ICIC Express Letters
Part B: Applications
Volume **9**, Number **12**, December **2018**

ICIC International ©2018 ISSN 2185-2766
pp. 1279–1286

GEAR CLASSIFICATION FOR DEFECT DETECTION IN VISION INSPECTION SYSTEM USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Imam Mustafa Kamal[1], Riska Asriana Sutrisnowati[2]
Hyerim Bae[3,*] and Taesoo Lim[4]

[1]Big Data Department
[3]Industrial Engineering Department
Pusan National University
2, Busandaehak-ro 63 beon-gil, Geumjeong-gu, Busan 46241, Korea
imamkamal@pusan.ac.kr; *Corresponding author: hrbae@pusan.ac.kr

[2]Dong-Nam Grand ICT R&D Center
Pusan National University
QB e-Centum, 90 Centum Jungang-ro Haendae-gu, Busan 48059, Korea
asriana.riska@gmail.com

[4]Computer Engineering Department
Sungkyul University
Sungkyul University-ro 53, Manan-gu, Anyang-si, Gyeonggi-do 430742, Korea
teshou@gmail.com

Received May 2018; accepted August 2018

Abstract. *For the purposes of accuracy and speed in industrial inspection, many companies heavily depend on human resources rather than automated systems. Nowadays, the most accurate method of image classification is deep learning. As demonstrated in ImageNet challenge, there is still no method that outperforms deep learning. Therefore, for automatic detection of defective gears, we propose the use of deep learning with two kinds of classification approaches, namely the Naïve approach and the fine-grained approach. The Naïve approach allows deep convolutional neural networks (CNN) to directly classify defects and non-defects in gear images, whereas the fine-grained approach harnesses an image processing technique before using CNN. Our experimental results show that there is a tradeoff between these two approaches: the Naïve approach is better in terms of processing time while the fine-grained approach is better in terms of accuracy.*
Keywords: Defect detection, Classification, Deep convolutional neural networks, Image processing

1. **Introduction.** Maximization of factory throughput is the goal of every manufacturing company. However, this cannot be achieved if non-defective products are classified as defective. Inspection for detection of defective products can be performed manually by humans and/or automatically with the help of sensors and computer programs. In the semiconductor industry and others that require high precision, manual inspection is deemed to be inadequate since products might be too small or too dangerous for humans to adequately inspect. One of the branches of computer science, computer (machine) vision has developed computers with "human-like" vision. Indeed, image recognition is a computer vision technology that can be used in machine vision inspection systems for inspection and labeling of images.

With machine vision inspection systems, images are classified as defective or normal (non-defective) with higher precision than is possible with manual inspection. The most

DOI: 10.24507/icicelb.09.12.1279

widely utilized methods of machine vision inspection are neural networks and deep learning. Initially, in 2012 with AlexNet [1], deep convolutional neural networks (CNN) was utilized to obtain a 15.4% top-5-tests error rate, a groundbreaking achievement at that time. Next, in 2013 ZF Net [8] was used to obtain an even lower, 11.2% error rate. In 2014 and 2015, error rates of 7.3% and 6.7% were achieved with VGG Net [7] and GoogleNet [2] respectively. Finally, in 2016, Microsoft ResNet [6] could bring down the error rate further to 3.6%. Such improvements in image recognition and classification (relative to the human error rate which ranges between 5% and 10%), were made possible by the design architecture typified by CNN. Some experiments have determined network layer numbers [2,3,5,7], while another study has designed a residual block for maintenance of vanishing/exploding gradients [6].

2. **Problem Statement and Preliminaries.** In this paper, we are dealing with gear images, which have several specific constraints. First, a "defective" or "non-defective" gear is determined by images displaying eight rectangular boxes, as shown in Figure 1. If any of those boxes indicates a defect, the gear will be categorized as defective and vice versa. Second, it is sometimes difficult to distinguish defective from non-defectivegear images at a glance. Third, the light intensities and sizes of the eight rectangular boxes differ: metal objects, when imaged by a camera, show reflections when captured by a camera, the rectangular boxes in the center tend to appear larger than those in the corners when the camera location is front and center. The key defect characteristics identified in this study were edge-hole, scratch, bump, chunks, and asymmetry. An edge-hole defect is indicated by a rectangular box showing one or more holes at the edge (refer to Figures 2(b_1) and 2(b_2)); a scratch defect, by a box showing a horizontal or vertical scratch (Figures 2(c_1) and 2(c_2)); a bump defect, by a box showing one or more bumps (Figures 2(d_1) and 2(d_2)); chunks, by a box showing breakage with one or more small chunks (Figures 2(e_1)

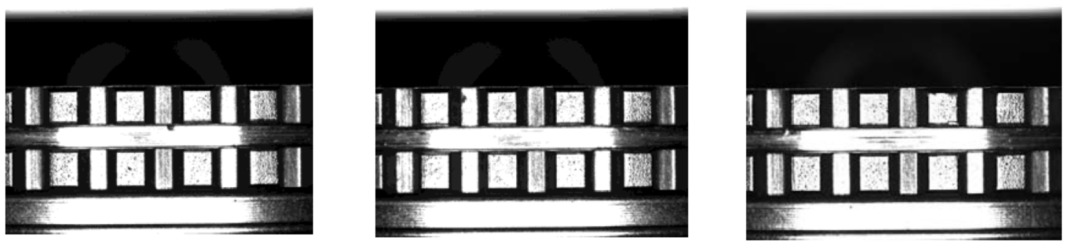

FIGURE 1. Overall gear images. Left: non-defective gear. Center: non-defective gear. Right: defective gear (note that the second upper rectangle from the right has scratch in its bottom-right corner).

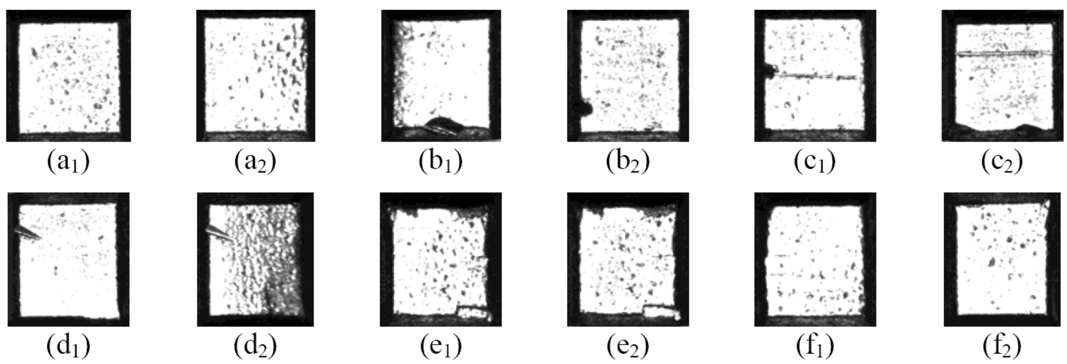

FIGURE 2. Important gear characteristics determining defective or non-defective status: (a) normal (non-defective), (b) edge-hole, (c) scratch, (d) bump, (e) chunks, (f) asymmetry

and 2(e_2)) and asymmetry, by a box having a shape that is not symmetric (Figures 2(f_1) and 2(f_2)).

This paper is organized as follows. Section 3 presents the related work. Sections 4 and 5 discuss the proposed method and experimental results, respectively. Finally, Section 6 draws conclusions.

3. **Related Works.** Given our problem wherein several rectangular boxes are to be identified as defective or non-defective, and the positioning of a gear that slightly position at an angle, here, we discuss several studies by other researchers. In [5], the authors used feature learning in CNN for fault detection in rotating machinery. Vibration signals from 2 perpendicularly-placed accelerometers are gathered and then they are used as the input for the convolutional neural network having 32 convolutional layers with channel of width 64 and 200 units of fully-connected layers. The results show that CNN get a very good result by 6% compared to the random forest classifier. In [1], the authors experimented on a CNN design for industrial inspection using hyper-parameter selection (i.e., manual preselection of parameters, such as filter size, that affect the architecture of the artificial networks). The networks had 3 layers for feature selection, a fully connected artificial neural networks with 2 layers as a final classifier, and 12 different outputs employing softmax regression. The authors stated that by using the hyper-parameters setting, the fault detection networks could be developed with minimal prior knowledge. In our case study, we do not have hyper-parameters because data are not available. In [10], the authors presented a one-class classifier for fault defection. This classifier provided a loss function for a penalty term based on Euclidean geometry in order to train deep CNN and mapped non-defective samples into a high-dimensional hypersphere where defective samples are kept far from the hypersphere. A radius for inspection accuracy is specified, according to which, images mapped outside it are considered to be defective. However, the number of defective products was small but there were many variations so that minimal interference by domain experts in classifying such defects would be required, which cannot be used in our case study. Therefore, our proposed method is specifically designed to tackle the problem in Section 2.

4. **Proposed Method.** In order to classify defect images based on the criteria presented in Section 2, we use two types of methodology employing CNN, namely the Naïve approach and the fine-grained approach.

4.1. **Naïve approach.** The Naïve approach entails two-class classification. We directly input the gear image into CNN which has two image-classification steps: a convolution layer extracts the important features from the input image, and a fully-connected network to classify them. During the training process, the weight values are automatically adjusted using gradient descent. As shown in Figure 3, in this research, the CNN architecture comprised three convolutional layers. The first two layers use 32 kernels of 3×3 size, and the third layer uses 64 kernels, also of 3×3 size. Padding, max-pooling, and ReLu as the activation function are applied in each convolutional layer. The two fully-connected layers each have 128 neurons, and the final layer has two neurons corresponding to defect and non-defect classes, respectively. AdamOptimizer is used instead of the classical stochastic gradient descent procedure to update network weights iteratively based on training data. The architecture was strongly inspired by previous researchers [7,8] with some slight modification in terms of output simplification (given that our problem is merely one of binary classifications).

4.2. **Fine-grained approach.** The CNN architecture of the fine-grained approach is similar to that of the Naïve approach. However, the gear image is pre-processed prior to going to CNN. Since the image that determines a defective or non-defective gear is

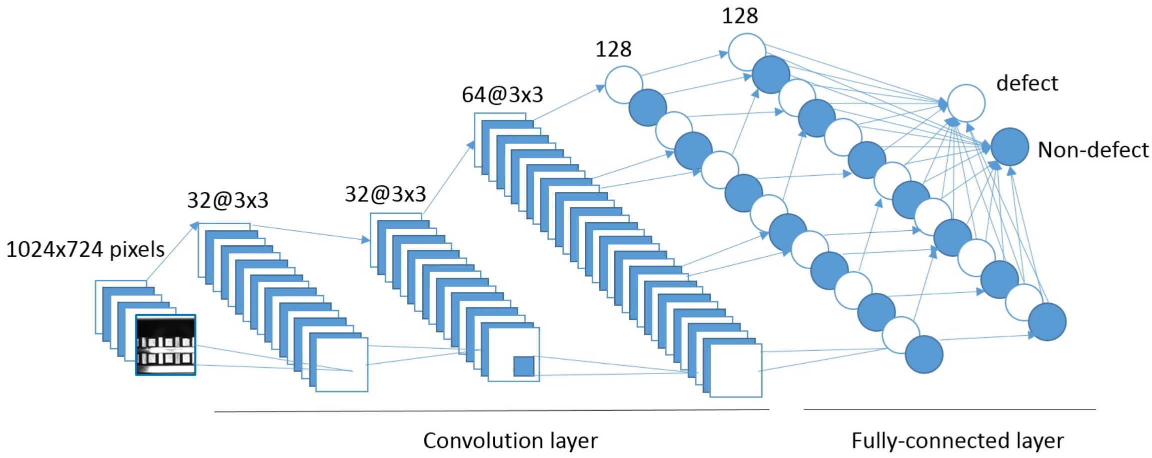

Figure 3. Principal architecture of deep CNN in Naïve approach

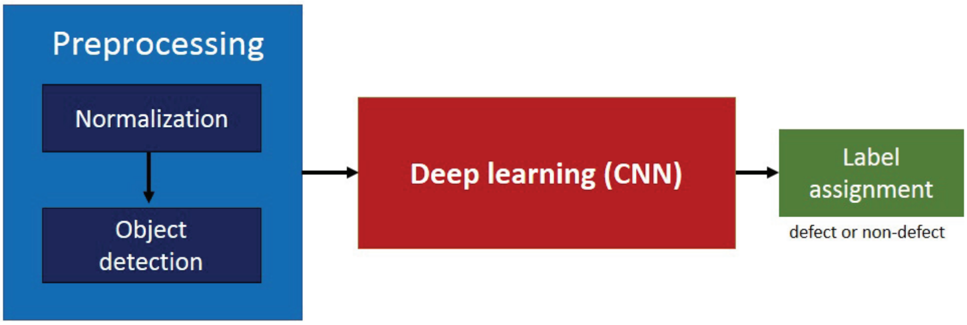

Figure 4. Designed system for defect detection in fine-grained approach

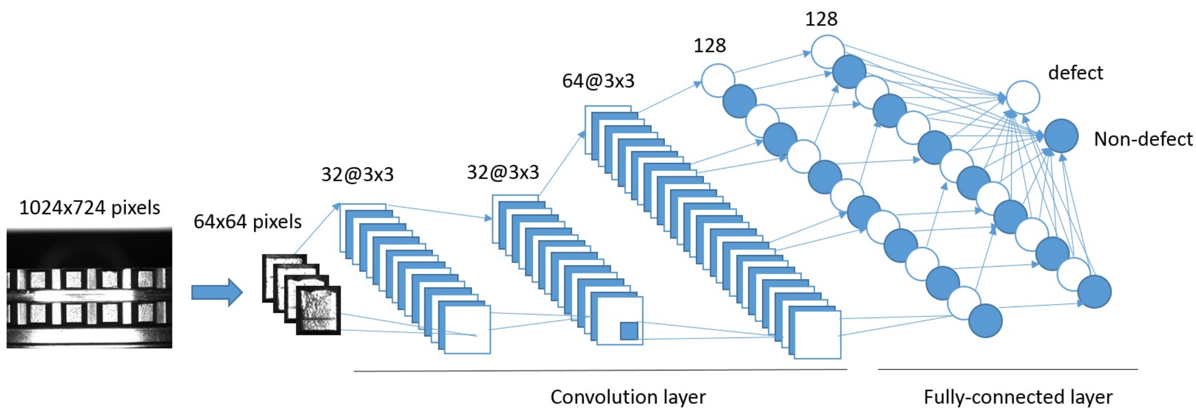

Figure 5. Principal architecture of deep CNN in fine-grained approach

comprised of eight rectangular boxes, we crop the boxes and, afterward, allow CNN to classify each box. If any of the boxes contains a defect it will be classified as a defective image. A more detailed workflow of the fine-grained approach is depicted in Figure 4, and the CNN architecture is shown in Figure 5.

For image pre-processing, we use gamma correction to normalize the different light intensities of the gear images and, thereby, control the overall contrast. Images that are too dark or too bright are normalized by Equation (1), where *gamma* is an input parameter. As shown in Figure 6, the best input parameter is 3; the higher the gamma value, the more correctly it detects rectangular boxes. The Canny edge detection algorithm, a multi-step algorithm that can detect edges with noise suppression [4] is used to detect and crop the boxes. The Canny edge detection algorithm, a multi-step algorithm that can

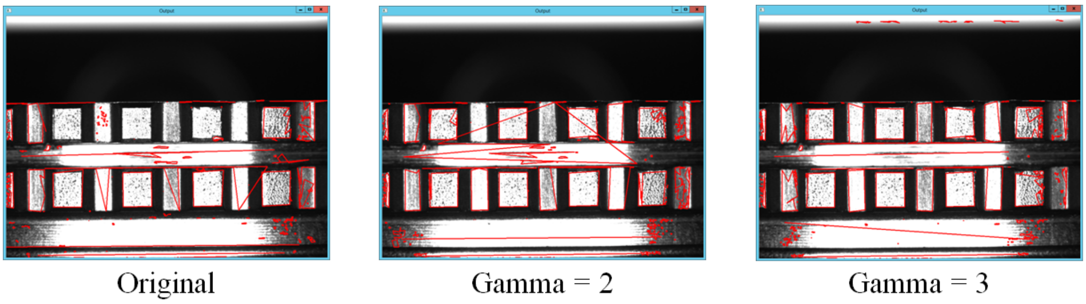

FIGURE 6. Image enhancement using different parameters of gamma correction and Canny detection to detect eight important rectangular boxes

detect edges with noise suppression [4], is used to detect and crop the boxes. The Canny algorithm detects the edges by using the following steps: preprocessing, calculating gradients, non-maximum suppression, and thresholding with Hysterysis. The preprocessing is used to remove the noise by using Gaussian blur. Afterward, the gradient magnitudes and directions are calculated at every single point in the image to find an edge. By harnessing the gradient magnitudes and orientations, it can determine the actual edge by using non-maximum supression. The output of the non-maximum suppression is thin edges, and this might be broken at various points. However, it can be fixed by filling in gaps with a threshold, which is called Hysterysis.

5. **Experimental Results.** Data consisting of 200 defective images and 200 non-defective images of 1024×724 pixels original size was gathered from an actual gear manufacturing company in South Korea. The experiment was conducted using a PC with an Intel Core i7 4790K CPU, 32 Gb RAM, a GPU NVIDIA GeForce GTX 1080Ti, Python and TensorFlow as the deep learning framework.

After the training process converged, we captured the output which flattened vector from the final convolutional layer. Afterward, we transformed it into matrix form. The matrix form as visualized in Figure 7 is a pretty good representation of the corresponding rectangular boxes.

Figure 8 represents the cost function for the 1500 epochs used to train our CNN within roughly 10 hours. As plotted in the figure, the training of the fine-grained approach tended to be more stable, faster to converge and to have a lower error rate compared with the Naïve approach. This result was due to the fact that we had implemented no image preprocessing method with the Naïve approach. Thus, it became difficult for the CNN model to distinguish between defective and non-defective-gear images. As noted earlier, at a glance, it is sometimes difficult to distinguish a defective-gear image from a

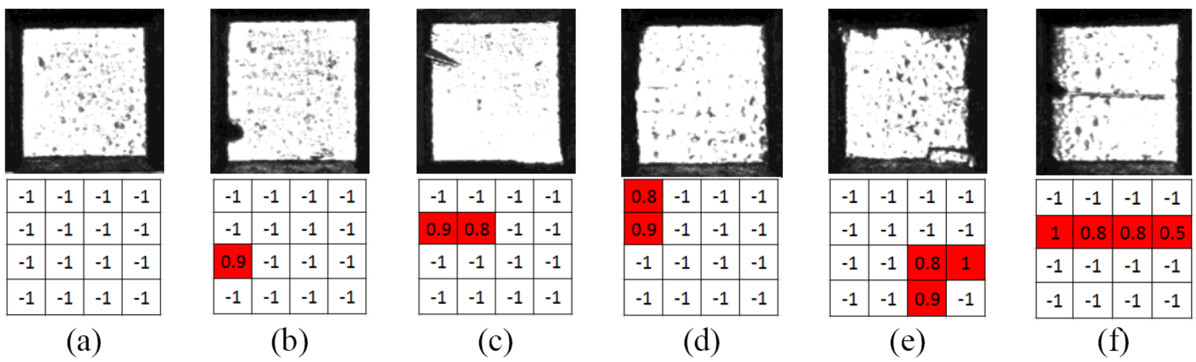

FIGURE 7. Feature extraction result after final convolution layer in matrix form: (a) normal (non-defective), (b) edge-hole, (c) bump, (d) asymmetry, (e) chunks, (f) scratch

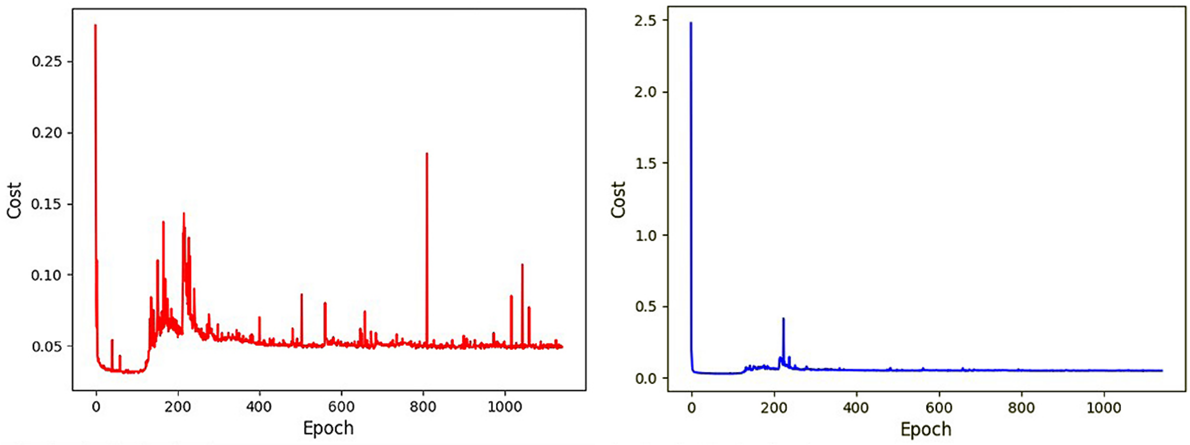

FIGURE 8. Cost function of CNN after 1500 epochs. Left: Naïve approach. Right: fine-grained approach.

non-defective-gear image.

$$corrected = 255 \times image^{\frac{1}{gamma}} \tag{1}$$

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2}$$

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

$$f\text{-}score = 2 \times \frac{recall \times precision}{recall + precision} \tag{5}$$

$$specificity = \frac{TN}{FP + TN} \tag{6}$$

A confusion matrix is a table that is often used to evaluate the performance of binary classification [9]. To measure how good our two approaches were, we compared them using precision, recall, f-score, and specificity, respectively. Precision is the number of correctly classified positive examples divided by the number of examples labeled by the system as positive. Recall is the number of correctly classified positive examples divided by the number of positive examples in the data, while the f-score is a combination of precision and recall; finally, specificity is used to assess how effectively a classifier identifies negative labels.

As shown in Table 1, the numbers of cases in which we correctly predicted "defective" gear, classified as true positive (TP), were 181 for the Naïve approach and 191 for the fine-grained approach. The numbers of gears that were predicted to be "defective" but were actually non-defective, classified as false positive (FP) were 19 and 9 for the Naïve and fine-grained approaches, respectively. The numbers of false negative (FN) images, predicted by the model to be "non-defective" but actually defective, were 13 and 5 for the

TABLE 1. Confusion matrix table

$N = 400$		Naïve approach		Fine-grained approach	
		Predicted		*Predicted*	
		Defect	*Non-defect*	*Defect*	*Non-defect*
Actual	*Defect*	181	13	191	5
	Non-defect	19	187	9	195

Naïve and fine-grained approaches, respectively. And, finally, the number of true negative (TN) images (correctly classified as defective) for the Naïve and fine-grained approaches were 187 and 191 respectively. From all of these results, we can infer that the fine-grained approach is superior to the Naïve approach.

In terms of production, our concern is to minimize the numbers of FP and FN gears. These measures are reflected in the f-score and specificity columns of Table 2. The greater the number of gears that are classified as false positive, the more the company will lose throughput and, therefore too, revenue. Meanwhile, the greater the number of gears that are classified as FN, the greater the number of defective products that will be sold to customers, the greater the number of complaints the company will receive, and finally, the worse its reputation will become. Table 2 compares the accuracy, precision, recall, f-score and specificity between the Naïve and fine-grained approaches, whose values were derived from Equations (2)-(6). From the experimental results, we can infer that the fine-grained approach outperforms for all of the criteria other than average processing time. The Naïve approach, however, is roughly seven times faster than the fine-grained approach. The fine-grained approach has a longer processing time because it uses two kinds of methods in the pre-processing stage, namely gamma correction for image normalization and Canny edge detection for image cropping. Hence, the fine-grained approach scans the gear image in more detail, whose process yields a better result.

Table 2. Comparison of methods' accuracy, precision, recall, f-score, and average processing time

Method	*Accuracy* (%)	*Precision* (%)	*Recall* (%)	*F-Score* (%)	*Specificity* (%)	*AVG Proc. Time* (s)
Naïve	92	90.5	93.3	91.9	90.8	**0.09**
Fine-grained	**96.5**	**95.5**	**97.4**	**96.5**	**95.6**	0.67

6. Conclusions and Future Work.

A fast and reliable vision inspection system is crucial in industrial inspection. If there are too many gear images classified as FN defective, the company will justifiably earn a bad reputation, and if there are too many images classified as FP defective, the company loses revenue. To tackle these issues, we conducted an experiment using CNN with two kinds of classification approaches, namely the Naïve and fine-grained approaches. The fine-grained approach provided good performance in terms of accuracy. However, it was very slow when compared with the Naïve approach. Therefore, in our future work we will minimize processing time by using a multi-thread or distributed GPUs technique.

Acknowledgment. This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean Government (MEST) (No. NRF-2015R1D1A1A09 061331).

REFERENCES

[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Proc. of the 25th International Conference on Neural Information Processing Systems*, vol.1, pp.1097-1105, 2012.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going deeper with convolutions, *Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[3] D. Weimer, B. Scholz-Reiter and M. Shpitalni, Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection, *CIRP Annals – Manufacturing Technology*, vol.65, pp.417-420, 2016.

[4] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.PAMI-8, no.6, 1986.

[5] J. Olivier, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. van de Walle and S. van Hoecke, Convolutional neural network based fault detection for rotating machinery, *Journal of Sound and Vibration*, vol.377, pp.331-345, 2016.

[6] K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition*, CoRR abs/1512.03385, 2015.

[7] K. Simonyan and A. Zisserman, *Very Deep Convolutional Neural Networks for Large-Scale Image Recognition*, CoRR abs/1409.1556, 2014.

[8] M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars (eds.), 2014.

[9] M. Sokolova and G. Lapalme, A system analysis of performance measures for classification tasks, *Information Processing and Management*, vol.45, no.4, 2009.

[10] M. Zhang, J. Wu, H. Lin, P. Yuan and Y. Song, The application of one-class classifier based on CNN in image defect detection, *Proc. of Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems CAS*, 2017.