A Performance Evaluation of Defect Detection by using Denoising AutoEncoder Generative Adversarial Networks

Kyosuke Komoto, Shunsuke Nakatsuka,
Hiroaki Aizawa, Kunihito Kato
Faculty of Engineering, Gifu University
1-1, Yanagido,
Gifu, 501-1193, Japan
komoto@cv.info.gifu-u.ac.jp, nakatsuka@cv.info.gifu-u.ac.jp, aizawa@cv.info.gifu-u.ac.jp, kkato@gifu-u.ac.jp

Hiroyuki Kobayashi, Kazumi Banno
N-TECH
134 Toyo Aza Kawahara Yorocho Yorogun,
Gifu, 503-1334, Japan
hkobaya@ntech.co.jp, kbanno@ntech.co.jp

***Abstract*—In this paper, we discuss a method to detect defects in industrial products by using Denoising AutoEncoder Generative Adversarial Networks. In previous methods, a defective area is detected by restoring a defective product image which added an artificial defect to a non-defective product image by Denoising AutoEncoder (DAE). Therefore, a defective area is detected by subtracted image of them. We discuss whether further accuracy improvement is possible by introducing a framework of adversarial learning to DAE in order to restore a defective image to a non-defective image clearer.**

Keywords—Deep Learning, Denoising AutoEncoder, Adversarial Learning.

I. INTRODUCTION

A lot of researches aimed at visual inspection using machine learning in industrial products are carried out. In General, the number of defective products is overwhelmingly smaller than that of good products. Also, defective products have many variations such as dents, scratches, and foreign objects. Therefore, it is difficult to design features of defective-products with human hands or statistically learn their feature. Thus, a method of learning Denoising AutoEncoder from only non-defective products images and restoring them into non-defective product images then using subtraction images of them to detect defective area has been studied. However, restored images with DAE becomes unclear when compared with original images, and detection accuracy may decrease in some cases.

There are methods that use Generative Adversarial Networks (GAN) that can clearly restore and complement images [1][2][3]. Therefore, in this research, we train DAE to remove artificial defects that are added to non-defective product images as noises and restore them into non-defective product images. Moreover, it is possible to clarify restoration with DAE by applying GAN to DAE. We call this method Denoising AutoEncoder - Generative Adversarial Networks (DAE-GAN). We discuss whether highly accurate detection is possible by applying DAE-GAN to visual inspection.

II. DENOISING AUTOENCODER

DAE is a neural network that takes a noised input image and is trained to remove noises. Thus, output image of DAE becomes denoised input image. DAE is trained by minimize Mean Squared Error (MSE) expressed in (1).

$$\mathrm{MSE}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{1}{M}\sum_{i=1}^{M}(\boldsymbol{x}^{(i)} - f(\tilde{\boldsymbol{x}}^{(i)}))^2 \tag{1}$$

Where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ represent images before and after adding noise. M is the size of a minibatch. $f(\cdot)$ is the output of DAE.

III. GENERATIVE ADVERSARIAL NETOWRKS

GAN consists of 2 neural networks, the Generator and the Discriminator as shown in Fig. 1. Define $G(\cdot)$ and $D(\cdot)$ as the Generator and the Discriminator, respectively. $\boldsymbol{x}$ and $\boldsymbol{z}$ represent sampled images from training data and sampled noise variables from the distribution p_z. The Discriminator is trained to determine whether an input image is an output image of the Generator or $\boldsymbol{x} \sim p_{data}$ that sampled from the training data. Hence output of the Discriminator will be a probability of an input image sampled from the distribution p_{data}. The Generator is trained to learn a distribution p_g that is close to p_{data} so that the Discriminator does not distinguish the Generator's output from an image sampled from p_{data}.

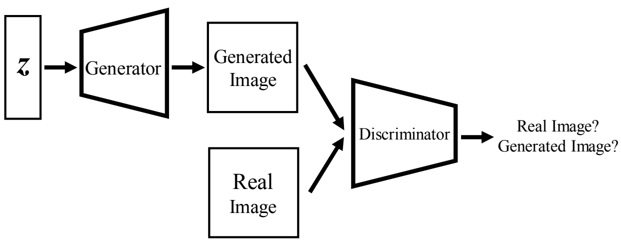


Fig. 1 GAN

Objective of GAN is denoted by (2). E represents expected value.

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}}[\log D(\boldsymbol{x})] + E_{z \sim p_z}[\log \log(1-D(G(z)))] \quad (2)$$

The Discriminator minimizes (3) to maximize (2).

$$L_d = -\frac{1}{M}\sum_{i=1}^{M}[\log D(\boldsymbol{z}^{(i)}) + \log(1-D(G(\boldsymbol{z}^{(i)})))] \quad (3)$$

The Generator minimizes (4) to minimize (2).

$$L_g = \frac{1}{M}\sum_{i=1}^{M}\log(1-D(G(\boldsymbol{z}^{(i)}))) \quad (4)$$

The Discriminator and the Generator are trained with gradient descent method with (3), (4). In this way, a training process in which Discriminator and Generator are trained mutually is called adversarial learning. Therefore, p_g will be close to p_{data} and the Generator outputs images that are close to training data.

IV. Denoising AutoEncoder - GAN

DAE-GAN has a configuration in which the generator of GAN is replaced with DAE. As shown in Fig. 2, the Generator restores a defective product image to a non-defective product image. The Discriminator determines whether its input is from the generator's restoration image or a real non-defective image. Therefore, it becomes possible for the generator to restore images of defective product to non-defective product in consideration of the distribution of non-defective products.

We define $\tilde{\boldsymbol{x}}$ is an image that added artificial defect on non-defective product image $\boldsymbol{x}$. The Generator minimizes (5) and (6). We introduce α, this is a hyper parameter that coordinates (5) and (6). Hence, the Generator minimizes (7).

$$L_{reconstruction} = \frac{1}{M}\sum_{i=1}^{M}(\boldsymbol{x}^{(i)} - G(\tilde{\boldsymbol{x}}^{(i)}))^2 \quad (5)$$

$$L_{adversarial} = \frac{1}{M}\sum_{i=1}^{M}\log(1-D(G(\tilde{\boldsymbol{x}}^{(i)}))) \quad (6)$$

$$L_g = L_{reconstruction} + \alpha L_{adversarial} \quad (7)$$

The Discriminator minimizes (8).

$$L_d = -\frac{1}{M}\sum_{i=1}^{M}[\log D(\boldsymbol{x}^{(i)}) + \log(1-D(G(\tilde{\boldsymbol{x}}^{(i)})))] \quad (8)$$

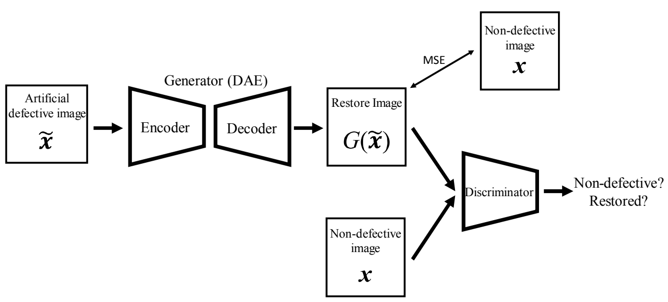
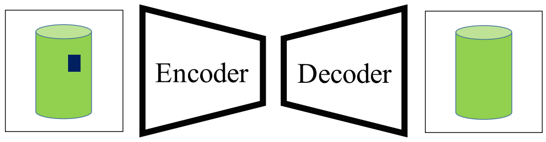


Fig. 2 DAE-GAN

The Generator and the Discriminator are trained mutually, output of DAE (Generator) becomes clearer because the Generator tries to deceive the discriminator.

V. Proposed Method

As shown in Fig. 4, the Generator (DAE) which is trained with DAE-GAN restores artificial defective product image to non-defective product image. As shown in Fig. 5, a defective area is detected by using the binarized subtraction image of these images.



(a) Artificial defective image

(b) Restore image

Fig. 4 Restoration with DAE

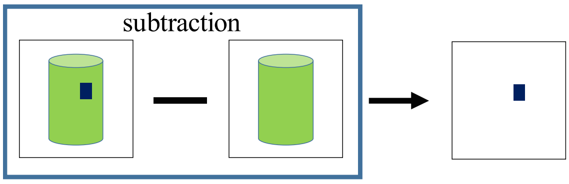


Fig. 5 Subtraction

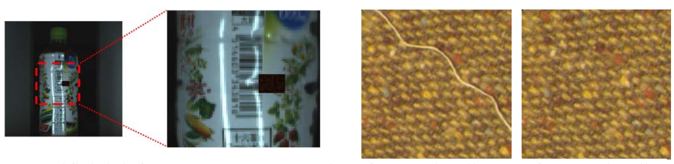
VI. Experiments

A. Dataset

We trained DAE-GAN with the plastic bottle product dataset and the textile product dataset.

We made a dataset of plastic bottle products by adding an artificial defect on images of them. An artificial defect is rectangle measuring from 5 to 15 pixels. Pixel values are (R, G, B) = (25±10, 15±10, 7.5±2.5) as uniform noises. We used 1860 of images for training, and we added artificial defects every epoch. We prepare 100 of each artificial defective product images for validation and testing. Fig. 3 (a) shows an example of artificial defective product image.

We made a dataset of textile products. We regard lint on textile as a defect. We made 100 pairs of images before and after putting lint taken at 1280 × 960 pixels. We used 80 pairs for training. Training images were cut out to an area of 224 × 224 pixels area, random position and random angle for each epoch. 80 pairs of cut images were used for an epoch. For validation and test dataset, we prepared 300 pairs of images which cut out at random positions and angles from 10 pairs of each of the remaining 20 pairs of images in advance. Fig. 3(b) shows an example of training images.



(a) Artificial defect (b) After putting lint Before putting lint

Fig. 3 An example of training data

B. Training

Tables 1, 2 show models of the Generator and the Discriminator. In the tables Conv(number of channels) represents Convolutional Layers. AveragePooling represents Average Pooling Layers. BN represents Batch Normalization layers. UpSampling represents UpSampling layers. FullyConnected (number of output units) represents Fully Connected layers.

We used Adam optimizer [4] for the Generator and the Discriminator. Initial parameters of Adam are alpha=0.000001 and beta1=0.1 in the Generator, alpha=0.0002 and beta1=0.5 in the Discriminator. In (7), we defined α with α=0.001 experimentally. Input images were 224 × 224 pixels. We trained DAE-GAN 1000 epochs for the plastic bottles dataset and 10000 epochs for the textile dataset.

C. Evaluation method

To evaluate detection of defects, it is necessary to evaluate both proper and false detection. Therefore, we evaluated Precision, Recall, F-measure of detection of a defective area.

We evaluated DAE-GAN every 10 epochs in the plastic bottles dataset, every 50 epochs in the textile dataset using validation datasets, and we adopted parameters when the F-measure is the maximum. The threshold value for binarization was adapted with the largest F-measure, 50 in the plastic bottle dataset, and 30 in the textile dataset.

VII. EXPERIMENTS RESULT AND DISCUSSION

A. Evaluation

We compared results of DAE-GAN and DAE. TABLE 3 shows the result of evaluation of the plastic bottle dataset and TABLE 4 shows the result of evaluation of the textile dataset.

In TABLE 3, Recall is the same in the result of DAE-GAN and DAE. However, Precision is higher in the result of DAE. Therefore, F-measure of DAE is higher than in DAE-GAN by 0.04. Thus, these results show DAE-GAN has more false detection than DAE.

In TABLE 4, all Precision, Recall and F-measure is better in the result of DAE than the result of DAE-GAN.

B. Detection of Defective Area

Fig. 6 shows an example of detection of defective area with DAE-GAN, input image, output image subtraction image, binarized image and ground truth image (GT). From the results of experiments, DAE-GAN can detect a defective area.

Fig. 7 shows comparison of restoration with DAE-GAN and DAE. From Fig. 7 (c) and Fig. 7 (d), DAE-GAN and DAE both can roughly detect a defective area.

C. DAE-GAN And DAE

Fig. 8 shows an enlarged image of comparison of restoration of DAE-GAN and DAE. When comparing Fig. 8 (b) with Fig. 8, visually, restoration with DAE-GAN is clearer than restoration of DAE especially in the area of characters.

TABLE 1 Model of the Generator

layer	Operation in the Generator
1	Input $\tilde{x}$
2	{Conv(32) + LeakyReLU + BN} ×2
4	AveragePooling
5	{Conv(64) + LeakyReLU + BN} ×2
7	AveragePooling
8	{Conv(128) + LeakyReLU + BN} ×3
11	AveragePooling
12	{Conv(256) + LeakyReLU + BN} ×3
15	AveragePooling
16	UpSampling
17	{Conv(256) + LeakyReLU + BN} ×3
20	UpSampling
21	{Conv(128) + LeakyReLU + BN} ×3
24	UpSampling
25	{Conv(64) + LeakyReLU + BN} ×2
27	UpSampling
28	Conv(32) + LeakyReLU + BN
29	Conv(3) + tanh

TABLE 2 Model of the Discriminator

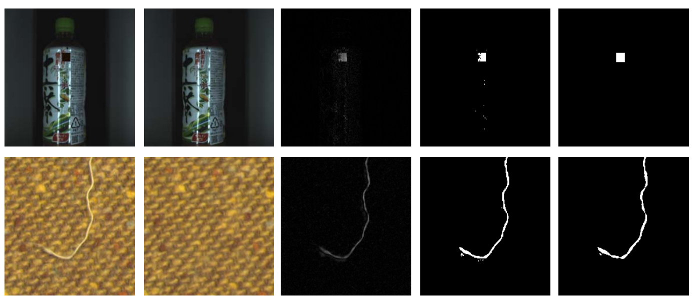
layer	Operation in the Discriminator
1	Input $G(\tilde{x})$
2	{Conv(16) + LeakyReLU} ×2
4	AveragePooling
5	{Conv(32) + LeakyReLU} ×2
7	AveragePooling
8	{Conv(64) + LeakyReLU} ×3
11	AveragePooling
12	{Conv(128) + LeakyReLU} ×3
15	AveragePooling
16	{Conv(256) + LeakyReLU} ×3
19	FullyConnected(2048) + ReLU
20	FullyConnected(2048) + ReLU
21	FullyConnected(1) + sigmoid

TABLE 3 Result of evaluation of plastic bottles dataset

	Precision	Recall	F-measure
DAE	0.79	0.59	0.68
DAE-GAN	0.68	0.59	0.64

TABLE 4 Result of evaluation of textile dataset

	Precision	Recall	F-measure
DAE	0.87	0.85	0.86
DAE-GAN	0.78	0.83	0.81



(a) Input (b) Output (c) Subtraction (d) Binarized (d) GT

Fig. 6 Detect defection with DAE-GAN

When using MSE of loss function of DAE, MSE has a precondition that non-defective images are added noises that follow normal distribution of average 0. Thus, output images become blurred. In contrast, loss function of adversarial learning doesn't consider the noise, therefore, output images become clearer. This phenomenon is reported by Isola *et al.* [5]. We confirmed this from the experiments.

In training of DAE, it can be confirmed that the training is progressing well due to the decrease in error. In contrast, training of DAE-GAN, loss function is the sum of mean squared error and loss of adversarial learning. Hence, it couldn't be possible that training is progressing well. For that reason, there is a possibility of insufficient of training and parameter tuning of DAE-GAN.

As shown in Fig. 8, restoration of DAE-GAN is clearer visually, thus, it appears superior to DAE. However, in the detection method of the experiments, we simply calculate subtraction image pixel-wise. Hence, blurred generated images have a larger difference in pixel value in the defective area than the clearer images as a result. Therefore, even if an image is clearer, accuracy is not always higher.

D. *Discuss of Datasets*

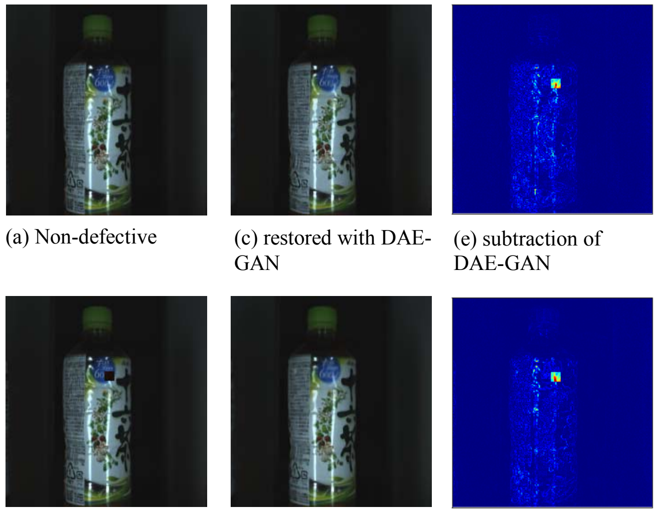
The dataset of plastic bottles images is taken by the same device and images of a single product. Hence variance among bottles is also very small. Also, the variation of change in the object in the images is only rotation and some Positional deviation. Therefore, sufficient restoration was possible with DAE.

As with the dataset of plastic bottles, the dataset of the textile products also used images of a single product and took them in the same environment. Therefore, variance between images is small. The reason why sufficient restoration was possible even in DAE may be that the texture pattern of the cloth product was simple.

In dataset that has small changes of pattern and small distribution of images, are not necessarily clearer images restored with DAE-GAN. However, it can be expected that restoration with DAE is not sufficient when applied to products with a large variance. Therefore, there is a possibility that the framework of GAN that can obtain clearer images effectively works.

VIII. Conclusion

In this paper, DAE-GAN introducing DAE with the framework of GAN is effective for detection of defective area of a product. We conducted experiments of detection of artificial defects of plastic bottle products and detection of lint on textile products and evaluated them. As a result, restoration with DAE-GAN is clearer than restoration of DAE. However, from the results of experiments, we concluded that when we detect defects from subtraction image of defective product image and restored product image, DAE-GAN is not always superior.



(a) Non-defective (c) restored with DAE-GAN (e) subtraction of DAE-GAN

(b) Artificial defective (d) restored with DAE (f) subtraction of DAE

Fig. 7 Comparison of DAE-GAN and DAE

(a) input (b) restoration with DAE-GAN (c) restoration with DAE

Fig. 8 Enlarged images of comparison of DAE-GAN and DAE

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Courville, Aaron Courville, and Yoshua Bengio: Generative Adversarial Nets. NIPS, pp. 2672–2680, 2014

[2] Avery Allen, and Wenchen Li: Generative Adversarial Denoising AutoEncoder for Face Completion https://www.cc.gatech.edu/~hays/7476/projects/Avery_Wenchen/

[3] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang: Generative Face Completion. CVPR , pp. 3911-3919, 2017

[4] Diederik P. Kingma, and Jimmy Ba: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980, 2014.

[5] Isola Phillip, Zhu Jun-Yan, Zhou Tinghui and Efros Alexei A: Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004, 2016