Fabric Defect Detection System Using Stacked Convolutional Denoising Auto-Encoders Trained with Synthetic Defect Data

Young-Joo Han 1,2 and Ha-Jin Yu 2,*

1 R&D Center, Vieworks, Anyang-si 14055, Korea; orora71@gmail.com
2 School of Computer Science, University of Seoul, Seoul 02504, Korea
* Correspondence: hjyu@uos.ac.kr; Tel.: +82-2-6490-2448

Received: 5 March 2020; Accepted: 27 March 2020; Published: 6 April 2020

Abstract: As defect detection using machine vision is diversifying and expanding, approaches using deep learning are increasing. Recently, there have been much research for detecting and classifying defects using image segmentation, image detection, and image classification. These methods are effective but require a large number of actual defect data. However, it is very difficult to get a large amount of actual defect data in industrial areas. To overcome this problem, we propose a method for defect detection using stacked convolutional autoencoders. The autoencoders we proposed are trained by using only non-defect data and synthetic defect data generated by using the characteristics of defect based on the knowledge of the experts. A key advantage of our approach is that actual defect data is not required, and we verified that the performance is comparable to the systems trained using real defect data.

Keywords: autoencoders; CNN; fabric defect detection; synthetic defect generation

1. Introduction

Recently, the importance of researching effective defect detection systems has been increasing. The defect detection systems using deep learning show better performance than the conventional feature-based defect detection system in complex patterns.

The methods of detecting defects based on deep learning are largely divided into the methods using supervised learning and unsupervised learning. In supervised learning, methods such as image segmentation [1], classification [2], and image detection [3] are used. In unsupervised learning, methods such as autoencoders are used.

Studies of defect detection using supervised learning are actively progressing. The methods have high defect detection rate, but they need sufficient defect data for training. Due to the industrial area’s characteristics, the process of accumulating data over a long period of time is necessary to actually apply these studies.

There are studies on defect detection based on unsupervised learning using autoencoders. The studies include autoencoders for pre-training method [4] and detecting abnormal regions using the output from the autoencoders [5,6]. Studies on detecting abnormal regions using the output of the autoencoders have the advantage of not requiring labeled defect data but have lower detection rate than methods using supervised learning.

It is difficult to get a large amount of actual defect data. However, experienced workers can express the shapes and the types of defects. In this paper, we propose an efficient approach of identifying defects without actual defect data by generating artificial defect data using types and characteristics of the defects known by experienced workers to train stacked convolutional autoencoders.
Our contributions are:

1. We define a defect detection system for environments where it is hard to obtain a large amount of data.
2. We define a method of generating synthetic defects by using characteristics of the defects based on the knowledge of the experienced workers. We can train an autoencoder where the input is a defect image generated artificially and the output is the corresponding clean image.

The paper is organized as follows: Section 2 describes the conventional ways of detecting defects using unsupervised learning. Section 3 describes the proposed system components and implementations. Section 4 describes our practical implementation and test results. Finally, our summary and conclusions are provided in Section 5.

2. Related Works

Defect detection using computer vision has been widely studied for automated inspection systems. It has gradually displaced the traditional manual methods. Also, the study of the defect detection method using deep learning is accelerating the replacement of the traditional manual methods. As previously stated in Section 1, studies of deep learning-based defect detection are categorized into two groups: supervised and unsupervised learning-based methods. However, supervised learning-based method has the problem of limited defected data. To overcome this problem, there have been various studies on detecting defects using unsupervised learning-based methods.

The studies using autoencoders have mainly conducted for unsupervised defect detection method. These studies use differences between original image and restored image from the autoencoders to detect defects. Mei et al. [5] adopted the multi-scale convolutional denoising autoencoder (MSCDAE) architecture. They use the various scales of the image generated by the Gaussian pyramid and utilize the salt-and-pepper noise model to the input image to train the denoising autoencoders of various sizes. They try to detect defects by combining outputs of the autoencoders with various sizes. This method shows 80% accuracy and 0.65 F-Score on fabric datasets.

Bergmann et al. [7] used perceptual loss function to get better performance of the autoencoder in defect detection. They note that existing methods lead to large residuals in edge regions that have slight localization inaccuracies. To solve this problem, they proposed a method using perceptual loss function based on structure similarity to detect defects. The loss function measures luminance, contrast, and structure between image patches. Besides that, there are various approaches to detect defects based on unsupervised learning without using autoencoders. These studies tried to detect defects by using wavelet transformation [8–10] or Gabor filter [11,12].

Also, there are studies using an unsupervised and a supervised manner simultaneously. Yundong Li et al. [6] adopt Fisher criterion-based stacked denoising autoencoder (FCSDA). They designed a Fisher criterion-based loss function in the feature space to overcome the limited defected data problem. They trained two autoencoders and use them as a classification network and a segmentation network. The classification network is trained in a supervised manner with labeled dataset, and the segmentation network is trained in an unsupervised manner. If the classification network judges an image patch as a defect, the segmentation network locates the defect.

The aforementioned studies about unsupervised defect detection have focused on the loss functions to train the autoencoders suitably or ensemble the results of the various autoencoders to get better performance. They all trained denoising autoencoders and used them for defect detection. However, they did not consider much about adding noise to the input of the denoising autoencoders in the training phase.

They use only typical type of noise, such as Gaussian noise or salt-and-pepper noise. In our work, we focused on the noise used in the training phase. We added not only Gaussian noise but also generated synthetic defects to the input of the denoising autoencoders for training. Our method achieved significant performance improvement in real-world patterned fabric data.
3. The Proposed System

3.1. System Components

Our defect detection system is comprised of four units: the defect generation unit, the pre-processing unit, the convolutional autoencoders unit, and the post-processing unit. Figure 1 shows overall architecture of the proposed system. The images coming from the camera are divided into small patches and processed one-by-one. In the defect generation unit, synthetic defects are generated using the characteristics of defects that the experienced workers described. Then, the generated defects are added to the non-defect training data. The preprocessing unit normalizes the images to make the dataset have a common scale. The trained convolutional autoencoder is used to reconstruct the input patches. Once the defective patches are used as input to the autoencoder, the defects can be located using the differences between the defective patches and the reconstructed images. The post-processing unit detects defects by using thresholding and morphology filtering.

![Figure 1. Overall architecture of the proposed defect detection system.](image)

3.2. Defect Generation Unit

The defect generation unit is comprised of two stages: the defect pattern generation stage and the defect merge stage.

3.2.1. Defect Pattern Generation Stage

During the defect input stage, synthetic defects that the experts produce are expressed as image data. The human experts have knowledge of the various defects that occur frequently and also the defects that are very rare or hard to detect but critical for the quality of the products.

If the defects in fabric are due to stain, an expert just draws shapes with the stain’s color. Similarly, in the case of the defects due to scratches, we just draw achromatic shapes.

Because synthetic defect patterns are generated by humans, only small amount of synthetic defect data are available. To solve this problem, we apply various data augmentation methods to the synthetic defect data which include: color inversion, color augmentation, translation, flip, etc.

3.2.2. Defect Merge Stage

During the defect merge stage, outputs of the defect pattern generation stage are combined into the non-defected data. To combine the images, we use Poisson image editing [13] method or Alpha compositing [14] method. It depends on the shape and type of the defects. In our experiments, we use alpha compositing with high transparency for the synthetic stain defects, and we use alpha compositing with low transparency and Poisson image editing for the synthetic scratch defects. Details about the
synthetic defects in our experiments are described in Section 4.4. After combining the images, additive Gaussian noise is added for robust defect detection.

3.3. Pre-Processing Unit

The pre-processing unit normalizes the images to have zero mean value and standard deviation equal to one. We use the z-score normalization [15] method. Let the pixels from an image are \( \{x_i\} \) and the data matrix \( X = [x_0, x_1, x_2, \cdots, x_n] \). The normalized image data \( x' \) can be computed as:

\[
x' = \frac{x_i - \bar{x}}{\sigma}
\]

where \( \bar{x} \) is the mean value of \( \{x_i\} \) and \( \sigma \) is the standard deviation of \( \{x_i\} \) of the entire dataset.

3.4. Convolutional Autoencoders Unit

The autoencoder is trained to restore the non-defected image data from defected image data. The network architecture of the autoencoder is RED30 proposed in [16].

3.4.1. Convolutional Autoencoders

RED30 consists of an encoder with 15 convolutional layers, a decoder with 15 deconvolutional layers and element-wise sum layers for skip connections. All of the convolutional layers and deconvolutional layers use ReLU as activation functions. The architecture of the convolutional autoencoder is shown in Figure 2.

![Figure 2. The overall architecture of the network (RED30).](image)

The encoder extracts the feature of the image data and converts it into a latent vector. Then, the decoder converts the latent vector into a restored image data [17]. In these processes, the defects in the image data are removed. Difference between the input image data and the restored image data are used to detect defects.

3.4.2. Skip Connections

When the convolutional autoencoder network goes deeper, it does not work well. This is because too much details are already lost in the encoder. To handle this problem, the skip connections are added between two corresponding convolution and deconvolution layers. Let the output from a convolution layer is \( X_1 \) and the output from the corresponding deconvolutional layer is \( X_2 \). The input to the next deconvolutional layer is computed as follows:

\[
F(X_1, X_2) = max(0, X_1 + X_2).
\]
3.4.3. Training

In the training phase, the autoencoder learns a mapping from the input image with defects to the original clean image. The autoencoder is trained to minimize the mean squared error (MSE) between original patches and restored patches.

3.5. Post-Processing Unit

In our scheme, the difference between the original image \( x_i \) and the output \( \hat{x}_i \) of the autoencoder can be used as a clue for defect detection. Since our system depends on the difference of the pixel values, we use the naive thresholding approach for detecting defects. However, defects having small differences and defects only having differences in specific channels are difficult to detect by naive thresholding approach. To solve the problem with defects in specific channels, first, we calculate the Euclidean distance \( L(x_i, \hat{x}_i) \) as in Equation (3) for each pixel. We use logarithmic transformations with a bias for solving problems with defects having small differences. The results of these methods represent the abnormality of a pixel, \( K(x_i, \hat{x}_i) \) as:

\[
L(x_i, \hat{x}_i) = \sqrt{\sum_k (x_{i,k} - \hat{x}_{i,k})^2} \tag{3}
\]

\[
K(x_i, \hat{x}_i) = \log(L(x_i, \hat{x}_i) + 1) \tag{4}
\]

where \( x_i \) and \( \hat{x}_i \) mean the pixels in the original image and the output of the autoencoder and \( k \) means index of the channel. Additionally, morphology operations are used to remove noise before determining the defected region using thresholding.

4. Experiments and Results

4.1. Dataset

In this section, we evaluated the performance of our proposed defect detection system. We compared our proposed system with a defect detection system using real data. We applied the proposed system to the real-world fabric samples with various defects. The fabric samples were captured by VTC-2K10.5G-C19 [18] which is produced by Vieworks. We captured three types of fabric: dataset 1, 2, and 3 as shown in Figure 5. Each of the datasets is comprised of 64,000 images of 256 \( \times \) 256 size, and we resized the image to 128 \( \times \) 128 for training.

To measure the performance of the system, we captured 6000 images of real defected fabric samples also. The defects in the fabric samples are produced during the manufacturing process, or artificially applied to the fabric using knives, awls, inks, etc.

4.2. Generated Synthetic Defects

We assume that the defects generated by experts are comprised of four types: hole, stitching, stain, and misprinting. By considering the characteristics of each type, we generate synthetic defects and apply them to the non-defected image. In the case of hole and stitching defects, we draw achromatic ellipses and lines, and apply them to non-defected images by alpha compositing with low transparency or Poisson image editing. Similarly, we drew colored shapes for stain and misprinting. These are applied to non-defected images by alpha compositing with high transparency. All of the generated defects are randomly located and scaled to random size. Also, we used three data augmentation methods for synthetic defects: flip, transformation, color augmentation. Figure 3 shows examples of the generated synthetic defects.
Figure 3. Samples of the generated synthetic defects in our experiment: (a) non-defect images, (b) generated synthetic defect images, and (c) defect applied images.

4.3. Experimental Conditions

We train our model using 64,000 training images. The training dataset consists of non-defected images. We train RED30 with the Adam optimizer [19], Xavier initialization [20], a learning rate of 0.0001, a weight decay of 0.0005, and a noise level of 0.25. The number of filters is 64 and the filter size of the convolution and deconvolution layers is $3 \times 3$.

4.4. Verification

We compare the performance of our model with a defect detection system trained using real data. The baseline system we implemented is U-Net [21]. We train a U-Net using 13,000 training images with actual defects. Also, we used 6,000 test images with actual defects to measure the performance of the U-Net. The test images are the same as the images used in our system. When training the U-Net, we have used Adam optimizer, Xavier initialization, and a learning rate of 0.0001. The input and output size of the U-Net are $128 \times 128$, $88 \times 88$ respectively.

4.5. Evaluation

To evaluate performance of the networks, we use recall, precision, and F-score (F1 score). These values are computed as:

Recall $= \frac{TP_p}{TP_p + FN_p}$ \hspace{1cm} (5)

Precision $= \frac{TP_p}{TP_p + FP_p}$ \hspace{1cm} (6)

$F1 \ Score = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$ \hspace{1cm} (7)

Recall means the portion of the cases that the defect detector indicates as the defects among actual defects, while precision means the portion of the cases that are the actual defects among the cases that the detector indicated as defect. Table 1 shows the comparison between the performance of the U-Net and the performance of our proposed system.
Table 1. Performance comparison of the two methods.

|          | Recall | Precision | F1 Score |
|----------|--------|-----------|----------|
|          | Ours   | U-Net [21]| Ours     | U-Net   |
| Dataset 1| 0.732  | 0.916     | 0.788    | 0.648   |
| Dataset 2| 0.678  | 0.783     | 0.871    | 0.857   |
| Dataset 3| 0.501  | 0.697     | 0.822    | 0.742   |

4.6. Results Analysis

Figure 4 shows some examples of the results of our proposed system. Also, Figure 5 shows the comparison between the performance of U-Net and the performance of our proposed system. In the case of the proposed method, the performance of defect detection drops when the difference in pixel values is small, while in the case of the baseline system, this phenomenon does not appear often. This is because our proposed system detects defects by using the difference between the input image and the restored image. It can make the defects with a small difference look like noise.

![Figure 4. Samples of the outputs of the convolutional autoencoders: (a) defected fabric sample images, (b) reconstructed images from convolutional autoencoder, and (c) results of the proposed system.](image-url)
5. Conclusions

In this paper, we proposed a defect detection system using synthetic defect data based on stacked convolutional autoencoders. The proposed autoencoder is trained by using only non-defect data and synthetic defect data generated by using the characteristics of defects described from the knowledge of experts. To verify the performance of our method, we compared the performance of the proposed system with U-Net trained with actual defect data, and showed that the proposed system using only non-defect data and synthetic data can detect actual defects comparably with that using real defect data. This method can be applied to many industrial and medical applications such as cancer detection using the knowledge of experienced doctors.

As a defect detection system using synthetic defect data generated by using the characteristics of defect based on the knowledge of the experts, our system has a limitation of the low detection-rate of unknown defects. However, the limitation is also shared with defect detection system based on real defect data. The system can be improved by iteration of the whole process. If the human expert forgets to mention a type of defect and the system makes some errors, the expert can add the type and some related types.

Our immediate future work includes enhancing the performance with larger and various learning datasets and an enhanced defect generating unit since the performance of our system can vary widely with different learning and test datasets. We can also improve the performance by solving the problem of low recall and precision when the difference between the input image and the restored image is very small. As one of the solutions, we may set a threshold in the tolerance according to the applications such that we can decide the limit of small defects that is allowed in a specific application. Finally, we will apply semisupervised learning which is based on the use of a large amount of unlabeled data but also employs a small amount of labeled data which may be available as in [22].
Author Contributions: Conceptualization, Y.-J.H.; methodology, Y.-J.H.; software, Y.-J.H.; validation, Y.-J.H.; formal analysis, Y.-J.H.; investigation, Y.-J.H.; resources, Y.-J.H.; data curation, Y.-J.H.; writing—original draft preparation, Y.-J.H.; writing—review and editing, H.-J.Y.; visualization, Y.-J.H.; supervision, H.-J.Y.; project administration, H.-J.Y.; funding acquisition, H.-J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2020R1A2C1007081).

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Yu, Z.; Wu, X.; Gu, X. Fully Convolutional Networks for Surface Defect Inspection in Industrial Environment. In *International Conference on Computer Vision System*; Springer: Cham, Switzerland, 2017; p. 10528.
2. Faghhi-Roohi, S.; Hajizadeh, S.; Núñez, A.; Babuska, R.; De Schutter, B. Deep convolutional neural networks for detection of rail surface defects. In Proceedings of the IJCNN, Vancouver, BC, Canada, 24–29 July 2016; pp. 2584–2589.
3. Li, Y.; Huang, H.; Xie, Q.; Yao, L.; Chen, Q. Research on a surface defect detection algorithm based on MobileNet-SSD. *Appl. Sci.* 2018, 8, 1678. [CrossRef]
4. Şeker, A.; YÜKSEK, A. Stacked autoencoder method for fabric defect detection. *Cumhur. Üniversitesi Fen-Edeb. Fakültesi Fen Bilimleri Derg.* 2017, 38, 342–354. [CrossRef]
5. Mei, S.; Wang, Y.; Wen, G. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors* 2018, 18, 1064. [CrossRef] [PubMed]
6. Li, Y.; Zhao, W.; Pan, J. Deformable patterned fabric defect detection with fisher criterion-based deep learning. *IEEE Trans. Autom. Sci. Eng.* 2016, 14, 1256–1264. [CrossRef]
7. Bergmann, P.; Löwe, S.; Fauser, M.; Sattlegger, D.; Steger, C. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv* 2018, arXiv:1807.02011.
8. Zhu, B.; Liu, J.; Pan, R.; Gao, W.; Liu, J. Seam detection of inhomogeneously textured fabrics based on wavelet transform. *Text. Res. J.* 2015, 85, 1381–1393. [CrossRef]
9. Li, P.; Zhang, H.; Jing, J.; Li, R.; Zhao, J. Fabric defect detection based on multi-scale wavelet transform and Gaussian mixture model method. *J. Text. Inst.* 2015, 106, 587–592. [CrossRef]
10. Hu, G.H.; Zhang, G.H.; Wang, Q.H. Automated defect detection in textured materials using wavelet-domain hidden Markov models. *Opt. Eng.* 2014, 53, 093107. [CrossRef]
11. Hu, G.H. Automated defect detection in textured surfaces using optimal elliptical Gabor filters. *Optik* 2015, 126, 1331–1340. [CrossRef]
12. Bissi, L.; Baruffa, G.; Placidi, P.; Ricci, E.; Scorzon, A.; Valigi, P. Automated defect detection in uniform and structured fabrics using Gabor filters and PCA. *J. Vis. Commun. Image Represent.* 2013, 24, 838–845. [CrossRef]
13. Pérez, P.; Gangnet, M.; Blake, A. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*; Association for Computing Machinery: San Diego, CA, USA, 2003; pp. 313–318.
14. Wikipedia. Alpha Compositing. Available online: https://en.wikipedia.org/wiki/Alpha_compositing (accessed on 14 January 2020).
15. Wikipedia. Feature Scaling. Available online: https://en.wikipedia.org/wiki/Feature_scaling (accessed on 14 January 2020).
16. Mao, X.; Shen, C.; Yang, Y.B. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2016; pp. 2802–2810.
17. Masei, J.; Meier, U.; Cireşan, D.; Schmidhuber, J. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*; Springer: Berlin, Germany, 2011; Volume 6791, pp. 52–59.
18. Vieworks. TDI Line Scan Cameras. Available online: https://vision.vieworks.com/en/camera/tdi_line_scan (accessed on 14 January 2020).
19. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* 2014, arXiv:1412.6980.
20. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*; JMLR.org: Brookline, MA, USA, 2010; pp. 249–256.
21. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.

22. Bruni, R.; Bianchi, G. Effective Classification using a Small Training Set based on Discretization and Statistical Analysis. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 2349–2361. [CrossRef]