2019 12th Asian Control Conference (ASCC)
Kitakyusyu International Conference Center
Japan, June 9-12, 2019

# Classification of Steel Surface Defect Using Convolutional Neural Network with Few Images

Min Su Kim[1], Taesu Park[1] and PooGyeon Park[2]

*Abstract*— **Classification of steel surface defects in steel making industry is essential for the detection of defects through the classification of defects and for the analysis of causes that make defects. This makes it possible to reduce the defect rate of the product, and drastically reduces the mass defect in the steel making process. Recently, Deep Learning has been used for defect detection using Convolutional Neural Network(CNN). Compared to the existing rule-based method, the defect classification using CNN achieved high performance. However, learning CNN requires hundreds or thousands of images. In the case of a defective image, it is difficult to obtain images of thousands or hundreds of images. To overcome these shortcomings, few-shot learning which need few images to train network can be used. Siamese Neural Network using CNN is used for few-shot learning. In this paper, we use few-shot learning with Siamese Neural Network using CNN structure with contrastive loss to classify defects with a small number of steel surface defect images.**

## I. INTRODUCTION

In the steel industry, classification of steel surface defect plays a very important role in finding the cause of defects of the manufacturing process. It is possible to eliminate the defects by utilizing the cause of defects of the product in real time for the manufacturing process. As a result, mass defect in the continuous manufacturing process can be drastically reduced. In addition, process optimization can be performed by analyzing the pattern of defects according to the operating conditions.

Recently, various machine learning methods have been used to solve classifying the steel surface defect problem. [1] used Principal component Analysis(PCA) and Self-Organizing Maps(SOM) to classify steel surface defects. There were also many researches that analyze defects through support vector machines: twin support vector machine [2], [3], hyper-spheres based support vector machine [4], [5]. In addition, surface defects were classified by various machine learning algorithms such as random forest [6] and bayesian network classifiers [7].

As Convolutional Neural Network(CNN) shows good performance in handling images, artificial neural network has been widely used for defect detection using CNN. In [8], CNN for steel surface defect using photometric images are proposed. [9] used cascaded auto-encoders and compact CNN to detect defects and classify them for automated fault detection. [10] proposed the CNN with linear activation unit for classifying of steel surface defects.

Common deep learning networks require hundreds, or even thousands, of images to learn a class. It's not easy to label all the data classes due to its cost and time. In addition, the data required for learning may not be sufficient. There may not even be a few images to be used for learning.

One way to solve this problem is a few-shot learning. Few-shot learning can learn quickly even if there are only a few training images per label [11], [12]. It applies the classification algorithm to data with very few samples in each class. In this methodology, rather than predicting the class itself, a pair of classes is generated and trained for the few-shot network with various pairs. Using the generated pair, features from each image are extracted and calculate the $L1$ distance of each feature. Based on the $L1$ distance, images included in the pair are determined that those are the same class or different classes [13], [14].

In [14], two images are given to a neural network to train whether they are in the same class or not. Then, put the test image on the network and let the support set guess which image belongs to the same class. To do this, the Siamese neural network is designed to put two images into same one network, and calculate the probability that the two images are the same class. In addition, the advantage of this method is that it shows high performance in revealing differences between images that have never been seen before.

In this paper, we use few-shot learning and Siamese neural network to classify defects with a small number of defect image of steel surface defect images. The features of steel surface defect images are extracted through CNN and the $L_1$ distance value of the feature is used to predict whether the class pairs are the same or not. We trained all classes of defects using few images for training network. Also, except one class, we confirmed that network can classify whether the class that excluded in training is different from the existing class.

## II. CLASSIFICATION MODEL

### A. Siamese Neural Network

Siamese neural network [15] is designed to put two images into one network and ultimately to get a probability of whether the two images are the same class. To do this, two images are given to the neural network to train whether they are in the same class or not. Then, put the test image on the network and let the support set, which is collection map of

[1]Min Su Kim is with the Department of Electrical Engineering, Pohang University of Science and Technology, Pohang, Republic of Korea kms6777@postech.ac.kr
[2]Taesu Park is with the Department of Electrical Engineering, Pohang University of Science and Technology, Pohang, Republic of Korea tspark@postech.ac.kr
[3]PooGyeon Park is with the Department of Electrical Engineering, Pohang University of Science and Technology, Pohang, Republic of Korea ppg@postech.ac.kr

each class, guess which image of support set is similar to the test image. In the support set, only one image is of the same class, and the rest of the image is of another class. The structure of Siamese neural network is shown in Fig.1. $L_1$ distance ,which is absolute distance, of features from each network which shares weights of structure is obtained. This value is then passed through the fully connected layer, and the sigmoid function takes a single value. Target value is 1 for the same class pair, 0 for the different class pair. As shown in equation (1), apply the sigmoid function to the weighted sum of the $L_1$ distance values of each feature to limit $P^j$ between 0 and 1 where $P^j$ is the final output of $j_{th}$ train image and $\alpha_i$ is the parameter learned automatically in the training process and multiplied by the distance value for each feature and $\sigma$ is the sigmoid activation function. $F^j_{1,L-1}(i)$ is $i_{th}$ feature of input 1 of $j_{th}$ image pair in $L-1_{th}$ layer.

$$P^j = \sigma(\sum_i \alpha_i |F^j_1(i) - F^j_2(i)|) \tag{1}$$

Then, we can obtain the loss using binary cross entropy for the predicted value and the target as in (2).

$$L(P^j, t^j) = t^j log P^j + (1 - t^j) log(1 - P^j) \tag{2}$$

where $t^j$ is target value of $j_{th}$ image pair which has 1 for same class and 0 for different class.

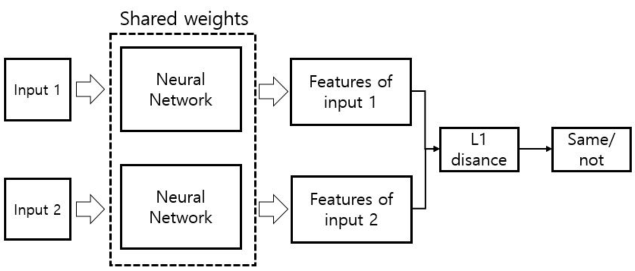


Fig. 1. Siamese network structure

## B. Contrastive Loss

For similar images, networks can be trained with similarity functions rather than binary cross entropy. The neural network learns the function of finding the similarity of two images [16]. Similarity function $d(img_1, img_2)$ returns the degree of difference between two images. If similarity function return value greater than any reference value $m$, which is called margin, the two images are very different from each other. Considering this reference value is a constructive loss. Contrastive loss considers how far the distances of the features of the two images are, unlike the binary cross-entropy which considers the probability that the two images are the same [17]. Therefore, when the feature has a similar distribution, the reference value can be adjusted and the accuracy of networks can be improved. Since the steel plate defect data also have a similar distribution of each label as the face image, we use contrastive loss. Contrastive loss is defined as (3).

$$\begin{aligned} L(F^j_1, F^j_2, t^j) = \\ &\frac{1}{2} t^j \left\{ max(0, m - |F^j_1 - F^j_2|) \right\}^2 \\ &+ \frac{1}{2}(1 - t^j)|F^j_1 - F^j_2|^2 \end{aligned} \tag{3}$$

## C. Proposed CNN Siamese model

Basically, we used the structure of Siamese network, CNN layer was used to extract each feature. Detailed filter size and number of filters are shown in Fig. 2. The network structure used in the model experiment is as follows. Considering the small pixel size of steel surface defects, we used a convolution layer with a small filter size. As mentioned above, considering that the defect images have a similar distribution, we used constrative loss rather than binary cross-entropy. We used constrative loss for the loss function and set the margin to 1.

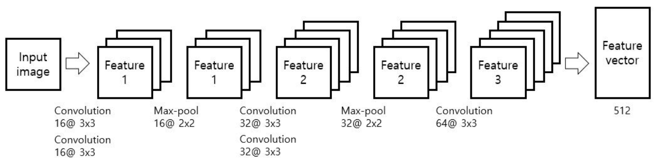


Fig. 2. Feature extraction in proposed CNN Siamese model

We used 3 x 3 filter size for each convolution layers and max-pooling layer was added after two convolution layers. Through the network, the channel expands to 64, and when passing through the last convolution stage, the image of 64 x 64 x 1 changes to 16 x 16 x 64. After flattening it, reduces its dimension into 512 feature vectors. Since one pair has two images, two 512 feature vectors are created via the network. Calculate the contrastive loss using two feature vectors.

# III. EXPERIMENTS

NEU (Northeastern University) surface defect dataset was used for the experiment. The NEU-CLS-64 data set consists of nine class defects, each of which is a 64 x 64 pixel gray scale image. Each class has 200 to 1,500 images. Sample images of each defect class is shown in Fig. 3.

## A. Learning with 9 class

For each of the nine classes, few-shot learning was performed using 5 or 10 images, and data of each class not used for learning was assumed to be unlabeled. We checked the accuracy of 200 unlabeled images. The simplest baseline for one-shot running is 1-nearest neighbor and random guessing. 1-nearest neighbor selecting the closest euclidean distance from the test image to the nearest training image as in (4).

$$Y(\bar{x}) = argmax_{c \in S} ||x_c - \bar{x}|| \tag{4}$$

where $\bar{x}$ is test image, $Y(\bar{x})$ is predicted class, $S$ is support set and $x_c$ image in support set. Random guessing randomly

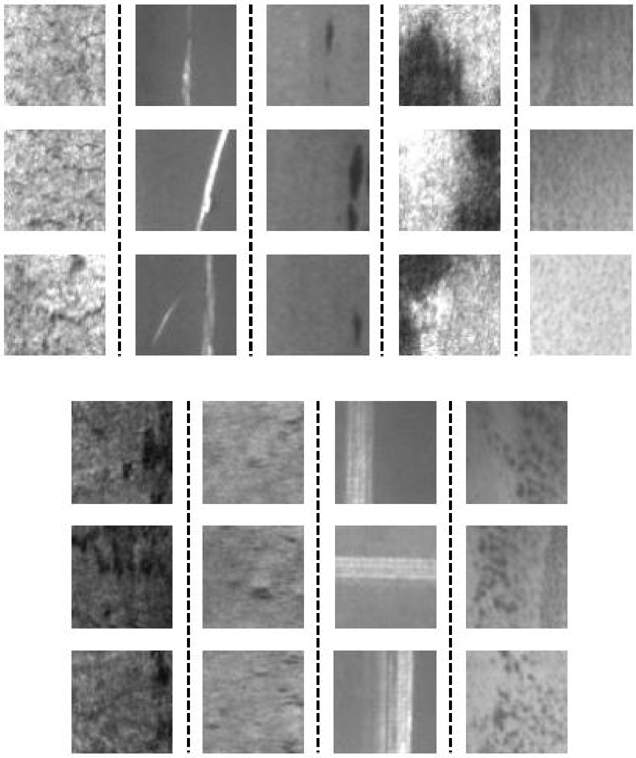

Fig. 3. Example of 9 class of steel surface defects. Images corresponds to each class of 1 - 9 from upper left to lower right.

infer labels for each class. When the number of labels is $n$, the accuracy is calculated as $1/n$. In our network, predicted class of test image is obtained as in (5)

$$Y(\bar{x}) = argmax_{c \in S} O(x_c, \bar{x}) \tag{5}$$

where $O(a, b)$ is $L_1$ difference of feature of $a$ and $b$.

Fig. 4 shows a pair of images used in learning. When learning the image (a), (b) and (c) are used as a pair. (b) is a support set such that only one image is the actual input, and (c) nine images are the same support set as the actual input. When a pair is formed, one same and eight not same data are input. Using five images of each class, the number of classes used for training was increased from 2 to 9, and the accuracy was calculated as Fig. 5 (a). When using 10 images of each class the accuracy was calculated as Fig. 5 (b). Accuracy is 85.1% and 86.5% for 9 classes, respectively. The accuracy according to the number of possible classes is as follows in TABLE 1.

TABLE I

ACCURACY OF FEW-SHOT LEARNING FOR STEEL SURFACE DEFECTS

| possible class | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 5 images | 95.1 | 94.3 | 91.2 | 90.8 | 88.1 | 86.6 | 85.7 | 85.1 |
| 10 images | 99.1 | 95.2 | 95.4 | 92.3 | 93.1 | 90.2 | 87.8 | 86.5 |

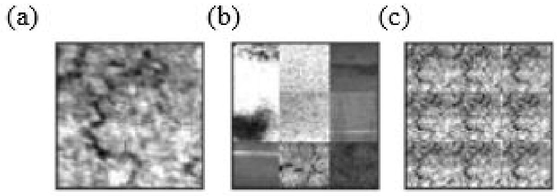


Fig. 4. Example of support for training: (a) input image of network, (b) Support set of different class, (c) Support set of same class

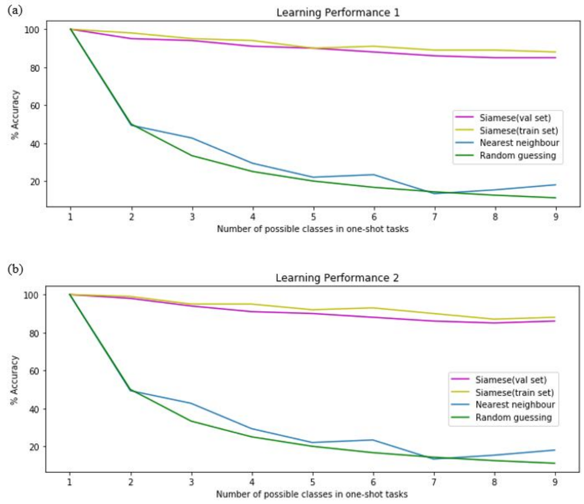


Fig. 5. Few-shot learning accuracy with number of possible classes: (a) 5 images per class, (b) 10 images per class

### B. Learning with 8 class

For each of the eight classes, we used 5 images randomly and assumed that the data of each class that was not used for learning was unlabeled. We checked the accuracy of 200 unlabeled images. One class that was not used for training was tested. After training the network with 8 classes first, we predicted whether the new class is the same as the existing class or not. The classes used for learning are as follows in Fig. 6. The ratio that distinguishes non trained class from the existing class is as follows in TABLE 2

TABLE II

ACCURACY OF NON-TRAINED CLASS

| Method | Test |
|---|---|
| Random Guessing | 11.1 |
| Nearest neighbour | 24.5 |
| Siamese net | 80.1 |

## IV. CONCLUSIONS

We use Few-shot learning with Siamese neural network using CNN structure with contrastive loss to classify defects with a small number of steel surface defect images. The NEU

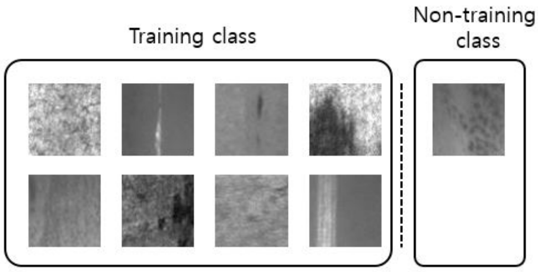


Fig. 6. Few-shot learning accuracy with number of possible classes: (a) 5 images per class, (b) 10 images per class

steel surface defect data set was used, and the network was learned by 5 and 10 small amount of images. Accuracy of network are of 85.1% and 86.5% was confirmed for 9 labels, respectively. In addition, 80.1% accuracy was also obtained for the untrained label.

## ACKNOWLEDGMENT

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICT Consilience Creative program(IITP-2019-2011-1-00783) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation)

This research was supported by Korea Electric Power Corporation. (Grant number:R18XA01)

## REFERENCES

[1] LUIZ, AO Martins; FLAVIO, LC Padua; PAULO, EM Almeida. Automatic detection of surface defects on rolled steel using computer vision and artificial neural networks. In: IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society. IEEE, 2010. p. 1081-1086.

[2] Chu M, Gong R, Gao S, et al. Steel surface defects recognition based on multi-type statistical features and enhanced twin support vector machine. Chemometrics and Intelligent Laboratory Systems, 2017.

[3] Chu M, Liu X, Gong R, et al. Multi-class classification method using twin support vector machines with multi-information for steel surface defects. Chemometrics and Intelligent Laboratory Systems, 2018, 176: 108-118.

[4] Gong R, Wu C, Chu M. Steel surface defect classification using multiple hyper-spheres support vector machine with additional information. Chemometrics and Intelligent Laboratory Systems, 2018, 172: 109-117.

[5] Chu M, Liu X, Gong R, et al. Multi-class classification method for strip steel surface defects based on support vector machine with adjustable hyper-sphere[J]. Journal of Iron and Steel Research International, 2018, 25(7): 706-716.

[6] Kwon B K, Won J S, Kang D J. Fast defect detection for various types of surfaces using random forest with VOV features. International Journal of Precision Engineering and Manufacturing, 2015, 16(5):965-970.

[7] PERNKOPF, Franz. Detection of surface defects on raw steel blocks using Bayesian network classifiers. Pattern Analysis and Applications, 2004, 7.3: 333-342.

[8] SOUKUP, Daniel; HUBER-MORK, Reinhold. Convolutional neural networks for steel surface defect detection from photometric stereo images. In: International Symposium on Visual Computing. Springer, Cham, 2014. p. 668-677.

[9] Tao, X., Zhang, D., Ma, W., Liu, X., and Xu, D. Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks[J]. Applied Sciences, 2018, 8(9): 1575.

[10] ZHOU, Shiyang, et al. Classification of surface defects on steel sheet using convolutional neural networks. Mater. Technol, 2017, 51.1: 123-131.

[11] FEI-FEI, Li; FERGUS, Rob; PERONA, Pietro. One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence, 2006, 28.4: 594-611.

[12] Miller, Matsakis, and Viola, Learning from One Example through Shared Densities on Transforms. Proc. Computer Vision and Pattern Recognition, 2000

[13] VINYALS, Oriol, et al. Matching networks for one shot learning. In: Advances in Neural Information Processing Systems. 2016. p. 3630-3638.

[14] KOCH, Gregory; ZEMEL, Richard; SALAKHUTDINOV, Ruslan. Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop. 2015.

[15] BROMLEY, Jane, et al. Signature verification using a" siamese" time delay neural network. In: Advances in neural information processing systems. 1994. p. 737-744.

[16] WEN, Yandong, et al. A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision. Springer, Cham, 2016. p. 499-515.

[17] CHOPRA, Sumit; HADSELL, Raia; LECUN, Yann. Learning a similarity metric discriminatively, with application to face verification. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005. p. 539-546.