Design Tool of Deep Convolutional Neural Network for Intelligent Visual Inspection

Fusaomi Nagata
Sanyo-Onoda City University

Kenta Tokuno
Sanyo-Onoda City University

Akimasa Otsuka
Sanyo-Onoda City University

Takeshi Ikeda
Sanyo-Onoda City University

Hiroaki Ochi
Sanyo-Onoda City University

Design Tool of Deep Convolutional Neural Network for Intelligent Visual Inspection

To cite this article: Fusaomi Nagata *et al* 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **423** 012073

Design Tool of Deep Convolutional Neural Network for Intelligent Visual Inspection

Fusaomi Nagata[1], Kenta Tokuno[1], Akimasa Otsuka[1], Takeshi Ikeda[1], Hiroaki Ochi[1], Keigo Watanabe[2], Maki K. Habib[3]

[1] Graduate School of Science and Engineering, Sanyo-onoda City University, 1-1-1 Daigaku-Dori, Sanyo-Onoda 756-0884, Japan
[2] Graduate School of Natural Science and Technology, Okayama University, 3-1-1 Tsushima-naka, Kita-ku, Okayama 700-8530, Japan.
[3] Mechanical Engineering Department, School of Sciences and Engineering, American University in Cairo, AUC Avenue P.O.Box 74, New Cairo 11835, Egypt.

Corresponding author's e-mail address : nagata@rs.tusy.ac.jp

Abstract. Recently, convolutional neural networks (CNNs) are used essentially to classify images as it helps to cluster them by similarity and perform recognition. In this paper, a design tool that helps to develop different deep CNNs (DCNNs) is presented. As an example, a DCNN is designed by using the developed tool to use it for vision based inspection to recognize undesirable defects such as crack, burr, protrusion and chipping which normally occur in the manufacturing process of resin molded articles. An image generator is implemented to efficiently produce many similar images for training. Similar images are easily generated by rotating, translating, scaling and transforming original images. The designed DCNN is trained by using the produced images and then tested through classification experiments. The usefulness of the design tool and the basic performance of the designed DCNN are introduced.

1. Introduction

Artificial neural network (ANN) which has four or more layer structure is called deep neural network (DNN) and it is recognized as a promising machine learning technique. Convolutional neural network (CNN) is a type of DNN and it is specialized mainly for image and optical character recognition. 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 allowed mobile robots to perform real-time gesture recognition. Weimer et al. proposed deep CNN (DCNN) architectures for automated feature extraction in industrial inspection process [2]. The DCNN automatically generates features from massive amount of training image data and demonstrates excellent ability to detect defects with low false alarm rates. Faghih-Roohi et al. also presented a different type of DCNN for automatic detection of rail surface defects [3]. It was concluded that a large CNN model performs a better classification with good results than the small and medium CNN, although the training requires a longer time. Further, 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. Meanwhile, the authors reported the effectiveness of DCNN in terms of ability to classify the images of resin molded articles into two categories OK or NG, in which the image samples in the training test set were successfully classified after the proposed additional training process [5].

However, it seems that it is not easy for junior engineers to design an adequate DCNN in detail and to efficiently train it by using a large number of images and their labels. To cope with the need, a user-friendly design tool for DCNN is proposed in this paper. As an example, a DCNN is designed by using the developed tool to use it for vision based inspection to recognize undesirable defects such as crack, burr, protrusion and chipping which occur in the manufacturing process of resin molded articles. An image generator is implemented to efficiently produce many similar images for training. Similar images are easily generated by rotating, translating, scaling and transforming original images. The designed DCNN is trained by using the produced images and then tested through classification experiments. The usefulness of the design tool and the basic performance of the designed DCNN are introduced.

2. Development of design tool for deep convolutional neural network

2.1. Design application for DCNN

In this study, a dialogue-based design tool with its user interface illustrated in Fig. 1 was first proposed as a test test environment for DCNN design development. Matlab including Statistics and Machine Learning, Neural Network and Parallel Computing Toolboxes are were used to develop this design tool. Through the developed dialogue interuction, adequate DCNNs can be designed and trained. Main fucntions of the design tool are explained in numerical order as shown in Fig. 1.

2.1.1. Loading of training and test images. A large number of training images and training test ones for target categories are loaded into the memory area of MATLAB. The allowable maximum number of categories are twelve. Each image has a sequential number as image1.jpg, image2.jpg and so on. File folders for training images and training test ones can be designated, respectively.

2.1.2. Check of training and test images. Loaded training images and training test ones are arrayed as, e.g., $200 \times 200 \times 3 \times 5000$ (width $\times$ height $\times$ channel $\times$ number of images) and $200 \times 200 \times 3 \times 500$, respectively. Hence, they can be given to the training process as a training set and a training test set, respectively.

2.1.3. Categorization using training images. This function checks how correctly all training images are categorized, i.e., not only categorization accuracy for each category but also associated with score representing the probability of recognition for each image.

2.1.4. Categorization using training test images. This function evaluates the generalization ability to training test images. The training test images are those images that they have not been used in the training process.

2.1.5. Categorization using an image (one by one). This button enables another test function by which the categorization of each image is individually evaluated based on the categorization result and its score.

2.1.6. Pre-training execution. This is a pre-training mode that aims to conduct learning of a designed DCNN with randomly initialized weights. The weights to be trained are included in the convolutional layers' filters and fully connected layers near the DCNN' s output layer. In the training process, values of the effective parameters such as max epochs, mini batch size, training rate and desired categorization accuracy are changed as needed .

2.1.7. Save and load a trained DCNN. These functions can save and load a DCNN with trained weights to and from the PC's hard disk, respectively.

2.1.8. Additional training execution. This training function provides additional training mode with trained weights in the past. For example, after a DCNN with saved trained weights is loaded, a successive training of the DCNN by using increased and reorganized training images can be resumed. The generalization ability to unlearned images, i.e., training test images, are pinpointedly enhanced by conducting this additional training.

2.1.9. Categorization using shuffled images. This button introduces another categorization function using a trained DCNN that tries to classify, e.g., a large number of OK images and four kinds of NG images shuffled in a file folder into five categories and copy them into the corresponding folders.

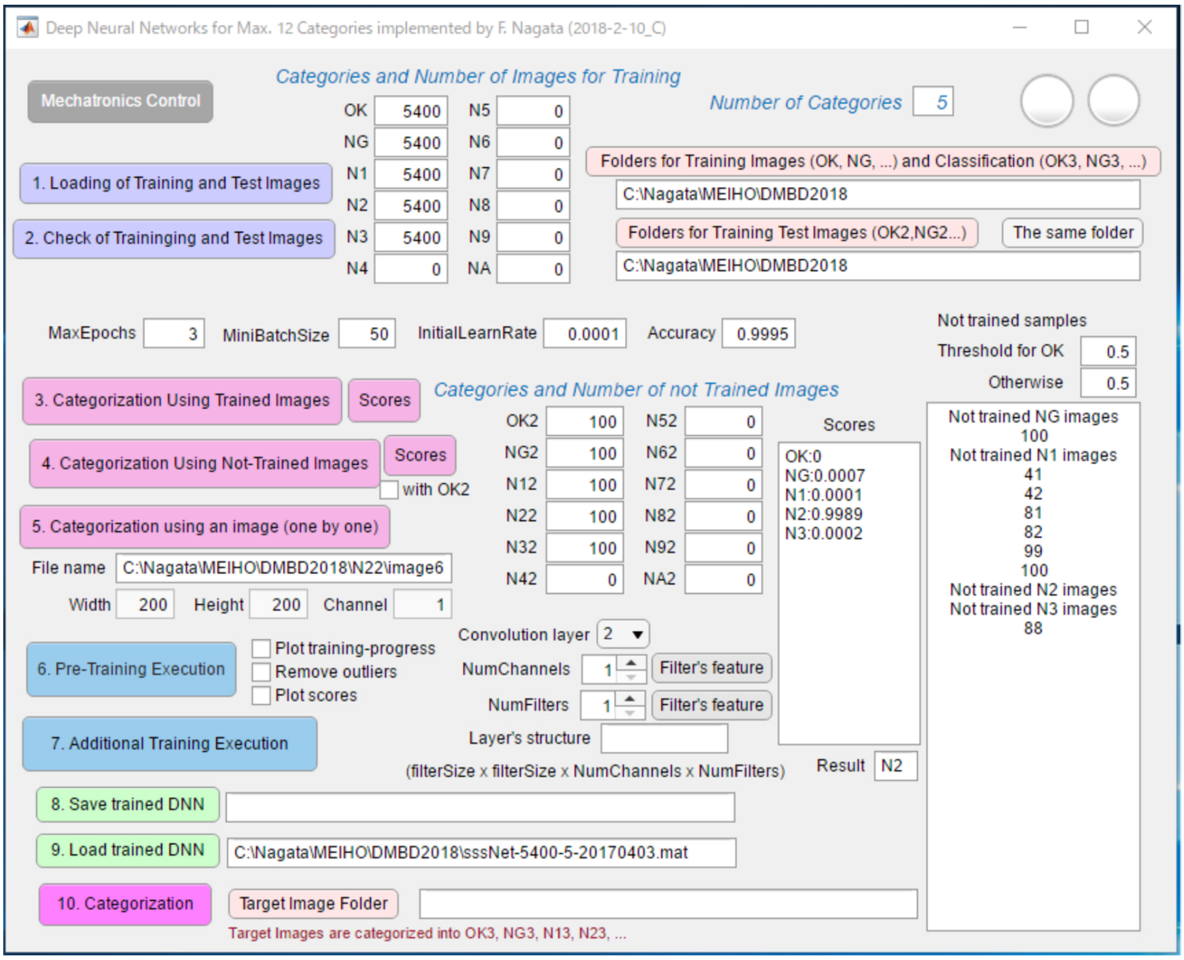

Fig. 1 Design tool developed for deep convolutional neural network (DCNN).

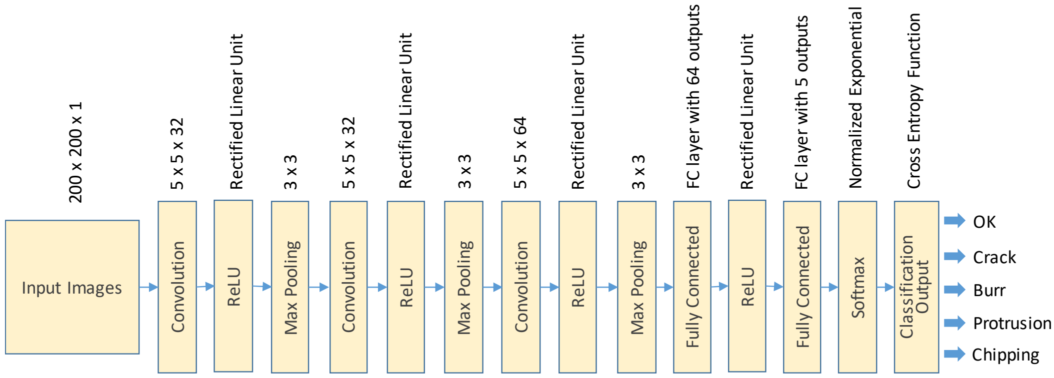

Fig. 2 DCNN with 15 layers designed for visual inspection of resin molded articles

2.2. *Design example of DCNN*

As an example, a DCNN with 15 layers is shown in Fig. 2. This example was designed by using the tool in order to inspect undesirable defects such as Crack, Burr, Protrusion and Chipping phenomena which occur in the production process of resin molded articles. These kinds of defects are small to the whole sizes of the resin molded articles, so that they are elusive and even experienced testing workers often miss them. The designed DCNN tries to detect such small defects.

A large number of images with those defects are required for the training of DCNN, so that a similar image generator was also implemented to efficiently produce many similar images with the same feature by rotating, translating, scaling or transforming original images with such defects. The designed DCNN can be trained by using the training set of the images.

2.3. *Training example*

In this trial test, gray-scale images (1020×5 categories) with the resolution of 200×200 were prepared by using the image generator and then training of the designed DCNN was conducted by using those images. The performance of the trained DCNN was simply checked by classifying training test sample images as shown in Fig. 3. Figure 4 shows their categorization scores. Although the trained DCNN had the ability to classify sample images except for "image2.jpg" and "image9.jpg" into the NG folder, it is presumed that the number of the training images with different type of features was insufficient. To gradually improve the generalization ablity, an additional training method to cope with the not well trained images was conducted, so that the recognition ability to images with similar features with not well trained images could be efficiently and pinpointedly improved.

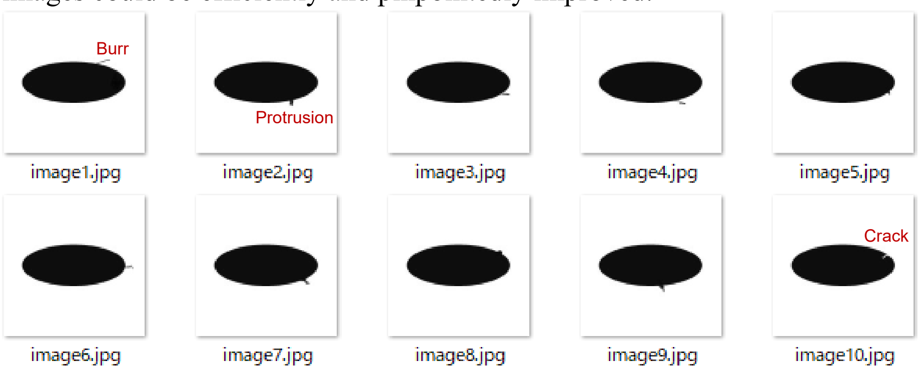

Fig. 3 Training test sample images which were not included in the training set.

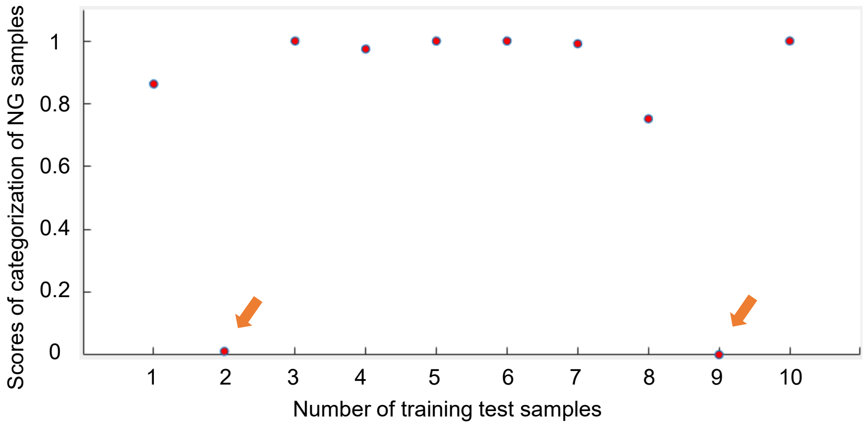

Fig. 4 Training test sample images which were not included in the training set.

3. Conclusions

In this paper, a user-friendly design application for DCNNs was proposed. As an example, a DCNN was designed by using the develoiped application to recognize undesirable defects such as Crack, Burr, Protrusion and Chipping that can be seen in the manufacturing process of resin molded articles. An image generator was also developed to efficiently produce many similar images for training. Similar images were easily generated by rotating, translating, scaling and transforming original images. The designed DCNN was trained by using the produced similar images and then evaluated through classification experiments. The proposed additional training process allowed the DCNN to pinpointedly improve the recognition ability. The usefulness of the proposed design application and the basic performance of designed DCNN were verified.

References

[1] J. Nagi, F. Ducatelle, G.A.D. Caro, D. Ciresan, U. Meier, A. Giusti, F.Nagi, J. Schmidhuber, L.M. Gambardella, “Max-pooling convolutionalneural networks for vision-based hand gesture recognition,” Procs. Of 2011 IEEEInternational Conference on Signal and Image Processing Applications(ICSIPA2011), pp. 342-347, 2011.

[2] D. Weimer, B. Scholz-Reiter, 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, B.D. Schutter, “Deep convolutional neural networks for detection of rail surface defects,” Procs. of the 2016 International Joint Conference on Neural Networks (IJCNN 2016), Vancouver, Canada, pp. 2584-2589, 2016.

[4] S. Zhou, Y. Chen, D. Zhang, J. Xie, 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] 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, March 2018.