석사학위논문

Master’s Thesis

기계학습 알고리즘을 활용한 크랙 진단

Crack detection using machine learning algorithm

2019

신소진（申昭眞 Shin, Sojin）

한국과학기술원

Korea Advanced Institute of Science and Technology

석사학위논문

기계학습 알고리즘을 활용한 크랙 진단

2019

신소진

한국과학기술원

기계항공공학부 기계공학과

기계학습 알고리즘을 활용한 크랙 진단

신소진

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2018년 12월 18일

심사위원장 이필승 (인)

심사위원 손 훈 (인)

심사위원 윤국진 (인)

Crack detection using machine learning algorithm

Shin, Sojin

Advisor: Lee, Phill-Seung

A thesis submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in Mechanical Engineering

Daejeon, Korea
December 18, 2018

Approved by

Lee, Phill-Seung
Professor of Mechanical Engineering

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my dissertation contains honest conclusions based on my own careful research under the guidance of my advisor.

MME
20173313

신소진 기계학습 알고리즘을 활용한 크랙 진단 기계공학과
2019년. 46+ⅲ 쪽. 지도교수: 이필승. (영문 논문)
Shin, Sojin. Crack detection using machine learning algorithm. Department of Mechanical Engineering. 2019. 46+ⅲ pages. Advisor: Lee, Phill-Seung. (Text in English)

초 록

본 논문은 유한요소모델과 기계학습 알고리즘을 이용한 크랙 진단 방법을 제안하였다. 그러나 기계 학습을 크랙 진단에 이용하기 위해서는 많은 데이터가 필요하다. 따라서 본 연구에서는 XFEM을 이용하여 다양한 크랙에 대응하는 변형 데이터를 효율적으로 생성하고, 변형 데이터가 주어지면 이에 대응하는 크랙 이미지를 생성하는 크랙 진단 방식을 제안하였다. 이를 위해 대표적인 생성 모델인 variational autoencoder(VAE)의 구조를 이용하였으며, 문제에 맞게 손실 함수를 수정하였다. 제안된 방식의 크랙 진단 결과는 변형 데이터를 이용하여 크랙의 위치와 형태를 진단할 수 있음을 보여주며, 하중에 독립적인 모드 형상을 이용한 크랙 진단 방식은 추후 진동 데이터를 이용한 크랙 진단의 기초연구가 될 것으로 기대한다.

핵 심 낱 말 기계학습, 균열 진단, XFEM, 생성 모델, VAE

Abstract

In this paper, we propose a crack detection method using finite element model and machine learning algorithm. However, in order to use machine learning for crack detection, a lot of data is needed. Therefore, in this study, we propose a crack detection method that efficiently generates deformation data corresponding to various cracks using XFEM and generates a corresponding crack image when deformation data is given. To do this, we use the structure of variational autoencoder (VAE), which is a representative model, and modified the loss function to fit the problem. The crack detection results show that the position and shape of the crack can be detected using the deformation data. The crack detection method using the mode shape independent of the load is expected to be a basic study of the crack detection using the vibration data.

Keywords machine learning, crack detection, XFEM, generative model, VAE

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1. Background

Cracks are one of the major causes of structural failure. Detection of crack in structure such as infrastructure, aircraft, and automobiles is very important because it is directly related to safety of user. Therefore, there is a need for techniques to identify and manage the presence of cracks before failures occur.

Thus, there are many nondestructive techniques for crack detection from data obtained from structures for crack detection. Typical examples include ultrasonic technology, radiation transmission technology, and eddy current technology. Ultrasonic technology is an inspection method to determine the position and size of discontinuity by analyzing the energy amount of ultrasonic waves reflected from a structure and the progress time of ultrasonic waves. The radiation transmission technique is a method of detecting defects by recording the difference in density between the normal region and the abnormal region of the radiation on a two-dimensional image. Eddy current testing technology is an inspection method that analyzes the change of alternating current called eddy current due to crack. These inspection methods have to analyze the data change of the external stimulus at the crack location in order to detect the crack, so it is difficult to have experienced experience and research.

Therefore, researches have been carried out to detect crack using artificial neural network for data analysis. Kornelija Zgonc and Jan D. Achenbach used ultrasonic response results to detect the size of cracks in the rivet hole [1]. Since a lot of data is required to train the network, they use synthetic data generated by 2D FEM together with experimental data. T. Chady and M. Enokizono used the eddy current response results obtained from the newly proposed sensor [2]. They studied the results of eddy current response on neural networks using simple specimen crack specimens. And depth results of cracks are shown by image information. In [3], neural network training was performed using only the vibration response data obtained from the FE model to study the artificial neural network. From these results, we extracted 13 features such as maximum response and minimum response from the ultrasonic vibration response data obtained from the FEM numerical analysis. As a result, it is suggested that the position and length of cracks can be detected for simple straight cracks.

Recently, deep learning technology has become popular due to the improvement of computer hardware performance, and deep learning is being used for crack detection. A neural network study was developed to predict the fatigue crack length and residual fatigue life [4]. The previous studies have not been limited to the shape of the structure since the feature length is extracted from the data obtained by the external stimulus and the position and length of the crack are detected by learning it in the artificial neural network. However, there is a disadvantage in that it is not possible to intuitively obtain image information such as the size and position of cracks in the structure. Therefore, there has been a study on the genetic algorithm method that predicts the current state in the direction of minimizing the monitoring difference and the predicted result of the XFEM model on the fixed structure in which the crack existed [5]. This work has the advantage of detecting many cracks in a three-dimensional structure, but it has a disadvantage in that efficiency is low because a large number of calculations are needed to detect cracks in one case. In addition, many studies for crack diagnosis have been made [6-15].

In the above studies, detection was done in a way that indicates the position and size of the crack numerically, not the crack image on the structure. Therefore, there was a disadvantage that the shape and position of the cracks could not be easily identified. The cracks that could be represented by images were relatively simple because they required the preparation of specimens or data from experiments. However, actual cracks appear on the structure in general form, and detection is required. Therefore, data for various cracks are required, and a method for efficiently generating data is required.

As image recognition and generation technology using deep learning has been developed recently, studies using image generation networks such as GAN and VAE are being conducted. In particular, research has been conducted on the use of deep running in the field of topology optimization, where computational efficiency is a problem [16]. In addition, research has been carried out to draw an image of a two-dimensional hand gesture, that is, a pose, from an image of a human hand gesture [17]. In addition, research has been carried out to generate a sketched image as a colored image, and research on generating images from sentences has also been conducted [18-20]. This study will also use this machine learning algorithm.

1.2. Research purpose

In this study, we want to show the possibility of image-detecting the cracks from external signals for the cracks in the structure. Therefore, the goal to overcome the limitation of the existing research is as follows.

- Data generation: efficient generation of data using XFEM model
- Data analysis : automatic feature extraction using deep learning
- Result : visualization of results using generation model

This study used a simple 2D rectangle structure and readily available deformation data to identify the feasibility of achieving these goals. First, the proposed method automatically generates crack data in various directions and sizes from the XFEM. Next, deformation data is processed into image in order to train deep learning network. Finally, the position and shape of the crack are detected as an image from deformation image not using for the network training. In this basic flow, several deep learning models were applied to crack detection and its performance was evaluated.

1.3. Organization of Dissertation

This paper is structured as follows:

In chapter 1, the research background is presented. We introduced research purpose and this dissertation.

In chapter 2, we proposed three deep learning networks that generate corresponding crack image when given deformation image. The performance of the proposed method was validated by quantitative evaluation of crack images generated using test data.

In chapter 3, There is a limitation that the previously used deformation data is dependent on the force condition. Therefore, we used the mode image independent of load as input data for training the proposed training network, and the performance was validated by quantitative evaluation.

Lastly, in chapter 4, we discuss about present conclusions and future works.

Chapter 2. Crack detection using deformation image

In this chapter, methods are proposed for detecting the corresponding crack image when given a deformation as input data. The performance of these methods are also verified. The methods are listed in order of improvement, and the method with best performance is selected.

2.1. Overview of proposed method

We proposed a method to detect the corresponding crack when given the external deformation of the structure. The method used machine learning algorithm for crack detection. Accordingly, it took the deformation and crack image pairs as a training dataset for deep learning algorithm. At this time, this problem can be regarded as a 1:1 match between deformation and crack. However, the proposed method aims to detect crack for new deformations that are not in the training dataset using a small amount of training data.

For this, the method is required to generate crack corresponding to the deformation of various cases. Therefore, generative model was used. The generative model is different from the discriminant model which accounts for most of the machine learning. In the discriminant model, when data x is input, training proceeds to know class c to which x belongs. That is, the objective is to obtain $p(c\,|\,x)$. The generative model, on the other hand, is not aimed at obtaining class c of data x. This model is interested in finding the distribution $p(x,c) = p(c\,|\,x)\cdot p(x)$ (Bayes' Rule) of data x belonging to class c. Finally, training proceeds to find $p(c\,|\,x)$ and $p(x)$, which is the distribution of x, and new data x' is generated from this distribution.

In this study, Variational Autoencoder (VAE) and Generative adversarial network (GAN), which are representative production models, were used. Accordingly, we briefly reviewed how VAE and GAN proceed to train $p(x)$.

2.1.1. Overview of generative model

2.1.1.1 Variational Autoencoder

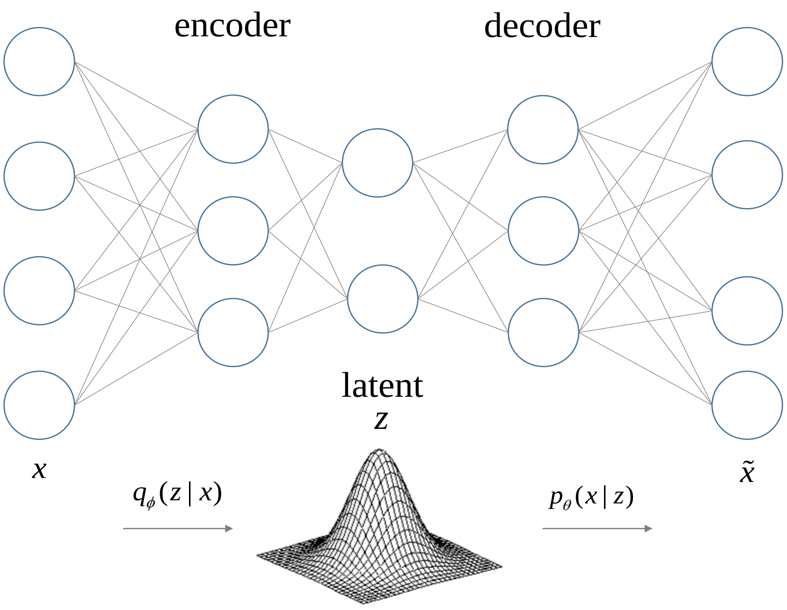


Figure 2.1 Variational Autoencoder

The training model of this study derived from VAE [21]. VAE is similar to Autoencoder (AE) as shown in Figure 2.1. These models train low-dimensional representation or distribution of data from structures that produce output such as input. However, VAE aims to generate data x from continuous distribution z, unlike AE which obtains low-dimensional representation z of high dimensional data. In other words, it is possible to create new data that is not in the existing training dataset if distribution z is obtained well.

Specifically, to obtain the previously defined $p(x)$, we have to find the distribution $p(z)$ of z and the distribution $p(x|z)$ that produces data x from z. However, $p(z)$ is not used as a sampling function because of the disadvantage that it is difficult to train [22]. Instead, it uses a distribution $p(z|x)$ that produces x properly when given x as input data. However, since this distribution is also unknown, one of the probability distributions $q_\phi(z|x)$ that can be easily used is selected so that the distribution is similar to $p(z|x)$.

Therefore, the following equation is used to obtain $p(x)$.

$$\begin{aligned}\log(p(x)) &= \int \log(p(x)) q_\phi(z \mid x) dz \\ &= \int \log(\frac{p(x,z)}{q_\phi(z \mid x)}) q_\phi(z \mid x) dz + \int \log(\frac{q_\phi(z \mid x)}{p(z \mid x)}) q_\phi(z \mid x) dz\end{aligned} \tag{2.1}$$

Where the first term of Equation (2.1) is called the evidence lower bound $ELBO(\phi)$ and the second term is the distance between the two probability distributions $q_\phi(z \mid x)$ and $p(z \mid x)$. (KL-divergence)

The training should proceed so that $q_\phi(z \mid x)$ is similar to $p(z \mid x)$. However, since KL-divergence is difficult to calculate, the training should proceed in the direction in which the KL-divergence decreases, that is, in the direction in which the $ELBO(\phi)$ increases.

For this, equation of this $ELBO(\phi)$ is as follows.

$$\begin{aligned}ELBO(\phi) &= \int \log(p(x \mid z)) q_\phi(z \mid x) dz - \int \log(\frac{q_\phi(z \mid x)}{p(z)}) q_\phi(z \mid x) dz \\ &= \mathrm{E}_{q_\phi(z|x)}[\log(p(x \mid z)] - KL(q_\phi(z \mid x) \parallel p(z))\end{aligned} \tag{2.2}$$

Finally, the following equations are considered for training in terms of variational inference and maximum likelihood using $ELBO(\phi)$.

$$\arg\min_{\phi,\theta} \sum\nolimits_i -\mathrm{E}_{q_\phi(z|x_i)}[\log(p(x_i \mid g_\theta(z)))] + KL(q_\phi(z \mid x_i) \parallel p(z)) \tag{2.3}$$

The first term is the reconstruction loss for the output data (same as the input data), and the second term is the KL-divergence for $q_\phi(z \mid x)$ and $p(z)$. Therefore, training is proceeded in such a way that the reconstruction loss is minimized so that the output becomes equal to the input, and the distance between the two distributions is minimized. It is the original VAE theory that the distribution of data x can finally be obtained in this process.

2.1.1.2 Generative adversarial network

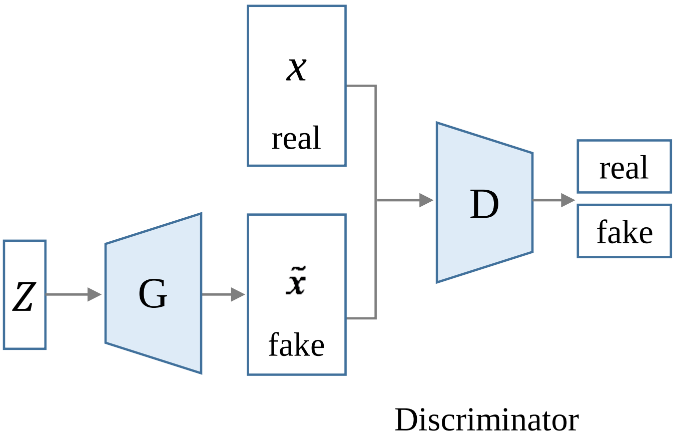


Figure 2.2 Generative adversarial network

The network used in this study is also derived from GAN [23]. The GAN trains to generate fake data similar to real data from noise by encoding noise z. Therefore, there exists a generator that generates fake data from random latent vector z, and there exists discriminator which distinguish fake data from real data. The purpose of the generator is to make the discriminator recognize fake data as real data. The purpose of discriminator is to judge real data as real and fake data as fake.

Therefore, the objective function of the GAN is as follows.

$$\min_G \max_D V(D,G) = \mathrm{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathrm{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2.4}$$

In Equation (2.4), discriminator proceeds so that a large value (close to 1) appears for the real data input and a small value (close to 0) for the fake data input. On the other hand, generator receives the noise z and trains that the fake data is judged as the real data by the discriminator. That is, the parameter is adjusted so that $1-D(G(x))$ value is maximized.

In actual training, parameters are updated separately without training two networks at the same time. As a result, the quality of the fake data generated by the generator at the beginning of training is not good. In this reason, discriminator distinguishes fake data well. For this reason, the parameters of the generator should be updated quickly at the beginning of the training.

Therefore, we use the objective function as follows to accelerate the generator training.

$$\min_G V(G) = \mathrm{E}_{z \sim p_z(z)}[\log D(G(z))] \tag{2.5}$$

From the viewpoint of reconstruction input data, VAE has the disadvantage of producing fake data similar to training data. On the other hand, GAN generates relatively new data well, but it has a disadvantage that training is difficult. There are two reasons why learning is difficult. The first is due to 'mode collapsing' [24], which is a phenomenon in which the training model does not fully reflect the real data distribution and loses its diversity. Next, if the generator's training is difficult and the force balance between the discriminator and the generator is broken, adversarial training of the two networks can no longer proceed. Research has been proposed to solve this problem. However, in this study, rather than creating a new data, it is important to generate crack data suitable for the deformation even though it is similar to the training data.

2.1.1.3 Adversarial Autoencoder

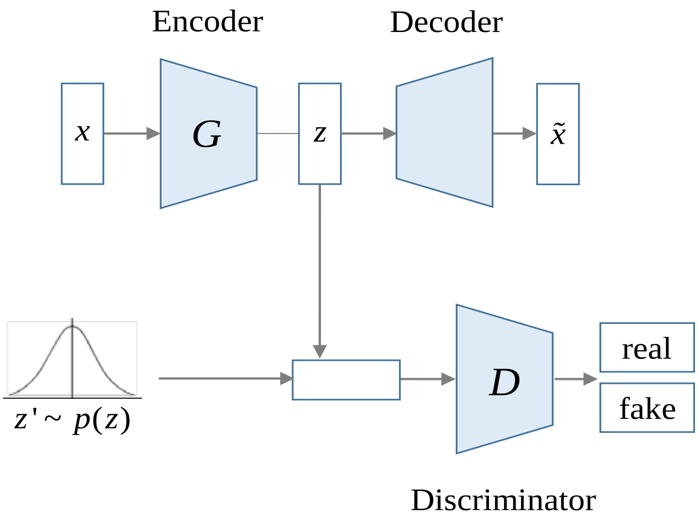


Figure 2.3 Adversarial Autoencoder

One of training networks of this study, the adversarial encoder and decoder network, derived from Adversarial Autoencoder (AAE) [25], which is a combination of GAN with VAE in Figure 2.3. AAE was created to compensate for disadvantages inherent in VAE. Therefore, a network that acts as a GAN discriminator in the existing VAE structure is added. For the calculation of KL-divergence, VAE assumes the prior probability distribution $p(z)$ a normal distribution that is easy to sample and proceeds with training to make $q_\phi(z\,|\,x)$ similar to $p(z)$. A problem arises, when the real data distribution does not follow the normal distribution. However, in the case of GAN, there is no need to assume a specific probability distribution in the model. This is because the GAN is taught to minimize the difference between the distribution of real data and the distribution of fake data generated by the generator. In other words, when the KL-divergence (Regularization) of VAE is replaced with GAN-loss (discriminator), model can use distributions other than the normal distributions.

The training process of AAE is as follows.

$$AE(\phi,\theta,x) = -\mathrm{E}_{q_\phi(z|x)}[\log(p_\theta(x\,|\,z))] \tag{2.6}$$

First, the KL-divergence term from the loss function of VAE is removed. Therefore, loss function of AE network is defined only with reconstruction loss as in Equation (2.6). That is, AE network proceeds to minimize the reconstruction loss.

$$-D(\phi,\lambda,x,z) = -\log d_\lambda(z) - \log(1-d_\lambda(q_\phi(x))) \tag{2.7}$$

Next, the discriminator parameters was updated in such a way as to discriminate real input z' from $p(z)$ as real, and fake input z form $q_\phi(z\,|\,x)$ as fake to minimize the loss function constructed as Equation (2.7).

$$-G(\phi,\lambda,x,z) = -\log(d_\lambda(q_\phi(x))) \tag{2.8}$$

Finally, a generator loss function is constructed as Equation (2.8) to determine the fake input as a real input by the discriminator. Generator of AAE is encoder in Figure 2.3, and training proceeds in the direction of minimizing the generator loss. As a result, when data x is input, the encoder generates the latent vector and the decoder reconstructs $\tilde{x}$ from the sampled value. AAE has the advantage of obtaining a result close to real data distribution because there is little restriction on the use of probability distribution. Therefore, this network is also used in this study because it has the merit of getting clear results which is characteristic of GAN.

2.1.2. Concept of proposed crack detection method

Figure 2.4 shows the concept of the proposed method. In step 1, the FE model is used to obtain the image of deformation corresponding to crack. To do this, boundary condition and force condition are defined and deformation of 2D rectangular structure with random crack is generated as image data using XFEM code [26]. Also, crack is also made the same size image. After that, pair of these two images is used as training data. In step 2, the crack detection network is trained using these data. There are three networks used in this study, and the basic framework is all the same.

Network 1 follows the behavior of the original VAE, and when deformation image is given, the training proceeds to generate the corresponding crack image. Network 2 and Network 3 add a discriminator in the same frame as Network 1. And these models have a difference according to the method of defining the loss function. Network 2 is trained to generate a crack image corresponding to the deformation image in the AAE frame. Network 3 adds a discriminator to the frame of Network 1. In this case, the loss is added to the loss function that updates the parameters of the encoder and decoder network, and training proceeds in a direction to minimize this loss. In step 3, we use the networks trained in step 2 for performance verification. Therefore, when a new deformation image is used that is not used as training data, it is checked whether a corresponding crack image is generated.

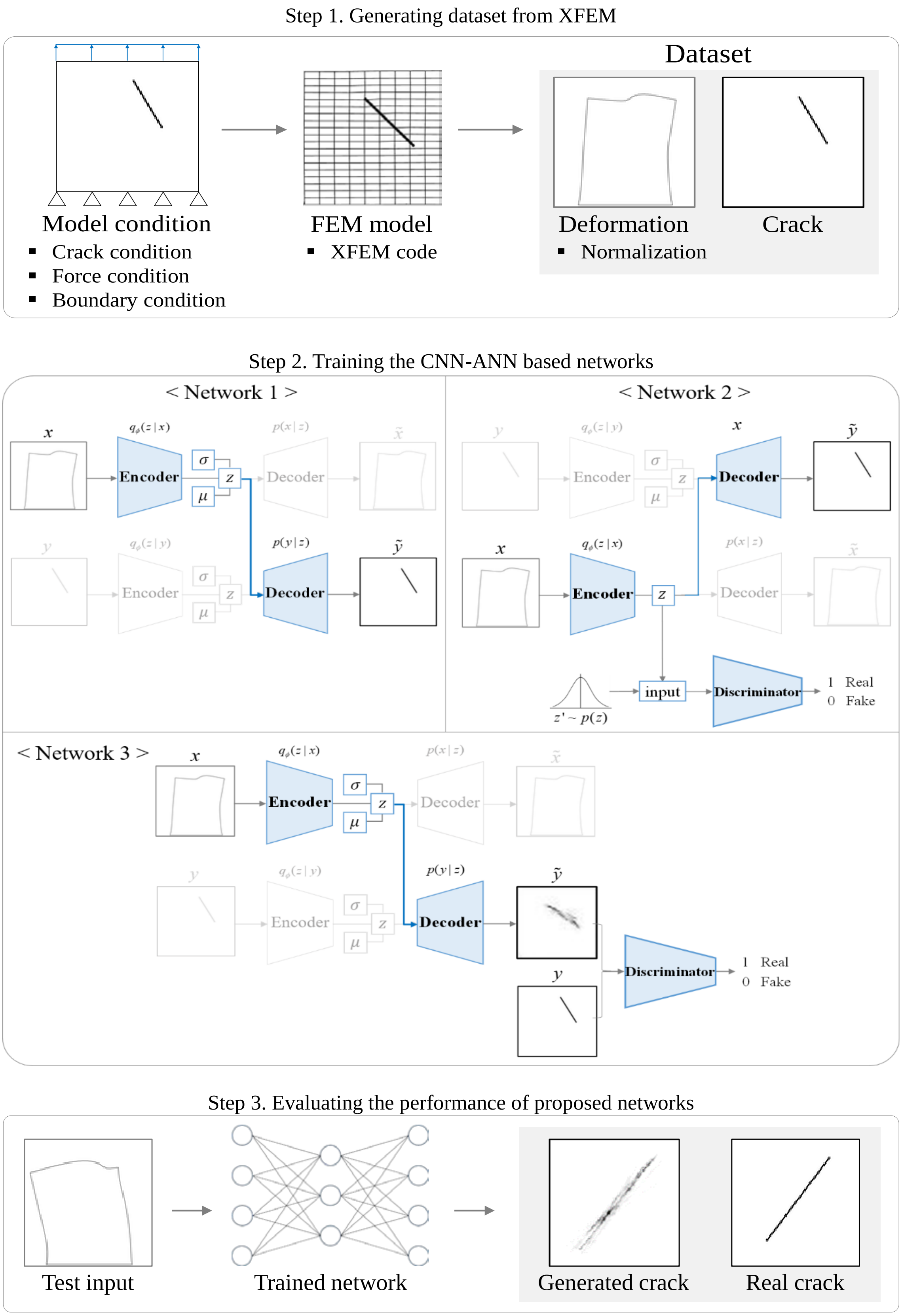


Figure 2.4 Concept of the proposed method

2.2. Generating the dataset

2.2.1. Generating the data with XFEM

The deformation data was generated in XFEM code [26]. To generate training data, a rectangular structure with the conditions in Table 2.1 was considered.

Table 2.1 Conditions of FE model of deformation data

Conditions	
Height	2
Width	2
Properties	200 / 0.3
Force	100

For the purpose of detection the internal crack, crack of random shape and location were created inside the rectangular structure. In addition, as shown in Figure 2.5, deformation data of Type 1 and Type 2 were generated by changing the force and boundary conditions acting on structures with the same crack. Type 1 fixed the lower surface and gave a uniform distribution force upwards. Type 2 provided a simply supported boundary condition in the y and x directions of the lower and left sides, respectively, and gave a uniform distribution force perpendicular to each plane direction on the top and right sides.

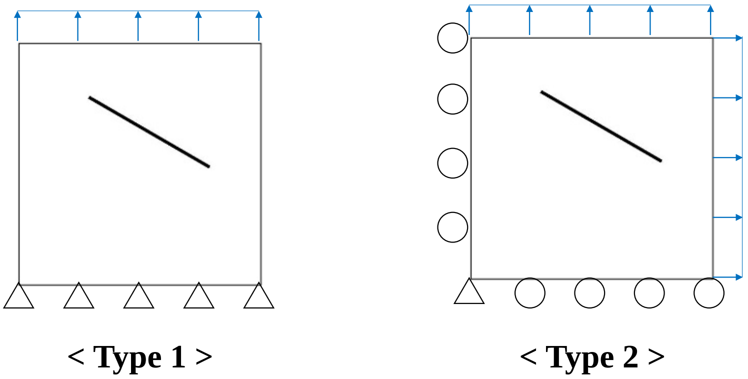


Figure 2.5 Boundary condition and force condition of Type 1 and Type 2

2.2.2. Refining the data with normalization

Make the result obtained from the FE model of the preceding conditions as image. At this time, as shown in Figure 2.6, the deformation is too small. Therefore, it is difficult to find out features in deep learning training with this data. Accordingly, the total deformation is normalized so that the maximum deformation is 20% of the diagonal length ($2\sqrt{2}$) so that it can be used for training.

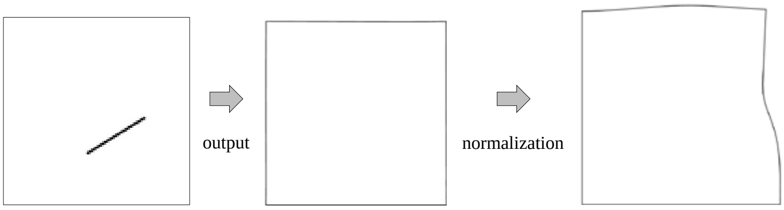


Figure 2.6 Output of XFEM and normalization result

A total of 2000 random crack and paired deformation data was created. Unnecessary parts were cut off for training. In order to improve training speed and performance, the size of data was converted from 1200×900 pixels to 100×100 pixels. We used the .png file to prevent data loss during the process of refining with training data. Finally, 2000 pairs of data used for training are shown in Figure 2.7.

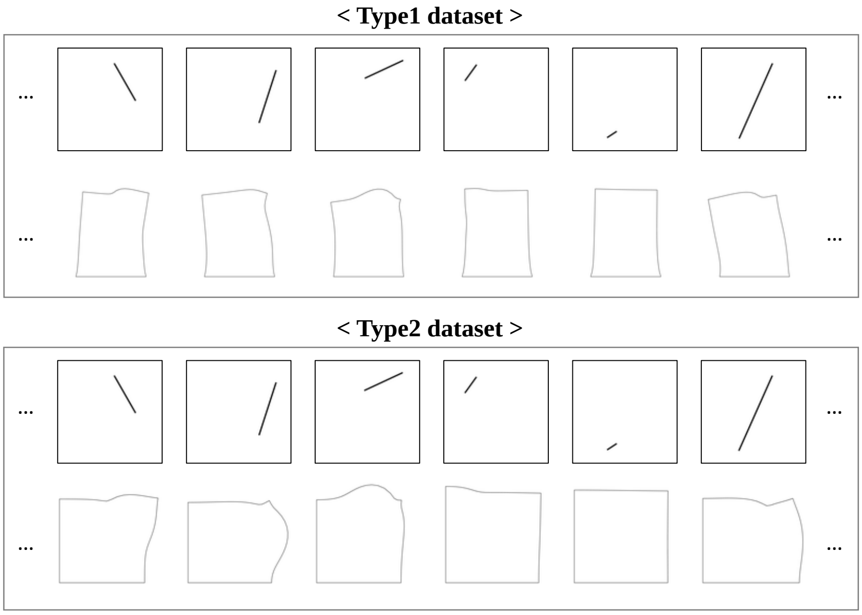


Figure 2.7 Training datasets of deformation

2.3. Training model

In this study, we used not only Artificial Neural Network (ANN) for training model. In order to take advantage of Convolutional Neural Network (CNN), which is excellent for image recognition, ANN and CNN network were used in combination. The CNN was used for image recognition in the encoder section, which trains the characteristics of the image. The decoder that generates the crack image used the ANN model because its shape was just straight and simple. In this way, the networks that have been improved and applied sequentially for crack detection are introduced, and the performance is summarized as a result.

2.3.1. Variational encoder and decoder network (Network 1)

In this study, it is necessary to generate accurate crack image corresponding to deformation, not completely new image. Therefore, we adopted the structure of VAE which shows a reliable result even though there is a tendency to overfitting the training data rather than GAN. However, VAE has the same structure of input and output so that it can automatically find the latent space of data when only input data is given, whereas the problem is to generate output data different from input. Therefore, only the operating structure in VAE was taken, and the model was called a variational encoder and decoder network (Network 1).

2.3.1.1 Detail of Network 1

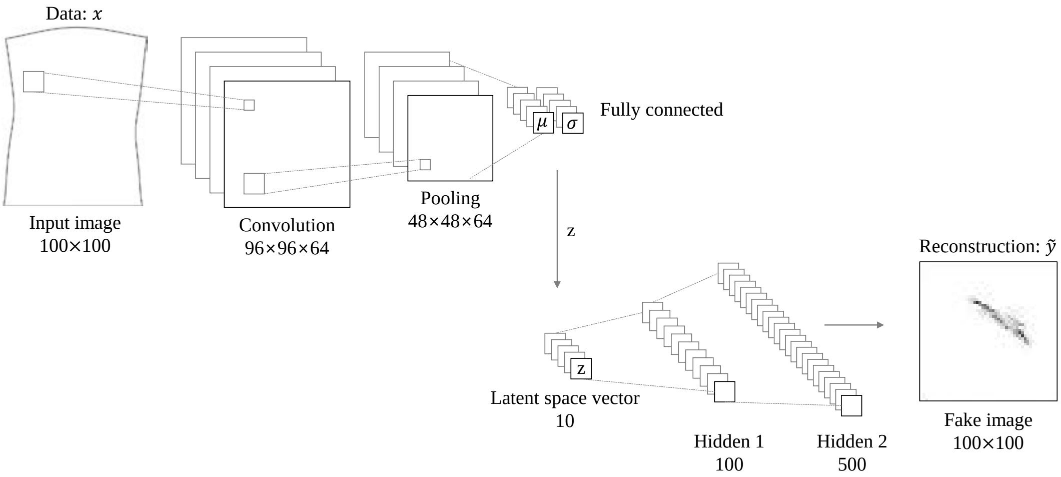


Figure 2.8 Detail of variational encoder and decoder network

In this study, the model should be reconfigured to create a crack corresponding to the deformation. To do this, we first need to find the latent space of the deformation and the latent space of the crack. Therefore, the VAE model of each image was constructed separately, and the mapping function between the latent space of the deformation and the latent space of the crack was trained. However, in this study, it can be assumed that the two data share a similar latent space because deformation and crack correspond to each other 1:1. Therefore, the unified latent space was constructed so that deformation and crack share one latent space. Specifically, Figure 2.8 shows the training of two identical VAE networks. First, VAE network for deformation is an encoder network of CNN structure. When deformation is input, 10 latent vectors are output. Next, VAE network for the crack is a decoder network of the ANN structure. When the result of the preceding encoding is input, a crack image having the same size as the deformation image is generated. For the implementation of this training procedure, the objective function of the original VAE is modified as follows [17].

$$\begin{aligned}\log(p(y)) &= \int \log(p(y)) q_\phi(z \mid x) dz \\ &= \int \log(\frac{p(y)p(z \mid y)q_\phi(z \mid x)}{p(z \mid y)q_\phi(z \mid x)}) q_\phi(z \mid x) dz \\ &= \int \log(\frac{q_\phi(z \mid x)}{p(z \mid y)}) q_\phi(z \mid x) dz + \int \log(\frac{p(y)p(z \mid y)}{q_\phi(z \mid x)}) q_\phi(z \mid x) dz \\ &= \int \log(\frac{q_\phi(z \mid x)}{p(z \mid y)}) q_\phi(z \mid x) dz + \int \log(\frac{p(y \mid z)p(z)}{q_\phi(z \mid x)}) q_\phi(z \mid x) dz\end{aligned} \tag{2.9}$$

The purpose of finding the distribution $p(x)$ of the data is similar to VAE. However, in this problem, the desired output y is not x, which is the same value as the input, so we need to find the distribution of the new output y. Therefore, the equation starts with $p(y)$. As a result, the first term represents the distance between two probability distributions of $q_\phi(z \mid x)$ and $p(z \mid y)$, and the distance between these two distributions should be minimized. In other words, pay attention to the second term. And the training is proceeded to the direction in which this value is maximized.

This second term can be thought of as a new evidence lower bound $ELBO(\phi)$.

$$\begin{aligned}ELBO(\phi) &= \int \log(p(y \mid z)) q_\phi(z \mid x) dz - \int \log(\frac{q_\phi(z \mid x)}{p(z)}) q_\phi(z \mid x) dz \\ &= \mathrm{E}_{z \sim q_\phi(z \mid x)}[\log p(y \mid z)] - KL(q_\phi(z \mid x) \parallel p(z))\end{aligned} \tag{2.10}$$

The same result as Equation (2.2) of VAE was obtained, except that input and output were different.

Finally, the following equation was considered for training.

$$\arg\min_{\phi,\theta} \sum_i \mathrm{E}_{q_\phi(z|x_i)}[\log(p(y_i \mid g_\theta(z)))] - KL(q_\phi(z \mid x_i) \,||\, p(z)) \tag{2.11}$$

Like VAE, the first term is the reconstruction loss and should be minimized so that the input x and output y are equal. The second term is the KL-divergence of $q_\phi(z \mid x)$ and $p(z)$, and an ideal sampling function can be obtained by minimizing the distance between two distributions. By maximizing the $ELBO(\phi)$ obtained by adding these two losses, the model can be finally constructed so that the deformation and the crack share one latent space. As a result, when a new deformation is input, a network for outputting a corresponding crack can be completed. The loss function defined on the basis of this theory is as follows. Where x is deformation image, y is crack image, G is generator network of VAE, and D is ANN based discriminator network.

$$\begin{aligned} Loss_{Recon} &= -\sum (y \log G(x) + (1-y)\log(1-G(x))) \\ Loss_{KL} &= 0.5\sum(\mu^2 + \sigma^2 - \log\sigma^2 - 1) \\ Loss &= -(Loss_{\mathrm{Re}con} + Loss_{KL}) \end{aligned} \tag{2.12}$$

Therefore, the training was performed by simultaneously updating the parameters of the encoder and decoder in the direction of reducing loss such as Equation (2.12).

2.3.1.2 Result of Network 1

The optimization function and hyper-parameter used in the training process are as follows.

- Optimizer : ADAM optimizer (Kingma and Ba 2014)
- Learning rate : 0.0005
- Batch size : 100
- Epoch : 2000
- Filter size : 5×5

It trained the network using 2000 data pairs of deformation and crack images. Next, for verification, we confirmed that when a new deformation image was input, it generates a corresponding crack image.

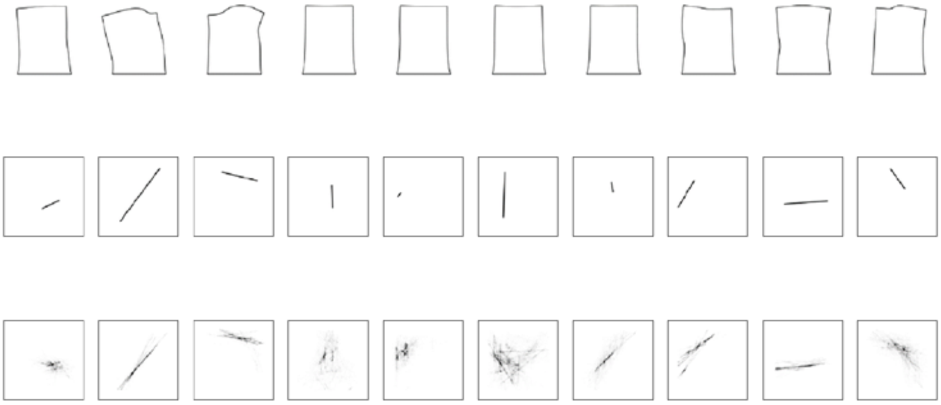


Figure 2.9 Result of Network 1 using Type 1 dataset

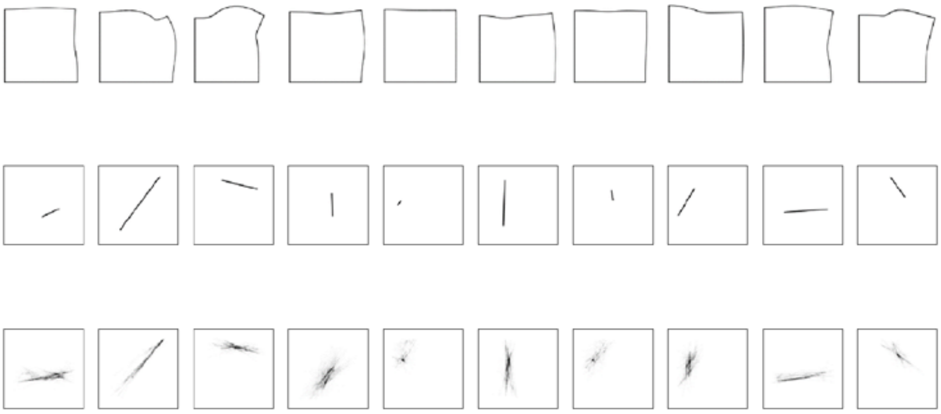


Figure 2.10 Result of Network 1 using Type 2 dataset

2.3.1.3 Discussion of Network 1

Figure 2.9 shows the result of Type 1 data, and Figure 2.10 shows the result of Type 2 data. As a result, there is no difference in deformation for the other cracks, as in the fifth and seventh for small cracks, so it can be confirmed that training is difficult. Also, we can see that the result is similar to the training data set, which is characteristic of VAE. On the other hand, in the case of crack not in the training data set, the detection performance is lowered as in the seventh. To solve this problem, there is a method of increasing the number of training data. However, it is less efficient, the number of suitable training data should be selected. It also shows the blurry output characteristic of VAE. In order to mitigate this, Network 2, which is modified AAE, was constructed.

2.3.2. Adversarial encoder and decoder network (Network 2)

Network 1 had a blurry result. In addition, since random z sampling is performed, there is a disadvantage in that result is not constant for each test. To overcome these limitations, we used the AAE network. Therefore, AAE has both the advantages of VAE, which is relatively easy to learn and generates physically meaningful values, and GAN, which shows clear results. In this study, only the operating structure of AAE was used, and the model was named adversarial encoder and decoder network.

2.3.2.1 Detail of Network 2

AAE is characterized by the same input and output. Therefore, in order to apply to this problem, we have to modify it to output the crack corresponding to the deformation. Similar to Network 1, a unified latent space was constructed so that deformation and crack share a single latent space. Therefore, as shown in Figure 2.11, three networks including the encoder network of deformation image, the decoder network of crack image, and discriminator network are constructed.

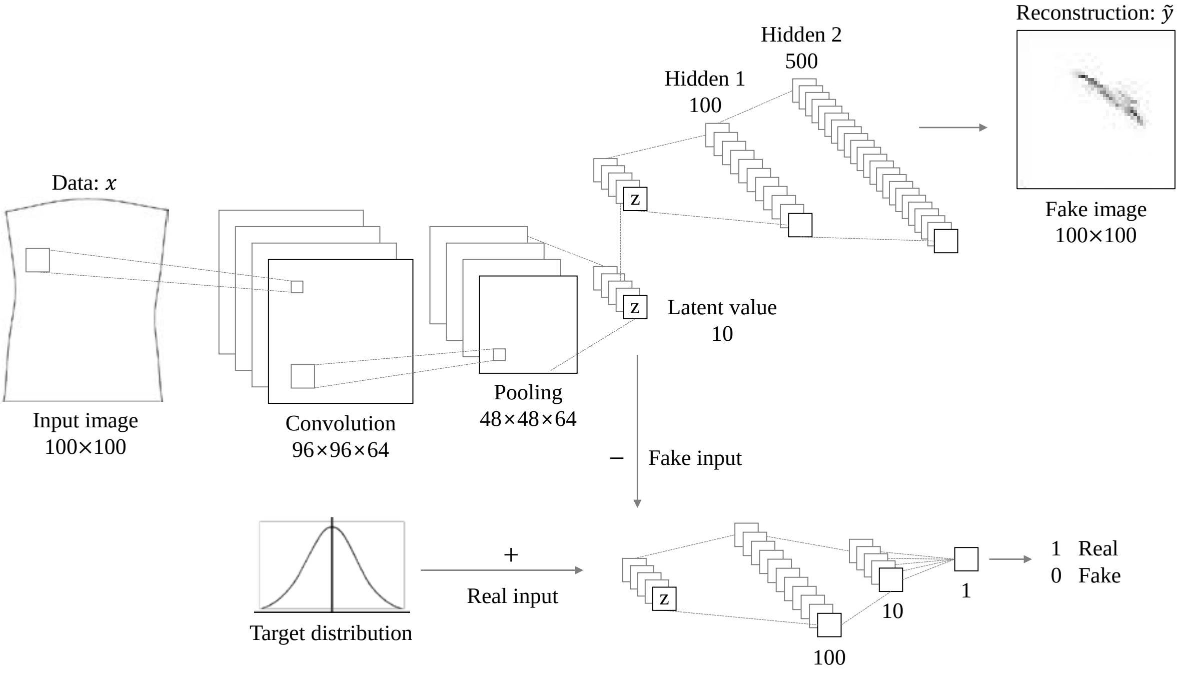


Figure 2.11 Detail of adversarial encoder and decoder network

As shown in Figure 2.11, the AE network for deformation image is an encoder network of CNN structure. When the deformation image is input, 10 latent vectors are output. Next, the AE network for the crack image is a decoder network of the ANN structure, and a crack image is generated from the result of the preceding encoding. This process is similar to that of the previous Network 1. However, the adversarial encoder and decoder network used discriminator instead of KL-divergence, so the reconstruction loss is equal to Equation (2.13).

$$Loss_{AE} = -\mathrm{E}_{q_{\phi}(z|x)}[\log(p_{\theta}(y\,|\,z))] \tag{2.13}$$

First, when deformation is given as an input, the training proceeds to minimize the reconstruction loss. Next, the ANN based discriminator network serves to distinguish fake latent value z of real latent value z' sampled from the real distribution. The loss function of the discriminator network is expressed by Equation (2.14), and the training proceeds in the direction of minimizing loss.

$$-Loss_{D} = -\log d_{\lambda}(z) - \log(1 - d_{\lambda}(q_{\phi}(x))) \tag{2.14}$$

Finally, the generator loss function is constructed as Equation (2.15) to make the discriminator discriminate as real for fake latent value z .

$$-Loss_{G} = -\log(d_{\lambda}(q_{\phi}(x))) \tag{2.15}$$

The generator of Network 2 represents the ENCODER part, and the training was performed by updating the parameter in the direction of minimizing this loss.

2.3.2.2 Result of Network 2

The optimization function and hyper-parameter used in the training process are as follows.

- Optimizer : ADAM optimizer (Kingma and Ba 2014)
- Learning rate : 0.0005 (For generator, encoder and decoder network)
- Learning rate : 0.0001 (For discriminator)
- Batch size : 100
- Epoch : 2000
- Filter size : 5×5

The initial output of generator is not good, so discriminator can easily distinguish it. Therefore, we used a smaller learning rate than the generator to slow the training of the discriminator. The training was conducted using 2000 pairs. When a deformation image is given to trained network, it is encoded to obtain z . The result is then used as a decoder input to generate a crack image.

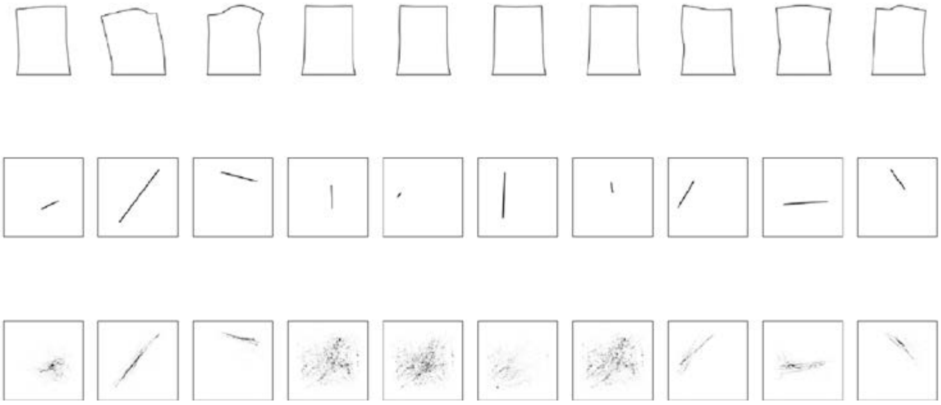


Figure 2.12 Result of Network 2 using Type 1 dataset

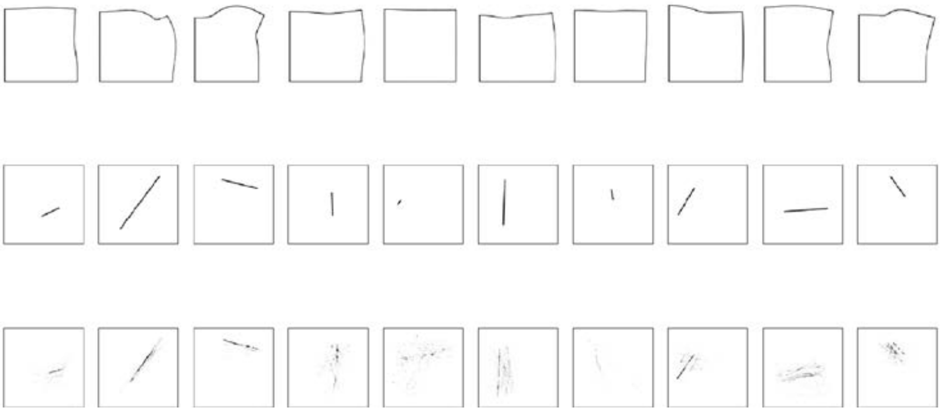


Figure 2.13 Result of Network 2 using Type 2 dataset

2.3.2.3 Discussion of Network 2

Figure 2.12 shows the results of Type 1 data and Figure 2.13 shows the results of Type 2 data. First, compared to Network 1, the blurring was somewhat resolved, but some cracks were not clear. There was no significant improvement in small cracks. However, since the latent value obtained as an encoding result is fixed, there is an advantage that a fixed value can be obtained. This model is fundamentally a structure where discriminator and generator are adversarial to each other, so training does not work well compared with Network 1. The results also showed no significant improvement. Therefore, to improve the efficiency and performance, Network 3 was proposed after adding discriminator to Network 1.

2.3.3. Variational encoder and decoder network with discriminator (Network 3)

Network 2 solved some of the blurry problems, but it was difficult to train. To overcome this limitation, we added a discriminator to Network 1. Like Network 1, the training is easy, and the network is constructed to have the advantages of VAE and GAN. Therefore, we added a discriminator to the operating structure of Network 1 and named it variational encoder and decoder network with discriminator.

2.3.3.1 Detail of Network 3

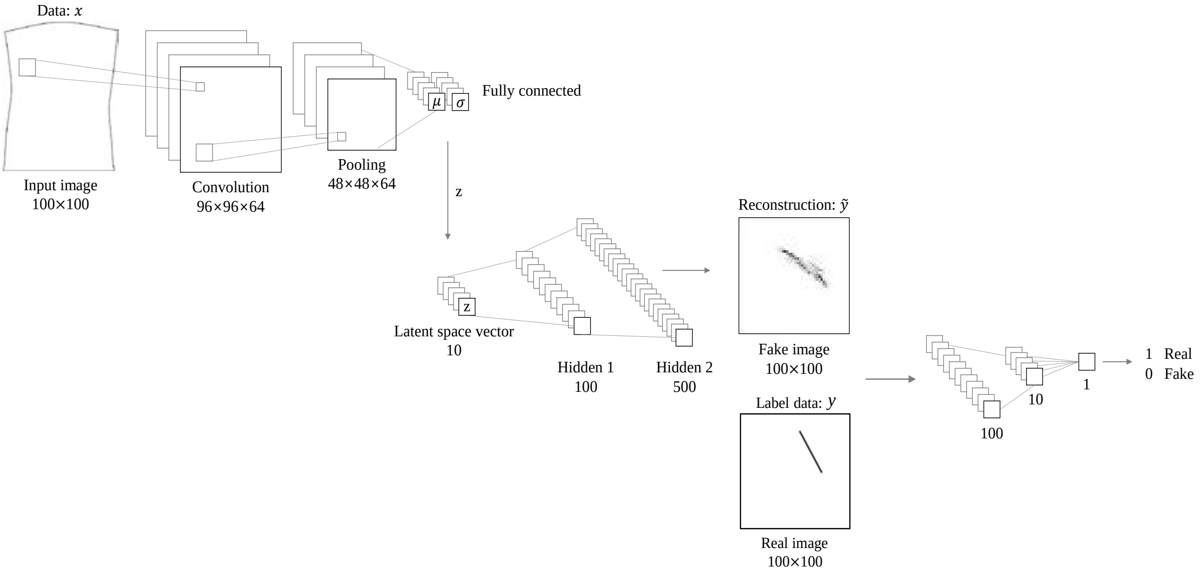


Figure 2.14 Detail of variational encoder and decoder network with discriminator

Reconstruct the model by adding a discriminator to Network 1 as shown in Figure 2.14. First, we constructed a VAE model for deformation and cracks separately. And then, take a structure of Network 1 that allows the deformation and crack to share a single latent space. Next, the encoder and decoder network thus constituted is used as generator. Like Network 1, the generator reduces the dimensions of the image to 10 latent vectors when a deformation image is given. Next, increase the dimension to the same size. Thereafter, training proceeds to generate a crack image. In this case, the fake image generated by the generator is used as input to the discriminator along with the real crack image, so the loss function is redefined.

Network 1 proceeds to minimize the reconstruction loss and KL-divergence (KL-loss) in order to generate a crack image from the deformation image. Network 3 is further constructed with GAN loss to train the discriminator to discriminate the fake image as real. As a result, deformation and crack share latent space, and when deformation is inputted, the network that outputs crack can be completed. The Loss function of the generator and the discriminator defined on the basis of this theory is finally shown in Equation (2.16). Where x is deformation image, y is crack image, G is the generator network of the VAE structure, and D is the ANN based discriminator network.

$$\begin{aligned} Loss_{Recon} &= -\sum (y \log G(x) + (1-y)\log(1-G(x))) \\ Loss_{KL} &= 0.5\sum (\mu^2 + \sigma^2 - \log\sigma^2 - 1) \\ Loss_{GAN} &= \sum \log D(G(x)) \end{aligned} \tag{2.16}$$

$$\begin{aligned} Loss_{G} &= -(Loss_{\mathrm{Re}con} + Loss_{KL} + Loss_{GAN}) \\ Loss_{D} &= -\sum \log D(y) + \log D(1-G(x)) \end{aligned}$$

Therefore, as in Equation (2.16), the encoder and decoder parameters of the generator network are simultaneously updated in the direction of minimizing the G loss. It also updates the discriminator network parameters to minimize the D Loss.

2.3.3.2 Result of Network 3

The optimization function and hyper-parameter used in the training process are as follows.

- Optimizer : ADAM optimizer (Kingma and Ba 2014)
- Learning rate : 0.0005 (For generator, encoder and decoder network)
- Learning rate : 0.0001 (For discriminator)
- Batch size : 100
- Epoch : 2000
- Filter size : 5×5

For the same reason as Network 2, the learning rate of generator and discriminator was different. The training was conducted using 2000 data pairs. After that, test data was generated and the trained network was verified. At this time, the network encodes the input deformation to obtain the latent value z. Then, the z is used as a decoder input to generate a crack image.

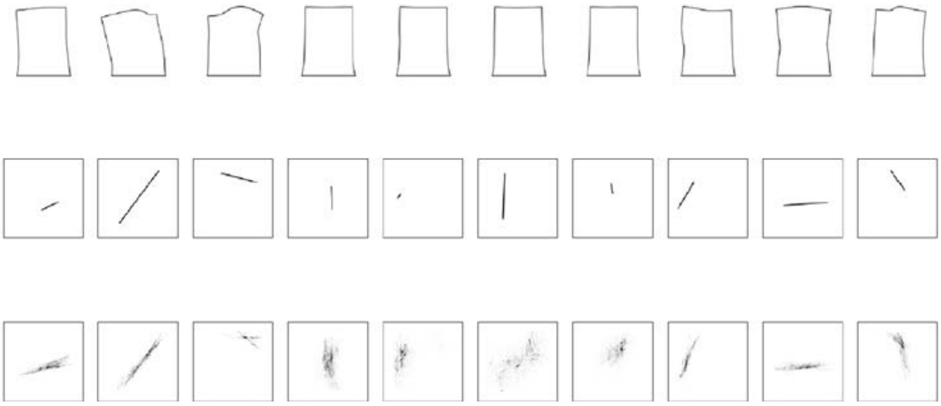


Figure 2.15 Result of Network 3 using Type 1 dataset

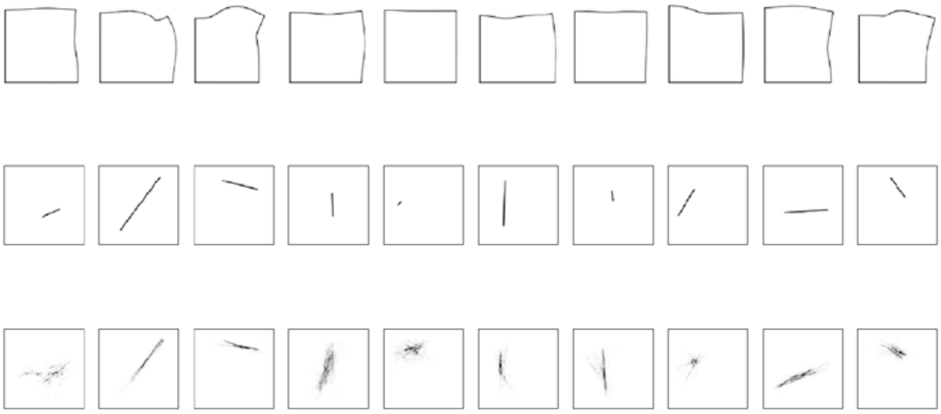


Figure 2.16 Result of Network 3 using Type 2 dataset

2.3.3.3 Discussion of Network 3

Figure 2.15 is the result of Type 1 data, and Figure 2.16 is the result of Type 2 data. Because it is the same structure as Network 1, it can be seen that the characteristic of Network 1 also appears in Network 3. That is, it outputs different results for each test, and it can be confirmed that detection is difficult for small cracks. Also, as with Network 1, you can see that the results are similar to the cracks in the training dataset. However, compared to Network 1, blurry result was somewhat resolved. Therefore, we confirmed that Network 3 outputs relatively accurate results. Furthermore, compared to Network 2, training is relatively easy. However, the current comparison is to qualitatively compare the results of the proposed Networks. Next, we propose a measure for quantitative performance comparison.

2.4. Evaluating the performance of trained network

In chapter 2.3, performance was checked for each network, but it was very qualitative. Therefore, in this chapter, we propose the following measures to quantitatively evaluate the performance of the training network.

2.4.1. Quantitative evaluation

2.4.1.1 Quantitative evaluation using least square method

The detection results show that the position and shape of the cracks are comparable, but they are blurry. Therefore, the least square method was modified to solve this problem. When we know the straight line of the crack, we used the vertical distance from the straight line as the error. The reason for not using pixel-wise MSE at this time is that although the network did not detect the exact position, we wanted to find the error when finding similar position and shape cracks.

$$\begin{aligned} Error_{(1)} &= \sum_{i}^{m} 1-\sqrt{weight}_{i} \\ Error_{(2)} &= \sum_{i}^{10000-m} \sqrt{\text{distance}_i} \times \sqrt{weight_i} \\ Error_{total} &= Error_{(1)} + Error_{(2)} \\ Accuracy &= 1-\frac{Error_{total}}{10000} \end{aligned} \tag{2.17}$$

Therefore, the straight line at 100×100 domain is obtained by using both ends of the crack. Then, the coordinates of the cracks detected by the network are the components (i, j) of the 100×100 matrix. Next, we use the distance formula between the coordinates of the crack and the straight line. That is, the distance between the detection value and the actual crack straight line is numerically expressed. At this time, we used the method that divides the error into the longest distance ($100\sqrt{2}$) in the domain so as to be between 0 and 1. In addition, the detection result values are closer to 0 in white, and closer to 1, they are in black. Therefore, the detected values along with the distance from the actual crack should be calculated with weighting. For this, the error was calculated by dividing into (1) the area where the actual crack existed and (2) the area where the crack did not exist. At this time, since the weight values are mostly concentrated from 0 to 0.4, the square root was taken to adjust the scale. Therefore, the error can be obtained by Equation (2.17) using the values obtained in (1) and the values obtained in (2).

2.4.1.2 Quantitative evaluation using histogram compare method

The accuracy was calculated using the least square method. However, there is a disadvantage that measurement of size is not possible. In order to compensate for this, we have considered the use of different measures together. In order to compensate for this, we have considered the use of different measures together. A histogram comparison method was used to measure the size of the crack. The actual crack data and detection data are projected on the x and y-axes and the histograms are compared. To do this, a data histogram is obtained by using dot product the vector with all components 1 on the left and right sides. Also, since the weights are mostly concentrated in values between 0 and 0.4, we take the square root to adjust the scale. Then, using the histogram correlation method of Equation (2.18), accuracy between 0 and 1 was shown.

$$\begin{aligned} d(H_1, H_2) &= \frac{\sum_i (H_1(i) - \bar{H}_1)(H_2(i) - \bar{H}_2)}{\sqrt{\sum_i (H_1(i) - \bar{H}_1)^2 \sum_i (H_2(i) - \bar{H}_2)^2}} \\ \bar{H}_k &= \frac{1}{N} \sum_j H_k(j) \end{aligned} \tag{2.18}$$

In the equation, N represents the total number of histogram bins, and i and j represent the i-th and j-th components of the histogram, respectively.

2.4.1.3 Result

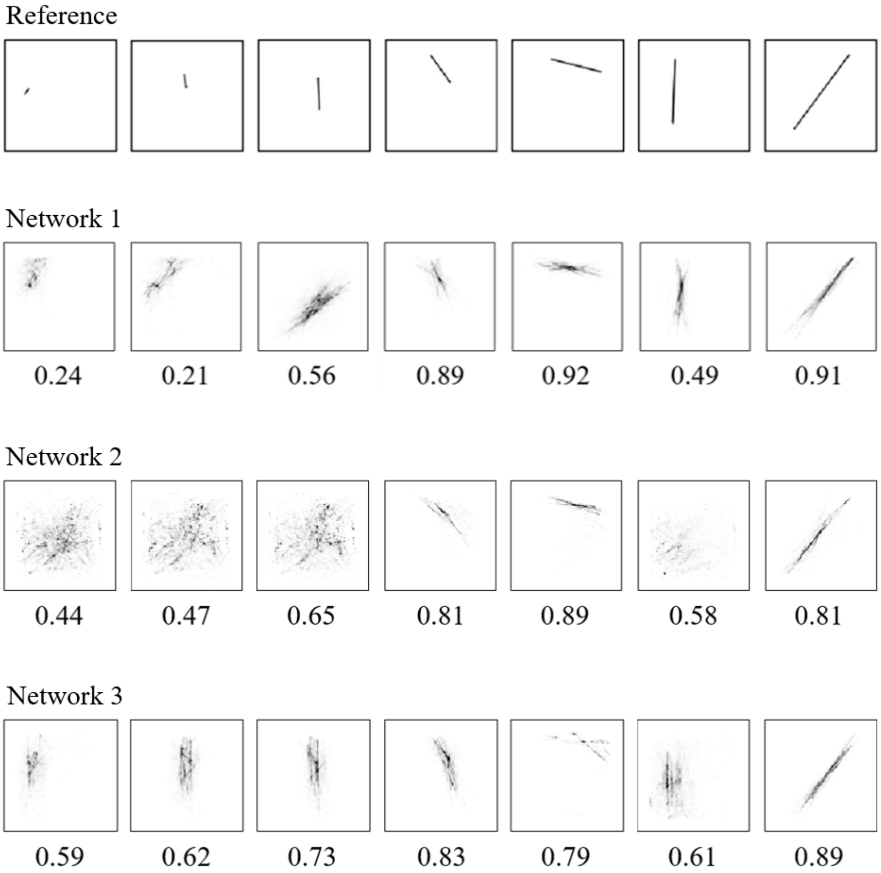


Figure 2.17 Quantitative result of Type 1 dataset

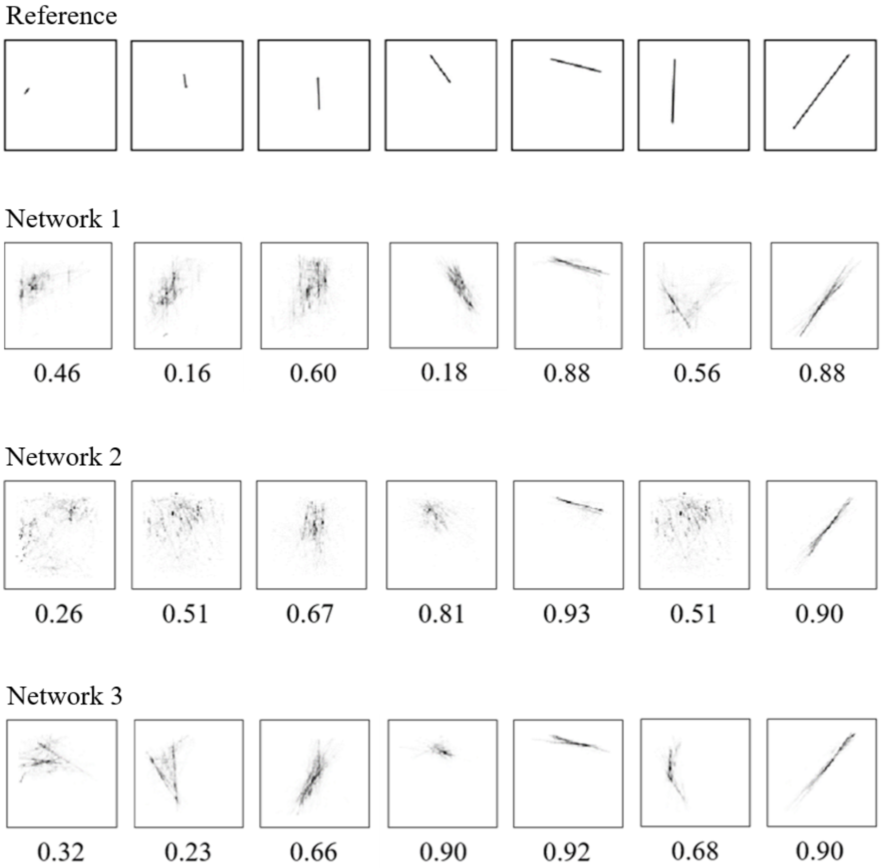


Figure 2.18 Quantitative result of Type 2 dataset

2.4.1.4 Discussion

Using the least square method and the histogram compare method, the accuracy is calculated as shown in Figure 2.17 and Figure 2.18. Figure 2.17 shows the result of training using Type 1 data. The results show that the accuracy is low for small crack. This is because deformation is very small regardless of the location of the cracks. Even though the vertical cracks are large, the same phenomenon occurs because the deformation is similar to the vertical tensile stress. In addition, the result of Network 3 is clearer than that of Network 1, and it was quantitatively accurate. Also, for Network 2, the accuracy was not very good for some cases of cracks. Figure 2.18 shows the result of training using Type 1 data. In this case, Network 3 is more accurate than Network 1 and Network 2 in terms of accuracy.

However, the proposed measure has the disadvantage of knowing the slope of the crack. Therefore, we used accuracy and recall to measure the accuracy. To measure precision and recall, a value less than a certain value is determined as 0, and a value equal to or greater than a predetermined value is determined as 1, and the result is clarified. As a result, the detection image is as shown in Figure 2.19, and the data values are composed of only 0 (white) and 1 (black). The first row in Figure 2.19 shows the actual cracks, and the second row is the network-detected crack. Finally, the third row is a refined result for quantitative evaluation.

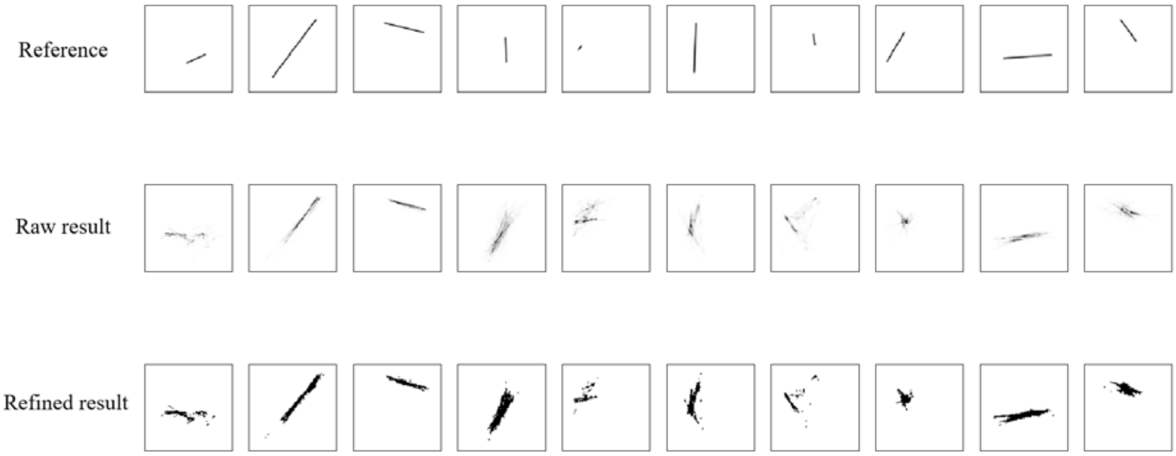


Figure 2.19 Refined result

To evaluate the detection performance, a detection rate (recall) was used first. Recall represents the ability to capture objects without missing in cognitive technology. Thus, this problem is a measure of whether the system has detected cracks in the pixel region where real cracks exist. Therefore, it is expressed as follows.

$$recall = \frac{crack_{real} \bigcap crack_{fake}}{crack_{real}} \tag{2.17}$$

However, if there is a crack in all pixels, the recall is 100%, so recall is not enough to evaluate performance. In other words, recall and precision must be considered simultaneously to evaluate the performance of detection technology. Therefore, we calculated the precision of crack detection results. Precision is used in machine learning to have a similar meaning to accuracy. Precision is the accuracy when the system detects a crack. That is, it is a measure to judge whether a pixel that the network diagnoses as a crack is a pixel where a real crack exists.

$$precision = \frac{crack_{real} \bigcap crack_{fake}}{crack_{fake}} \tag{2.19}$$

To evaluate the performance of detection technology, recall and precision must be considered simultaneously. Therefore, we want to express the precision-recall performance of the algorithm as a number using F-measure. The F-measure is calculated as the harmonic mean of the precision and recall. It can be expressed as follows.

$$F - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.20}$$

In this study, both precision and recall are very important. However, in fact, the value of precision is very small because it is represented by approximate position in detection. Therefore, both F-measure and recall were calculated. For performance verification, 100 Type 2 data were used and the quantitative evaluation values were averaged. The results of these performance tests are shown in Table 2.2.

Table 2.2 Result of precision and recall

	Recall	F-measure
Network 1	0.362	0.16
Network 2	0.366	0.18
Network 3	0.282	0.14

First, it was confirmed whether the quantitative evaluation was successful. As mentioned earlier, in the case of crack detection of this problem, it is very important to get the correct value. However, it is difficult to obtain accurate cracks in the actual exact position. That is, the value of precision is very small because the crack is represented by the approximate position. Therefore, the F-measure value is very small. On the other hand, the recall, which is the approximate location accuracy, was relatively high. Unlike the actual qualitative results, the results were evaluated with pixel by pixel, so the accuracy was low even though the approximate position was well detected. Therefore, this measure is considered to need more complement. Next, the network was compared using quantitative evaluation results. First, we can confirm that Network 3 has good performance for Type 1 data, which is difficult to detect for vertical cracks due to boundary condition and force direction, except for vertical cracks. On the other hand, in case of Type 2 data, the result of Network 2 is relatively accurate. However, it can be seen that Network 1 and Network 3 show more accurate results quantitatively. Comparing the network performance based on the qualitative comparison and the quantitative comparison, it can be confirmed that Network 3 shows more accurate results and quantitatively accurate results. At this time, Network 2 is somewhat clearer than the result of Network 1, but there is no big difference. It also does not detect size or location for small cracks. In other words, there are significant differences in performance. In addition, Network 2 is difficult to train, so Network 1 and Network 3 are used in this study. Finally, Figure 2.20 shows the detection results using Type 2 of the actual system in the case of the crack-free structure. When the quantitative evaluation was carried out except 0.03 or less pixels, a reasonable result was obtained.

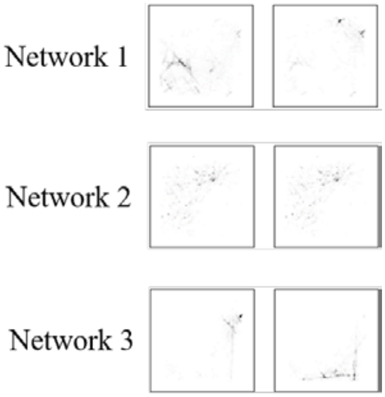


Figure 2.20 Result of crack-free structure

2.5. Validation of network performance

Convergence test was performed for verification of Network 1 and Network 3. Total 2000 data, 4000 data and 8000 data were used for the learning and the quantitative accuracy was averaged using a total of 1000 test data. The result is shown in Figure 2.21.

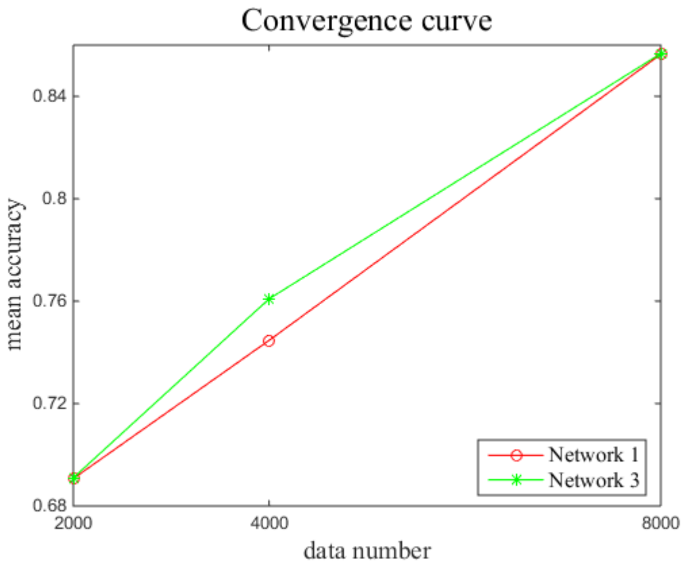


Figure 2.21 Convergence curve of quantitative results

In the convergence curve, it is confirmed that the accuracy improves as the number of data actually increases. However, Network 3 has higher accuracy than Network 1 only when it is trained using 4000 data. This is because this convergence curve compares the results of 1500 iterations. That is, in the case of Network 3 having a larger number of parameters, the training was not completed for all the parameters, and the results were different from each other. Therefore, if the number of training iterations is increased, the result is expected to improve. Finally, the number of latent values was increased by increasing the latent value from 5. The results of training Network 3 using Type 2 data (2000) are as follows. 5 was 0.7394, 10 was 0.6910, and finally 20 was 0.6787. Therefore, the best result was obtained when the latent value was 5.

2.6. Conclusion

In this study, we attempted to solve the problem of crack detection by generating a crack corresponding to the deformation of a two-dimensional rectangular structure. Therefore, the generative model was trained. Actually, rather than using VAE, which has the same input and output structure, it is more structured to use a model called Pix2pix using GAN, an image matching method, to implement crack images for deformation. Pix2pix is a structure in which generator and discriminator grow hostile in the way that generator generates an image from noise when input data is given and discriminator distinguishes it from a real image. Therefore, there is an advantage that a new image can be generated. However, as a result of actual training, there was a disadvantage that it was difficult to learn by falling into mode collapsing phenomenon as shown in Figure 2.22.

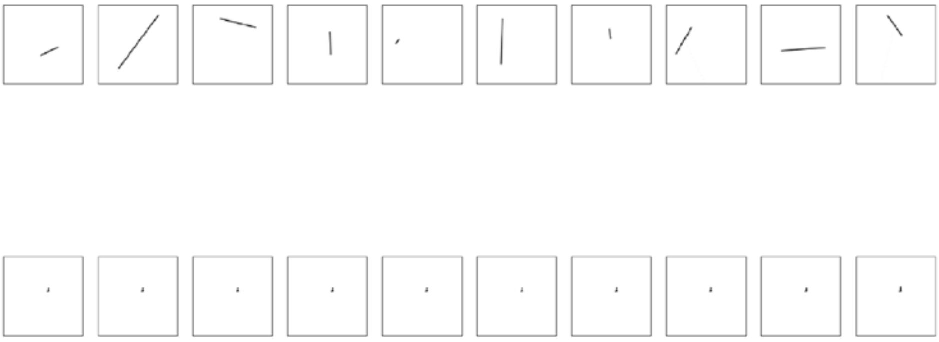


Figure 2.22 Result of Pix2pix

In this study, it is also important to create an image suitable for the deformation even if it is close to the training data, rather than creating a new image that is the advantage of the GAN. Therefore, based on the structure of VAE, discriminator was added to compensate for the disadvantages of VAE. Therefore, we proposed the structure of Network 1 ~ 3. In addition, through validation, we confirmed that the proposed network can finally detect cracks from the deformation image. However, in the case of deformation images, there is a limitation that cracks can be detected only for actual force conditions. In other words, there is a disadvantage in that if it is applied to an actual problem, data must be generated in accordance with the condition of the load and the training should proceed. To alleviate this limitation, we will use a static-mode shape that is load-independent in the next chapter. We also trained the Network 1 and Network 3 networks using the mode shape data.

Chapter 3. Crack detection using mode image

In this chapter, training and testing were conducted when the mode shape was used as input data in the crack detection model proposed in chapter 2.

3.1. Generating the dataset

Since the deformation data is load-dependent, there are limitations. In other words, a dataset must be configured for each load-condition to train the network. Therefore, in this chapter, static mode is used to construct load-independent dataset.

3.1.1. Generating the data with XFEM

Mode data was also generated using the XFEM code [26]. Consider a rectangular structure with the same conditions as in Table 3.1.

Table 3.1 Conditions of FE model of mode data

Conditions	
Height	2
Width	2
Properties	200 / 0.3

Therefore, to obtain the static mode, only the boundary conditions are considered except for the load condition. This is because the actual structure has a boundary condition and can be constructed as a model. Next, we used the boundary conditions of Type 2, which had good results in chapter 2, as shown in Figure 3.1.

In this condition, we obtain the eigenvector of the stiffness matrix obtained from Finite Element (FE) model. In this case, the first mode with the smallest eigenvalue represents the mode shape in which the structure is most easily changed, so that the first mode is used. Considering the training results of the mixed data in chapter 2, crack detection performance improves when additional features are added. Therefore, the mode shape is obtained from the eigenvector when only the first mode is input, and the mode shape is obtained from the eigenvector when both the first mode and the second mode are input.

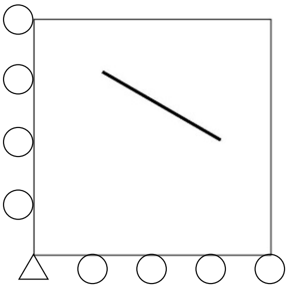
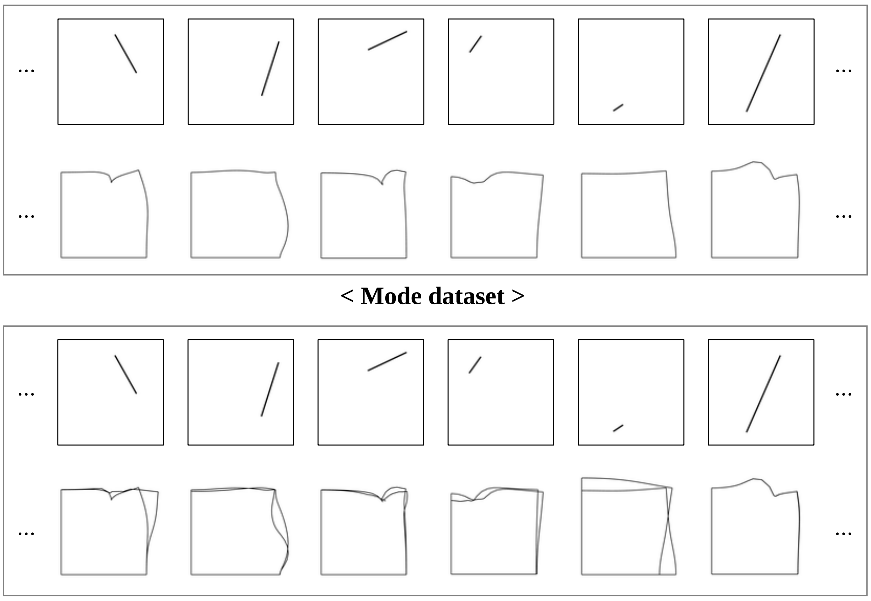


Figure 3.1 Boundary condition of load-independent dataset

3.1.2. Refining the data with normalization

For training, XFEM results are imaged. However, the mode shape of actual XFEM result is small deformation. Also, the overall mode deformation is smaller than deformation of the enrich node. Therefore, normalization is performed so that the maximum deformation of the outer node is 10% of the diagonal length ($2\sqrt{2}$) since the outer line of the structure is used for training. In this way, 2000 image data sets were created and the data were refined. The data of 2000 pairs used for training are divided into mode data and mixed mode data according to the number of modes used, as shown in Figure 3.2.



< Mode dataset >

< Mixed mode dataset >

Figure 3.2 Training datasets of mode shape

3.2. Training result

In chapter 2, the network is sequentially improved for crack detection. In this study, we selected the training of Network 1 and Network 3 and conducted training. Then we verified the performance of the trained network. Figure 3.3 shows the results of Network 1 and Network 3 training with mode data. Figure 3.4 shows the result of Network 1 and Network 3 training mixed mode data using both primary mode and secondary mode.

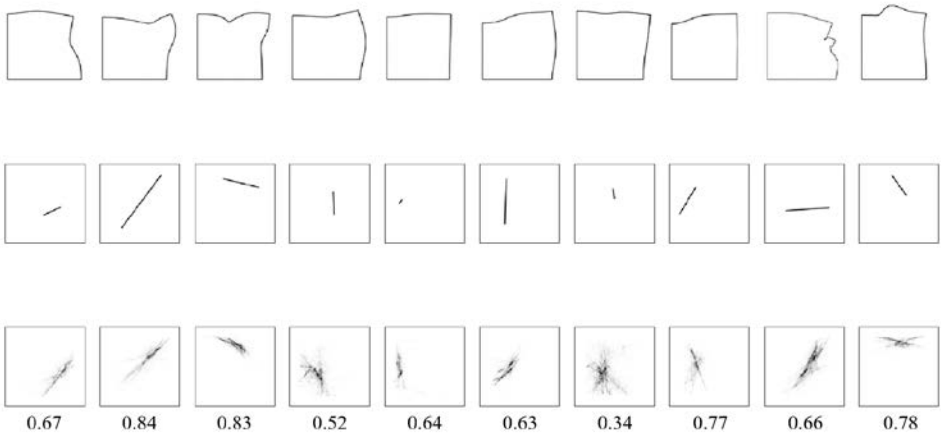


Figure 3.3 Result of Network 1and Network 3 using mode dataset

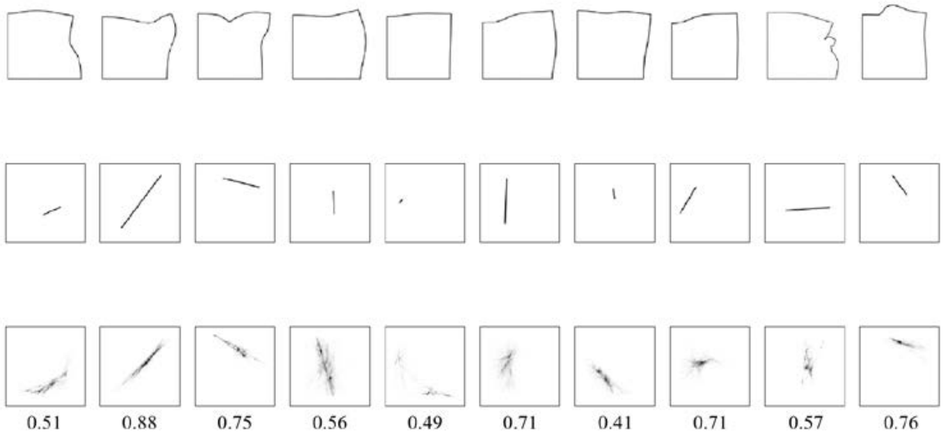


Figure 3.4 Result of Network 1 and Network 3 using mixed mode dataset

3.2.1. Discussion

Compared with the result of deformation data in chapter 2, it can be confirmed that the result is not good. This is because the actual deformation of the structure consists of superposition of various modes. Therefore, it is difficult to detect crack with just one mode. Therefore, as a result of checking the data, different cracks may correspond to each other in a similar mode shape as shown in Figure 3.5, which makes training of the network difficult.

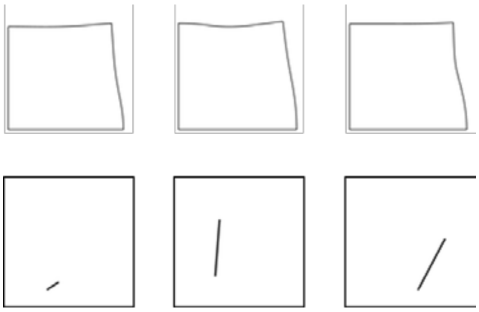


Figure 3.5 Similar mode shapes corresponding to different cracks

Therefore, the network is trained by using mixed mode data. As a result, we could obtain good results in Network 1 and Network 3 when using mixed mode data. Therefore, it is expected that more accurate crack detection will be possible by using superposition of modes of first mode, second mode, third mode.

Next, comparing the network, we can confirm that the result of Network 3 is clearer than Network 1. That is, the blurry result is resolved. However, the quantitative accuracy showed similar results. Unlike Network 2, which trains to minimize generator loss, Network3 is training to minimize overall loss by adding generator loss to ELBO. Therefore, the actual generator loss increases as training progresses. However, there is an advantage that GAN is applied because the training progresses because the generator and the discriminator are hostile.

3.3. Closure

In this chapter, we have trained Network 1 and Network 3 proposed in chapter 2 using static mode shape data of structures for crack detection. In other words, it is a crack detection method that generates an image of a corresponding crack from the mode shape of a two-dimensional rectangular structure. Because the deformation data in chapter 2 is load-dependent data, the efficiency is not good, so the training is performed using load-independent data.

Therefore, the training was performed using the primary mode shape. The results were worse than the training using deformation. This is because, as mentioned above, the characteristics of the crack cannot be found only in a few low-order modes. Therefore, mixed mode data was generated using both the primary mode shape and the secondary mode shape, and training was conducted. As a result, the performance was improved, and the feasibility of the method of generating data by superposition more modes was confirmed.

Chapter 4. Conclusion

In this study, we attempted to solve the problem of detecting internal crack of structure from external data (deformation, mode shape) by using generative model of deep learning. Therefore, when a deformation image of a structure is given, a training model is constructed to detect crack. First, training data and test data were generated using XFEM code. The deformation images and the corresponding crack images were generated by varying the force and boundary conditions on the 2×2 rectangular two-dimensional structures with random cracks. For the load-independent data training, a mode shape of the structure using only boundary conditions and a corresponding crack image were generated. In this way, 2000 pairs were used as training data, respectively. For the test data, 10 pairs of deformation and mode shape images were prepared for cracks not included in the training data. The deformation(mode) image is entered into the CNN based encoder network. At this time, the dimension is reduced to a latent vector, and it is trained to generate a crack image in the decoder network. This process is required to obtain both the distribution of the deformation(mode) image and the crack image. However, since deformation and crack are closely related to each other in this problem, it can be assumed that deformation and crack have the same latent vector. Therefore, a training model is constructed so that the deformation image and the crack image share the same latent vector. In the process, we minimize the distance between $q_\phi(x \mid z)$ to obtain the latent vector of deformation and $p(z \mid y)$ to obtain the latent vector of crack. In other words, the encoder and decoder parameters are updated so that the two latent vectors are equal. Using this encoder and decoder structure, a network for crack detection was constructed.

Network 1 is derived from a variational encoder and decoder network. Network 2 is an adversarial encoder and decoder network with GAN added to Network 1, and it tries to solve blurry problem which is a disadvantage of VAE. Network 2 solved the problem of blurry, which was the limitation of Network 1, but it became difficult to train with GAN structure. In addition, some cracks showed satisfactory performance, but for small cracks, they did not even notice the shape. Therefore, in order to solve this problem, training was conducted with Network 3 where a discriminator network was added to Network 1. In order to validate the performance of the three networks, a new deformation(mode) image was input and a crack image was output.

First, I have reviewed data. When the deformation image was used for the training data, Type 2 was better than Type 1. This is because vertical tensile load is applied to Type 1, so vertical cracks have similar deformation. Therefore, Type 2 data, which was able to better characterize crack, could detect crack more accurately. Small crack also had a small deformation size and could not obtain accurate result. Next, we use a mode shape which is load-independent data to overcome the constraint of deformation data. However, the detection using the mode shape showed worse performance than the deformation image. This is because the characteristics of cracks cannot be completely expressed by using only the primary mode. Therefore, the training was performed with the image that overlapped the primary mode and the secondary mode, and the better results were obtained. Through this, we confirmed the possibility of crack detection using mode data.

Next, the training model showed similar results for Network 1 and Network 3, which are structurally similar. By comparison, Network 3, which added a discriminator, performed a little better. At this time, Network 2 has a merit that Network 2 shows one result, not a random result in structure, and high-accuracy detection performance in some cases of cracks. However, the GAN feature makes training difficult, and in some cases the accuracy of cracks is not good. Therefore, Network 2 is excluded in order to select a network that guarantees consistent performance regardless of what data is input. Therefore, a convergence test was conducted to validate the adopted Network 1 and Network 3. At this time, the best used data was Type 2, and the performance of the training model was confirmed using 2000 pairs, 4000 pairs, and 8000 pairs of data. As a result, we can confirm that the performance increases for both Network 1 and Network 3 as the number of data increases. Also, it was confirmed that the detection performance was improved for the cracks that were difficult to detect. However, it is necessary to study whether the number of data is appropriate. As the actual number of data increases, accuracy improves, but training efficiency decreases, so it is important to find the right number of data.

First, since the study was conducted using a two-dimensional image, there is a limitation in actually detecting visible crack. Therefore, this study is not practical in itself, but it is meaningful in that it confirms the feasibility of crack detection using 3D crack data in the future. Still, I was concerned about the application of this study. In the case of a laminated structure with a thin plate type structure, it was judged that the result of this study could be applied because the cracks of each plate are invisible in reality. In fact, this study proceeded with two-dimensional data because there is no 3D finite element model implemented. However, if we later generate crack data for a 3D structure, we will expect that internal crack detection using the FE model and machine learning algorithm is possible.

Second, the deformation of the real structure is very small and difficult to measure. In addition, there is a limitation in that it uses lab-level data to understand the conditions. To solve this problem, we used load-independent static mode. However, static mode shapes also have limitations that are difficult to obtain from structures. Therefore, the results of this study are useful in confirming the feasibility of detecting crack through machine learning of mode data. Therefore, this result is expected to be a cornerstone of crack detection using dynamic mode(vibration) data.

Third, all of the data used in this study were normalized for data refining because of their small deformation. Therefore, the size information of the crack can be ignored in the same type of crack at the same position. In order to compensate for this, we will use conditional variational autoencoder (CVAE) in future studies. That is, the maximum deformation is known, so labeling it as a condition can solve this problem. There is also a limitation that different cracks correspond to similar deformations. Therefore, in order to solve this problem, a method of constructing a training model so that data of Type 1 correspond to data of Type 2 will be used. However, in order to guarantee the detection performance, we think that the application field should be considered for very small cracks.

Finally, in this study, we used CNN and ANN instead of CNN based neural network. However, in the case of image recognition, CNN shows better performance in most studies. Furthermore, the performance of the training model using CNN and ANN was better than the actual ANN based results. Therefore, it is necessary to use full CNN neural network in future studies. In addition, to make the training model robust, we will use training data to synthesize noise in deformation data. We will also check whether crack detection is possible for arbitrary deformations that are not generated by the FE model.

Despite the limitations of this study mentioned above, crack detection through machine learning of FE model data is intuitive and interesting. It is useful in detecting random cracks when compared with conventional crack detection studies using machine learning. Also, since the FE model data is used without using laboratory data, there is an advantage that data can be efficiently collected. Although there is a shortcoming in practicality when compared with current commercially available technologies. However, there is a limit to the commercialized crack diagnosis method that the actual person must directly collect data and analyze it. Therefore, we expect that this study will be a practical study through future work if we use external data (vibration) considering actual measurement.

Bibliography

[1] Zgonc, K., & Achenbach, J. D., *A neural network for crack sizing trained by finite element calculations*. NDT & E International, 29(3), 147-155, (1996).

[2] Chady, T., Enokizono, M., & Sikora, R., *Crack detection and recognition using an eddy current differential probe.* IEEE transactions on magnetics, 35(3), 1849-1852, (1999).

[3] Liu, S. W., Huang, J. H., Sung, J. C., Lee, C. C., *Detection of cracks using neural networks and computational mechanics.* Computer methods in applied mechanics and engineering, 191(25-26), 2831-2845, (2002).

[4] Lim, H. J., & Sohn, H., *Online fatigue crack quantification and prognosis using nonlinear ultrasonic modulation and artificial neural network*. In Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018, Vol. 10598, p. 105981L, (2018).

[5] Agathos, K., Chatzi, E., & Bordas, S. P., *Multiple crack detection in 3D using a stable XFEM and global optimization*. Computational Mechanics, 1-18, (2018).

[6] Kawashima K., Omote R., Ito T., Fujita H., Shima *T., Nonlinear acoustic response through minute surface cracks: FEM simulation and experimentation*. Ultrasonics, 40(1-8), 611-615, (2002).

[7] Pawar P. M., Ganguli R., *Genetic fuzzy system for damage detection in beams and helicopter rotor blades*. Computer methods in applied mechanics and engineering, 192(16-18), 2031-2057, (2003).

[8] Kumar D. S., Mahapatra D. R., Gopalakrishnan S., *A spectral finite element for wave propagation and structural diagnostic analysis of composite beam with transverse crack*. Finite Elements in Analysis and Design, 40(13-14), 1729-1751, (2004).

[9] Aydin, K., *Vibratory characteristics of Euler-Bernoulli beams with an arbitrary number of cracks subjected to axial load*. Journal of Vibration and Control, 14(4), 485-510, (2008).

[10] Dutta D., Sohn H., Harries K. A., Rizzo P, *A nonlinear acoustic technique for crack detection in metallic structures*. Structural Health Monitoring, 8(3), 251-262, (2009).

[11] Li T., Almond D. P., Rees D. A. S., *Crack imaging by scanning pulsed laser spot thermography*. NDT & E International, 44(2), 216-225, (2011).

[12] Schlichting J., Maierhofer C., Kreutzbruck M. *Crack sizing by laser excited thermography*. NDT & E International, 45(1), 133-140, (2012).

[13] Thatoi D. N., Ojha A. A., Bhanjadeo A., Mahapatra R., Sahoo S., Mohapatra S., *Application of Artificial Intelligence Techniques for Detection of Cracks-A Review*. International Journal of Engineering and Technology, 5(1), 57, (2013).

[14] Rao P. S., Mahendra N. V. D., *Vibration Based Damage Identification Method for Cantilever Beam Using Artificial Neural Network*. In International Conference on Experimental Vibration Analysis for Civil Engineering Structures (pp. 85-93). Springer, (2017).

[15] Liu R., Yang B., Zio E., Chen X. *Artificial intelligence for fault diagnosis of rotating machinery: A review.* Mechanical Systems and Signal Processing, 108, 33-47, (2018).

[16] Yu Y., Hur T., Jung J., Jang I. G., *Deep learning for determining a near-optimal topological design without any iteration*. Structural and Multidisciplinary Optimization, 1-13, (2018).

[17] Spurr A., Song J., Park S., Hilliges O., *Cross-modal deep variational hand pose estimation*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 89-98, (2018).

[18] Isola P., Zhu J. Y., Zhou T., Efros A. A., *Image-to-image translation with conditional adversarial networks*. arXiv preprint, (2017).

[19] Smith E., Meger D. *Improved adversarial systems for 3d object generation and reconstruction*. arXiv preprint arXiv:1707.09557, (2017).

[20] Zhang H., Xu T., Li H., Zhang S., Huang X., Wang X., Metaxas D. *Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks*. arXiv preprint, (2017).

[21] Kingma D. P., Welling M., *Auto-encoding variational bayes*. arXiv preprint arXiv:1312.6114, (2013).

[22] Doersch C., *Tutorial on variational autoencoders*. arXiv preprint arXiv:1606.05908, (2016).

[23] Radford A., Metz L., Chintala S., *Unsupervised representation learning with deep convolutional generative adversarial networks.* arXiv preprint arXiv:1511.06434, (2015).

[24] Goodfellow I., *NIPS 2016 tutorial: Generative adversarial networks.* arXiv preprint arXiv:1701.00160, (2016).

[25] Makhzani A., Shlens J., Jaitly N., Goodfellow I., Frey B., *Adversarial autoencoders*. arXiv preprint arXiv:1511.05644, (2015).

[26] G. KIM, *Two-dimensional crack analysis using the extended finite element method with polynomial enrichment technique*, Master Thesis, Korea Adv. Inst. Science, Techn., Daejeon, Republic of Korea, (2018).