Artificial Life and Robotics (2021) 26:35–41
https://doi.org/10.1007/s10015-020-00618-2

ORIGINAL ARTICLE

Detection of minute defects using transfer learning-based CNN models

Kento Nakashima[1] · Fusaomi Nagata[1] · Hiroaki Ochi[1] · Akimasa Otsuka[1] · Takeshi Ikeda[1] · Keigo Watanabe[2] · Maki K. Habib[3]

Received: 14 February 2020 / Accepted: 25 June 2020 / Published online: 6 July 2020

Abstract
In this paper, a design and training tool for convolutional neural networks (CNNs) is introduced, which facilitates to construct transfer learning-based CNNs based on a series-type network such as AlexNet, VGG16 and VGG19 or a directed acyclic graph (DAG)-type network such as GoogleNet, Inception-v3 and IncResNetV2. Minute defect detection systems are developed for resin-molded articles by transfer learning of AlexNet. AlexNet has the shallowest layer structure and the smallest number of weights within the six powerful networks, so that it is selected as the first CNN for evaluation. In the transfer learning process, after the last fully connected layers are replaced according to the number of categories needed for new tasks, an additional fine training is conducted using training images including small typical defects. In experiments, transfer learning-based AlexNet_6 and AlexNet_2 are obtained to deal with six and binary classification tasks, respectively. Then, our originally designed 15 layers CNNs named sssNet_6 and sssNet_2 are also prepared and trained for comparison. Finally, AlexNet_6 and sssNet_6, AlexNet_2 and sssNet_2 are quantitatively compared and evaluated through classification experiments, respectively.

Keywords Convolutional neural network (CNN) · CNN design tool · Transfer learning · Minute defect detection

1 Introduction

In this decade, there have been many researches that applied convolutional neural network (CNN) to the defect detection process of industrial products. For example, Tada et al. studied the defect detection on the inner surface of automobile tires using CNN [1]. Young et al. proposed a vision-based approach, which used a deep CNN in order to detect concrete cracks seen at bridges and dams without conducting calculation of the defect features [2]. Tokuno et al. have developed a CNN design tool and researched on a CNN-based visual inspection system to classify images of resin-molded articles as shown in Fig. 1 into non-defective or defective [3, 4], so that the cascade-type CNNs were proposed and the effectiveness was confirmed through experiments [5].

As for transfer learning, Xian et al. reported the defect detection system for capsules based on parameter transfer learning [6], in which weights and offsets of deep belief network (DBN) trained for solar cells were used as the initial values for another DBN for capsules. As a result, training time and number of iterations could be reduced. Ferguson et al. challenged a task to realize the automatic localization of casting defects with CNNs [7], in which several different types of CNN architectures including a transfer learning approach were compared to localize casting defects in X-ray images.

The objective of this study is to compare the performance of transfer learning-based CNNs with that of not-transferred series-type CNNs in classifying simulated images of resin-molded articles as shown in Fig. 1, and to obtain related novel knowledge and future direction for higher classification accuracy and reliability. The images in Fig. 1 contain

This work was presented in part at the 25th International Symposium on Artificial Life and Robotics (Beppu, Oita, January 22–24, 2020).

✉ Fusaomi Nagata
nagata@rs.socu.ac.jp

[1] Graduate School of Engineering, Sanyo-Onoda City University, 1-1-1 Daigaku-Dori, Sanyo-Onoda 756-0884, Japan

[2] Okayama University, Okayama, Japan

[3] American University in Cairo, Cairo, Egypt

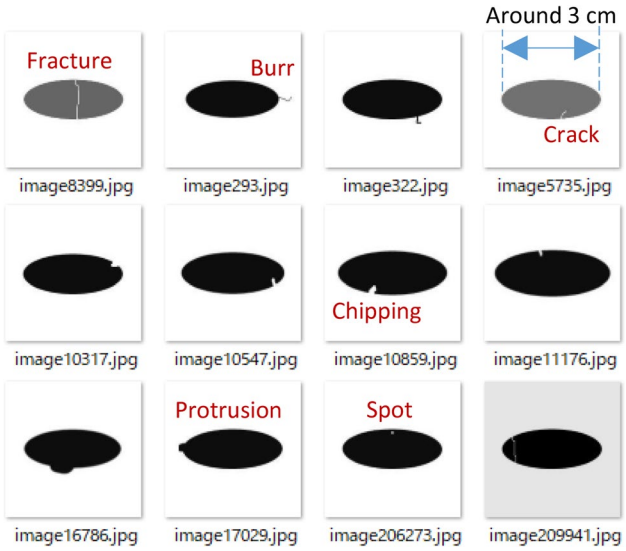

Fig. 1 Gray-scaled sample images (200 × 200) including small defects for training, validation and test

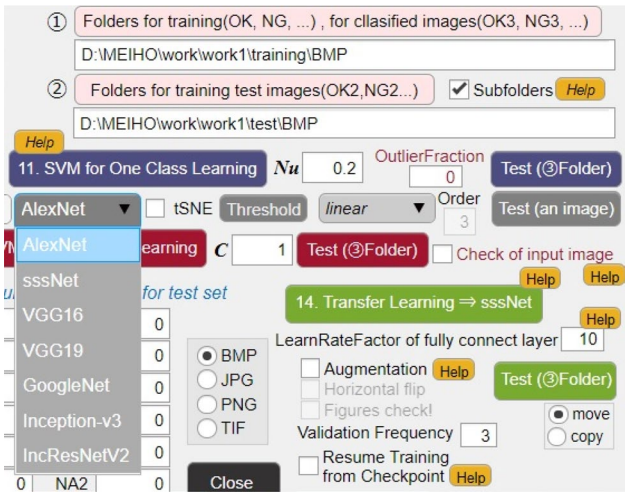

Fig. 2 A part of main dialogue developed on MATLAB system to user-friendly design transfer learning-based CNNs

five typical defects such as burr, crack, chipping, protrusion and spot. The resolution and format are 200 × 200 pixels and 256 gray scale, respectively. A data set consisting of these images is used for transfer learning. The detection performance of minute defects seems to be more influenced by the pixel sizes, i.e., resolutions of a target object and its defect in each image rather than their actual sizes.

In this paper, the authors develop a CNN design tool which can easily construct transfer learning-based CNNs. Figure 2 shows a part of the main dialogue of the developed software. As shown in Fig. 2, series networks such as AlexNet, VGG16, VGG19 and directed acyclic graph (DAG) networks such as GoogleNet, Inception-v3, IncResNetV2 can be used for transfer learning. In preliminary experiments, these CNN models were briefly tested in order to check to be able to newly design transfer learning-based CNNs. Consequently, since the 200 × 200 gray-scaled images as shown in Fig. 1 targeted here are tried to be classified into six categories at most, AlexNet with the shallowest layer structure and the smallest number of weights within the six networks is selected to design transfer learning-based CNNs which are compared with not-transferred CNNs named sssNet. Two kinds of CNNs obtained with AlexNet-based transfer learning are tried to be designed to detect minute defects seen in resin-molded articles with higher accuracy than the already proposed method [8]. In the first experiment, transfer learning-based AlexNet_6 and AlexNet_2 are designed and trained for six and two classifications, respectively. In the second experiment, our original 15-layered CNNs named sssNet_6 [8] and sssNet_2 are designed and similarly trained for six and two classifications, respectively. AlexNet_6 and sssNet_6, AlexNet_2 and sssNet_2 are quantitatively compared and evaluated by classifying test images, respectively. The usability of the CNN design tool available for transfer learning is also evaluated.

2 Defect detection using transfer learning of AlexNet

AlexNet as shown in Fig. 3 is used as an already trained superior CNN, which was the winner of ImageNet Large Scale Visual Recognition Challenge 2012. AlexNet was originally designed to classify input images into 1000 different object classes, including keyboards, mice, pencils and many kinds of animals [9], and trained using over 1.3 million images in the ImageNet database. In the transfer learning in this paper, the last three layers consisting of fully connected, softmax and classification output are replaced according to the number of categories needed for new tasks, then the reconfigured CNN is finely trained on target images of the articles.

2.1 Transfer learning based on AlexNet

In this subsection, CNNs based on the transfer learning from AlexNet are designed. As shown in Fig. 3, AlexNet is composed of a total of 25 layers including five convolutional layers and three fully connected ones. When an input image of 227 × 227 × 3 is given to the first convolution layer, 96 feature maps are generated by 96 filters with the size of 11 × 11 × 3. The second convolution layer receives the feature maps from the first convolution layer and generates 256 feature maps using 256 filters of 5 × 5 × 48. The following convolution layers after the second one have 384 filters of size 3 × 3 × 256, 384 filters of size

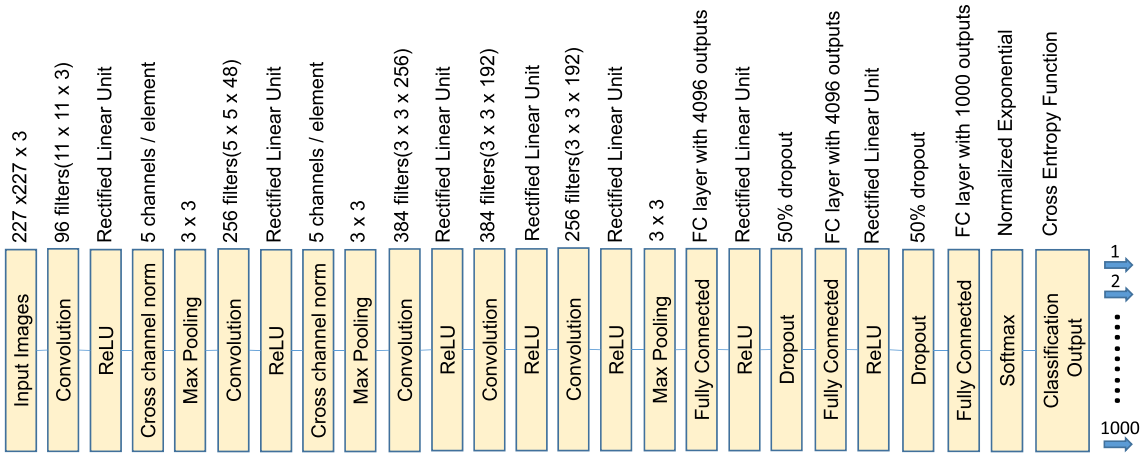

Fig. 3 Architecture image of original AlexNet used for transfer learning

$3 \times 3 \times 192$, 256 filters of size $3 \times 3 \times 192$, respectively. Although AlexNet has grouped type convolutional layers, generally speaking, filters in each convolutional layer has channels as many as feature maps in the previous convolutional one; feature maps as many as the filters are generated in the convolutional layer. The fully connected layers following the fifth convolutional layer have $4096 \times 4096 \times 1000$ weights.

In the transfer learning, the weights of the pretrained CNN model AlexNet are set as initial values for a different task and the last fully connected layers are replaced to fit the target task, then the transferred CNN model is additionally fine-trained. It is known that not only fine-tuning of a transferred CNN model is much faster and easier than that using randomly initialized weights from scratch, but also it can be completed with a smaller number of training images.

2.2 Design, fine-tuning and classification experiment

First of all, AlexNet_6, which was reconfigured for six classification by executing transfer learning of AlexNet, was designed. Figure 4 shows the structure of the designed AlexNet_6. A data set for training, verification and test was prepared for the transfer learning. 60,000 images consisting of defective-free 10,000 articles and $10,000 \times 5$ types of defective ones were used for training. In order to prevent undesirable over learning during the training process, 18,000 images ($3,000 \times 6$ categories) were also used for verification at the end of every training iteration. The iteration on training and verification was continued until the accuracy and loss to the training data reached to 0.999 and 0.001, respectively. It was confirmed that setting the learning rate, e.g., 0.0001 in convolutional layers smaller than that ,e.g.,

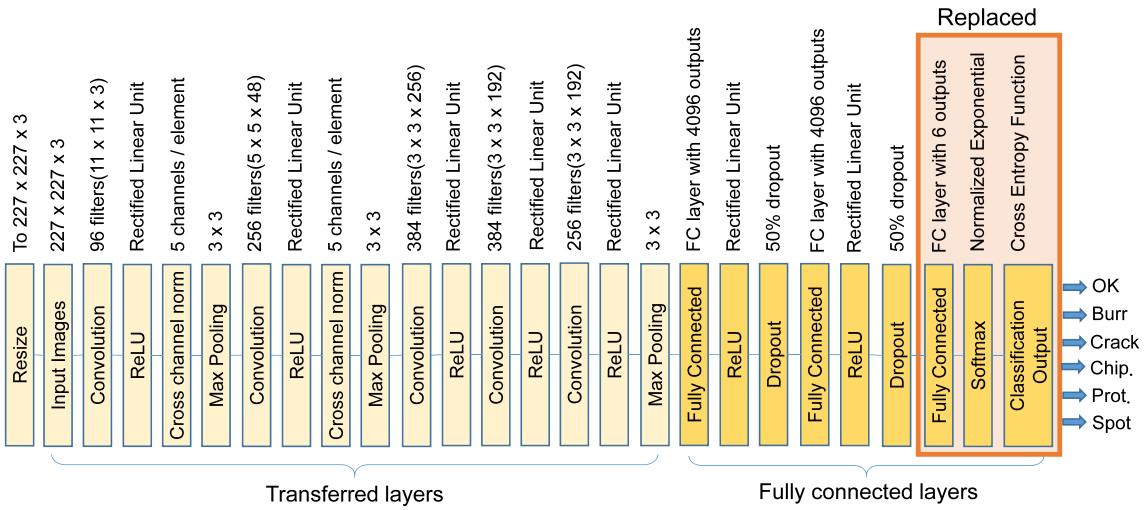

Fig. 4 Transfer learning-based AlexNet_6 designed using the proposed CNN design application shown in Fig. 2

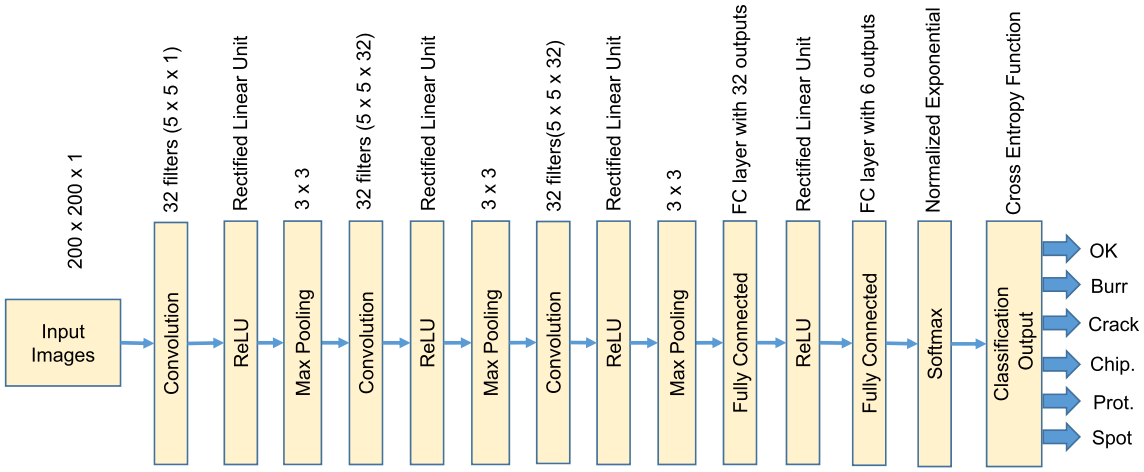

Fig. 5 Our original CNN named sssNet_6 designed using the proposed application shown in Fig. 2

Table 1 Number of misclassified images using AlexNet_6

OK	Burr	Crack	Chip.	Prot.	Spot
34	108	73	76	347	76

0.001 in replaced fully connected layers enabled faster convergence in training process. Similarly, AlexNet_2 was also designed and trained to deal with binary classification problem of defective article or non-defective one. Main specification of PC used for training is as follows: CPU, Core i7; main memory, 48 GB; GPU, GTX-1080 with 8 GB memory.

Next, 1000 test images were prepared for each class in order to evaluate the generalization performance of the trained AlexNet_6 and AlexNet_2. Then, classification experiments were conducted using the test images which had not been included in the training data set. Table 1 shows the numbers of images misclassified by AlexNet_6. Table 2 shows the confusion matrix that represents the overall recognition result of AlexNet_6, in which the vertical and horizontal axes mean actual and predicted classes, respectively. Table 3 shows the numbers of images misclassified by AlexNet_2. It is observed from Table 2 that 89 images were misrecognized as non-defective articles in spite of including some defects. In addition, 13 images were misclassified as non-defective articles despite including some defects as shown in Table 3.

3 Defect detection using CNNs trained based on actual target images

3.1 Design of our original CNN named sssNet

In this subsection, a comparative evaluation is described. Figure 5 shows the CNN named sssNet_6 consisting of 15 layers capable of a classification of 6 categories[8]. Similarly, sssNet_2 is also designed for a binary classification which can classify images into non-defective or defective. Note that although the two CNNs are shallower than AlexNet, they are originally trained with randomly initialized weights from scratch using actual target images as shown in the Fig. 1.

Table 3 Number of misclassified images using AlexNet_2

OK	NG				
	Burr	Crack	Chip.	Prot.	Spot
0	12	0	0	1	0

Table 2 Confusion matrix by AlexNet_6 (row: true labels, column: predicted labels)

	OK	Burr	Crack	Chip.	Prot.	Spot
OK	966	0	0	0	0	34
Burr	36	892	0	25	21	26
Crack	11	5	927	4	0	53
Chip.	14	6	53	924	3	0
Prot.	28	215	13	91	653	0
Spot	0	0	76	0	0	924

Table 4 Number of misclassified images using sssNet_6[8]

OK	Burr	Crack	Chip.	Prot.	Spot
0	248	378	364	485	262

In training, the input images as shown in Fig. 1 are given to the first layer and normalized. The second, fifth and eighth convolutional layers have filters composed of $5 \times 5 \times 1 \times 32$, $5 \times 5 \times 32 \times 32$ and $5 \times 5 \times 32 \times 32$ structures, respectively. In the convolution layers, feature extraction elements called filters were applied to the input images or feature maps while shifting the number of pixels specified by a parameter called stride from the upper left to the lower right.

3.2 Classification experiment by sssNets

sssNet_6 and sssNet_2 were trained for six and binary classifications, respectively. Note that training images and other termination conditions were the same as the experiments using AlexNet_6 and AlexNet_2 in the previous section. Then, in order to evaluate the generalization performance of trained sssNet_6 and sssNet_2, classification experiments were conducted using the same test images as the experiments using AlexNet_6 and AlexNet_2.

Table 4 shows the number of images misrecognized by sssNet_6. Table 5 shows the confusion matrix indicating the overall recognition result of sssNet_6. Also, Table 6 shows the numbers of images misclassified by sssNet_2. It is observed from Table 5 that 123 images were misrecognized as non-defective articles in spite of including small burr and protrusion. Also, for example, when sssNet_2 evaluated 1,000 images containing a burr, 61 images were misrecognized as non-defective as shown in Table 6. In addition, 109 images were misclassified as non-defective products in spite of including one of defects as shown in Table 6.

Table 6 Number of misclassified images using sssNet_2

OK	NG				
	Burr	Crack	Chip.	Prot.	Spot
1	61	26	0	22	0

Table 7 Comparison results of AlexNet_6 and sssNet_6

	Accuracy rate	Precision rate
AlexNet_6	0.8810	0.8869
sssNet_6	0.7105	0.7277

4 Comparison results and discussion

4.1 Classification into six categories

AlexNet_6 is quantitatively compared with the sssNet_6 using the results obtained in the previous section. Table 7 shows the results based on the criteria of accuracy and precision rates between AlexNet_6 and sssNet_6. First of all, the meaning of each rate is explained. For example, T_{Ok} is the number of images in which non-defective articles are correctly classified. On the other hand, F_{Bu}, F_{Cr}, F_{Ch}, F_{Pr} and F_{Sp} mean the number of images in which non-defective articles are misclassified into burrs, cracks, chips, protrusions and spots, respectively. The accuracy rate A_c of non-defective articles is given by

$$A_c = \frac{T_{Ok}}{T_{Ok} + F_{Bu} + F_{Cr} + F_{Ch} + F_{Pr} + F_{Sp}}. \tag{1}$$

Accuracy rates of other categories can be calculated in the same manner. Table 7 shows a comparison result between AlexNet_6 and sssNet_6, in which mean values among the accuracy rates of the six categories are written.

Next, another criterion P_r called precision rate is explained. P_r is calculated by

$$P_r = \frac{T_{Ok}}{T_{Ok} + F_{Ok}}, \tag{2}$$

where F_{Ok} means the number of images of defective articles which are unfortunately misclassified as non-defective. The

Table 5 Confusion matrix by sssNet_6 (row: true labels, column: predicted labels)[8]

	OK	Burr	Crack	Chip.	Prot.	Spot
OK	1000	0	0	0	0	0
Burr	93	752	97	2	37	19
Crack	0	0	622	0	0	378
Chip.	0	25	172	636	144	23
Prot.	30	306	64	66	515	19
Spot	0	0	262	0	0	738

Table 8 Comparison results of AlexNet_2 and sssNet_2

	Accuracy rate	Precision rate
AlexNet_2	0.9978	1.0000
sssNet_2	0.9817	0.9998

precision rate of other categories can be obtained in the same manner. Note that the precision rates in Table 7 are also the mean values of six categories.

It is confirmed from Table 7 that AlexNet_6 is superior to sssNet_6 in terms of both accuracy and precision. Besides this comparison, it is also confirmed from Tables 1 and 4 that the number of images misclassified by AlexNet_6 is fewer than that of sssNet_6.

4.2 Classification into two categories

AlexNet_2 and sssNet_2 are similarly compared on the accuracy and precision rates. Table 8 shows the compared results; T_P is the number of images of which defective articles are correctly classified; T_N means that of which non-defective articles are correctly classified; F_P is that of which defective articles are misclassified as non-defective; F_N is that of which non-defective articles are misclassified as defective. Using these variables, accuracy rate A_c of non-defective articles is given by

$$A_c = \frac{T_P + T_F}{T_P + T_N + F_P + F_N}. \tag{3}$$

Also, precision rate P_r of non-defective articles is calculated by

$$P_r = \frac{T_P}{T_P + F_P}. \tag{4}$$

It is confirmed from Table 8 that AlexNet_2 is superior to sssNet_2 in terms of both the criteria, i.e., accuracy and precision rates. Besides this comparison, it is also confirmed from Tables 3 and 6 that the number of images misclassified by AlexNet_2 is fewer than that of sssNet_2.

Various types of CNNs have been designed and introduced in this decade. When CNNs are newly designed for defect detection processes in industrial fields, it is strongly required from the aspect of maintenance, renovation and additional training that the networks should have shallower layer structures and a smaller number of weights as much as possible. It has been empirically obtained from the results as shown in Table 8 that CNNs should have at least three basic blocks consisting of convolution, ReLU and max pooling layers as shown in Fig. 5 in order to cope with the binary classification task of the 200 × 200 gray-scaled images. However, the structure shown in Fig. 5 was not effective in the six classification problem as shown in Table 7. The articles in the test images have similar minute defects or non-defective, so that accurate classification seemed to be difficult as shown in Table 7. In future work, the number of basic blocks in sssNet is planned to be increased.

4.3 Identification of the most activated area

Figure 6 shows examples of visualization of the most activated defective area when sssNet_2 is used. The developed option dialogue provides the effective function to identify the most activated area of each input image in classification by convoluting the generated feature maps. However, it was difficult for AlexNet-based transfer learning CNN to validly emphasize the most activated area of each input image in classification, because the convolutional layers of AleXnet were originally trained using a large number of images of 1,000 categories, which are quite different from the target images simulating resin-molded articles.

5 Conclusions

In this paper, a CNN design and training tool which can facilitate to construct transfer learning-based CNNs was developed for visual inspection. One of series networks as Alexnet, VGG16, VGG19 or DAG networks as GoogleNet, Inception-v3, IncResNetV2 can be selected for the transferred CNN. In the test trials of AlexNet-based transfer learning, the last fully connected layers were easily replaced and finely trained according to the number of actual categories needed in new tasks. It was confirmed that the additional fine training could be efficiently completed using target training images by means of setting the learning rate

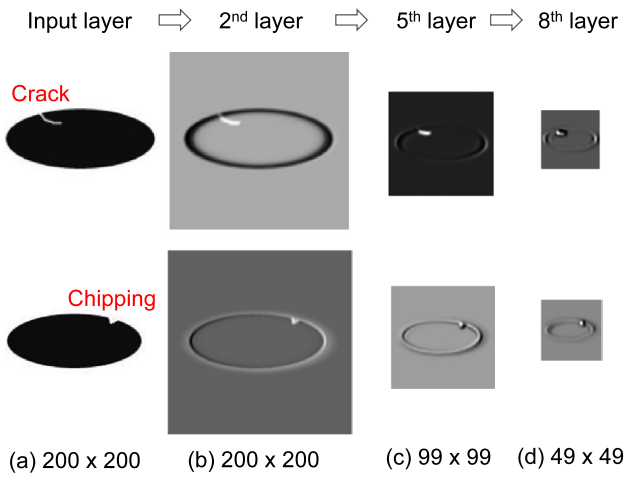

Fig. 6 Examples of visualization of the most activated area when sssNet_2 is used

of convolutional layers smaller than that of fully connected layers. It was also confirmed from classification experiments that transfer learning-based AlexNet_6 and AlexNet_2 were superior to sssNet_6 and sssNet_2 trained with randomly initialized weights from scratch in terms of both six and binary classification problems. It was expected from these results that transfer learning is an effective method to design superior CNNs with higher accuracy for visual inspection systems in spite of small number of training images and less training time. In future work, other more powerful series networks and DAG ones are planned to be applied to transfer learning for comparison.

However, on the other hand, it was difficult for the AlexNet-based transfer learning CNN to clearly identify the most activated area of each input image in classification. If some effective visualization function of the most activated defective area in actual target images is required, it seems that the CNN to solve the classification problem should be originally designed without using the transfer learning approach and trained using randomly initialized weights and actual target images.

Acknowledgements This work was partially supported by JSPS KAKENHI Grant Number 16K06203 and MITSUBISHI PENCIL CO., LTD.

References

1. Tada H, Sugiura A (2017) Defect classification on automobile tire inner surfaces using convolutional Neural networks. Comput Commun Control Autom pp 17–18
2. Cha Y, Choi W, Buyukozturk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civ Infrastruct Eng pp 1–18
3. Tokuno K, Nagata F, Otsuka A, Watanabe K (2018) Defect inspection system using deep convolutional neural networks. Proc Jpn Soc Mech Eng pp 35–39 (in Japanese)
4. Tokuno K, Nagata F, Otsuka A, Watanabe K (2018) Defect inspection system using deep convolutional neural networks. In: Proceedings of the 2018 JSME conference on robotics and mechatronics (ROBOMECH2018) C2A2–K14, 3 pages, (in Japanese)
5. Tokuno K, Nagata F, Otsuka A, Watanabe K, Habib MK (2019) Design tool of convolutional neural network (CNN)- design of cascade-type CNN and Its application to defect detection. In: Proceedings of 24th international symposium on artificial life and robotics, pp 733–737
6. Ri-Xian L, Ming-Hai Y, Xian-Bao W (2015) Defects detection based on deep learning and transfer learning. Metall Mine Ind 7:312–321
7. Ferguson M, Ak R, Lee YTT, Law KH (2017) Automatic localization of casting defects with convolutional neural networks. In: Proceedings of 2017 IEEE international conference on big data (Big Data), pp 1726–1735
8. Nakashima K, Nagata F, Watanabe K (2019) Basic research on detection of defective products with minute defects using convolution neural network (CNN) and support vector machine (SVM). In: Proceedings of the 2019 JSME conference on robotics and mechatronics (ROBOMECH2019), 2A1-Q05, pp 4 (in Japanese)
9. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Neural information processing systems conference, pp 1–9