Design and implementation of convolutional neural network-based SVM technique for manufacturing defect detection

Fusaomi Nagata*

Graduate School of Engineering,
Sanyo-Onoda City University,
1-1-1 Daigaku-dori,
Sanyo-Onoda 756-0884, Japan
Email: nagata@rs.socu.ac.jp
*Corresponding author

Maki K. Habib

Mechanical Engineering Department,
School of Sciences and Engineering,
American University in Cairo,
AUC Avenue, P.O. Box 74,
New Cairo 11835, Egypt
Email: maki@aucegypt.edu

Keigo Watanabe

Graduate School of Natural Science and Technology,
Okayama University,
3-1-1 Tsushima-naka, Kita-ku,
Okayama 700-8530, Japan
Email: watanabe@sys.okayama-u.ac.jp

Abstract: This paper introduces the design, implementation, training and testing of deep convolutional neural network (DCNN)-based support vector machines (SVMs). These DCNN-based SVMs are designed using software tools developed by the authors that enable them to construct, train, and test the DCNN-based SVMs and effectively facilitate vision-based inspection to detect different undesirable manufacturing defects. Two pretrained DCNNs are used for this purpose: the sssNet is developed by the authors and was trained using many actual and simple target images consisting of seven categories, and the standard AlexNet that was trained by a large number of images consisting of 1,000 categories. In this work, the pretrained sssNet and AlexNet are used as feature vector extractors in training and testing. The generated feature vectors are used as inputs to train SVMs for the final binary classification represented as accept (OK) or reject (NG) category.

Keywords: convolutional neural network; CNN; support vector machine; SVM; one-class learning of SVM; two-class learning of SVM; defect inspection system; template matching; edge extraction.

Reference to this paper should be made as follows: Nagata, F., Habib, M.K. and Watanabe, K. (2021) 'Design and implementation of convolutional neural network-based SVM technique for manufacturing defect detection', *Int. J. Mechatronics and Automation*, Vol. 8, No. 2, pp.53–61.

Biographical notes: Fusaomi Nagata is currently a Professor at the Department of Mechanical Engineering, Faculty of Engineering, Sanyo-Onoda City University, Japan. His research interests include intelligent control of industrial robot and its applications, development of network-based multiple mobile robots system and its intelligent control.

Maki K. Habib is currently a Full Professor at the American University in Cairo, Egypt. His research interests include human adaptive and friendly mechatronics, autonomous navigation, humanitarian demining, intelligent control, telecooperation, distributed teleoperation and collaborative control, wireless sensor networks and ambient intelligence, and biomimetic robots.

Keigo Watanabe is a Specially Appointed Professor at the Department of Intelligent Mechanical Systems, Graduate School of Natural Science and Technology, Okayama University, Japan. His research interests include intelligent signal processing and control using soft computing, bio-inspired robotics, and non-holonomic systems.

This paper is a revised and expanded version of a paper entitled 'Fusion method of convolutional neural network and support vector machine for high accuracy anomaly detection' presented at 2019 IEEE International Conference on Mechatronics and Automation (ICMA 2019), Tianjin, China, 4–7 August 2019.

1 Introduction

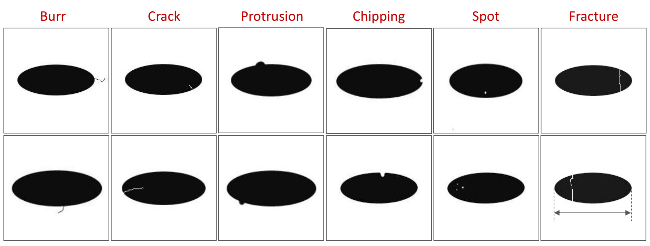
The performance of the automatic defect detection systems based on AI approaches, deep learning and convolutional neural network (CNN) exceeds any skilled inspectors, and it is expected to have a high demand for such systems. For example, the accurate online detection of many manufacturing processes such as welding defects detection is still challenging due to different types of defects associated with it. Recently, Zhang et al. introduced a deep learning-based online detection approach detecting defects in robotic arc welding process and associated with aluminium alloy using CNN and welding images (Zhang et al., 2019). Also, railway track fasteners have an important function to steadily fix tracks on the ballast bed. The need for automatic defect detection of such fasteners is required to ensure track safety and to reduce maintenance cost. To challenge this need, Wei et al. contributed by developing methods based on techniques of image processing and deep learning networks (Wei et al., 2019). Besides, Zhong et al. proposed a transfer learning approach based on CNN and support vector machines (SVMs) and applied his development to the process of gas turbine fault diagnosis (Zhong et al., 2019). Moreover, Fu et al. proposed an effective and compact CNN model that can strengthen low-level features training and incorporate multiple receptive fields in order to realise quick accurate and reliable classification of steel surface defects (Fu et al., 2019).

However, it is not easy for beginners and novice engineers to efficiently construct a reliable visual inspection system based on the AI learning frameworks made with c++, Python on Caffe, Chainer, TensorFlow, etc. (Kato et al., 2018). For example, the Convolutional Architecture for Fast Feature Embedding (CAFFE) developed by Berkeley Vision and Learning Center is a deep learning framework supported by open source libraries. Practical results for designing deep neural networks and image recognition in the Caffe framework are used and demonstrated (Komar et al., 2018). Chainer provided by Preferred Networks, Inc. is an open source software library to effectively conduct the calculation and training of neural networks. The characteristic function is to dynamically generate data structure needed for back propagation algorithms, which enables the software construction of complex and deep neural networks (Tokui et al., 2015). TensorFlow provided by Google is also an open source software library to support developments based on machine learning, numerical analysis, deep learning and so on (Ertam and Aydin, 2017).

User-friendly software tools have been developed by Nagata et al. to facilitate the design of applications based on DCNNs (Nagata et al., 2018b). For example, generally DCNNs have several blocks. These blocks mainly consist of convolutional, ReLU and pooling layers. The layers accept image files in the former hidden layers that lead to fully-connected layers and an output layer represented by softmax function layer. The developed software tools by the authors enable researchers, relevant students and engineers to design, train and test DCNNs without the necessity to engage with the complex details of using programming language in the development, such as C++ or Python.

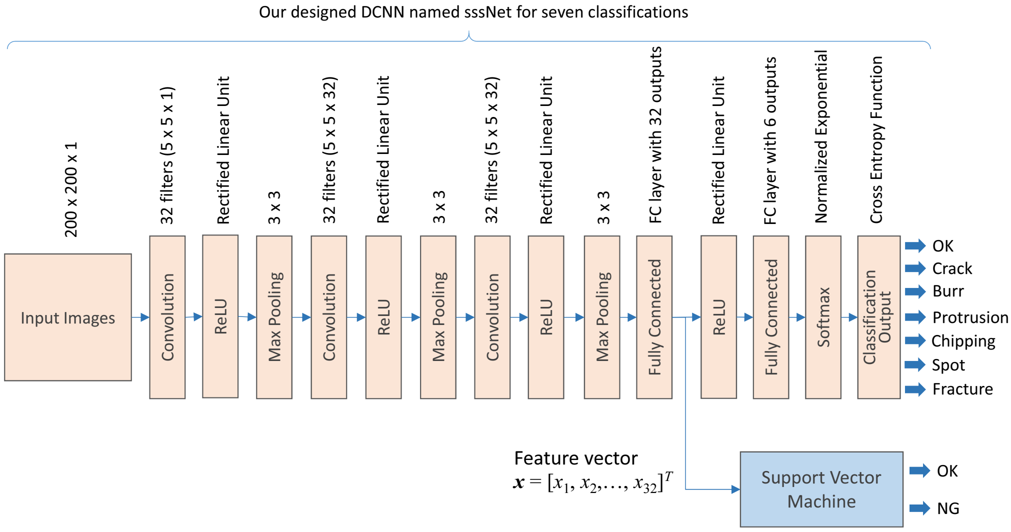
Besides the CNNs, Zhang et al. (2006) reported the use of one-class learning-based SVM, in which one class SVM is concluded as an effective method to deal with unsupervised training data. Chittilappilly and Subramaniam presented industrial defect detection using SVM approach (Chittilappilly and Subramaniam, 2017), in which feature vectors extracted from denoised and restored images are applied to SVM classifier for identifying fault in the industrial applications. Nagata et al. also introduced a design and training tool to support the development of CNN-based SVMs (Nagata et al., 2019a).

Figure 1 Six types of defect examples that may appear during manufacturing process of resin moulded articles with horizontal length of about 30 mm (see online version for colours)



This paper presents binary classifications using DCNNs, SVMs and template matching method for vision-based inspection to detect manufacturing defects. Firstly, using the developed software tools, two one-class learning-based SVMs are designed and trained using only OK images with no defect for the purpose to distinguish it from images with manufacturing defects. It is assumed in this paper that the manufacturing defects include burr, crack, chipping, protrusion, spot and fracture which normally appear during the manufacturing process of resin moulded articles. Examples of typical defects are shown in Figure 1. As it is known that it is difficult for conventional shallow NNs to be applied to the defect detection system which has to classify many similar images into defective or non-defective.

Figure 2 The proposed SVM obtained from one-class learning with sssNet and trained for binary classification (see online version for colours)



Note: The input to the SVM is the feature vector generated by the first fully-connected layer of the designed sssNet.

Due to its great ability as feature vector extractors, two pretrained DCCNs are used for this purpose: the sssNet is developed by the authors and was trained using many actual, single and target images consisting of seven categories, and the standard AlexNet that was trained by a large number of images consisting of 1,000 categories.

In this work, the pretrained sssNet and AlexNet are firstly used as feature vector extractors in training and testing of two one-class learning using SVMs. They are called as one-class learning using SVMs processes with AlexNet and processes with sssNet, respectively. In one-class learning process, only OK images are used. Then, the AlexNet is also used as a feature vector extractor in training and testing of a two-class learning-based SVM. In two-class learning process, both OK and NG images are used. The feature vectors generated by the DCNNs are used as inputs to train SVMs for the final binary classification represented as accept (OK) or reject (NG) category. The performance of both SVMs trained through one-class learning using AlexNet and sssNet, and also the SVM trained through two-class learning using AlexNet are evaluated and compared through binary classification experiments. Then, a template matching method is adopted and integrated to extract actual target areas from the original images used for training and testing, which enables to not only enhance the accuracy and reliability of the binary classification but also reduce the calculation load.

2 SVMs integrated with two types DCNNs

Defect inspection of manufactured product conducted by human is in general unstable and insufficient, and accordingly the error in the classification tends to lead to serious problems. Hence, AI in terms of deep learning can contribute to develop automatic inspection of defects that is able to recognise and remove the undesirable defective products from the good ones. This section introduces two types of DCNN-based SVMs with aim to perform binary classification, which are designed, trained and tested using the software tools developed on MATLAB. Neural Network Toolbox, Parallel Computing Toolbox for GPU, Deep Learning Toolbox, Statistics and Machine Learning Toolbox provided by MathWorks are used for the development. The experiments show that input images to the trained SVMs can be classified into OK or NG categories with a high accuracy.

2.1 The Design of sssNet-based SVM

The sssNet is the authors' original DCNN and it is developed and trained using many actual and simple target images as input images for the purpose to classify it into seven categories, these categories are: OK with no defect, and the other six are with different types of defect: crack, chipping, protrusion, burr, spot or fracture as shown in Figure 1. The output of the sssNet is represented by the probability called score. The sssNet can also extract a multidimensional feature vector $\boldsymbol{x} = [x_1, x_2, \dots, x_{32}]^T$ from each inputted image. Since, the SVM with its extension by the Kernel method is able to solve nonlinear classification problems, the SVM is integrated with the sssNet to get such a DCNN-based SVM that can characteristically extract the multidimensional feature vectors from the input images and then can classify the vectors in terms of binary classification, i.e., OK or NG. Figure 2 introduces the designed DCNN-based SVM for binary classification in which the feature vector $\boldsymbol{x}$ extracted from the first fully connected layer (11th layer) in the sssNet is given to the input layer of the SVM. The Gaussian kernel function shown in equation (2) was used with one-class learning of the DCNN-based SVM, where the feature vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{5,100} \in \Re^{32\times 1}$ were extracted from 5,100 OK input images that were only used with the unsupervised learning of sssNet-based SVM while NG images were not used at all.

The unclassified, uncategorised or unlabeled training data is used to conduct unsupervised learning such as one-class learning, so that no defect NG images are used for SVM training. The algorithm of sequential minimal optimisation (SMO) developed by Platt (1998) is used to optimise the dual problem of quadratic programming (QP) of the obtained SVM using Lagrange multiplier method. In optimising one-class learning DCNN-based SVM, a parameter $\nu (0 < \nu < 1)$ is used to determine the upper limit rate of the number of outliers, which are positioned between an origin and a hyper plane, to that of all samples $\boldsymbol{x}$. In this test trial, ν is set to 0.5.

When an extracted feature vector $\boldsymbol{x} \in \Re^{32\times 1}$ obtained from a test image with the designed sssNet is introduced into the trained SVM, the SVM output is called the score and it corresponds to the value $f(\boldsymbol{x})$ that is obtained by

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i G(\boldsymbol{x}_i^*, \boldsymbol{x}) + b \tag{1}$$

where $\boldsymbol{x}_i^* \in \Re^{1\times 32} (i = 1, 2, ..., N)$ represents the obtained support vectors; N indicates to the number of the support vectors obtained through the training associated with the OK category training data set; b and $\alpha_i \ (i = 1, 2, ..., N)$ are the bias and the Lagrange multipliers respectively that represent the parameters of SVM estimated from the training process. $|b|$ is regarded as the distance from the origin to the hyper plane. y_i is the label set to 1 indicating for one-class learning case. $f(\boldsymbol{x})$ represents the signed distance to the decided hyper plane, while $G(\boldsymbol{x}_i^*, \boldsymbol{x})$ indicates to the kernel function given by

$$G(\boldsymbol{x}_i^*, \boldsymbol{x}) = \exp\left(- \left\| \frac{\boldsymbol{x}_i^* - \boldsymbol{x}_s}{k} \right\|^2 \right) \tag{2}$$

where k and $\boldsymbol{x}_s$ are the kernel scale and the standardised input vector respectively and they are calculated by

$$\boldsymbol{x}_s = (\boldsymbol{x} - \boldsymbol{x}_\mu) \oslash \boldsymbol{x}_\sigma \tag{3}$$

with

$$\boldsymbol{x}_\mu = \frac{\sum_{j=1}^{5,100} \boldsymbol{x}_j}{5,100} \tag{4}$$

$$\boldsymbol{x}_\sigma = \left[\frac{1}{5,100} \sum_{j=1}^{5,100} (\boldsymbol{x}_j - \boldsymbol{x}_\mu)^{\circ 2} \right]^{\circ \frac{1}{2}} \tag{5}$$

where $\oslash$, $\circ 2$, $\circ \frac{1}{2}$ represent the Hadamard operators for elementwise division, the power and the root, respectively.

The experiments show that time of few minutes was required to complete training and consequently k, N and b were obtained numerically as 1.1875, 2,621 and –1.0639, respectively. The designed SVM after been trained with one-class learning is shown in Figure 4. Then, the binary classification can be conducted by the trained SVM using a feature vector obtained from a test image. The binary categories are estimated by checking the sign of $f(\boldsymbol{x})$, so that $f(\boldsymbol{x}) > 0$ means OK category and that of $f(\boldsymbol{x}) < 0$ means NG category.

Also, it is necessary to mention that the nth-order polynomial given by the following equation

$$G(\boldsymbol{x}_i^*, \boldsymbol{x}) = \left[1 + \frac{(\boldsymbol{x}_i^*)^T}{k} \frac{\boldsymbol{x}_s}{k} \right]^n \tag{6}$$

And this is considered also as another form of kernel function. It is important to note that there is a need to check in advance the matching between feature vectors and kernel functions.

2.2 AlexNet-based SVM design

AlexNet is a well-known DCNN type that was designed to be specialised to perform mainly image recognition. This type of DCNN is already pre-trained using a very large number of images, more than one million images covering 1,000 categories. Hence, it can classify an input image into one of the 1,000 categories. AlexNet also is used to extract from each inputted image a feature vector $\boldsymbol{x} = [x_1, x_2, ..., x_{4,096}]^T$. Accordingly, the feature vector obtained from AlexNet has 4,096 elements that facilitate the ability to deal with one thousand classifications. Figure 3 presents the proposed AlexNet-based SVM for binary classification to whose input layer the feature vectors obtained from the 2nd fully connected layer (20th layer) in the AlexNet are submitted. The same conditions used with the case of sssNet-based SVM one-class learning were applied to train this AlexNet-based SVM. For this one-class learning in which 5,100 of only OK images $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{5,100} \in \Re^{4,096\times 1}$ are used for AlexNet-based SVM unsupervised learning, it took several minutes to complete the training. Once completing the training, the parameters k, N and b were obtained as 26.7690, 2,667 and –1.0635, respectively.

2.3 Experimental classifications with two SVMs trained through one-class learning

Once both one-class learning of the two SVMs were trained, experimental classifications are performed to check the generalisation capability to differentiate between OK and NG images. Normally the used images for testing are images that were not used during the training process. The classification results are shown in Figure 5 and they are represented in terms of histograms. The obtained results are associated with the sssNet-based SVM presented in Figure 2. The output values $f(\boldsymbol{x})$ obtained from the trained SVM with the sssNet is denoted by the horizontal axis, and the vertical axis denotes the number of image samples. It can be observed from the histogram presented in Figure 5 that the SVM has the ability to well separate NG images from OK ones.

Figure 3 The binary class SVM with its feature vector generated from AlexNet (see online version for colours)

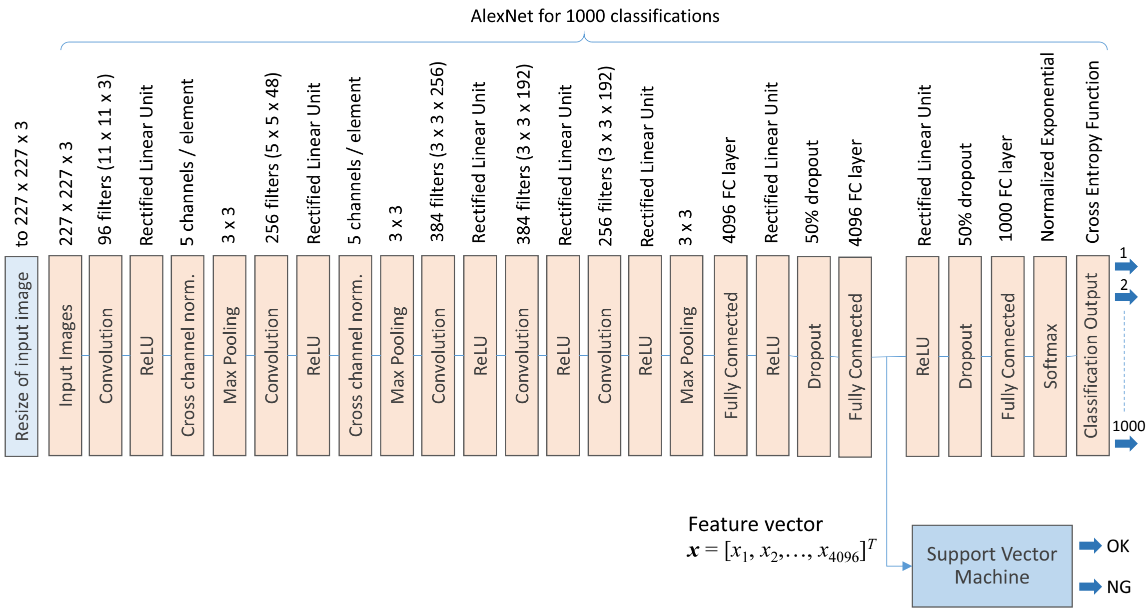
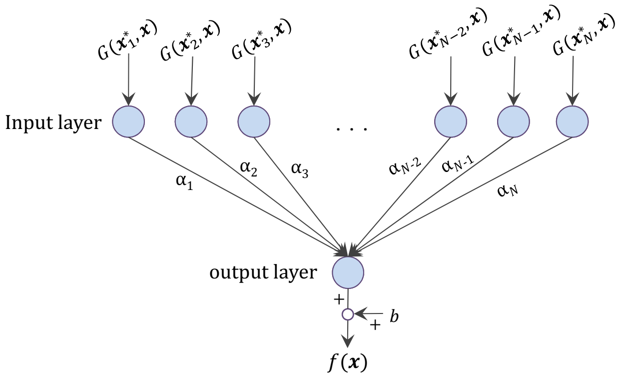


Figure 4 Introduced the SVM as given by equation (1) after optimised through one-class learning (see online version for colours)



In addition, Figure 6 introduces the classification results of the AlexNet-based SVM outlined in Figure 3. It can be observed from Figure 6 that the classification results of AlexNet-based SVM have also the ability to separate the NG images from the OK ones with almost similar reliable performance as what was obtained with the sssNet-based SVM. Such similar performance and discrimination ability have been achieved in spite of having different length of the generated feature vectors, i.e., 32 elements from the sssNet and 4,096 elements from AlexNet. As for the case of having target images with grey-scaled resolution $200 \times 200 \times 1$ as illustrated in Figure 1, the size of having 4,096 elements feature vector submitted to the SVM show high redundancy. Table 1 includes the comparison of misclassified images number for both the sssNet and the AlexNet. Test images of 35,000 images (30,000 images with 5,000 image of each defect, and 5,000 OK image) are used to test each of the DCNN-based SVM obtained through one-class learning. The results show that all of the OK images were classified correctly, and it also show that the performance superiority of the sssNet-based SVM as the table shows the misclassified images among the six types of different defected images used in the test images.

Figure 5 Classification results using the sssNet-based SVM shown in Figure 2, in which horizontal and vertical axes denote the output from the SVM and the number of classified images, respectively (see online version for colours)

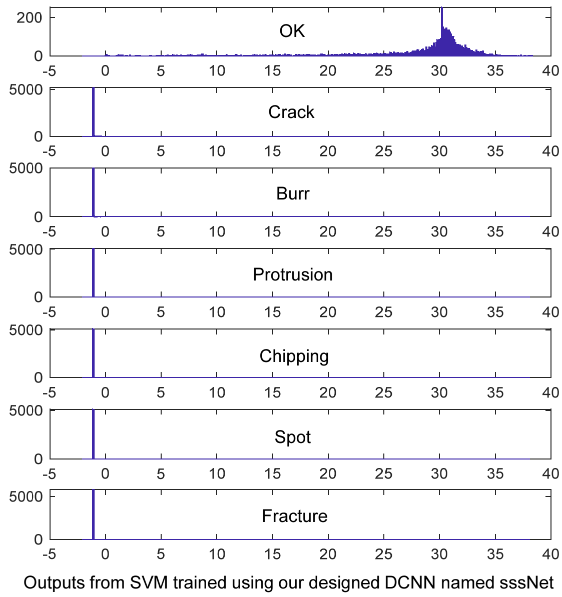
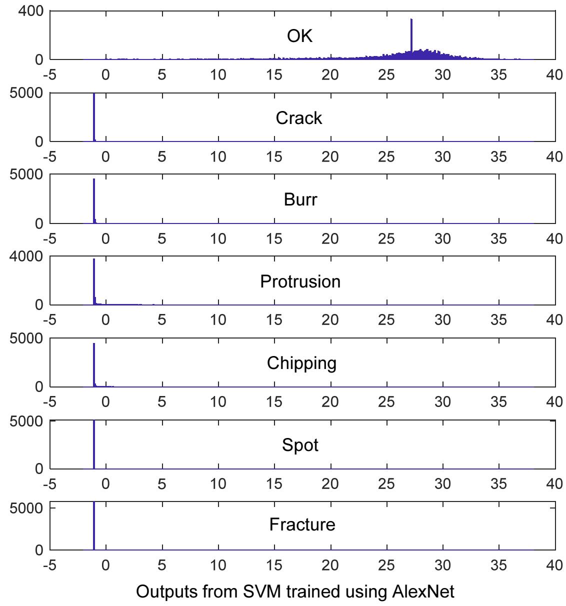


Table 1 Comparison of each of misclassified images number between sssNet and AlexNet.

SVM	*Burr*	*Crack*	*Chipping*	*Knob*	*Spot*	*Fracture*
sssNet	13	4	1	0	0	0
AlexNet	167	20	298	127	0	0

Figure 6 Classification results using the AlxNet-based SVM shown in Figure 3, in which horizontal axis denotes the output from the SVM and vertical axis denotes the number of classified images (see online version for colours)



In the next section, a design of a two-class learning-based SVM is presented, which is also called as the SVM obtained through supervised learning.

3 Another AlexNet-based SVM trained with two-class learning

The two types of trained SVMs presented in the previous section were obtained using one-class learning, that means using only OK images with no defect for the training. This section introduces the design and comparison of another SVM trained through two-class learning. In this case, the SVM is trained using two classes of images consisting of OK images and NG images. For this purpose, the AlexNet is used as a feature vector extractor. Once having the training process done, classification function $f(\boldsymbol{x})$ of the SVM is calculated using equation (1), where the label y_i respectively indicates to 1 or –1 according to OK or NG when dealing with two-class learning.

$f(\boldsymbol{x})$ represents the signed distance from $\boldsymbol{x}$ to the decision boundary that is called the hyper plane. In case that the $G(\boldsymbol{x}_i^*, \boldsymbol{x})$ is the linear kernel, it is given by

$$G(\boldsymbol{x}_i^*, \boldsymbol{x}) = \frac{(\boldsymbol{x}_i^*)^T}{k} \frac{\boldsymbol{x}_s}{k} \tag{7}$$

Accordingly, the function formulated by equation (1) can be simplified into

$$f(\boldsymbol{x}) = \frac{\boldsymbol{x}_s^T}{k} \boldsymbol{\beta} + b \tag{8}$$

where k represents the scale of the kernel, $\boldsymbol{\beta} \in \Re^{4,096 \times 1}$ indicates to the fitted linear coefficient vector, while b denotes to the bias. The parameters of the SVM such as k, $\boldsymbol{\beta}$ and b are the estimated through training. Figure 7 illustrates the the designed SVM structure after been trained with two-class learning. The feature vector $\boldsymbol{x} \in \Re^{4,096 \times 1}$ is the output obtained from the second fully connected layer within the structure of AlexNet.

Figure 7 The SVM given by equation (8) after been trained with two-class learning (see online version for colours)

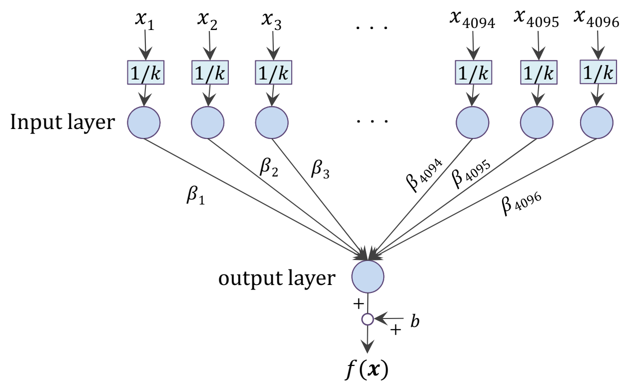


Table 2 Confusion matrix checking the trained situation of the SVM based on two-class learning

Actual \ Predicted	*Anomaly*	*Normal*
Anomaly	30,573	27
Normal	8	5,092

Table 3 Confusion matrix checking the generalisation of the SVM based on two-class learning

Actual \ Predicted	*Anomaly*	*Normal*
Anomaly	5,987	13
Normal	3	997

The SVM shown in Figure 7 was trained using images of 5,100 normal OK images and 30,600 (with 5,100 × 6 categories) anomalies. The anomaly images include the six types of defects highlighted shown in Figure 1. The numerical values of the SVM parameters k, N and b were determined through the training as 63.8168, 698 and 4.9334 respectively. By using the training images data again, the trained situation was checked, and the result given by the confusion matrix is presented in Table 2. From Table 2, the accuracy and precision indicators are calculated as 0.9990 and 0.9997, respectively.

The generalisation ability of the two-class learning-based SVM was also checked using 1,000 normal test images and 6,000 (with 1,000 × 6 categories) for the anomalies, and the obtained result by the confusion matrix is introduced in Table 3. Table 3 shows that the accuracy and precision are calculated as 0.9977 and 0.9995, respectively. This demonstrates that it is possible to observe the desirable generalisation, but, ability to achieve the complete classification with a misclassified rate of 0 could not be reached. Table 2 shows that the decision boundary called as the hyper plane is determined by a soft margin concept in the training process in which a certain degree of misclassification is allowable to avoid falling in the situation of extreme over fitting. That is the reason of why the misclassification at such degree as shown in Table 3 is unavoidable.

4 Template matching to narrow images including features

4.1 Image extraction based on normalised cross correlation

The template matching technique is introduced in this section. When applying a template image with size of (M, N) as shown in Figure 8, a padding operation is used to generate space around a target image content and also inside any of its defined borders. This operation creates extra space around the target original image. The total space formed by the padded area and that of the original target image are called an enlarged target image.

The correlation coefficient $\alpha(u, v)$ between a template image and an area of the same size within the target image enlarged using padding operation is calculated by Lewis (2001)

$$\alpha(u,v) = \frac{s_{it}(u,v)}{s_i(u,v)s_t(u,v)} \tag{9}$$

$$s_{it}(u,v) = \sum_{y=v}^{v+N-1} \sum_{x=u}^{u+M-1} \{f(x,y) - \bar{f}_{u,v}\} \{t(x-u, y-v) - \bar{t}\} \tag{10}$$

$$s_i(u,v) = \sqrt{\sum_{y=v}^{v+N-1} \sum_{x=u}^{u+M-1} \{f(x,y) - \bar{f}_{u,v}\}^2} \tag{11}$$

$$s_t(u,v) = \sqrt{\sum_{y=v}^{v+N-1} \sum_{x=u}^{u+M-1} \{t(x-u, y-v) - \bar{t}\}^2} \tag{12}$$

where (u, v) is the upper-left position coordinate of the template image in the expanded target image; the standard deviations are described by $s_i(u, v)$ and $s_t(u, v)$; $s_{it}(u, v)$ represents the covariance;$f(x, y)$ is the normalised value of the greyscale at the (x, y) position described in the expanded target images coordinate frame; $t(x - u, y - v)$ is the normalised greyscale value at the position $(x - u, y - v)$ described in the template image coordinate frame; M and N are the template image width and height respectively; $\bar{t}$ is the greyscale mean value of the template image; $\bar{f}(u, v)$ is also the greyscale mean value in the area just below the template image. Equation (9) is used to generate sequentially the correlation coefficients $\alpha(u, v)$ of the template image by raster scanning it from top left to bottom right within the expanded target image. Once the scanning is completed, best correlated area to the template image is extracted according to the maximum correlation coefficient $\alpha(u, v)$ value. Figure 9 introduces examples of images extracted obtained through the use of template matching technique.

Figure 8 The configuration of a target image, padding area and the template image of (M, N) in size (see online version for colours)

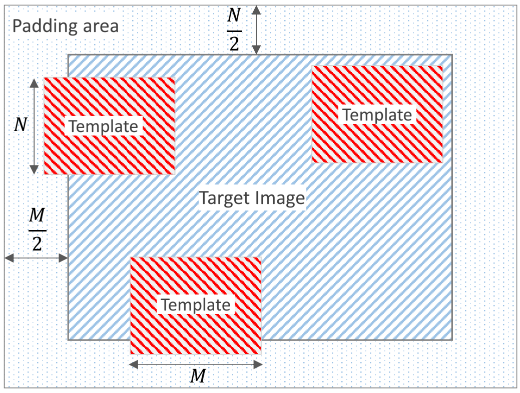
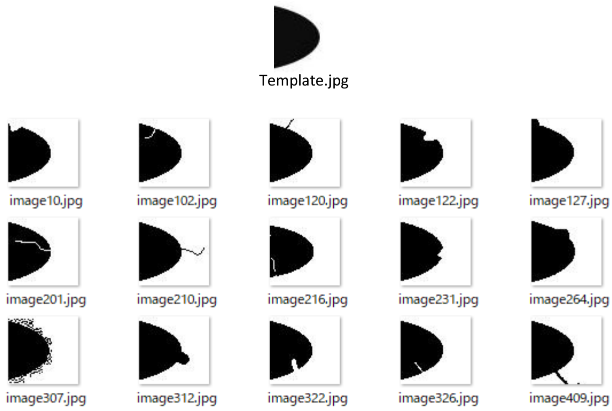


Figure 9 Examples of extracted images obtained through the use of template matching technique



4.2 Experiments

The effectiveness of the template matching method is evaluated using 3,000 OK images without defect and some of these images are presented in Figure 10. The original OK images are first trimmed before using them. Then, the designed AlexNet-based SVM shown in Figure 3 was trained through one-class learning using the 3,000 trimmed OK images. As a result of this, the training helped to find the parameters k, N and b as 50.0116, 1,526 and -5.6925 respectively. After once training was done, the one-class learning-based SVM with the Alexnet was evaluated using 120 test images including examples as shown in Figure 11. OK images with no defect and NG images with only one type of defects as shown in Figure 1 were used as part of the test images. While the result of OK or NG classification associated with the used test images is introduce in Figure 12. This result is obtained in associated with the test images shown in Figure 11, where horizontal plus and minus numerical values represented as scores are used to classify the test images as OK or NG, respectively. The experimental classification results indicate that all images used in testing are successfully classified as OK or NG category.

Figure 10 Examples of the 3,000 OK images for training the SVM with AlexNet as a feature extractor

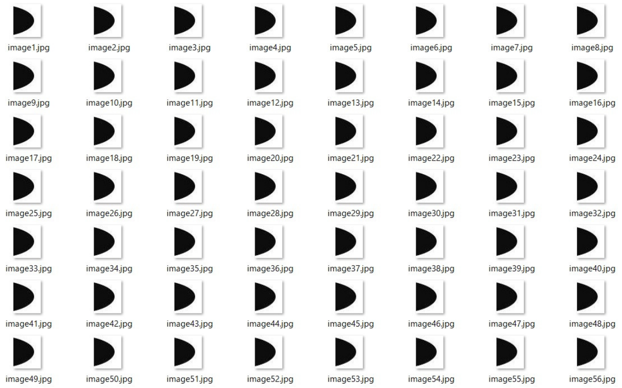


Figure 11 Some of 120 test images for evaluating the one class learning-based SVM trained using images shown in Figure 10

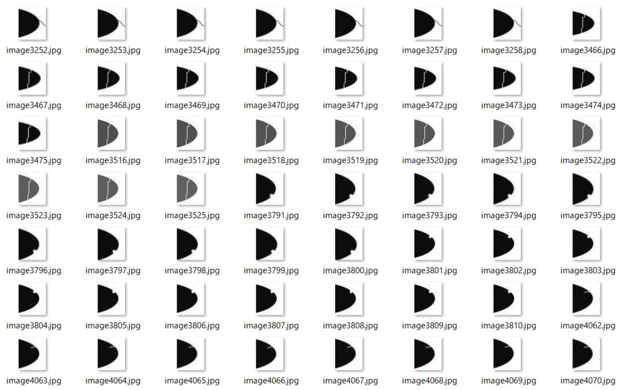
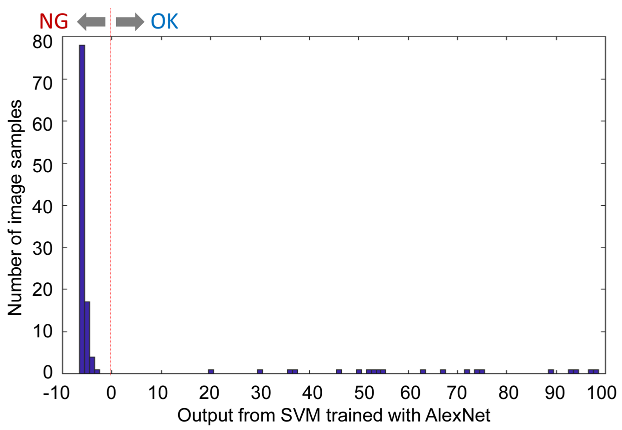
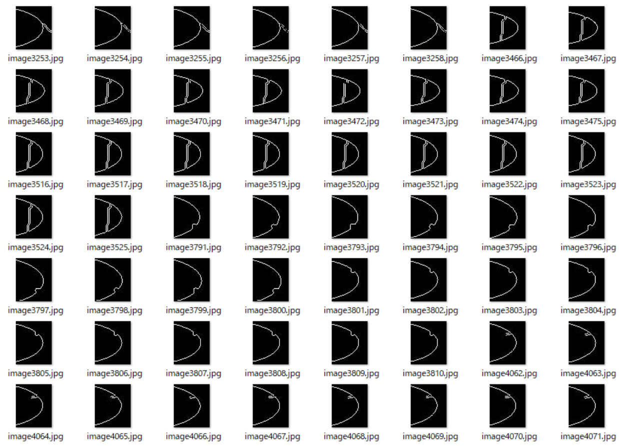


Figure 12 Binary classification result of 120 test images as shown in Figure 11 (see online version for colours)



In the same manner, the edge extraction function as shown in Figure 13 is also provided by the proposed tool, in which for example, Sobel filter, Prewitt filter or Roberts filter (Topno and Murmu, 2019) is selectable. If dominant defect features for binary classification are mainly included around the periphery of an object, this function will be also effective in terms of the reductions of not only the resolution of images but also the calculation load in training process.

Figure 13 Examples of training images converted by edge extraction function



5 Conclusions

This paper presented the design, implementation, training and testing of deep convolutional neural network (DCNN)-based support vector machines (SVMs) to enhance the performance of binary classification. Software tool was developed to facilitate the development, testing and evaluation. Two types of pretrained DCNNs with multiple classifications were used in this work. The first DCNN named sssNet was developed by the authors to inspect real defects associated with the manufacturing process of resin moulded articles and it was trained using many actual and simple target images consisting of seven categories. The objective of training is to enhance the ability for generalisation. The second one is well known, standardised and generalised DCNN called AlexNet which was trained by a large number of images consisting of 1,000 categories. The pretrained sssNet and AlexNet are firstly used as feature vector extractors in training and testing the developed two types of one-class learning-based SVMs. They are called as one-class learning-based SVM processes with sssNet, and as one-class learning-based SVMs processes with AlexNet. Only OK images are used in both cases. In addition, the AlexNet is also used as a feature vector extractor in training and testing of a two-class learning-based SVM using both images with no defects and images with defects. For all one class and two-class learning-based SVMs, the generated compressed feature vectors were used as inputs for the SVMs to train them for the final binary classification represented as accept (OK) or reject (NG) category. The three DCNN-based SVMs were successfully designed, trained, tested and evaluated.

Both performance obtained from one-class learning of SVMs with AlexNet and sssNet, and the SVM obtained through the two-class learning with AlexNet are evaluated and compared through training and classification experiments. The obtained results indicated that the developed sssNet enhanced the accuracy and the reliability of improved binary classification. Beside this, the adopted template matching method was used with the AlexNet-based SVM to narrow the area within the image reflecting the important featured associated with the images used in the training, so that to reduce the required computational load until having the SVM to perform desired binary classification. As a follow up with this successful work, it is possible to develop a cascade-type SVM that has multiple SVMs use for high accuracy binary classification.

Currently, the authors are interested in utilising the transfer learning method of acknowledged DCNNs such as AlexNet, VGG16, VGG19, etc. called series network and GoogleNet, Inception-V3, Inception-ResNet-V2, etc. called directed acyclic graph (DAG) network. In future work, another option dialogue is planned to be developed to interactively design original DCNNs obtained using the transfer learning method.

References

Chittilappilly, A.J. and Subramaniam, K. (2017) 'SVM based defect detection for industrial applications', *Procs. of 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS2017)*, pp.782–786.

Ertam, F. and Aydin, G. (2017) 'Data classification with deep learning using Tensorflow', *Procs. of 2017 International Conference on Computer Science and Engineering (UBMK)*, pp.755–758.

Fu, G., Sun, P., Zhu, W., Yang, J., Cao, Y., Yang, M.Y. and Cao, Y. (2019) 'A deep-learning-based approach for fast and robust steel surface defects classification', *Optics and Lasers in Engineering*, Vol. 121, pp.397–405.

Kato, S., Inagaki, Y. and Aoyama, M. (2018) 'A structural analysis method of OSS development community evolution based on a semantic graph model', *Procs. of 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pp.292–297.

Komar, M., Yakobchuk, P., Golovko, V., Dorosh, V. and Sachenko, A. (2018) 'Deep neural network for image recognition based on the Caffe framework', *Procs. of 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pp.102–106.

Lewis, J.P. (2001) 'Fast normalized cross-correlation', *Industrial Light & Magic*, 7pp.

Nagata, F., Tokuno, K., Ochi, H., Otsuka, A., Ikeda, T., Watanabe, K. and Habib, M.K. (2018a) 'A design and training application for deep convolutional neural networks and support vector machines developed on MATLAB', *Procs. of the 6th International Conference on Robot Intelligence Technology and Applications (RiTA2018), Lecture Notes in Mechanical Engineering (LNME)*, pp.27–33.

Nagata, F., Tokuno, K., Watanabe, K. and Habib, M.K. (2018b) '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.

Nagata, F., Mitarai, K., Tokuno, K., Otsuka, A., Ikeda, T., Ochi, H., Watanabe, K. and Habib, M.K. (2019a) 'Binary classification method using deep convolutional neural networks and support vector machines', *Procs. of 24th International Symposium on Artificial Life and Robotics*, pp.780–784.

Nagata, N., Tokuno, K., Nakashima, K., Otsuka, A., Ikeda, T., Ochi, H., Watanabe, K. and Habib, M.K. (2019b) 'Fusion method of convolutional neural network and support vector machine for high accuracy anomaly detection', *Procs. of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA2019)*, pp.970–975.

Platt, J. (1998) *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*, Technical Report MSR-TR-98-14, pp.1–24.

Tokui, S., Oono, K., Hido, S. and Clayton, J. (2015) 'Chainer: a next-generation open source framework for deep learning', *Workshop on Machine Learning Systems at Neural Information Processing Systems (NIPS)*, pp.1–6.

Topno, P. and Murmu, G. (2019) 'An improved edge detection method based on median filter', *Procs. of 2019 Devices for Integrated Circuit (DevIC)*, pp.378–381.

Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L. and Li, Y. (2019) 'Railway track fastener defect detection based on image processing and deep learning techniques: a comparative study', *Engineering Applications of Artificial Intelligence*, Vol. 80, pp.66–81.

Zhang, X., Gu, C. and Lin, J. (2006) 'Support vector machines for anomaly detection', *Procs. of 2006 6th World Congress on Intelligent Control and Automation*, pp.2594–2598.

Zhang, Z., Wen, G. and Chen, S. (2019) 'Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding', *Journal of Manufacturing Processes*, Vol. 45, pp.208–216.

Zhong, S.S., Fu, S. and Lin, L. (2019) 'A novel gas turbine fault diagnosis method based on transfer learning with CNN', *Measurement*, Vol. 137, pp.435–453.