Development of Variational Auto-Encoder based Anomaly Detection Technique for Pattern Images of Honeycomb Structure

OSHIMA Takahisa : Inspection & Measurement Group, Technology Platform Center, Technology & Intelligence Integration
SUZUKI Kazumasa : Inspection & Measurement Group, Technology Platform Center, Technology & Intelligence Integration
MAEDA Munehiko : Manager, Intelligent Sensing & Electronics Systems Group, Technology Platform Center, Technology & Intelligence Integration
IMAI Wataru : Quality Assurance Department, IHI AEROSPACE Co., Ltd.
KIMURA Kenshi : Quality Assurance Department, IHI AEROSPACE Co., Ltd.

Debonding defects in honeycomb sandwich panels can be detected by visualizing the inside of the structure nondestructively. However, it is difficult to obtain enough defective product images required for automatic detection by rule-based methods such as image processing. To solve the problem, we developed an automatic detection method with the Variational Auto-Encoder, which is a deep learning technique. With this method, even if only non-defective product images can be used as training data, we confirmed that an anomaly detection model could be generated and that the automatic detection could be performed. In addition, further high-precision detection model was realized by defining unique loss function in the training phase.

1. Introduction

Honeycomb sandwich panels are commonly used in aircraft parts for weight reduction. The panels are manufactured mainly by bonding face sheets, made of composite material, and a hollow columnar honeycomb core, made of aluminum or other foil. If a face sheet separates from the honeycomb core due to damage from outside or a manufacturing defect, the strength and stiffness of the structural member are reduced. For this reason, nondestructive inspection techniques are used to detect debonding defects. A typical inspection technique is ultrasonic testing, with the pulse echo method being commonly used. This involves ultrasonic waves being emitted from an ultrasonic probe set on the material surface and being reflected from any defect existing in the material. When these waves are received, their intensity and arrival time provide information about the defect. Honeycomb sandwich panels can be inspected by using changes in the intensity of the reflected ultrasonic waves that occur depending on the presence or absence of the honeycomb core at the adhesive interface between the face sheet and honeycomb core. Recently, a combination of the phased array method[1], which uses an array probe that contains multiple ultrasonic wave sensors, and a signal processing technology called the total focusing method (TFM)[2] has enabled imaging in which the focus can be set to any point in the structure. **Figure 1** shows an example of ultrasonic testing of honeycomb sandwich panels.

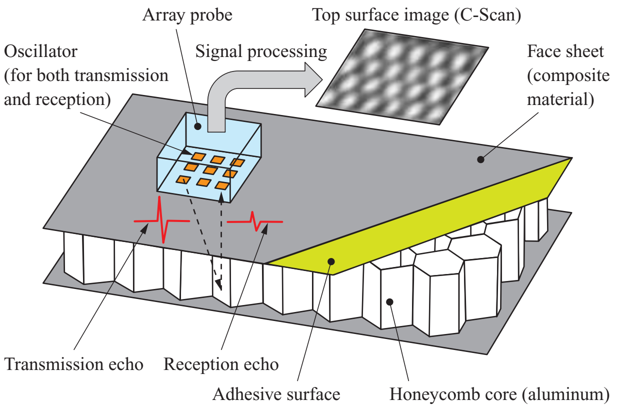


Fig. 1 Example of ultrasonic testing of honeycomb sandwich panel

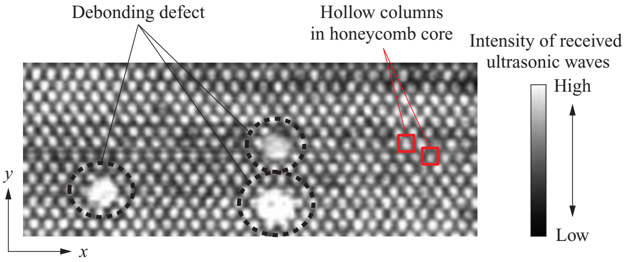


Fig. 2 Ultrasonic echo image of honeycomb sandwich panel and its defects

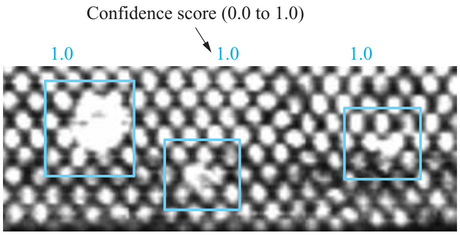
Figure 2 shows an example of imaging of the adhesive surface using ultrasonic testing. A debonding defect was artificially inserted. The whiter the area shows, the higher the intensity of the received ultrasonic waves is. Areas with good bonding are indicated in black, corresponding to low intensity, because ultrasonic waves are transmitted to the honeycomb core via the adhesive surface. When the adhesive surface is in good condition, it reflects a regular honeycomb pattern, giving a visual representation in which white dots are arranged regularly and separately from each other. However, if a debonding defect is present, an irregular pattern is observed, in which neighboring dots blur and merge into each other. The inspector checks this image visually to detect debonding defects, but there is strong demand for the inspection process to be automated. This is because the characteristics relating to use of structural members often make the inspection area large and inspection takes considerable time. However, the properties of the image depend greatly on the condition of the contact between the ultrasonic array probe in **Fig. 1** and the inspection target, and on the internal condition of the structural member. It is therefore difficult to conduct the inspection automatically because of difficulty in defining defects using common rule-based image processing approaches, such as filtering and shape detection. In addition, the number of defective products is generally much smaller than that of non-defective products, so that inability to collect sufficient detection target data is a problem.

This paper describes an anomaly detection model which enables the defect detection from honeycomb pattern images. This model was developed using deep learning, which makes it possible to solve target tasks with a flexibility close to that of humans, and which has been expected to have applications in the manufacturing industry in recent years. Specifically, honeycomb pattern images were reconstructed by an unsupervised learning method called the Variational Auto-Encoder (VAE), which does not require defective product data, with any change occurring in the process of reconstruction being detected as a defect. For this task, this paper also proposes a new loss function for conventional training methods, with the aim of enhancing detection model accuracy.

2. Anomaly detection using deep learning

2.1 Supervised and unsupervised learning methods

If it is possible to define defect data, for this study, an object detection approach can be selected as a supervised learning method. Typical object detection approaches include R-CNN[3], SSD[4], and YOLO[5]. **Figure 3** shows an example of output for defect detection using R-CNN. For this training, 100 dummy defect images were created by image editing. Object detection is multi-task processing by which the location of the object (bounding box), class of the object, and a confidence score for classifying the object as in that class (0.0 to 1.0) are output at the same time. For example, only one class is used in **Fig. 3** because defect is the only classification. In general, this method has the advantage of being able to detect defects directly and accurately, but also has the disadvantage of not being able to deal with changes in the definition of the defect when manufacturing conditions for the product change, so that training data needs to be collected again.



(Note) □ : Location of object (bounding box)

Fig. 3 Output example with object detection method (R-CNN)

Unsupervised learning methods are generally used for clustering to extract regularities from the input data, and typically include the k-means method and self-organizing map. An advantage is that because only non-defective product data is handled, the learning model is allowed to learn the range of normal data, making it easy to deal with unexpected defects. However, it is also necessary to note the disadvantage that if the data learned by the model lacks diversity, then the range of normal data becomes narrow, resulting in frequent false detection. Because images are handled in this study, an approach was selected that is capable of generating images — typified by Auto-Encoder[6] and AnoGAN (Anomaly Detection with Generative Adversarial Networks)[7] — and Auto-Encoder was selected in consideration of the number of non-defective product data we have for performing training.

2.2 Anomaly detection using Auto-Encoder

Auto-Encoder is not a technique dedicated to anomaly detection. It was originally used to learn initial parameter values as a means of addressing the vanishing-gradient problem, which was a problem with multi-layered networks[6], and is one of the techniques that formed the foundation of the present third AI boom. Currently, Auto-Encoder is used mainly for dimensional compression and generative models. **Figure 4** shows a simple three-layered Auto-Encoder network model. It has the following three features:

(1) The model is trained so that the input and output are the same.

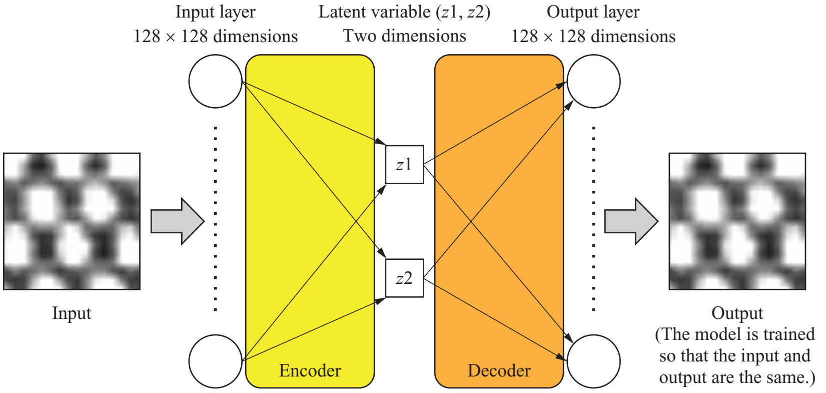
(2) It has an arrangement of symmetric layers called the encoder and decoder.

(3) In the intermediate layers, the encoder, which can obtain more dimensionally compressed representations called latent variables than the input, performs dimensional compression, and the decoder functions as a generative model for the new image.

Figure 5 shows an anomaly detection method using Auto-Encoder. Using only non-defective product images for training allows the encoder to extract the features of non-defective products as latent variables, and the decoder to restore non-defective product images based on those variables. Consequently, if a defective product image is given to the trained model as input, it is eventually converted to a non-defective image and produced as output. At this time, an image highlighting the defective area can be obtained by extracting the differences between the input and output, so that anomaly detection can be performed with any appropriate type of post-processing.

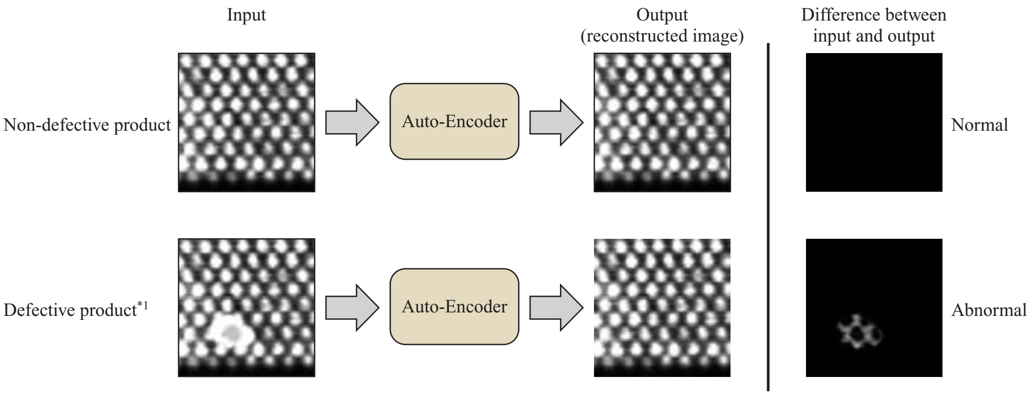
2.3 Anomaly detection by VAE

The following gives an overview of VAE, which performs anomaly detection in the same manner as Auto-Encoder. **Figure 6** shows a network model of VAE. A major feature in



(Notes) Number of pixels 128 × 128 pixel
Bias, activation functions, etc. are omitted.
○ : Input node
z1, z2 : Latent variable
⟶ : Weight

Fig. 4 Example of Auto-Encoder network model with three layers



(Note) *1 : Defects have not been learned, and are therefore not restored.

Fig. 5 Anomaly detection mechanism by Auto-Encoder

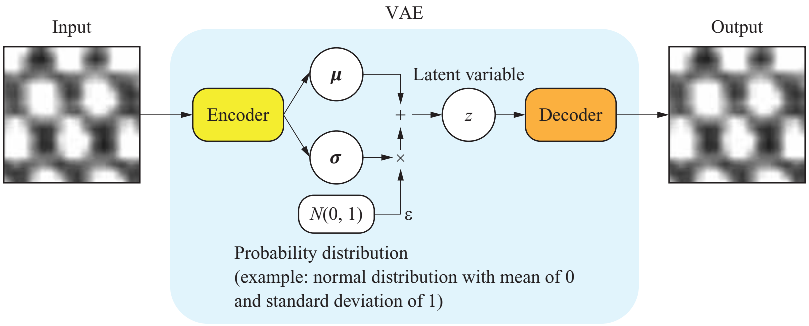


Fig. 6 VAE network model

comparison to Auto-Encoder is that VAE uses a probability distribution to represent latent variables. The input is not directly converted to latent variables; instead, based on the assumption that the latent variables are sampled from a certain probability distribution, the encoder calculates the mean and variance. However, with latent variable sampling, error back-propagation is not possible during model training. Therefore, an approximation expression like Equation (1) is used (Reparametrization Trick(8)).

$$z = \boldsymbol{\mu} + \varepsilon \boldsymbol{\sigma}^2 \qquad \cdots\cdots (1)$$

z : Latent variable
$\boldsymbol{\mu}$: Mean
$\boldsymbol{\sigma}$: Standard deviation
ε : Sampling from a normal distribution with a mean of 0 and standard deviation of 1

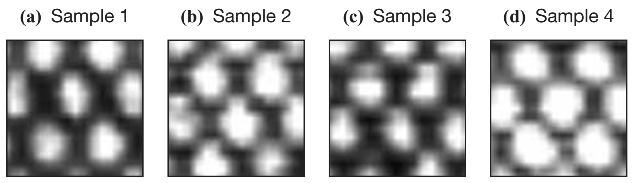
With Auto-Encoder, an input image is deterministically converted to latent variables, and it is difficult to understand the structure of the data space that those variables create.

With VAE, however, latent variables are limited to a probability distribution structure, and therefore, those that are close to each other in the probability distribution have similar feature values and provide similar images when processed by the decoder. For this reason, with VAE, it is easy to create continuously changing images. **Chapter 3** reports the results obtained by developing a VAE model and applying it to honeycomb pattern images.

3. Experimental results

3.1 Training data

Figure 7 shows examples of honeycomb pattern images of non-defective products used for training. The total number of images was 1 200, each being a gray scale image consisting of 128 × 128 pixels. These images were obtained by dividing an ultrasonic visualization image of a honeycomb sandwich panel with a certain size into multiple images, and then upsampling them to the specified size. Too fine division results in inputting an image showing only a white blurred area when a large defective area was evaluated. Therefore, in order to obtain a sufficient number of training data sets, the scale was adjusted so that one image contained seven to eight honeycomb cells.



(Note) Number of pixels : 128 × 128 pixel

Fig. 7 Examples of training data sets

3.2 Construction model and training

Keras[9], which is an open-source deep learning library, was used as the framework for implementation. **Figure 8** shows the structure of the VAE network model that was developed. Because the images handled were relatively simple, the structure of both the encoder and decoder consisted of a stack of only two or three convolutional and fully connected layers. The kernel size of the convolutional layers was 7 × 7 pixels. The latent variables were vectors of 32 dimensions, and the probability distribution was normal. The numbers of batches and epochs during training were 32 and 30, respectively. As a loss function, Equation (2), which is commonly used for VAE, was adopted.

$$L_{total} = L_{recon} + L_{KL} \quad \cdots\cdots(2)$$

$$L_{recon} = \sum\left\{-p\log q-(1-p)\log(1-q)\right\} \quad \cdots\cdots(3)$$

$$L_{KL} = \frac{1}{2}\sum(-2\boldsymbol{\sigma}+\boldsymbol{\mu}^2+\boldsymbol{\sigma}^2-1) \quad \cdots\cdots(4)$$

L_{total} : Loss function

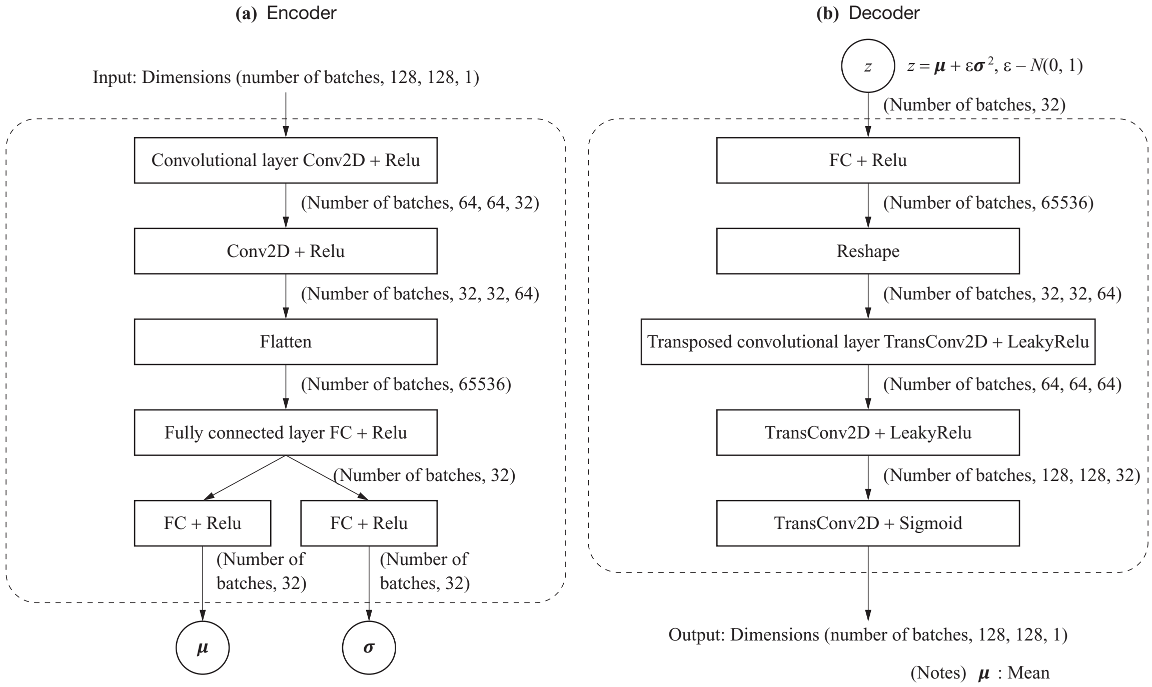
L_{recon}: Reconstruction loss due to cross entropy between pixels (differences between input and output images for entire model)

L_{KL} : Kullback-Leibler distance (scale that represents the degree of consistency between the latent variable probability distribution and normal distribution [mean: 0, standard deviation: 1]). RMSProp[10] was used as the optimization algorithm.

p : Pixel value of input image

q : Pixel value of output image

$\boldsymbol{\mu}$: Mean vector of probability distribution for latent variable sampling


(a) Encoder
Input: Dimensions (number of batches, 128, 128, 1)
Convolutional layer Conv2D + Relu
(Number of batches, 64, 64, 32)
Conv2D + Relu
(Number of batches, 32, 32, 64)
Flatten
(Number of batches, 65536)
Fully connected layer FC + Relu
(Number of batches, 32)
FC + Relu
FC + Relu
(Number of batches, 32)
(Number of batches, 32)
μ
σ
(b) Decoder
z
z = μ + εσ², ε − N(0, 1)
(Number of batches, 32)
FC + Relu
(Number of batches, 65536)
Reshape
(Number of batches, 32, 32, 64)
Transposed convolutional layer TransConv2D + LeakyRelu
(Number of batches, 64, 64, 64)
TransConv2D + LeakyRelu
(Number of batches, 128, 128, 32)
TransConv2D + Sigmoid
Output: Dimensions (number of batches, 128, 128, 1)

(Notes) $\boldsymbol{\mu}$: Mean
$\boldsymbol{\sigma}$: Standard deviation
z : Latent variable

Fig. 8 Architecture of constructed VAE network model

$\boldsymbol{\sigma}$: Standard deviation vector of probability distribution for latent variable sampling

3.3 Inference results

Figure 9 shows the results of inference for evaluation data sets having the same format as the training data. Defects were artificially introduced into the honeycomb sandwich panel. From the non-defective product images, it can be seen that the reconstructed images, which constitute the output of the model, have almost the same pattern, number of honeycomb cells, and honeycomb size as the input images. When an image containing a defect is input, the reconstructed image, from which the defect has been removed, shows only the normal honeycomb pattern because the model does not possess feature representation of the defect. This is advantageous for obtaining an image in which only the defect is highlighted when differences between the input and output of the model are extracted.

However, there were some cases in which, when an image containing a defect was reconstructed, not only the defect image was removed, but also the features of the original honeycomb pattern were lost (bottom of **Fig. 9-(b)**). **Chapter 4** proposes a method of mitigating this problem.

4. Improvement of VAE using shape feature value

Conventional Auto-Encoder and VAE use the cross entropy between the input and output pixels as a term of the loss function to represent restoration error during training (1st term in Equation (2)). To prevent the loss of the original honeycomb pattern and appearance of a pattern not contained in the original image, we evaluated not only individual pixels in the input and output images but also the gradient between neighboring pixels, and added a shape feature value to the loss function (Equation (5))[(11)]. Equation (6) shows the terms that appear in the shape feature value equation.

$$L_{total} = L_{recon} + L_{KL} + L_{shape} \quad \cdots\cdots(5)$$

$$L_{shape} = \frac{1}{2}\sum\left\{L_H\left(\frac{\partial p}{\partial x},\frac{\partial q}{\partial x}\right) + L_H\left(\frac{\partial p}{\partial y},\frac{\partial q}{\partial y}\right)\right\}$$

$$L_H(s, t) = -s\log t - (1-s)\log(1-t) \quad \cdots\cdots(6)$$

L_{shape} : Shape feature value

x : Horizontal pixel position in image

y : Vertical pixel position in image

Figure 10 shows the results of inference after re-training with the shape feature value added to the loss function. From the figure, it can be seen that with conventional VAE, when a defect is contained in an input image, the honeycomb pattern is deformed in the reconstructed image, but with the proposed method, the features of the honeycomb pattern are maintained. Therefore, for anomaly detection using differences between the input and output, more accurate extraction of defects is possible, and improved detection performance can therefore be expected.

5. Conclusion

We developed a non-defective product training model using VAE with the aim of automatically detecting debonding defects from the ultrasonic visualization images of honeycomb sandwich panels. We confirmed that debonding defect images can be obtained by extracting the differences between the input and output of the model after training. In addition, we added a new evaluation term, the shape feature

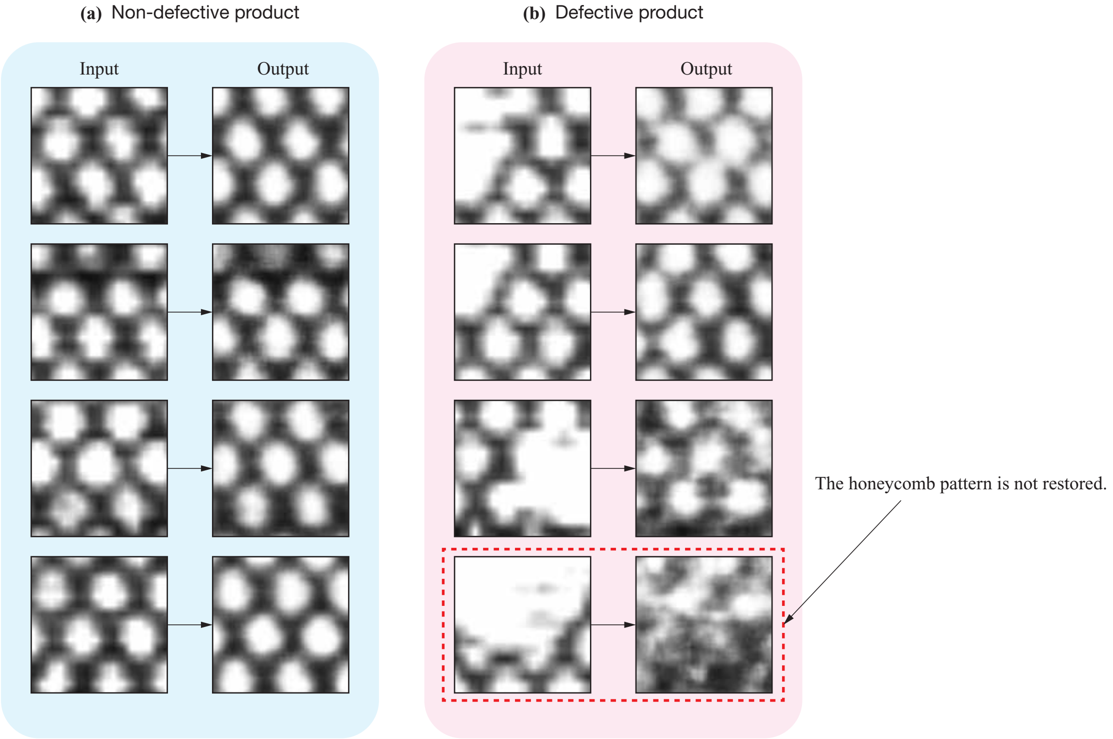
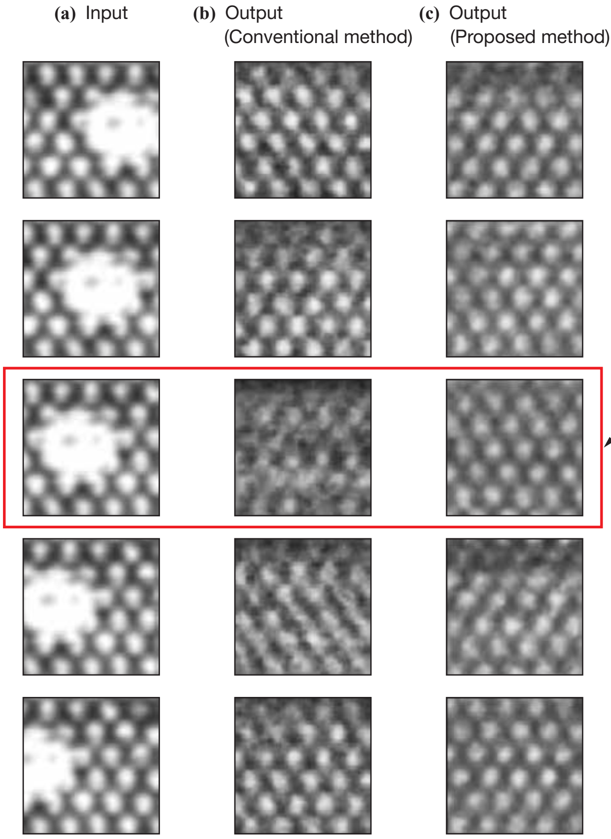

(a) Non-defective product
(b) Defective product
Input
Output
Input
Output
The honeycomb pattern is not restored.

Fig. 9 Inference results



This case shows the clearest improvement with the proposed method (the honeycomb pattern in the defective area is restored).

Fig. 10 Improvement of honeycomb pattern reconstruction by the proposed method with defective image input

value, to the loss function for training, resulting in improvement in quality of the reconstructed images used for detection. These results allowed us to expect to improve detection performance. Going forward, we are aiming to develop post-processing techniques for quantitatively evaluating the size and severity of defects and to utilize these techniques at manufacturing sites.

REFERENCES

(1) T. Oshima, Y. Nagai, M. Hato, H. Hatanaka, M. Tagami, M. Yokoyama, T. Asano, N. Fukushima and Y. Takeda : Examination of Ultrasonic Testing for Stainless Steel Weld Joints, IHI Infrastructure Technical Report, Vol. 5, 2016

(2) T. Oshima, H. Kawasaki and H. Hatanaka : Examination of Flaw Evaluation Methods in the Full Matrix Capture and Total Focusing Method, The Japanese Society for Non-Destructive Inspection (JSNDI), Proceedings of JSNDI Fall Conference, 2017, pp. 27-28

(3) S. Ren, K. He, R. Girshick and J. Sun : Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Advances in Neural Information Processing Systems, Vol. 28, 2015

(4) W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu and A. C. Berg : SSD: Single Shot MultiBox Detector, arXiv:1512.02325, 2016

(5) J. Redmon, S. Divvala, R. Girshick and A. Farhadi : You Only Look Once: Unified, Real-Time Object Detection, arXiv:1506.02640, 2016

(6) G. E. Hinton and R. R. Salakhutdinov : Reducing the Dimensionality of Data with Neural Networks, Science, Vol. 313, Iss. 5 786, 2006, pp. 504-507

(7) T. Schlegl, P. Seeböck, S. M. Waldstein, U. S. Erfurth and G. Langs : Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery, arXiv:1703.05921, 2017

(8) D. P. Kingma and M. Welling : Auto-Encoding Variational Bayes, arXiv:1312.6114, 2013

(9) Keras Documentation : Keras: Python Deep Learning Library, < https://keras.io/ja/ >, accessed 2019-12-5

(10) S. Ruder : An overview of gradient descent optimization algorithms, arXiv:1609.04747, 2016

(11) G. Sato, M. Maeda, K. Suzuki, Y. Uchida, T. Oshima and K. Yonekura : Improvement of VAE Reconstruction Error using Shape Feature Terms, Information-Based Induction Sciences and Machine Learning, The 21st Information-Based Induction Sciences Workshop (IBIS2018), D1-36