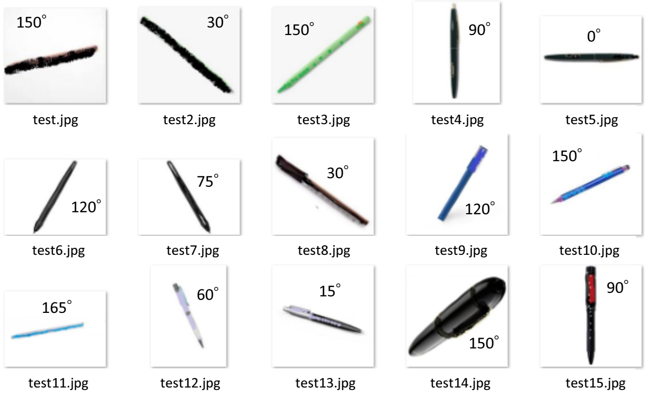


Fig. 7. Classification results of test images using transfer learning based CNN.

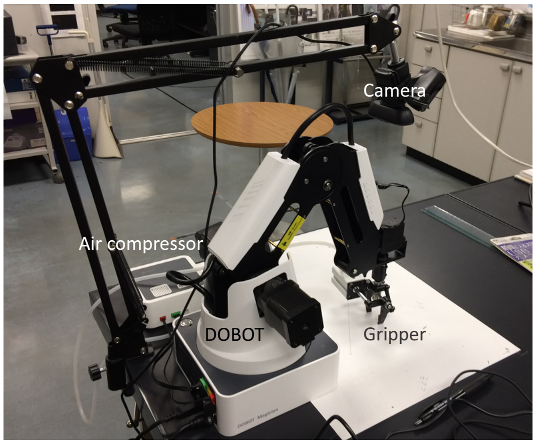
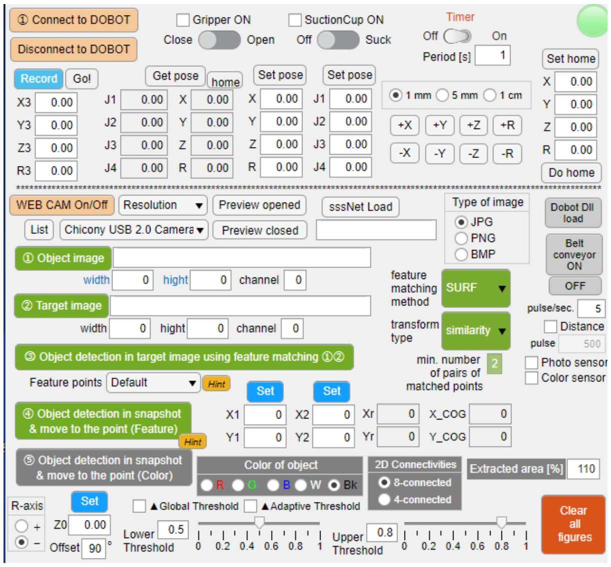


Fig. 8. Experimental setup based on an articulated robot.



☐ Gripper ON — Close / Open
☐ SuctionCup ON — Off / Suck
Off / On
Period [s] 1

X3 0.00 | Y3 0.00 | Z3 0.00 | R3 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
X 0.00 | Y 0.00 | Z 0.00 | R 0.00
J1 0.00 | J2 0.00 | J3 0.00 | J4 0.00
● 1 mm ○ 5 mm ○ 1 cm
X 0.00 | Y 0.00 | Z 0.00 | R 0.00

Chicony USB 2.0 Camera ▾
● JPG ○ PNG ○ BMP
width 0 hight 0 channel 0
width 0 hight 0 channel 0
feature matching method SURF ▾
transform type similarity ▾
pulse/sec. 5
☐ Distance
pulse 500
min. number of pairs of matched points 2
☐ Photo sensor
☐ Color sensor
Feature points Default ▾
X1 0 | X2 0 | Xr 0 | X_COG 0
Y1 0 | Y2 0 | Yr 0 | Y_COG 0
○ R ○ G ○ B ○ W ● Bk
● 8-connected ○ 4-connected
Extracted area [%] 110
○ + ● −
Z0 0.00
Offset 90 °
☐ ▲Global Threshold ☐ ▲Adaptive Threshold
Lower Threshold 0.5 (0 0.2 0.4 0.6 0.8 1)
Upper Threshold 0.8 (0 0.2 0.4 0.6 0.8 1)

Fig. 9. Developed control dialogue for DOBOT.

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 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

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 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 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. Consequently, desired position $[x_d\ y_d]^T$ in robot coordinate system to move the gripper to the COG position can be obtained by

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

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

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. The part of the connected component is further cropped centering the COG from the 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.

VI. Conclusions

In this paper, a CNN acquired by transfer learning of AlexNet, which is the winner of ImageNet LSVRC2012, is 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 transfered CNN has been able to recognize the orientation of an object in images with 12 kinds of degrees. The effectiveness and promise of the CNN are evaluated using test images imitating thin resin mold articles. Then, we actually applied this transfer learning-based CNN to a small articulated robot to demonstrate a pick and place task which required orientation information of

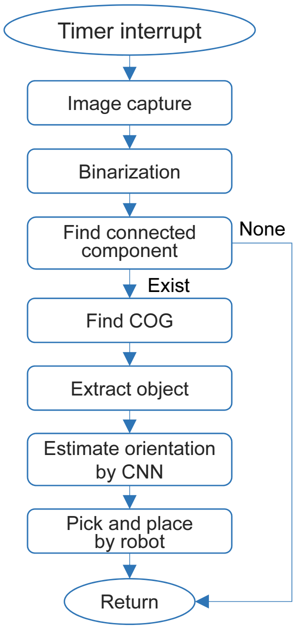


Fig. 10. Flowchart to realize pick and place task using transfer learning-based CNN.

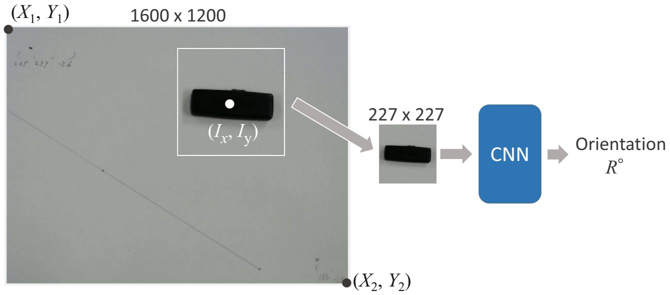


Fig. 11. Procedure to extract the orientation of a workpiece from an input image.

target objects. In future work, we are planning to apply the proposed system to an actual production line of resin molded articles with various shapes. Because orientation information is essential for industrial robots to successfully play a pick and place task.

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.