DOI: 10.1007/s40684-016-0039-x ISSN 2288-6206 (Print) / ISSN 2198-0810 (Online)

Machine Learning-Based Imaging System for Surface Defect Inspection

Je-Kang Park[1], **Bae-Keun Kwon**[2], **Jun-Hyub Park**[3], and **Dong-Joong Kang**[1,#]

1 School of Mechanical Engineering, Pusan National University, 2, Busandaehak-ro, 63beon-gil, Geumjeong-gu, Busan, 46241, South Korea
2 LG Electronics, 170, Seongsanpaechong-ro, Seongsan-gu, Changwon-si, Gyeongsangnam-do, 51533, South Korea
3 Department of Mechatronics Engineering, Tongmyong University, 428, Sinseon-ro, Nam-gu, Busan, 48520, South Korea
# Corresponding Author / Email: djkang@pusan.ac.kr, TEL: +82-51-510-2356, FAX: +82-51-514-1118

KEYWORDS: Defect detection, Deep neural networks, Machine learning, Machine vision

Modern inspection systems based on smart sensor technology like image processing and machine vision have been widely spread into several fields of industry such as process control, manufacturing, and robotics applications in factories. Machine learning for smart sensors is a key element for the visual inspection of parts on a product line that has been manually inspected by people. This paper proposes a method for automatic visual inspection of dirties, scratches, burrs, and wears on surface parts. Imaging analysis with CNN (Convolution Neural Network) of training samples is applied to confirm the defect's existence in the target region of an image. In this paper, we have built and tested several types of deep networks of different depths and layer nodes to select adequate structure for surface defect inspection. A single CNN based network is enough to test several types of defects on textured and non-textured surfaces while conventional machine learning methods are separately applied according to type of each surface. Experiments for surface defects in real images prove the possibility for use of imaging sensors for detection of different types of defects. In terms of energy saving, the experiment result shows that proposed method has several advantages in time and cost saving and shows higher performance than traditional manpower inspection system.

Manuscript received: April 15, 2015 / Revised: May 28, 2016 / Accepted: June 2, 2016
This paper was presented at ISGMA 2015

1. Introduction

Machine vision has become faster and more precise along with the development of small and cost-effective hardware and has become widely used in various industrial fields. Various systems using machine vision,[1] such as pose control of robots using visual servoing[2] and, on-line process control using part component recognition,[3] increase the efficiency of factory automation and the possibility of a smart factory. In particular, the surface inspection systems using vision sensors has taken the spotlight as a tool performing a non-contact inspection that is not influenced by the type of target, condition of the surface, electromagnetic field, or temperature. Moreover, machine vision can perform defect detection, texture and shape classification, and other additional inspections at the same time. In the past, the surface inspection was performed to detect defects on non-texture surfaces such as flat intensity with no variation, but it is now used on more complicated-texture surfaces. Over the past few decades, inspection systems can gain the ability to detect difficult defects such as tiny and blurred artifacts, which have been conventionally inspected with the naked eye of humans, and to perform more advanced inspections like texture regulation and alignment measurement. In conventional studies, defects on complicated-texture surfaces of wood,[4] fabric,[5] and metal[6] have been detected by analyzing the wavelet of surface backgrounds, and many researchers have studied such methods as ICA,[7] Gabor filter[6] and Co-occurrence matrix[8] so far.

Recently, vision systems have made remarkable progress in performance by combining machine learning algorithms. Conventional methods using the frequency and regularity of textures were greatly affected by hand-crafted parameters set up by users, but machine learning-based methods can provide probable results through learning the features of training data. Accordingly, they show better results, and many studies are being conducted on defect detection using learning algorithms. For a surface defect inspection, machine learning is largely divided into the classification part of objects and backgrounds and the extraction part of image features. A. Cord[9] performed a road defect detection based on adaptive-boosting by using textural patterns as features, and S. Ding[10]

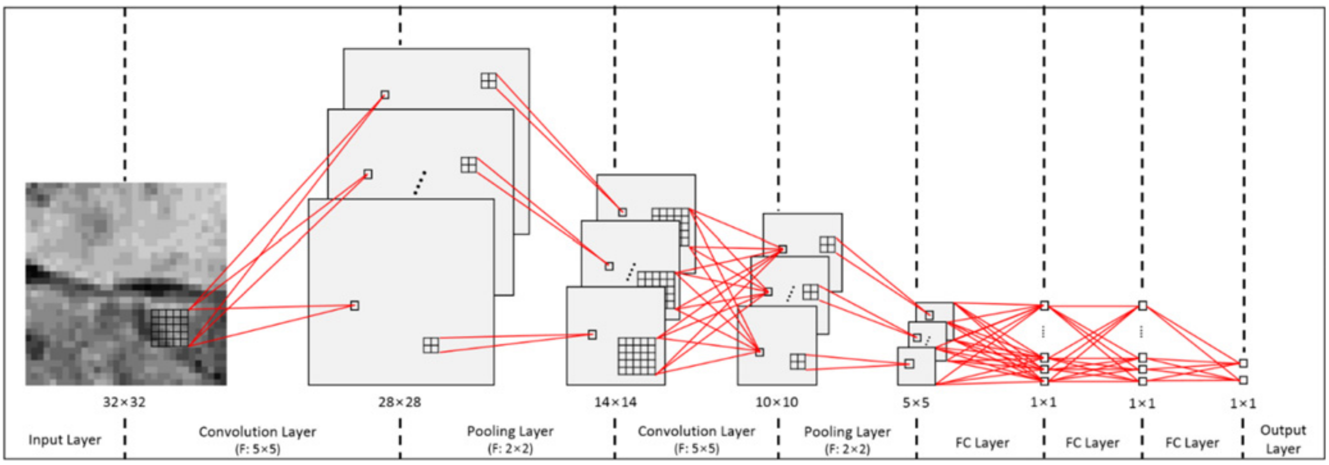


Fig. 1 The Structure of CNN in case of experiment No. 2 in Table 2. Red edges show connection weights between nodes and learned during training of network

introduced a fabric defect inspection system using SVM (Support Vector Machine) that had learned HOG (Histogram of Gradient) features.

In modeling image features, it is necessary to find discriminative features that can appropriately classify objects and backgrounds because the results of training may differ depending on the feature types. Therefore, feature extraction algorithms are quite important for machine learning, and choosing or developing features with high discriminative power determines the reliability of a defect inspection system. A machine learning-based defect inspection adopts a statistical classification, but users have to select features, and its performance can be limited by how optimally features are selected.

In this paper, we used CNN (Convolution Neural Network) to solve the shortcomings of the existing machine learning. As a kind of machine learning and a special type of neural network with deep layer architecture, CNN performs feature extraction and recognition at the same time on a single network, while conventional machine learning methods obtain recognition results in two separated modes, which are learning effective features and extracting feature vectors from input patterns through a feature extraction algorithm. CNN proceeds with increasing the abstraction level by extracting features from low-level layers to high-level layers. Therefore, lower layers of the network perform extraction of detailed features, higher layers combine features of low level, and fully connected (FC) layers in the output part of the network perform classification. As a result, CNN can directly recognize patterns without handcrafted feature extraction. Since feature extraction and classification are simultaneously performed in a neural network, features fit for the classification are automatically carried out, further improving performance. This paper have built and tested several types of deep networks to select adequate structure for surface defect detection because most of conventional CNN-based approaches are too much complex for surface pattern inspection and have been used for 3D object recognition such as cars, humans, animals, etc.

2. Convolution Neural Network

2.1 Related Research

As a neural network developed to imitate the process of human brain processing visual images, CNN was proposed for Neocognitron[11] in 1980, and since LeCun[12] applied the gradient-based method to CNN, it has been widely used in the field of image recognition. However, composing a deep neural network requires a great deal of sample data and training time, and since Shallow Neural Network was not able to classify complex data, its use was gradually reduced. Recently, however, since GPU-based parallel processing has become popular and big data can be collected easily, CNN has attracted more attention again.[13] CNN shows high performance when applied to natural language processing,[14] speech recognition,[15] and image processing,[16] each of which requires big data with many classes. Even though CNN has shown favorable results in the field of classification, very few studies have been were conducted in industrial applications such as defect detection and surface inspection systems. Some researchers such as Soukup[17] and Yao[18] used CNN for defect detection; however Soukup et al. have applied on a limited type of surfaces such as steel defects from photometric stereo system. Yao et al. have suggested the hand-crafted image features such as energy, variance, and entropy to enhance defects pattern. However, Yao's method is not end-to-end convolutional network but cellular neural network for only classification.

2.2 Construction of CNN

A neural network is composed of more than one layer, and each layer contains many nodes. Nodes in each layer combine the output of the lower layer in a form of weighted sum. Each layer performs feature abstraction by the value of weight, and the relation between layers is expressed as a weight matrix. Output X_j^n between layer 'n' and the layer 'n-1' is calculated through Eq. (1) based on the output X_j^{n-1} of the n-1 layer and weight w_{ij}^n between the two layers. At this point, θ_i^n is the bias of node I, and f is the sigmoid function.

$$X_j^n = f\left(\sum_i w_{ij}^n X_i^{n-1} + \theta_j^n\right) \tag{1}$$

When the weight is trained through sufficient data, it is possible to extract high-level features of the upper layer by combining low-level features from the output of the lower layer. As data put into the bottom layer pass through the other layers, feature abstraction is gradually performed, further leading to feature selection in a higher level. As a

Algorithm 1 Training process of CNN (for one batch iteration)

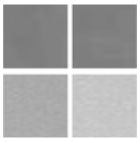
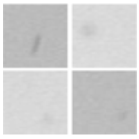
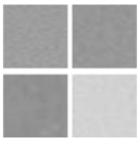
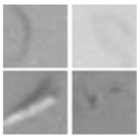
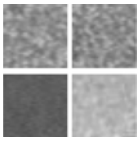
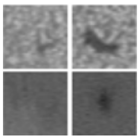
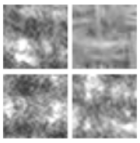
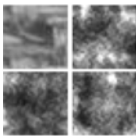
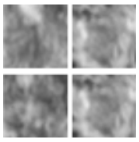
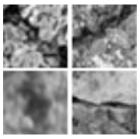
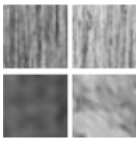
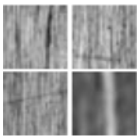
```
# Notation
$F^{conv}(X,w)$ : Eq. (2)
$F^{pool}(X)$ : Eq. (3)
$F^{FC}(X,w)$ : Eq. (1)
$F^{soft\ max\ loss}(X)$ : Eq. (4,5)
$b$ : batch size
$w^{layer}$ : weights of corresponding layer
$\alpha$ : learning rate
Structure of CNN : 2 convolution, 2 pooling, 2 FC layers
Input : $I_k$ $(k=1,2,...,b)$
# Estimation step (feed-forward process)
for $k = 1$ to $b$
    $X_k \leftarrow F^{conv1}(I_k, w^{conv1})$
    $X_k \leftarrow F^{pool1}(X_k)$
    $X_k \leftarrow F^{conv2}(X_k, w^{conv2})$
    $X_k \leftarrow F^{pool2}(X_k)$
    $X_k \leftarrow F^{FC1}(X_k, w^{FC1})$
    $X_k \leftarrow F^{FC2}(X_k, w^{FC2})$
    $L_k \leftarrow F^{soft\ max\ loss}(X_k)$
end for

# Training step (feed-backward process)
for $k = 1$ to $b$
    $w^{FC2} \leftarrow w^{FC2} - \frac{\alpha}{b}\frac{\partial F^{FC2}(X_k, w^{FC2})}{\partial w^{FC2}}$
    $w^{FC1} \leftarrow w^{FC1} - \frac{\alpha}{b}\frac{\partial F^{FC1}(X_k, w^{FC1})}{\partial w^{FC2}}$
    $w^{conv2} \leftarrow w^{conv2} - \frac{\alpha}{b}\frac{\partial F^{conv2}(X_k, w^{conv2})}{\partial w^{conv2}}$
    $w^{conv1} \leftarrow w^{conv1} - \frac{\alpha}{b}\frac{\partial F^{conv1}(X_k, w^{conv1})}{\partial w^{conv1}}$
end for
```

result, with more layers, more high-level features are extracted, and the performance of classification improves as well. However, it requires a large number of sample data and more training time as well.

CNN consists of an input layer where data can be input, an output layer where the class type can be output, and many hidden layers responsible for feature detection, as shown in Fig. 1. The output layer with FC layers is responsible for classifying high-level features extracted from the hidden layer. The hidden layer is composed of convolution layers and pooling layers. In the network, the input layer handles data input and the output layers classify the feature with two or three FC layers. The number of FC layers may depend on the classification performance. Usually, two FC layers are enough to classify the features.

Eq. (1) shows general form for node calculation of convolutional neural network layer. In CNN, layers that are calculated by Eq. (1) are called FC layer. Convolution layers and pooling layers are calculated similar to Eq. (1) but have connection constraints between nodes in two layer.

Table 1 Sample image types for CNN training

	Class C_b^1 Silicon wafer Background		Class C_d^1 Silicon wafer Defect
	Class C_b^2 Solid paint Background		Class C_d^2 Solid paint Defect
	Class C_b^3 Pearl paint Background		Class C_d^3 Pearl paint Defect
	Class C_b^4 Fabric Background		Class C_d^4 Fabric Defect
	Class C_b^5 Stone Background		Class C_d^5 Stone Defect
	Class C_b^6 Wood Background		Class C_d^6 Wood Defect

$$X_j^n = f\left(\sum_{i-\frac{m}{2}}^{i+\frac{m}{2}} w_{ij}^n X_i^{n-1} + \theta_j^n\right) \tag{2}$$

$$X_j^n = \max\left(\left[X_{i-\frac{m}{2}}^{n-1}, X_{i+\frac{m}{2}}^{n-1}\right]\right) \tag{3}$$

Where Eqs. (2) and (3) are calculation process of convolution layer and pooling layer, respectively. In Eqs. (2) and (3), m is mask size and refer to range of input nodes that are used in calculation of output nodes. In other words, unlike the FC layer, convolution layers can learn mask values that extract feature from image. Every node in a convolution layer is connected to the image window of a lower layer. Nodes in the layer share the same weight with each other, that is, the connection edges between layers have the same role as the convolution operation of image processing. At this point, the edge weight is image mask. In image processing, a fixed value mask is used, but in CNN, the value of a mask is automatically trained from the sample data, further improving efficiency. Nodes in a pooling layer select the maximum value in small window from the output of the convolution layer that is called max pooling operation. As maximum pooling ignores small changes in the location of features, it absorbs location errors. In other words, by keeping features well-expressed through the selection of maximum values and reducing the size of images, it is possible to carry out a strong search on the scale. Classification performance gradually increases while going through several pooling

Table 2 Error rates of CNN structures with two convolution layers and two pooling layers

Exp. No.	Layer structure	# of Node / Kernel size	Error rate / Training time
No. 1	Conv. layer 1	Output node: 6 Kernel size: 5	3.698% / 388.3 sec
	Pooling layer 2	Kernel size: 2	
	Conv. layer 3	Output node: 12 Kernel size: 3	
	Pooling layer 4	Kernel size: 2	
No. 2	Conv. layer 1	Output node: 6 Kernel size: 5	2.414% / 380.4 sec
	Pooling layer 2	Kernel size: 2	
	Conv. layer 3	Output node: 12 Kernel size: 5	
	Pooling layer 4	Kernel size: 2	
No. 3	Conv. layer 1	Output node: 6 Kernel size: 5	4.713% / 387.5 sec
	Pooling layer 2	Kernel size: 2	
	Conv. layer 3	Output node: 12 Kernel size: 3	
	Pooling layer 4	Kernel size: 3	
No. 4	Conv. layer 1	Output node: 6 Kernel size: 7	2.335% / 391.3 sec
	Pooling layer 2	Kernel size: 2	
	Conv. layer 3	Output node: 12 Kernel size: 4	
	Pooling layer 4	Kernel size: 2	
No. 5	Conv. layer 1	Output node: 6 Kernel size: 7	2.312% / 379.7 sec
	Pooling layer 2	Kernel size: 2	
	Conv. layer 3	Output node: 12 Kernel size: 6	
	Pooling layer 4	Kernel size: 2	
No. 6	Conv. layer 1	Output node: 6 Kernel size: 6	3.391% / 369.5 sec
	Pooling layer 2	Kernel size: 3	
	Conv. layer 3	Output node: 12 Kernel size: 5	
	Pooling layer 4	Kernel size: 2	

layers, and classes are determined in the FC layers.

2.3 Training of CNN

In the training steps, CNN should learn optimal weights of all layers. First, training samples are composed of images I_k $(k=1,2,...,m)$ and class label vector c_k. If I_k belongs to i-th class then c_k^i is 1 and the other elements of c_k are 0. Feed-forward process of CNN estimates class vector c_k' from I_k and the elements of c_k' has probability distribution through Eq. (5). This means that c_k' and c_k should be similar to each other when CNN has been learned. Therefore, the loss functions that should be minimized is as follows.

$$L = \sum_{k=1}^{m} c_k \log h(x_k) + (1-c_k)\log(1-h(x_k)) \qquad (4)$$

$$h(x_k^i) = \frac{e_k^i}{\sum_i e_k^i} \qquad (5)$$

Where Eq. (5) is called softmax function. The softmax function generates a normalized exponential distribution from output nodes in last FC layer. In Eq. (4), $h(x_k)$ means an estimated class vector c_k'. Therefore, the loss function Eq. (4) is increasing when difference between c_k and c_k' is large. Conversely Eq. (4) is decreasing when difference between c_k and c_k' is small.

CNN has many hidden layers. To learn all the weights in the layers, loss should be back-propagated from the last FC layer to the first convolution layer. Batch gradient decent, generally used to train neural network, can propagate an error by chain rule. Therefore, to train the CNN, we use batch gradient decent algorithm and overall training process is summarized in Algorithm 1.

3. Experimental Results

Since CNN can perform feature extraction and classification at the same time, it does not need an additional feature extraction module. To maximize the classification performance, however, it is quite important how to compose convolution layers and pooling layers. With more hidden layers, the classification performance could improve, but the speed of training will slow down, and with fewer hidden layers, CNN is not able to show its maximum performance. On the contrary, too complex structure of many layers and parameters might lead to over-fitted model for even simple classification problems of two or three classes. In this paper, we find the optimal composition of layers through some experiments and verify the performance of this CNN-applied surface inspection method through comparative experiments with the other methods. The proposed CNN was composed of a consecutive linear operation. Therefore, we didn't use any neural network libraries and we have implemented CNN by just use of some basic linear algebra functions in the Matlab. We have performed the experiments on the Intel® i5 desktop without parallel processing device such as GPU acceleration.

At the learning steps, we set a learning rate by 0.01 and batch size by 32. All the images used for our experiments were grayscale with a size of 32×32 pixels, and we used 6 kinds of surfaces, including wafer, solid color paint, pearl color paint, fabric, stone, and wood, as well as 12 kinds of datasets with and without defects. Each dataset has 2,000 images, and Table 1 shows sample images arranged. Lower indexes b and d show backgrounds and defects, respectively.

3.1 Efficiency of CNN for Defect Detection

The overall performance of CNN, such as classification and training time, varies depending on the composition of hidden layers. Therefore, it is essential to compose layers in the optimal way. The composition of layers relates to the number of convolution layers and pooling layers, the number of nodes in a convolution layer, the kernel size of a convolution mask, the kernel size of a pooling mask, and so on. Elements affecting the number of layers are the size of an image and the number of sample data. Lastly, the size of an image affects the mask of a layer, and the number of classes affects the number of nodes.

For surface inspection, it is necessary to determine the size of a sample image that is large enough to express small-sized defects as well as textures, so this study used images with the size of 32×32

pixels, which is the size most frequently used for texture learning. Since the number of classes to be classified was already fixed, we conducted experiments on the optimal conditions by changing the number of layers, nodes, and kernels with the limited number of training samples. In these experiments, we used six different kinds of surfaces by using two different datasets of backgrounds and defects. After dividing a total of 4,000 images into 3,000 training sets and 1,000 test sets, we conducted a cross validation four times. The error rate and training time are the mean values of 24 iterations of experiments and error rate was calculated as follows.

$$error\ rate = 100 \times \frac{N_{incorrect\ class}}{N_{test\ samples}}(\%) \qquad (6)$$

Where $N_{test\ samples}$ is the number of test samples and $N_{incorrect\ class}$ is the number of false positives and false negatives. In addition, since too small-sized kernels increase the error rate, we used the kernel of the first convolution layer with a size of at least 5 × 5 pixels.

The first experiment was conducted with CNN composed of two convolution layers and two pooling layers. For the 6 cases of experiments, parameter settings of convolution and pooling layers are shown in Table 2. After selecting the two experimental conditions showing the best performance, we added one more convolution layer and pooling layer as a further experiment. In the current dataset, however, more layers were not used to compose CNN due to the limitation on the number of training samples. Parameter sets about these experiments are shown in Table 3. Through this experiment, we found several properties of CNN. First, when a convolution layer was composed of small kernels, it showed a tendency to increase the error rate. When 3 × 3 pixels were included in the convolution kernel, the error rate increased in all cases of the first to the sixth experiments, and the fifth experiment using the largest-size kernels showed the most remarkable results. We also found that the convolution kernels in the first layer had most effect. When carrying out the seventh and eighth experiments by adding more layers to the fourth and fifth experiments, we found out that the error rate decreased as the number of layers increased. However, when comparing the fifth experiment with the seventh, we found that the composition of kernels had more effect on the results than the number of layers. It was also found that training time was directly affected by the number of layers rather than the numbers of kernels and nodes. In an experiment with the same number of layers, training time was found not to be affected by the conditions of nodes or kernels. It was more important that the error rate decreased as the number of layers increased, but because of the training time increasing too much, the severe increase in training time was found to restrict the number of layers in terms of training efficiency.

3.2 Comparative Experiments

In this section, we conducted a comparative experiment on conventional inspection methods by with using the optimized CNN. We compared PSO-ICA, a well-known for inspection algorithm of silicon wafer, and a Gabor-filter showing high performance in texture patterns, and also compared a machine learning-based method of applying VOV features to RF (Random Forest).[19] PSO-ICA extracted the frequency information of backgrounds by using the background dataset, and RF was prepared to learn images with the same size of 32 × 32 pixels as that of CNN. The first experiment was conducted to analyze the reliability of the six kinds of surfaces separately, and the second experiment was conducted to train more than two surfaces of different type at the same time and compare them. The average error rate was calculated through the cross validation. Figs. 2 and 3 show the results of these experiments.

Table 3 Error rates of CNN structures with three convolution layers and three pooling layers

Exp. No.	Layer structure	# of Node / Kernel size	Error rate / Training time
No. 7	Conv. layer 1	Output node: 6 Kernel size: 7	2.319% / 711.9 sec
	Pooling layer 2	Kernel size: 2	
	Conv. layer 3	Output node: 12 Kernel size: 4	
	Pooling layer 4	Kernel size: 2	
	Conv. layer 5	Output node: 24 Kernel size: 3	
	Pooling layer 6	Kernel size: 2	
No. 8	Conv. layer 1	Output node: 6 Kernel size: 7	2.304% / 698.1 sec
	Pooling layer 2	Kernel size: 2	
	Conv. layer 3	Output node: 12 Kernel size: 6	
	Pooling layer 4	Kernel size: 2	
	Conv. layer 5	Output node: 24 Kernel size: 3	
	Pooling layer 6	Kernel size: 2	

PSO-ICA is a method designed for wafer, and it shows the best performance on smooth surfaces. It also shows the most reliable defect inspection results in wafer containing low-contrast defects. However, as the texture gets stronger, it shows less reliability, and shows the worse results on the surface of fabric, stone, and wood. Conversely, the Gabor filter was designed to inspect the surface of materials with strong textures. As a result, as the texture gets stronger, it shows better results, especially on the surfaces of stone and wood. Since it is not a learning-based method, the Gabor filter has the merit of not needing a preparation step for various kinds of surfaces. RF does not show the most remarkable results on a specific surface, but it shows favorable reliability on average, compared to the other methods, except the proposed method.

Our method shows the best results on all the surfaces, except wafer and solid surface. Moreover, the results showed no major difference with those of PSO-ICA on wafer and solid surface. As learning-based methods, both RF and CNN showed better results on average than the other methods and were found to be less sensitive to changes on the surface. However, it was found that both of the methods had less reliability on the surfaces of fabric, stone, and wood, while hardly distinguishing backgrounds from defects because of strong textures on the surface.

The second experiment was conducted on seven cases where more than two surface types were mixed, and to improve the classification, all the sample groups were tuned. The Gabor filter was excluded from this experiment because it was not a training-based method.

In Case 1, when trained about wafer and solid together, ICA and RF

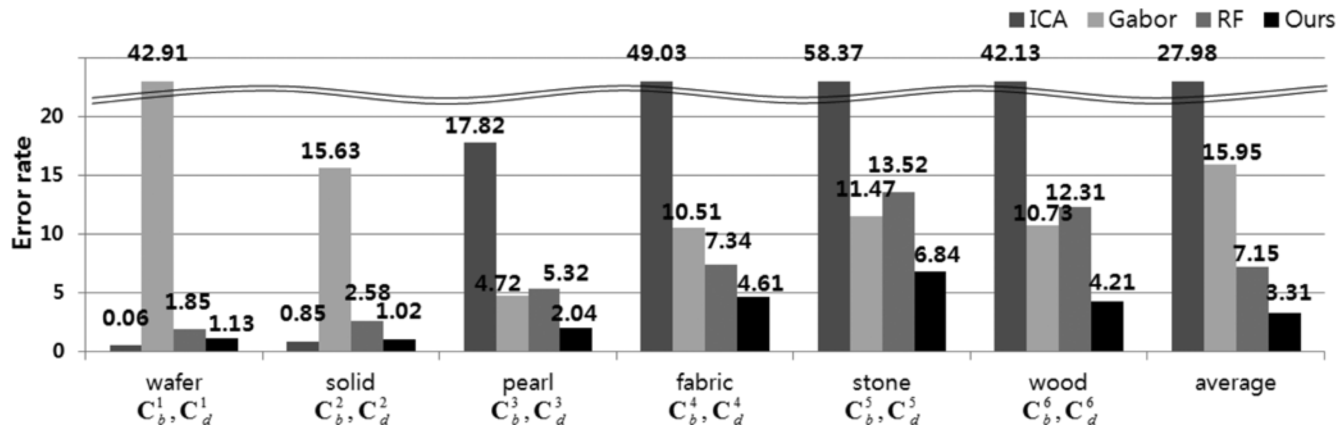


Fig. 2 Error rate of comparative experiment with two classes

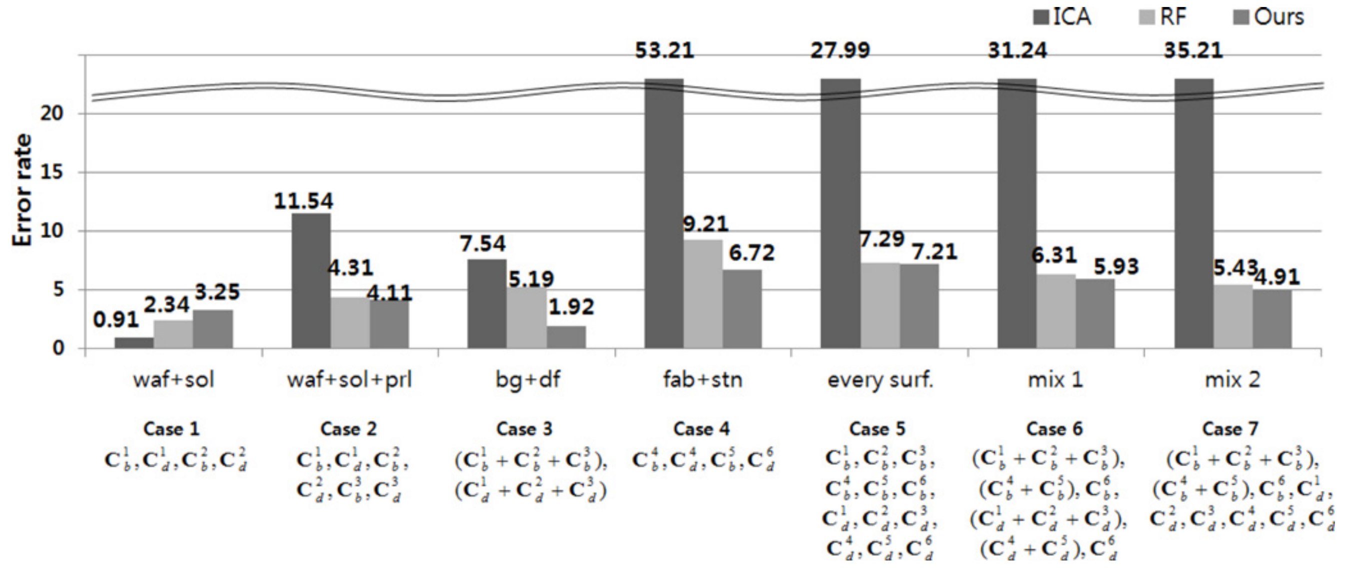


Fig. 3 Error rate of comparative experiment for mixed classes

showed no major difference in results, while CNN showed an increase in the error rate. As a result of analyzing the samples trained, classifying C_b^1 and C_b^2 having a similar texture to each other provides an increase in the error rate. When trained about wafer, solid, and pearl at the same time, ICA and RF showed similar results to those when they had been separately trained, but the error rate was found to increase in CNN. To supplement such shortcomings, therefore, we conducted Case 3 where backgrounds and defects of three surface types were mixed, which showed a decrease in the error rate. Such phenomena took place mainly on smooth surfaces where the texture was hardly distinguished, and in Case 4 where fabric and stone were mixed, there was no sharp increase found in CNN because it directly extracted features to distinguish the texture. Finding such features by a handcrafted method is a very difficult and time-consuming job.

In Case 5 where CNN was trained about all the 12 classes, it showed no remarkable performance increase compared to RF. To bring about a more advanced effect, we mixed more classes. In Case 3, three kinds of classes were mixed into one, and 6,000 images were used in one class, but in Cases 6 and 7, cases of single and mixed classes were trained together, so the number of images was limited to 2,000 for a combined class. In Case 6, only similar surfaces were mixed, and the same number of classes of backgrounds and defects were trained. However, as different defects mixed in a single class, there was no major improvement in the performance. In Case 7, however, we tried to maximize the classification performance by combining only similar background classes with separating their defect classes, which showed the most remarkable results when the defects of all the six surface types were searched.

CNN shows high reliability when classifying a single surface, but when searching various surfaces, it shows partially poor performance. This problem has two causes.

One is the shortage of sample data for training. The reliability of CNN increases by the depth of layers, but indispensable requirement for more layers is more sample data. Since CNN creates features by analyzing big data, its performance is directly associated with the number of sample data. When there is shortage of sample data, over-fitting takes place, further leading to decrease in reliability. To compose CNN in general, each class is recommended to have at least over 10,000 sample data at least. However, it is actually difficult to find images suitable for different environments without public dataset. In such case, we have a way of increasing the number of samples by shifting or rotating images, but there is possibility that a drift may take place due to similar samples

Table 4 Qualitative evaluation about the proposed method and conventional manpower inspection

	Proposed methods	Manpower inspection
Accuracy	98%	98.5%
Speed	5285 Samples/min	20 Samples/min
Duration	All day	Limited hour
Defect type	Various	Specific defect

being added. Furthermore, when sample data do not include wide range of the possible defects, the reliability of new defects cannot be assured.

The other cause is an increase in the size of images. When an image gets larger, we can expand the depth of neural network by adding more layers. Since more layers leads to exponential increase in the computing cost, however, simple expansion of the depth without data analysis will increase the learning cost without improving the reliability. The ICA, Gabor filter and RF require about 0.097, 0.265 and 0.014 second to detect defect for 640 × 480 pixels image with sliding window search, respectively. However CNN takes 0.217 second and it is slower than ICA and RF. Even though deep networks such as CNN require slightly much time on CPU, repetitive convolution-pooling structure is effectively parallelized and the computing time is greatly reduced with GPU device.

3.3 Analysis of Energy Saving

In most industrial fields, conventional defect inspection systems are still depends on human inspection. Existing systems are not able to inspect various types of surface defect and vulnerable to changes in illumination. The proposed method can overcome this limitation and save human hiring cost for inspection. We have performed an evaluation qualitatively and the results are summarized in Table 4.

According to Table 4, both methods have no significant difference in inspection accuracy. In contrast, the efficiency of the proposed method with respect to other factors (Inspection Speed, Work Duration, Acceptable Type of Defects) show higher performance than the manpower inspection. Especially, the inspection speed and work duration are most influential factors to reduce human hiring cost and increase productivity. Since the proposed method shows higher performances on these factors than manpower inspection, an actual effect of manpower saving can be expected.

4. Conclusions

This paper proposes a new surface defect inspection method using CNN, and the proposed method showed superior defect detection results from classification performance of deep neural networks. While users should model feature vectors in case by case through conventional machine learning methods, CNN directly acquires feature vectors from the convolution layer of the neural network. As a result, it is possible to apply high-level features that users hardly design, and to show great reliability by overcoming the limitations of manual modeling. In particular, this method does not need individual feature modeling to reflect types of surfaces with different textures, as it has quite high efficiency to apply. Moreover, when a new surface is added, this method does not require the development of new features and has a strong an advantage of applying parameters without tuning.

ACKNOWLEDGEMENT

We would like to acknowledge the financial support from the Basic Science Research Program through the NRF funded by the Ministry of Education, Science and Technology of South Korea (NRF-2013R1A1A2060427, NRF-2011-0017228).

REFERENCES

1. Aleixos, N., Blasco, J., Navarron, F., and Moltó, E., “Multispectral Inspection of Citrus in Real-Time Using Machine Vision and Digital Signal Processors,” Computers and Electronics in Agriculture, Vol. 33, No. 2, pp. 121-137, 2002.

2. Espiau, B., Chaumette, F., and Rives, P., “A New Approach to Visual Servoing in Robotics,” IEEE Transactions on Robotics and Automation, Vol. 8, No. 3, pp. 313-326, 1992.

3. Cheng, Y. and Jafari, M. A., “Vision-Based Online Process Control in Manufacturing Applications,” IEEE Transactions on Automation Science and Engineering, Vol. 5, No. 1, pp. 140-153, 2008.

4. Funck, J., Zhong, Y., Butler, D., Brunner, C., and Forrer, J., “Image Segmentation Algorithms Applied to Wood Defect Detection,” Computers and Electronics in Agriculture, Vol. 41, No. 1, pp. 157-179, 2003.

5. Yang, W., Li, D., Zhu, L., Kang, Y., and Li, F., “A New Approach for Image Processing in Foreign Fiber Detection,” Computers and Electronics in Agriculture, Vol. 68, No. 1, pp. 68-77, 2009.

6. Kumar, A. and Pang, G. K., “Defect Detection in Textured Materials Using Gabor Filters,” IEEE Transactions on Industry Applications, Vol. 38, No. 2, pp. 425-440, 2002.

7. Tsai, D.-M. and Lai, S.-C., “Defect Detection in Periodically Patterned Surfaces Using Independent Component Analysis,” Pattern Recognition, Vol. 41, No. 9, pp. 2813-2832, 2008.

8. Latif-Amet, A., Ertüzün, A., and Erçil, A., “An Efficient Method for Texture Defect Detection: Sub-Band Domain Co-Occurrence Matrices,” Image and Vision Computing, Vol. 18, No. 6, pp. 543-553, 2000.

9. Cord, A. and Chambon, S., “Automatic Road Defect Detection by Textural Pattern Recognition Based on AdaBoost,” Computer-Aided Civil and Infrastructure Engineering, Vol. 27, No. 4, pp. 244-259, 2012.

10. Shumin, D., Zhoufeng, L., and Chunlei, L., “AdaBoost Learning for Fabric Defect Detection Based on HOG and SVM,” Proc. of ICMT on Multimedia Technology, pp. 2903-2906, 2011.

11. Fukushima, K., “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position,” Biological Cybernetics, Vol. 36, No. 4, pp. 193-202, 1980.

12. Le Cun, Y., Bottou, L., Bengio, Y., and Haffner, P., “Gradient-Based Learning Applied to Document Recognition,” Proc. of the IEEE, Vol. 86, No. 11, pp. 2278-2324, 1998.

13. Hinton, G. E., Osindero, S., and Teh, Y.-W., “A Fast Learning Algorithm for Deep Belief Nets,” Neural Computation, Vol. 18, No. 7, pp. 1527-1554, 2006.

14. Collobert, R. and Weston, J., “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning,” Proc. of the 25th ICML, Vol. 25, pp. 160-167, 2008.

15. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., et al., “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” IEEE Signal Processing Magazine, Vol. 29, No. 6, pp. 82-97, 2012.

16. Sonka, M., Hlavac, V., and Boyle, R., “Image Processing, Analysis, and Machine Vision,” Cengage Learning, pp. 407-409, 2014.

17. Soukup, D. and Huber-Mork, R., “Convolutional Neural Networks for Steel Surface Defect Detection from Photometric Stereo Images,” in: Advanced is Visual Computing, George, B., Richard, B., (Eds.), Springer, pp. 668-677, 2014.

18. Yao, S. and Hai-Ru, L., “Detection of Weft Knitting Fabric Defects Based on Windowed Texture Information and Threshold Segmentation by CNN,” Proc. of IEEE on Digital Image Processing, pp. 292-296, 2009.

19. Kwon, B.-K., Won, J.-S., and Kang, D.-J., “Fast Defect Detection for Various Types of Surfaces Using Random Forest with VOV Features,” Int. J. Precis. Eng. Manuf., Vol. 16, No. 5, pp. 965-970, 2015.