The Twenty-Fourth International Symposium on Artificial Life and Robotics 2019 (AROB 24th 2019),
The Fourth International Symposium on BioComplexity 2019 (ISBC 4th 2019),
B-Con Plaza, Beppu, Japan, January 23-25, 2019

Design Tool of Convolutional Neural Network (CNN) –Design of Cascade-Type CNN and Its Application to Defect Detection–

Kenta Tokuno[1], Fusaomi Nagata[1], Akimasa Otsuka[1]
Keigo Watanabe[2], Maki K. Habib[3]
[1]Sanyo-Onoda City University, Japan
[2]Okayama University, Japan
[3]The American University in Cairo, Egypt
[1]nagata@rs.socu.ac.jp, [1]f118608@ed.socu.ac.jp, [1]otsuka_a@rs.socu.ac.jp
[2]watanabe@sys.okayama-u.ac.jp , [3]maki@aucegypt.edu

Abstract

In this decade, convolutional neural network called CNN has been attracting attention due to its high ability of image recognition and other applications. In this paper, a design and training tool for convolutional neural network is developed. The tool requires no knowledge and experience about C++ or Python. As a test trial, a multi-class CNN is designed using the tool to detect defects such as crack, burr, protrusion, chipping and spot which occurs in the manufacturing process of resin molded articles. The multi-class CNN is trained with a large number of training images of each category. Then, a cascade-type CNN consisting of multiple binary-class CNNs is proposed to classify target test images into OK category or NG category including the defects such as crack, burr, protrusion, chipping and spot. A metrics using the miss-classification rate is applied in order to compare the performances of the conventional multi-class CNN and the proposed cascade-type CNN.

1 INTRODUCTION

Deep neural networks (DNNs) have been not sufficiently available up to around 2012 from the middle of 1990's, because such problems as weight tuning to enhance the network performance and the difficulty of application development for image recognition had not been overcome. However, the effectiveness of deep learning has been recognized widely from around 2012 due to the progress of computational ability using GPU and the improvement of network architecture using auto encoder and Rectified Linear Unit (ReLU). In particular, CNN has been gathering attention as one of the most powerful approaches for image recognition and applied to various kinds of system solutions [1, 2, 3].

The authors already developed a CNN design tool for defect inspection [4, 5]. In this paper, a multi-class CNN id designed to inspect small defects such as

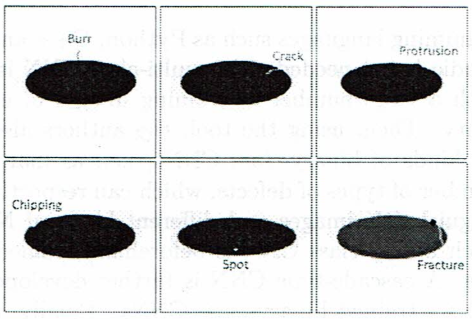
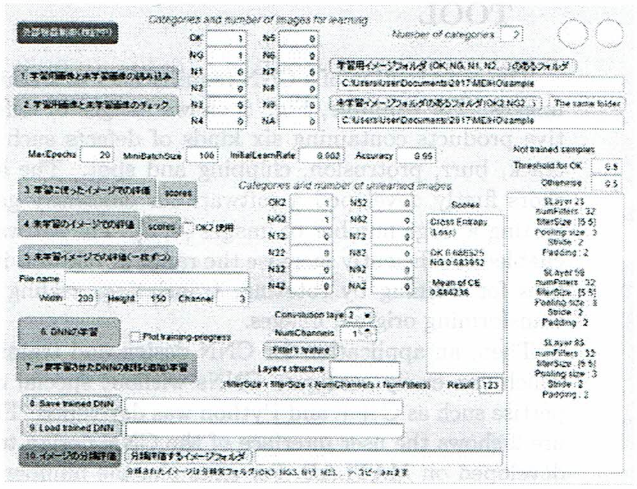


Figure 1: Training image samples including typical defects such as burr, crack, protrusion, chipping, spot and fracture.


学習画像と未学習画像の読み込み
1. 学習用画像と未学習画像の読み込み
2. 学習用画像と未学習画像のチェック
3. 学習に使ったイメージでの評価
scores
4. 未学習のイメージでの評価
scores
OK2 使用
5. 未学習イメージでの評価(一枚ずつ)
6. DNNの学習
Plot training-progress
Filter's feature
7. 一度学習させたDNNの継続(追加)学習
8. Save trained DNN
9. Load trained DNN
10. イメージの分類評価
分類評価するイメージフォルダ
分類されたイメージは分類先フォルダ(OK3, NG3, N13, N23, ...)へコピーされます

Categories and number of images for learning
OK 1, NG 1, N1 0, N2 0, N3 0, N4 0, N5 0, N6 0, N7 0, N8 0, N9 0, NA 0
Number of categories 2
学習用イメージフォルダ (OK, NG, N1, N2, ...) のあるフォルダ
C:\Users\User\Documents\2017\MEH\Osample
未学習イメージフォルダのあるフォルダ(OK2, NG2, ...) The same folder
C:\Users\User\Documents\2017\MEH\Osample
MaxEpochs 20 MiniBatchSize 100 InitialLearnRate 0.003 Accuracy 0.95
Not trained samples
Threshold for OK 0.5
Otherwise 0.5
Categories and number of unlearned images
OK2 1, NG2 1, N12 0, N22 0, N32 0, N42 0, N52 0, N62 0, N72 0, N82 0, N92 0, NA2 0
Scores
Cross Entropy (Loss)
OK 0.686525
NG 0.683952
Mean of CE 0.684238
File name
Width 200 Height 150 Channel 3
Convolution layer 2
NumChannels 1
Layer's structure
(filterSize x filterSize x NumChannels x NumFilters) Result 123
SLayer 2S numFilters: 32 filterSize: [5 5] Pooling size: 3 Stride: 2 Padding: 2
SLayer 5S numFilters: 32 filterSize: [5 5] Pooling size: 3 Stride: 2 Padding: 2
SLayer 8S numFilters: 32 filterSize: [5 5] Pooling size: 3 Stride: 2 Padding: 2

Figure 2: User-friendly CNN design tool developed on MATLAB system.

crack, burr, protrusion, chipping and spot phenomena seen in the manufacturing process of resin molded articles . In designing the CNN, the expertise about pro-

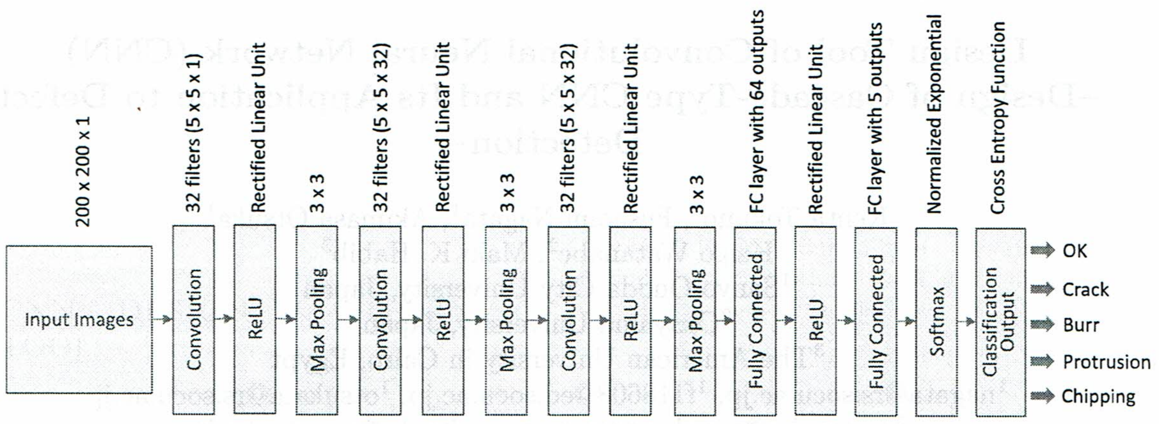


Figure 3: Multi-class CNN with 15 layers and 5 outputs which is designed using the application shown in Fig. 2.

gramming languages such as Python, C++ and Visual Studio is not needed. The multi-class CNN is trained with a large number of training images of each category. Then, using the tool, the authors also design six kinds of binary-class CNNs, i.e., as many as the number of types of defects, which can respectively distinguish OK images and different kinds of NG ones. Each binary-class CNN is beforehand trained one by one. A cascade-type CNN is further developed using the six trained binary-class CNNs. Finally, the conventional CNN for multi-class classification and the proposed cascade-type CNN are evaluated and compared based on the rate of miss classification.

2 CNN DESIGN & TRAINING TOOL

The construction of a CNN requires a lot of training images. For example, Fig. 1 shows images of defective products containing six kinds of defects such as crack, burr, protrusion, chipping and spot. The authors firstly developed a software for efficiently generating a large number of images [4, 5]. The software enabled to efficiently increase the range of relevant images for training by rotating, translating, scaling or transforming original images.

Then, an application for CNN design and training which can easily configure CNNs without special expertise such as C ++ and Python was developed. Figure 2 shows the user interface of the CNN design tool developed on MATLAB. For example, the number of categories can be increased up to twelve; parameters of max epoch, mini batch size, learning rate, classification accuracy can be changed; optimized learning rate for each training data can be examined. In addition, a function to classify test images into each category based on the output from Softmax function and another function to visualize the channels constituting the filters of each convolution layer can be provided.

Figure 3 shows a multi-class CNN with 15 layers and 5 outputs which is designed using the application shown in Fig. 2. The first layer is set for input images of 200×200×1 (height×width×channel) given by a matrix with zero-center normalization. The second and fifth layers are convolution ones which have 32 filters with the resolution of 5×5×1 and 5×5×2, respectively. In the convolution layers, the filters are applied to images while being slided from the left top to the right bottom within each image based on the value of the stride. The eighth layer is also a convolution layer which has 64 filters with the resolution of 5×5×32. The values of stride and padding in the convolution layers are set to [1 1] and [2 2 2 2], respectively. The convolution layers perform the translation invariance. The third, sixth, ninth and twelfth layers are activation functions called ReLU given by

$$f(u) = \max(0, u) \tag{1}$$

$$f'(u) = \begin{cases} 1 & (u > 0) \\ 0 & (u \leq 0) \end{cases} \tag{2}$$

where $\boldsymbol{y}_n = [y_{n1}\ y_{n2}\ y_{n3}\ y_{n4}\ y_{n5}]^T$ ($||\boldsymbol{y}_n|| = 1$) is the output from the thirteenth fully connected layer corresponding to the n-th image. In this case, the cross entropy as a loss function is given by

$$\bar{E} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{5} t_{nk}\log(y_{nk}) \tag{3}$$

where $\boldsymbol{t}_n = [t_{n1}\ t_{n2}\ t_{n3}\ t_{n4}\ t_{n5}]^T$ means the n-th desired output for five categories, i.e., $[1\ 0\ 0\ 0\ 0]^T$, $[0\ 1\ 0\ 0\ 0]^T$, $[0\ 0\ 1\ 0\ 0]^T$, $[0\ 0\ 0\ 1\ 0]^T$, $[0\ 0\ 0\ 0\ 1]^T$. The cross entropy is also used to tune the weights in back propagation algorithm during training process.

3 MULTI-CLASS CNN

After preparing 50000 training images consisting of 10000 of normals and 40000 of four kinds of anomalies, the multi-class CNN was trained using the 50000 images. The multi-class CNN with trained weights was saved into a file named sssNet1. In order to evaluate the generalization property of the sssNet1, a classification test using 200 test images of non-defective and 200 × 4 images of four kinds of defectives was done. As the result, non-defective images and defective images with a defect of chipping could be classified with a recognition rate of 100 %. The other three categories could be classified with recognition rates about 97 %. Figure 4 shows the categorization scores of test images with four kinds of defects (four categories × 200 samples). In this case, in order to easily evaluate the mis-recognized result, each image is scored as $y_n = [y_{n1}\ y_{n2}\ y_{n3}\ y_{n4}\ y_{n5}]^T$ ($||y_n|| = 1$) by the sssNet1 and stored into the corresponding folder of label with the highest probability among five outputs y_{ni} from the Softmax function. It was observed that the numbers of 31, 62, 69 and 119 in the images with a burr, 40, 81, 120, 139, 140, 156, 167 and 180 in the images with a crack, and 15, 16, 40, 115, 116, 140, 197 in the images with protrusion were mis-classified.

Next, those 19 (=4+8+7) images mis-classified in the classification test were newly added into each original training data set to make them 10004, 10008 and 10007 respectively, so that the updated training data set consequently consists of 50019 images containing images without a defect. The additional training means to successively retrain the already trained sssNet1 without initializing the weights. The retrained sssNet1 is named sssNet2. Then, some test images with the features of burr, crack, protrusion and chipping were tested using the sssNet2. The result showed that the test images could be successfully classified with a recognition accuracy of 100 %.

By the way, it was observed from the result of evaluating the test images using the sssNet1 that several test images with a small feature of crack were mis-classified as non-defectives. This seems to be caused by that few images with similar small defect features were included in the training data set of the crack. In training a CNN, even if training images belong to the same category, the more training images with different types of characteristics, the more generalization ability and recognition accuracy can be enhanced.

4 DESIGN OF CASCADE-TYPE CNNS

Although it may be allowed for an inspection process of resin molded articles to mis-classify non-defective as defective, mis-classifying defective as non-defective must be prevented. Defective products must not be mixed into the lot of non-defective products. This case is one of binary classification problems, so that it is not necessary to recognize the types of the defects. All needed is to clearly classify a target image into non-defective or defective.

In this section, a cascaded-type CNN architecture as shown in Fig. 5 is proposed for binary classification. The proposed architecture consists of multiple CNNs

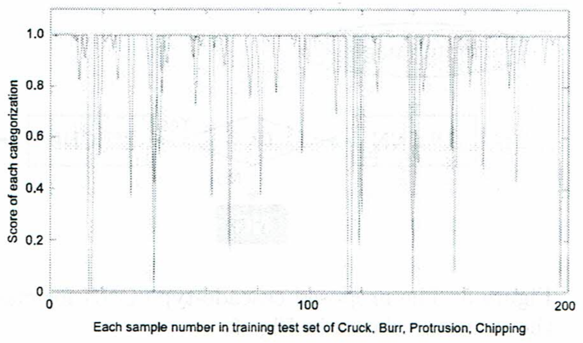


Figure 4: Categorization scores of test images with 4 kinds of defects (4 categories × 200 samples).

with 15 layers and two outputs as shown in Fig. 6. The number of the binary-classifiers CNNs prepared is the same as that of types of the defects to be classified. The advantage of the binary-classifier CNN is that it can be effectively trained in a shorter time than the multi-class CNN as shown in Fig. 3. Firstly, the CNN as shown in Fig. 3 is expanded to another multi-class CNN with 6 outputs for comparison which can classify images into six categories, i.e., non-defective, burr, crack, chipping, protrusion and spot. Then, a cascaded-type CNN consisting of 5 kinds of binary-classifier CNNs is proposed as shown in Fig. 6. The 5 kinds of binary-classifier CNNs classify a test image into non-defective or burr, crack, chipping, protrusion and spot, respectively.

In preliminary training, the cascade-type CNN and the multi-class CNN with 6 outputs are respectively trained using 5000 images×6 categories (one non-defective and 5 defectives). In evaluation experiment, classification performances by the multi-class CNN and the cascade-type CNN were compared using each 400 test images of non-defective and 5 defectives. The evaluation was carried out with an erroneously classified rate which is the ratio between "the number of test images with defects recognized to be non-defective" and "the number of all test images with defects". Table 1 shows the comparison of mis-classified rate of images with defects using the conventional multi-class CNN and the proposed cascade-type CNN. If all the 5 binary-class CNNs in the cascade-type CNN recognize a test image as non-defective, then the image is finally classified into a file folder on non-defective. In other words, if even one binary-class CNN classifies a test image as an anomaly, then the image is immediately judged to be a defective.

ζ is a threshold value. Among the outputs $y_n = [y_{n1}\ y_{n2}]^T$ from the softmax layers in the binary-classifier CNNs, if y_{n1} which is the score of non-defective is larger than ζ, then the image is judged as non-defective. Adjusting this threshold ζ to a larger value can make it difficult to be classified as a non-defective article. It is observed from the result shown

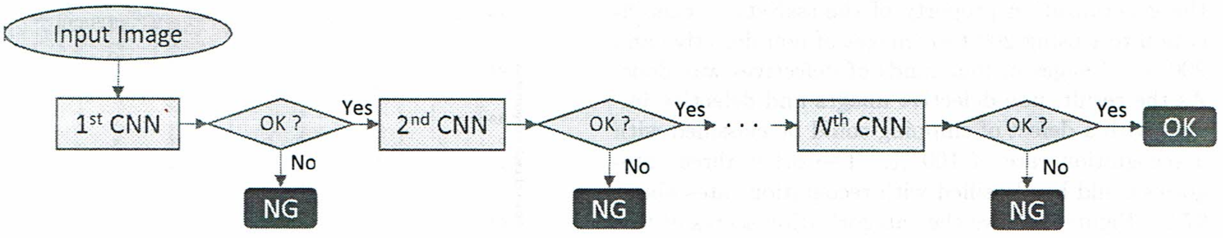


Figure 5: The proposed cascade-type CNN for classifying images into OK or NG category, in which each CNN has the structure shown in Fig. 6.

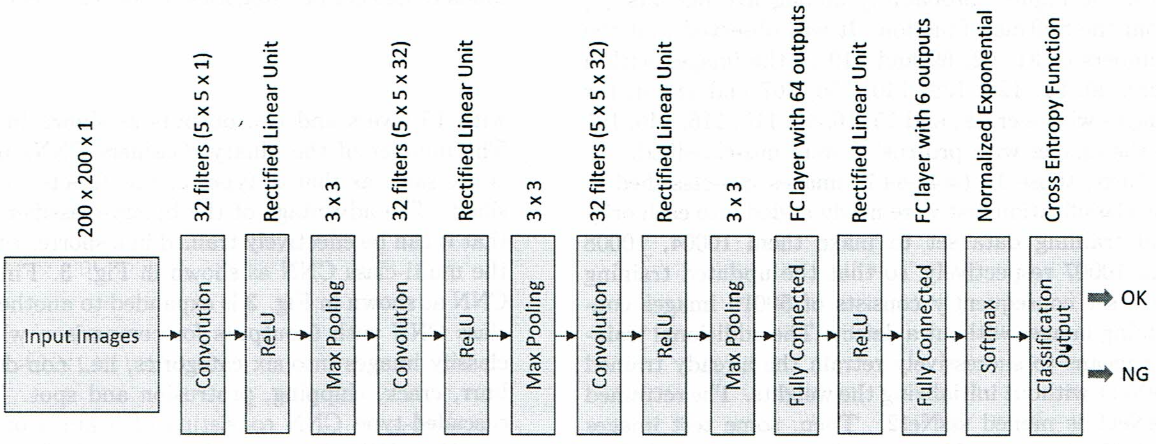


Figure 6: CNN with 15 layers and 2 outputs for binary-classifying images into OK or NG category.

Table 1: Comparison of mis-classified rate of images with defects using the conventional multi-class CNN and the proposed cascade-type CNNs as shown in Fig. 5.

CNN typeDefect	Burr	Crack	Chipping	Protrusion	Spot
Multi-class CNN	0.098	0.070	0.007	0.033	0.003
Cascade-type CNN					
$\zeta = 0.95$	0.035	0.033	0.000	0.015	0.003
$\zeta = 0.97$	0.018	0.025	0.000	0.015	0.003

in the Table 1 that the proposed cascade-type CNN could reduce the mis-classified rate of test images with defects except spot.

5 CONCLUSIONS

In this paper, a design and training tool for convolutional neural network (CNN) has been introduced. The design tool requires no knowledge and experience about C++ or Python. The authors designed a multi-class CNN with 15 layers and 5 outputs using the tool to detect undesirable defects such as crack, burr, protrusion, chipping and spot which occur in the manufacturing process of resin molded articles. After training the CNN using a large number of training data, the performance was evaluated by simple classification tests. As a result, the usefulness of the proposed CNN design and training tool was confirmed. Then, a novel architecture of cascaded-type CNN was proposed to reduce the mis-classified rate. The superiority of the cascaded-type CNN was observed through experiments compared with a multi-class CNN with 15 layers and 6 outputs. The proposed cascaded-type CNN is planned to be applied to actual physical inspection processes in future work.

References

[1] 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)*, pp. 2584–2589, Vancouver, Canada, July 2016.

[2] A. Ferreira, G. Giraldi, "Convolutional neural network approaches to granite tiles classification," *Expert Systems with Applications*, Vol. 84, No. 30, pp. 1–11, 2017.

[3] 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.

[4] F. Nagata, K. Tokuno, A. Otsuka, T. Ikeda, H. Ochi, H. Tamano, H. Nakamura, K. Watanabe and M.K. Habib, "Design tool of deep convolutional neural network for visual inspection," *Procs. of The Third International Conference on Data Mining and Big Data (DMBD2018), Springer-Nature LNCS conference proceedings 10943*, pp. 604–613, Shanghai, China, June 2018.

[5] F. Nagata, K. Tokuno, K. Watanabe and M.K. Habib, "Design application of deep convolutional neural network for vision-based defect inspection," *Procs. of 2018 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1701–1706, Miyazaki, Oct. 2018.